Approximation and Hardness: Beyond P and NP

Pasin Manurangsi

May 16, 2019

Approximation and Hardness: Beyond P and NP

by

Pasin Manurangsi

A dissertation submitted in partial satisfaction of the

requirements for the degree of

Doctor of Philosophy

in

Computer Science

in the

Graduate Division

of the

University of California, Berkeley

Committee in charge:

Professor Luca Trevisan, Co-chair
Professor Prasad Raghavendra, Co-chair
Professor Nikhil Srivastrava

Spring 2019

Approximation and Hardness: Beyond P and NP

Abstract

Approximation and Hardness: Beyond P and NP

by

Pasin Manurangsi

Doctor of Philosophy in Computer Science

University of California, Berkeley

Professor Luca Trevisan, Co-chair

Professor Prasad Raghavendra, Co-chair

The theory of NP-hardness of approximation has led to numerous tight characterizations of approximability of hard combinatorial optimization problems. Nonetheless, there are many fundamental problems which are out of reach for these techniques, such as problems that can be solved (or approximated) in quasi-polynomial time, parameterized problems and problems in P.

This dissertation continues the line of work that develops techniques to show inapproximability results for these problems. In the process, we provide hardness of approximation results for the following problems.

- **Problems Between P and NP:** Dense Constraint Satisfaction Problems (CSPs), Densest k-Subgraph with Perfect Completeness, VC Dimension, and Littlestone's Dimension.
- **Parameterized Problems:** k-Dominating Set, k-Clique, k-Biclique, Densest k-Subgraph, Parameterized 2-CSPs, Directed Steiner Network, k-Even Set, and k-Shortest Vector.
- **Problems in P:** Closest Pair, and Maximum Inner Product.

Some of our results, such as those for Densest k-Subgraph, Directed Steiner Network and Parameterized 2-CSP, also present the best known inapproximability factors for the problems, even in the (believed) NP-hard regime. Furthermore, our results for k-Dominating Set and k-Even Set resolve two long-standing open questions in the field of parameterized complexity.

To my parents

Contents

Acknowledgments

Finishing up this dissertation is quite bittersweet; while I am certainly happy, the past four years here at Berkeley have been some of the most wonderful in my life, and I would prefer to never leave! There are so many people I would like to thank for making this happens, and I do apologize in advance if I (inevitably) miss some.

First and foremost, I will never be able to thank my advisors, Luca Trevisan and Prasad Raghavendra, enough for what they have done for me. From their guidance when I am (completely) lost to their moments of brilliance, they have truly shaped and inspired me as a researcher. Outside of theory, their perspective of the world, patience, kindness, humility–and of course great senses of humor–have a profound impact on me. Thank you very much Luca and Prasad!

Throughout my PhD, I have been very lucky to be mentored (and hosted) by: Dana Moshkovitz, Yury Makarychev, Madhur Tulsiani, Irit Dinur, Eden Chlamtác, Arnab Bhattacharyya, and Kai-Min Chung. I am grateful to all of them for teaching me so much, and for their continuing support both professionally and personally.

In addition to those listed above, I have had the great pleasure and honor to work with many amazing collaborators, without whom this thesis would not have been possible: Karthik C.S., Warut Suksompong, Bundit Laekhanukit, Aviad Rubinstein, Danupon Nanongkai, Haris Angelidakis, Parinya Chalermsook, Rajesh Chitnis, Marek Cygan, Guy Kortsarz, Daniel Reichman, Igor Shinkar, Suprovat Ghoshal, Euiwoong Lee, Andreas Feldmann, Jason Li, Michal Wlodarczyk, Anupam Gupta, Aravindan Vijayaraghavan, Piotr Faliszewski, and Krzysztof Sornat. I will miss (and, in some cases, have already missed) the times we spend trying out the craziest of ideas together!

I would also like to thank all my fellow theory students and postdocs at Berkeley for their friendship, our shared joyous moments–and their occasional late night companies at Soda Hall: Lynn Chua, Akshay Srinivasan, Aviad Rubinstein, Tselil Schramm, Di Wang, Jonah Brown-Cohen, Fotis Iliopoulos, Jingcheng Liu, Peihan Miao, Frank Ban, Siqi Liu, Seri Khoury, Grace Dinh, Aaron Schild, Tarun Kathuria, Sidhanth Mohanty, Chinmay Nirkhe, Arun Ganesh, Morris Yau, Elizabeth Yang, Richard Zhang, Manuel Sabin, Nick Spooner, Rotem Arnon-Friedman, Sam Hopkins, Antonio Blanca and Sam Wong. Without you guys, this journey would not have been nearly as fun, interesting and enjoyable!

Lastly, I thank my family, especially mom and dad, and my girlfriend, Palmy, for their unwavering love and support. I am most fortunate to have all of you with me throughout my life.

Chapter 1

Introduction and Overview

The most basic question regarding any computational problem is whether it admits an *efficient* algorithm. On this front, the theory of *NP-completeness* [Coo71; Kar72; Leo73], arguably one of the most important concepts in computer science, allows us to collectively explain why thousands of computational problems arising in wide variety of fields in science and engineering are likely to be computationally intractable. However, while powerful, the notion of "efficiency" of algorithms used in NP-completeness—whether their running times are polynomial in the input size—still falls short of providing satisfactory explanation to the complexity of a number of fundamental problems. For one, it fails to explain why some problems admit quasi-polynomial time algorithms, but yet do not seem to be solvable in polynomial time. Furthermore, while algorithms with running times $O(n)$ and $O(n^2)$ are both classified as efficient in this notion, the former would finish in seconds whereas the latter could take years on an input of size say 1GB, a scenario that has become increasingly common in the age of big data. Such blind spots have led to a relatively new area of *fine-grained complexity theory*, which seeks to understand the computational complexity of problems beyond whether they can be solved in polynomial time.

The aim of this thesis is to advance our understanding of optimization problems through the lens of fine-grained complexity and *approximation algorithms*; these are algorithms that are allowed to output estimated solutions rather than exact ones. Previous works have demonstrated that such a relaxation can lead to a drastic change in computational complexity: some NP-hard problems admit polynomial time algorithms with good approximation guarantees, and some $O(n^2)$-time problem can be approximated well in $O(n)$ time. However, this is not the case for every problem, as some problems remain intractable even when approximate solutions are allowed. That is, for certain approximation ratios, some NP-hard problems still do not admit polynomial time algorithms, and some $O(n^2)$-time problems still do not admit $O(n)$-time algorithms. The theory of probabilistic checkable proofs (PCP)–which has by now been developed into the field of *hardness of approximation*–provides justifications for the first type of inapproximability: indeed, many NP-hard problems remain NP-hard even when approximation is allowed. On the other hand, until the past few years, the second (fined-grained) phenomenon remained largely unexplained. This dissertation continues this latter body of work, and is divided into three parts, based on the complexity of the problems tackled: (I) problems between P and NP, (II) parameterized problems, and

(III) problems in P. Below we provide overviews of each section. To keep the discussion at a high level, we will be mostly informal; all notations and results will be formalized later on in this thesis.

1.1 Part I: Problems Between P and NP

First, as mentioned earlier, we consider problems which admits quasi-polynomial time (approximation) algorithms. These problems are unlikely to be NP-hard, as otherwise all problems in NP would be solvable in quasi-polynomial time, a scenario considered unlikely by most complexity theorists. So how, then, can we justify the nature of these problems that lie "between P and NP"?

Similar to how *polynomial-time reductions* lie at the heart of the theory of NP-completeness, *subexponential-time reductions* play a central role in establishing computational barriers for problems between P and NP. Suppose, for instance, that we would like to show that a problem $\mathscr{A}$ cannot be solved in $N^{o(\log N)}$ time, where N is the size of the input. One way to do this is to reduce an instance of 3SAT with n variables to an instance of $\mathscr{A}$ of size $N = 2^{O(\sqrt{n})}$. Now, if we had an $N^{o(\log N)}$-time algorithm for $\mathscr{A}$, then this algorithm would have solved 3SAT in time $(2^{O(\sqrt{n})})^{o(\sqrt{n})} = 2^{o(n)}$; the latter is believed to be unlikely, a belief formalized under the name Exponential Time Hypothesis (ETH)[1] [IP01; IPZ01]. In other words, assuming ETH, we have provided a matching running time lower bound for problem $\mathscr{A}$.

Such a reduction was arguably first pioneered in the context of hardness of approximation by Aaronson, Impagliazzo and Moshkovitz [AIM14]; they dub the reduction "birthday repetition", a name that will become clear shortly. Their reduction has since inspired hardness between P and NP for many problems, including Densest k-Subgraph with perfect completeness [Bra+17; Man17a], Nash Equilibrium and related problems [BKW15; Rub17b; BPR16; Rub16; Bha+16b; DFS16], Community Detection [Rub17a] and VC Dimension [MR17b]. Indeed, the first part of our thesis can be viewed as a study of the power of (variants of) birthday repetition.

1.1.1 Birthday Repetition Theorem and Dense CSPs

We start in the first chapter by considering the original birthday repetition construction from [AIM14]. The reduction is most intuitive when described in terms of (one-round) *two-prover games*. A two prover game $\mathcal{G}$ consists of

- Finite sets of questions X, Y and corresponding answer sets Σ_X, Σ_Y.
- A distribution $\mathcal{Q}$ over pairs of questions $X \times Y$.
- A verification function $P : X \times Y \times \Sigma_X \times \Sigma_Y \to \{0, 1\}$.

The game is played as follows: the verifier picks a random pair of questions (x, y) according to the distribution $\mathcal{Q}$, and sends x to the first prover and y to the second prover. The provers then respond back with answers $\sigma_x \in \Sigma_X$ and $\sigma_y \in \Sigma_Y$; the verifier accepts if the predicate $P(x, y, \sigma_x, \sigma_y)$

[1] See Hypothesis 1 for a more formal statement of ETH.

evaluates to one and rejects otherwise. The goal of the provers is to select a strategy that achieves the highest possible acceptance probability; this probability is referred to as the *value* of the game.

Two-prover games and, more specifically, a special class of two-prover games known as the *projection games* are the starting points for reductions in a large body of hardness of approximation results. In fact, the PCP Theorem [Aro+98; AS98] is equivalent to the NP-hardness of the following problem: given a game, distinguish whether its value is one, or is at most 0.99.

Interestingly, the two-prover games generated by the PCP Theorem is usually "sparse", in the sense that the support of $\mathcal{Q}$ is very small[2] compared to $|X| \cdot |Y|$. It turns out that this is not a coincidence: the "dense" case is much easier to approximate. Specifically, when $\mathcal{Q}$ is the uniform distribution on $X \times Y$ (for which the game is said to be a *free game*), the problem of distinguishing whether a given free game is satisfiable or whether its value is at most $(1 - \varepsilon)$ can be solved in $N^{O\left(\frac{\log N}{\varepsilon}\right)}$ time [AIM14][3].

In this light, Aaronson et al.'s birthday repetition can be viewed as a reduction from any two-prover game to a free game that, when initialize with appropriate parameters, yields the (almost) matching $N^{\widetilde{\Omega}(\log N)}$ running time lower bound for approximating the value of free games. For parameters $k, \ell \in \mathbb{N}$, the $(k \times \ell)$-birthday repetition $\mathcal{G}^{k \times \ell}$ of a two-prover game $\mathcal{G}$ consists of

- The set of questions in $\mathcal{G}^{k \times \ell}$ are $\binom{X}{k}$ and $\binom{Y}{\ell}$ respectively, i.e., each question is a subset $S \subseteq X$ of size k and subset $T \subseteq Y$ of size ℓ.
- The distribution over questions is the uniform product distribution over $\binom{X}{k} \times \binom{Y}{\ell}$.
- The verifier accepts if, for every pair of $(x, y) \in S \times T$ such that (x, y) form a valid pair of questions in $\mathcal{G}$, i.e., $(x, y) \in \operatorname{supp}(\mathcal{Q})$, the answers to x and y are accepted in $\mathcal{G}$.

Notice that, for small value of k, ℓ, say $k = \ell = 1$, the game $\mathcal{G}^{k \times \ell}$ is "almost trivial" because, for most of the pairs S, T, there will be no $x \in S, y \in T$ such that (x, y) forms a valid pair of questions in the original game $\mathcal{G}$. This means that, on such (S, T), the verifier for $\mathcal{G}^{k \times \ell}$ always accepts.

However, the situation becomes interesting as soon as $k, \ell = \Omega\left(\sqrt{N}\right)$. In this regime, the expected number of valid (x, y) for a random pair S, T is $\Omega(1)$. This is also a good point to note that this resembles the situation of the "birthday paradox": if there are $\sqrt{D}$ people whose birthdays are independently identically uniformly sampled from $\{1, \dots, D\}$, then the expected number of pairs of people sharing the same birthday is $\Theta(1)$. This indeed leads Aaronson et al. [AIM14] to give the construction the name "birthday repetition".

Under a mild non-degeneracy condition on $\mathcal{G}$, the above expectation statement can be turned into a probabilistic guarantee that, for most S, T, there exists at least one valid $(x, y) \in S \times T$. In other words, the verifier checks at least one constraint from the original game $\mathcal{G}$. Intuitively, this should mean that finding a good strategy for the new game $\mathcal{G}^{k \times \ell}$ is "not easier" than that of the original game $\mathcal{G}$. The main contribution of [AIM14] is to confirm this intuition, by showing that the value of $\mathcal{G}^{k \times \ell}$ is no more than the value of $\mathcal{G}$ plus $O(\sqrt{k\ell/N})$.

[2] In fact, $|\operatorname{supp}(\mathcal{Q})|$ can be assumed to be linear in $|X| + |Y|$ [Tre01; Din07].

[3] Here, we use N to denote the size of the instance, i.e., $N = |X||Y||\Sigma_X||\Sigma_Y|$.

In doing so, they immediately arrives at the $N^{\widetilde{\Omega}(\log N)}$ running time lower bound for approximating the value of free games: starting with a hardness of approximation result for approximating the value of a two-prover game $\mathcal{G}$ of size N, we can consider the birthday repetition game $\mathcal{G}^{k \times \ell}$ with $k = \ell = \Theta(\sqrt{N})$. The latter is of size roughly $\widetilde{N} = \binom{N}{\sqrt{N}} = 2^{O(\sqrt{N} \log N)}$. If we can approximate the value of $\mathcal{G}^{k \times \ell}$ well in time $\widetilde{N}^{o\left(\frac{\log \widetilde{N}}{\log \log \widetilde{N}}\right)}$, then we can approximate the value of the original game $\mathcal{G}$ in time $2^{o(N)}$, which would violate ETH.

As readers might have already noticed, the above paragraph glosses over a subtle but important fact: we have to start with $\mathcal{G}$ whose value is hard to approximate, meaning that we have to evoke the PCP Theorem to begin with. Hence, we have to also taken in the account the "size blow-up" in the PCP. Fortunately, there are known PCP constructions with small blow-ups [BS08; Din07; MR10]. Specifically, Dinur's PCP [Din07] can produce a two-prover game (or alternatively an instance of Gap-3SAT[4]) of size $N = n \operatorname{polylog}(n)$ when starting with a 3SAT formula of size n. As a result, the described approach still gives hardness of $\widetilde{N}^{\widetilde{\Omega}(\log \widetilde{N})}$ for approximating the value of free games.

While Aaronson et al.'s work [AIM14] appeared to have resolved the complexity of birthday repetition, there were in fact a few open questions remained. The main one, which was highlighted in [AIM14], is whether birthday repetition can *decrease* the value of a game (i.e. amplify the hardness gap). In particular, since the verifier in $\mathcal{G}^{k \times \ell}$ checks $\Theta(k\ell/N)$ pairs of questions from $\mathcal{G}$ in expectation, it was suggested in [AIM14] that the value of $\mathcal{G}^{k \times \ell}$ should decrease exponentially in $\Theta(k\ell/N)$. The main contribution of our first chapter is to confirm this conjecture. In doing so, we give an almost matching running time versus approximation ratio tradeoff curve for the problem of approximating the value of free games. Roughly speaking, we show that, to achieve an approximation ratio of $N^{1/i}$, one needs $N^{\widetilde{\Omega}(i)}$ time, which is tight. On a more technical level, our proof relies in the following fact from extremal graph theory: any dense graph must contain many copies of (small) complete bipartite subgraphs (bicliques). With this in mind, we carefully bound the number of bicliques in the "acceptance graph" for $\mathcal{G}^{k \times \ell}$. This technique turns out to be useful in the next chapter as well.

We also provide several additional results. For instance, we show a similar lower bound in terms of strong SDP relaxations (i.e. the Lasserre hierarchy) and we give an approximation algorithm with similar running time to that of [AIM14] that works for a more general case of dense constraint satisfaction problems (CSPs) and even when the instance might not be satisfiable.

1.1.2 Densest k-Subgraph (with Perfect Completeness)

In the second chapter, we consider the DENSEST k-SUBGRAPH (DkS) *with perfect completeness* problem, which can be viewed as an approximate version of the classic k-CLIQUE problem. In DkS with perfect completeness, we are given a graph G with a promise that it contains a k-clique. The goal is to find a subgraph of size k that is as dense as possible. This problem again admits

[4] In the Gap-3SAT problem, we are given a 3CNF formula and the goal is to distinguish between the case that it is satisfiable and the case where every assignment violates at least 1% of clauses.

a quasi-polynomial time approximation scheme. That is, there is an $n^{O\left(\frac{\log n}{\varepsilon}\right)}$-time algorithm that can find a k-vertex subgraph of density[5] $(1 - \varepsilon)$ [FS97; Bar15].

There is a straightforward, albeit incorrect, reduction from free games to DkS with perfect completeness: given a free game with question sets X, Y and answer sets Σ_X, Σ_Y, create a graph whose vertex set is $(X \times \Sigma_X) \cup (Y \times \Sigma_Y)$, and two vertices $(x, \sigma_x) \in X \times \Sigma_X$ and $(y, \sigma_y) \in Y \times \Sigma_Y$ are connected iff the verifier accepts (σ_x, σ_y) when the pair of questions (x, y) is drawn. The pairs of vertices whose questions are from the same set are always linked. Such a graph is sometimes referred to as the *labelled extended graph* of the game. It is obvious that, if the game is satisfiable, then there is an $(|X| + |Y|)$-clique in the graph. Unfortunately, it is possible that the graph has a dense subgraph, even when the value is very small; this reduction hence fails.

Remarkably, however, Braverman et al. [Bra+17] show that, if instead of starting from an arbitrary free game we start from a birthday repetition game $\mathcal{G}^{k \times \ell}$ with $k, \ell = \Omega(\sqrt{N})$, then the reduction in fact works, in the sense that any $(1 - \varepsilon)$-dense subgraph of size $(|X| + |Y|)$ (for a small constant $\varepsilon > 0$) translates back to a strategy of $\mathcal{G}$ with high value. Similar to before, their result immediately implies that $(1 - \varepsilon)$-approximation of DkS with perfect completeness requires $n^{\widetilde{\Omega}(\log n)}$ time assuming ETH, which nearly matches the aforementioned algorithms.

In light of our result in the previous section, it is natural to ask whether we can achieve "gap amplification" effect here as well. That is, can we prove hardness for DkS with perfect completeness with large factors? Unfortunately, there is a counterexample showing that this construction can achieve a factor of at most two (see the appendix of [Man17a]). The main contribution of this chapter is to overcome this barrier and achieve inapproximability ratios that are almost polynomial. Specifically, we show that, assuming ETH, no polynomial-time algorithm can approximates DkS with perfect completeness to within $n^{1/(\log \log n)^c}$ factor of the optimum. We also provide a finer trade-off between the approximation ratio and running time, although this is not yet tight.

Due to the mentioned counterexample, we need to modify the reduction to make our proof work. Roughly speaking, instead of starting with two prover games, we have to start with boolean CSPs and, instead of picking sets of "questions", we pick sets of variables instead. As alluded to above, the key step of our proof is to bound the number of (small) bicliques in the constructed graph, which is more challenging in this case than in the previous chapter because here we are considering the labelled extended graph as opposed to the acceptance graph before.

Interestingly, while our proof is tailored for the special case of DkS with perfect completeness, it does give the best known hardness for DkS, in which no promise of k-clique existence is given. For the general DkS problem, Bhaskara et al.'s state-of-the-art algorithm for the problem achieves only $O(n^{1/4+\varepsilon})$ approximation ratio and it is believed that the problem is hard to approximate to within a large (possibly even polynomial) factor. Despite this, previous attempts at proving hardness of approximation, including those under average case assumptions, fail to even come close to a polynomial ratio; the best ratios ruled out under any worst case assumption and any average case assumption were only any constant [RS10] and $2^{O(\log^{2/3} n)}$ [Alo+11] respectively. Thus, our results also present the best inapproximability factor so far for DkS.

[5] The density of an n-vertex graph for $n \geqslant 2$ is the number of its edges divided by $\binom{n}{2}$, which is a number between zero and one (inclusive).

1.1.3 VC Dimension and Littlestone's Dimension

The last chapter of the first part studies the complexity of (approximating) two fundamental quantities in learning theory: VC Dimension and Littlestone's Dimension. These dimensions capture the number of samples needed in the PAC learning model and the mistake bounds in the online learning model respectively. We consider the model in which a concept class is given explicitly in the input (as a binary matrix whose (x, C)-th entry is 1 iff element x belongs to concept C), and we would like to compute the dimensions. It is not hard to see that both quantities can be computed *exactly* in time $N^{O(\log N)}$, where N denote the size of the input (i.e. matrix). Assuming the randomized Exponential Time Hypothesis, we prove nearly matching lower bounds on the running time, that hold even for approximation algorithms for small constant factors.

It should be noted that, while the constructions in this chapter are inspired by the aforementioned birthday repetition, there are additional challenges, and the proof techniques also diverge quite significantly from the previous ones. However, this might not be completely coincidental: while birthday repetition has found applications for very different problems, these problems all share essentially the same quasi-polynomial time *algorithm*. The bottleneck in those problems is a bilinear optimization problem $\max_{u,v} u^\top Av$, which we want to approximate to within a (small) constant additive factor. To do this, it suffices to find an $O(\log n)$-sparse sample $\hat{v}$ of the optimal v^*; the algorithm enumerates over all sparse $\hat{v}$'s [LMM03; Aro+12; Bar15; Che+15b]. Indeed, the algorithms for both free games and DkS with perfect completeness are of this form.

In contrast, the problems we consider here have completely different quasi-polynomial time algorithms: for VC Dimension, it suffices to simply enumerate over all $\log |\mathcal{C}|$-tuples of elements (where $\mathcal{C}$ denotes the concept class and $\log |\mathcal{C}|$ is the trivial upper bound on the VC dimension) [LMR91]. Littlestone's Dimension can be computed in quasi-polynomial time via a recursive "divide and conquer" algorithm (See Section 5.4.4). We hope that our hardness in this section serves as a supporting evidence that the birthday repetition framework might find more applications for a wider range of problems in the future.

1.2 Part II: Parameterized Problems

The second part of this dissertation shifts the focus to the so-called *parameterized complexity* (or *multivariate complexity*), an area which emerged in the late eighties and early nineties to provide yet another approach to tackle NP-hard problems. To illustrate, let us consider three classic NP-hard problems from [Kar72]: VERTEX COVER, CLIQUE and DOMINATING SET (DOMSET). While all are NP-hard, their complexity seems to differ if we are looking to find a solution of small size k. In particular, whereas no $N^{o(k)}$-time algorithm is known for either CLIQUE or DOMSET, this is possible in time $2^k \cdot N^{O(1)}$ for VERTEX COVER; such an algorithm can be much faster than the trivial $N^{O(k)}$-time algorithm. Motivated by this, a parameterized problem with parameter k is said to be *fixed-parameter tractable (FPT)* if it can be solved in $f(k) \cdot N^{O(1)}$ time for some (computable) function f. This serves as the notion of "efficient algorithms" for parameterized complexity, in the same way that polynomial-time algorithms do in the theory of NP-completeness.

Since its inception, parameterized complexity has provided a fruitful platform for both algorithmic and intractability results. Turning back to k-CLIQUE and k-DOMSET once again, the lack of FPT algorithms for them can be explained: they are complete for the classes W[1] and W[2] respectively [DF95a; DF95b]. Hence, assuming these classes do not collapse to FPT, the two problems are intractable in this parameterized notion. In fact, under ETH, a stronger lower bound is known: not even $f(k) \cdot N^{o(k)}$-time algorithm exists for k-CLIQUE and k-DOMSET [Che+04; Che+06]. In other words, the trivial $N^{O(k)}$-time algorithm is essentially the best possible (up to the constant in the exponent).

Approximation has been suggested as a way to overcome these parameterized complexity barriers. However, even when considering approximation algorithms, no "non-trivial" result is known. On the other hand, despite the strong lower bounds established for exact algorithms, few inapproximability results were known for parameterized problems, until the past few years.

To understand the barrier in proving hardness of approximation for parameterized problems, let us first describe the standard strategy in proving tight running time lower bounds (e.g. from [Che+04; Che+06]). These reductions can be thought of as taking an instance of 3SAT with n variables and produces an instance of k-CLIQUE or k-DOMSET of size $N = 2^{O(n/k)}$. If we can solve either of these problems in $f(k) \cdot N^{o(k)}$ time, then we can also solve 3SAT in $2^{o(n)}$ time, violating ETH.

Suppose we try to take a similar path to prove hardness of approximation. The most natural approach would be to first apply the PCP theorem so that we have hardness of the gap version of 3SAT problem; using the best known PCP [Din07], this Gap-3SAT consists of $n' = n \cdot \text{polylog}(n)$ variables. Then, we can apply the reductions mentioned above to transform the Gap-3SAT instance to a k-CLIQUE or k-DOMSET instance. This gives an instance of size $N = 2^{O(n'/k)} = 2^{n \cdot \text{polylog}(n)/k}$. However, N is already super exponential and does not give any lower bound at all!

With this obstacle in mind, there seems to be two paths going forward: first, we can try to produce the gap in hardness of approximation via something different than the PCP Theorem. The first chapter of this second part takes this route and in the process obtain strong inapproximability results for k-DOMSET, which resolves a long-standing open question in parameterized complexity.

Second, we can just make a stronger assumption, that Gap-3SAT itself takes exponential time! This assumption, now known under the name Gap-ETH, composes quite nicely with existing reductions. Indeed, now that there is no polylog(n) size blow-up from the PCP Theorem, applying the current known reductions to Gap-ETH already implies that k-CLIQUE is hard to approximate to within a constant factor [Bon+15]. The main challenge here is thus the issue of *gap amplification*, e.g., how can we prove hardness of large factor for k-CLIQUE (or other problems). This is a main focus of this line of works, which appears in Chapters 7, 8, 9 and 10 in this dissertation.

1.2.1 Inapproximability of k-Dominating Set (via Distributed PCP)

Our results for k-DOMSET (and also k-CLIQUE in the next chapter) can be best stated via the notion of *total FPT inapproximability*. To motivate this notion, recall that the greedy algorithm for k-DOMSET achieves an approximation ratio of $(\ln n + 1)$ [Joh74; Chv79; Lov75; Sri95; Sla96]. In the setting of parameterized complexity, this can be quite bad: since we think of k as much smaller than n, then overhead factor of $O(\ln n)$ can even be unbounded in terms of k. The question, which has been asked multiple times in literature (see e.g. [DFM06; CGG06; Dow+08; CH10; DF13]), is whether we can get an $g(k)$-approximation algorithm for k-DOMSET in FPT time for some function g (i.e. even with say $g(k) = 2^{2^k}$).

The main contribution of this chapter is a negative answer to this question: we show that it is W[1]-hard to approximate k-DOMSET to within $g(k)$ factor for any function g. Furthermore, we strengthen the running time lower bounds under ETH and Strong ETH (SETH) to $f(k) \cdot n^{\Omega(k)}$ and $f(k) \cdot n^{k-\varepsilon}$ respectively; once again, these apply for any $g(k)$-approximation algorithm for k-DOMSET. In other words, there is little one can save in the running time compared to the trivial algorithm, even when approximation is allowed. Previously, the best known hardness of approximation of k-DOMSET due to Chen and Lin [CL16] rules out only any constant factor and $O(\log^{1/4} k)$ factor under W[1]-hardness and ETH respectively.

As touched upon briefly in our above discussion, our proof uses a different way to produce gap rather than the traditional approach of the PCP Theorem. In particular, we generalize the ***Distributed PCP*** framework of Abboud, Rubinstein and Williams [ARW17a] for proving hardness of approximation in P, to the context of parameterized complexity. Roughly speaking, our generalized view is that, if we start with a hypothesis that can be written in a certain form (see Section 6.3), then, to prove hardness of approximation for a variant of the label cover problem called MAXCOV, it suffices to give an "efficient" protocol for a certain multi-party communication problem. The hardness of approximation for k-DOMSET is then established by reducing from the label cover problem; such reductions were known in literature [Fei98; Cha+17] (see Section 2.11).

Stating the above connection/framework (even informally) requires a few additional notations and hence it will be left out from the introduction; for interested readers, Section 6.1 provides a brief overview of the framework without too much notational overhead.

1.2.2 Inapproximability from Gap-ETH I: k-Clique

Next, we consider the k-CLIQUE problem. For maximization problems such as k-CLIQUE, the notion of total inapproximability becomes slightly different. Specifically, it is now obvious to get a k-approximation for k-CLIQUE, by just outputting one vertex! As a result, such a maximization problem is said to be *totally inapproximable* if there is no $o(k)$-approximation in FPT time. In this chapter, we show that this is the case for k-CLIQUE. However, we need the stronger Gap-ETH assumption for this result, as opposed to just $W[1] \neq$ FPT or ETH in the previous chapter.

On a more technical level, the proof once again proceeds by first showing hardness of MAXCOV, with a stronger requirement that the constraints have a "projection property". Unfortunately, such a property does not hold for instances created in the previous chapter. However, we can construct a desirable instance relatively simply from Gap-ETH. The hardness for k-CLIQUE follows immediately via a classic reduction from the NP-hardness of approximation literature [Fei+91].

Apart from k-CLIQUE, we also consider the problem of Maximum Induced Subgraph with Hereditary Property (e.g. Maximum Induced Planar Subgraph). For this problem, Khot and Raman [KR00] prove a dichotomy that, for a specific property, the problem is either FPT or W[1]-hard. Here we extend this to show that, for the "hard" properties, the problem is even totally inapproximable, assuming Gap-ETH. An interesting aspect of our reduction (from k-CLIQUE) is that it is noticeably simpler than that of Khot and Raman; this demonstrates that having a gap in the starting problem can help simplify the reduction. Such a theme will come up again later in the thesis.

1.2.3 Inapproximability from Gap-ETH II: k-Biclique and Densest k-Subgraph

While the previous two chapters rely on hardness of (variants of) label cover as starting points for hardness of approximation, we take a different route in this chapter; our starting point will instead be the reduction from Chapter 4 (i.e. Section 1.1.2 above). As we mentioned above, the soudness in Chapter 4 proceeds by arguing that the constructed graph contains few small bicliques. Hence, if we subsample the graph by keeping each vertex independently at random with an appropriate probability, then, in the soundness case, the small bicliques should all disappear. It turns out that such an probability is still large enough that, in the completeness case, we are left with a large clique. In other words, this implies that the "CLIQUE-VS-BICLIQUE" problem is "totally FPT inapproximable". This problem can then be easily reduce to k-BICLIQUE, by "bipartizing" the graph.

It is not hard to observe that the total inapproximability of k-BICLIQUE implies some hardness of approximation for DENEST k-SUBGRAPH, where k is the parameter. In particular, a classic result of Kővári, Sós and Turán [KST54] says that any k-vertex graph which is t-biclique-free has density at most $k^{-\Omega(1/t)}$. Now, the total inapproximability of k-BICLIQUE implies that we cannot distinguish in FPT time a graph containing k-biclique and one which is say $(\log \log k)$-biclique-free. Then, in the former case we have a k-vertex subgraph that has density more than a half, while in the latter any k-vertex subgraph has density at most $k^{-\Omega(1/\log\log k)}$. This gives hardness of approximating DENSEST k-SUBGRAPH to within a factor of $k^{O(1/\log\log k)}$. Of course, $\log\log k$ can be replaced with any function that goes to infinity as $k \to \infty$, meaning that this approach can gives an inapproximability for DENSEST k-SUBGRAPH to within a factor of $k^{o(1)}$.

It should be noted however that this does not give "total FPT inapproximability" for DENEST k-SUBGRAPH, unlike our earlier results so far. Indeed, unfortunately, we do not manage to achieve total FPT inapproximability for any of the problems from this point onwards, although for some problems we still get pretty strong inapproximability results.

1.2.4 Inapproximability from Gap-ETH III: Parameterized 2-CSPs with Strong Soundness (via Agreement Testing Theorem)

In an attempt to prove an even stronger hardness for DENSEST k-SUBGRAPH and related problems, we consider a harder problem (i.e. easier to prove hardness) called PARAMETERIZED 2-CSPS. In this context, it is easiest to described the problem in terms of the *colorful* version of DENEST k-SUBGRAPH. Namely, in PARAMETERIZED 2-CSP, we are given a graph G and a partition of its vertices $V(G) = V_1 \cup \cdots \cup V_k$, the goal is to find k vertices each from a different partition that maximizes the number of edges they induced. Here k is once again the parameter. Note that this problem is exactly the same as DkS except that the vertices have to come from different partitions.

We show that the PARAMETERIZED 2-CSP problem is hard to approximate to within a factor of $k^{1-o(1)}$ assuming Gap-ETH. Interestingly, our result also gives the best known hardness of approximation in terms of k, even for the non-parameterized version. In this regime, it is strongly believe that the problem is NP-hard to approximate to within $k^{\Omega(1)}$ factor, but no such result is known; in fact, proving such an NP-hardness result seems quite challenging as it would resolve a well-known conjecture in the theory of PCP called the Sliding Scale Conjecture [Bel+93]. Please refer to Chapter 9 for discussions regarding the conjecture.

On a technical level, the main component in our proof is a "combinatorial agreement testing theorem", which can also be viewed as a derandomized direct product test. In particular, the question is of the following form: given boolean functions $f_1, \dots, f_k$ on domains $S_1, \dots, S_k \subseteq [n]$ and suppose that δ fraction of the pairs agree on their intersections, can we recover a global function $g : [n] \to \{0, 1\}$ that "roughly" agrees with a "large" ($\approx \delta k$) fraction of the given functions? Here $S_1, \dots, S_k$ are of size $\Omega_k(n)$ and are "random looking" subsets. We show that such a statement holds, even when δ is as small as $1/k^{1-o(1)}$ (Theorem 9.9), which is roughly optimal since nothing non-trivial can be said when $\delta \leqslant 1/k$. To the best of our knowledge, no prior derandomized direct product tests work for such a low agreement (when measure in terms of k).

Our agreement testing theorem almost immediately yields the aforementioned hardness for 2-CSPs, by taking $S_1, \dots, S_k$ to be the subsets of the variables of the starting Gap-3SAT formula, let the i-th partition contains every function $f : S_i \to \{0, 1\}$, and let the constraints (i.e. edges) check whether the two functions agree and that they do not violate any clauses. In the completeness case, it is clear that we can pick k functions that are the restrictions of the global satisfying assignment; this yields a k-clique. On the other hand, in the soundness case, our agreement testing theorem implies that, if $\delta \geqslant 1/k^{1-o(1)}$ fraction of pairs of the k selected functions agree, we can recover $g : [n] \to \{0, 1\}$ that agree with many of the f_i's. When setting parameters appropriately, a simple counting argument implies that g must satisfy almost all clauses, which is a contradiction.

There are two consequences of our hardness of approximation of PARAMETERIZED 2-CSPS:

- First, due to a known reduction [DK99; CFM18], our result implies hardness of approximation for the DIRECTED STEINER NETWORK (DSN) problem with factor $k^{1/4-o(1)}$ where k denotes the number of demand pairs (and k is the parameter). This is the first $k^{\Omega(1)}$ hardness for the problem (even in the non-parameterized regime).

- Secondly, we show, by rephrasing our 2-CSP instance in terms of label cover with a projection property and using the known reduction from label cover the set cover [Fei98], that the k-UNIQUE SET COVER is hard to approximate to within a factor of $k^{1/2-o(1)}$. This hardness will be useful in the next chapter.

Unfortunately, we still do not know how to translate the techniques developed for PARAMETERIZED 2-CSPs back to DkS, and even proving $k^{0.001}$-factor inapproximability for the latter remains open.

1.2.5 Inapproximability from Gap-ETH IV: k-Even Set and k-Shortest Vector

In the next chapter, we consider the k-EVEN SET and k-SHORTEST VECTOR problems. The k-EVEN SET problem is a parameterized variant of the MINIMUM DISTANCE PROBLEM of linear codes over $\mathbb{F}_2$, which can be stated as follows: given a generator matrix $\mathbf{A}$ and an integer k, determine whether the code generated by $\mathbf{A}$ has distance at most k, or in other words, whether there is a nonzero vector $\mathbf{x}$ such that $\mathbf{Ax}$ has at most k nonzero coordinates.

In the k-SHORTEST VECTOR problem, we are given a lattice whose basis vectors are integral and an integer k, and the goal is to determine whether the norm of the shortest vector (in the ℓ_p norm for some fixed p) is at most k.

The question of whether k-EVEN SET and k-SHORTEST VECTOR are fixed-parameter tractable has been repeatedly raised in literature; in fact, they were two of the few remaining open questions from the seminal book of Downey and Fellow [DF99]. We stress here that the parameterized complexity of these two problems were open even for *exact* algorithms. In this chapter, we negatively answer this question by showing that, assuming Gap-ETH, there are no FPT algorithms for the two problems. Our lower bound holds even against approximation algorithms; the inapproximability ratios we can rule out for k-EVEN SET is *any* constant factor, whereas for k-SHORTEST VECTOR we only rule out *some* constant factor.

Similar to the NP-hardness of approximation proofs for both problems [DMS03; Kho05], our first step is to show that their non-homogenous counterpart, the k-NEAREST CODEWORD and k-NEAREST VECTOR problems, are hard to approximate to within large factor. This is established via a known reduction of Arora et al. [Aro+97] from k-UNIQUE SET COVER, for which we show inapproximability in the previous chapter.

The second step of the proof is to reduce from hardness of approximating k-NEAREST CODEWORD and k-NEAREST VECTOR to k-EVEN SET and k-SHORTEST VECTOR respectively. In the case of k-SHORTEST VECTOR, the same proof as Khot's NP-hardness proof [Kho05] works in the parameterized settings as well. As for k-EVEN SET, the NP-hardness of approximation reduction of Dumer, Micciancio and Sudan [DMS03] does not immediately work. While our final reduction is still heavily inspired by their reduction, we need to define a new set of properties of error-correcting codes, which are used as a gadget in the reduction. (See Section 10.3.1.) We then show that a known family of codes (in particular, the BCH code) satisfies these properties.

Once again, we stress that it is crucial to have *hardness of approximation* of k-NEAREST CODEWORD and k-NEAREST VECTOR for the reductions to work. This brings us back to the point brought up earlier in Section 1.2.2 that starting with hardness of approximation can help make the reductions easier. Indeed, even if one does not care about approximation algorithms at all, obtaining hardness of approximation might still be useful in facilitating subsequent reductions, as is demonstrated here in the case of k-EVEN SET and k-SHORTEST VECTOR.

1.3 Part III: Problems in P

1.3.1 Closest Pair and Maximum Inner Product

Finally, we consider problems within P. We mentioned above that the Distributed PCP framework was developed by Abboud et al. [ARW17a] to prove hardness of approximation of problems in P. To be more specific, the canonical problems that they prove inapproximability results for are the BICHROMATIC MAXIMUM INNER PRODUCT (BMIP) and the BICHROMATIC CLOSEST PAIR (BCP) problems [ARW17a; Rub18], which serve as the sources of other hardness results shown in their paper(s). In both problems, we are given two sets $A, B \subseteq \{0,1\}^d$ of n points in d dimensions. The goal of BCP (resp. BMIP) is to find a pair of points $\mathbf{a} \in A, \mathbf{b} \in B$ that minimizes (resp. maximizes) their distance $\|\mathbf{a} - \mathbf{b}\|_2$ (resp. inner product $\langle \mathbf{a}, \mathbf{b} \rangle$). Here we think of d as $n^{o(1)}$. Both problems can be trivially solved in $O(n^{2+o(1)})$ time. The results of [ARW17a; Rub18] states that, in $O(n^{2-\varepsilon})$ time, BCP and BMIP cannot even be approximated to within $(1+\delta)$ and $2^{\log^{1-o(1)}(n)}$ factors respectively where $\delta > 0$ is a positive constant depending on ε. Their results and our results discussed below hold under the Strong Exponential Time Hypothesis (SETH); see Hypothesis 2.

As some readers might have already noticed, the "bichromatic" in the problems' names come from the fact that there are two sets in the input, i.e., one for each "color", and we are only allowed to pick one point from each color. These are different than the (originally studied) "monochromatic" versions of the problems, where the input is just a single set and we can pick any two (distinct) points from the set. Interestingly, despite the aforementioned strong inapproximability results for BCP and BMIP, it was not even known whether (monochromatic) CLOSEST PAIR (CP) and MAXIMUM INNER PRODUCT (IP) can be solved *exactly* in subquadratic time. This was indeed highlighted as an open question in several recent works [ARW17b; Wil18a; DKL18].

In this penultimate chapter, we partially answer this question by showing that for every $p \in \mathbb{R}_{\geqslant 1} \cup \{0\}$, under SETH, for every $\varepsilon > 0$, the following holds:

- No $O(n^{2-\varepsilon})$-time algorithm can solve CP in $d = (\log n)^{\Omega_\varepsilon(1)}$ dimensions in the ℓ_p metric.
- There exists $\delta = \delta(\varepsilon) > 0$ such that no $O(n^{1.5-\varepsilon})$-time algorithm can approximate CP to a factor of $(1+\delta)$ in $d = O_\varepsilon(\log n)$ dimensions in the ℓ_p-metric.
- No $O(n^{2-\varepsilon})$-time algorithm can approximate MIP to within a factor of $2^{\log^{1-o(1)}(n)}$ (for $d = n^{o(1)}$ dimensions).

In particular, our first result is shown by establishing the computational equivalence of the BICHROMATIC CLOSEST PAIR problem and the (monochromatic) CLOSEST PAIR problem (up to n^ε factor in the running time) for $d = (\log n)^{\Omega_\varepsilon(1)}$ dimensions.

At the heart of all our proofs is the construction of a dense bipartite graph with low *contact dimension*, i.e., we construct a balanced bipartite graph on n vertices with $n^{2-\varepsilon}$ edges whose vertices can be realized as points in a $(\log n)^{\Omega_\varepsilon(1)}$-dimensional Euclidean space such that every pair of vertices which have an edge in the graph are at distance exactly 1 and every other pair of vertices are at distance greater than 1. This graph construction is inspired by the construction of locally dense codes introduced by Dumer, Miccancio and Sudan in [DMS03], which was also the inspiration/template for our hardness of k-EVEN SET in the previous chapter!

1.4 Discussion and Future Directions

Although we provide several open problems in each of the chapters, these are usually problems closely related to the study in that particular chapter. In the last chapter of this thesis (Chapter 12), we provide a more high level view of the limitations of current techniques and discuss what we feel are interesting directions to explore in the future.

1.5 Bibliographic Notes

Chapter 3 is based on a work co-authored with Prasad Raghavendra which was published at ICALP 2017 [MR17a]. However, the version in this thesis contains a significant improvement: the main birthday repetition theorem (Theorem 3.5) has better parameter dependencies and the proof is completely different from that in [MR17a]. The new proof in fact relies on the techniques originally developed for DENSEST k-SUBGRAPH in [Man17a], on which Chapter 4 is based. This chapter closely follows the conference version of [Man17a] published in STOC 2017, with the exception that the running time-vs-approximation ratio tradeoff is stated more explicitly here (see Theorem 4.2). The fifth chapter is based on a joint work with Aviad Rubinstein published in COLT 2017 [MR17b]; the changes from the conference version are minimal.

The sixth chapter is based on a work co-authored with Karthik C.S. and Bundit Laekhanukit from STOC 2018 [KLM18]. Chapters 7 and 8 are extracted from a joint work with Parinya Chalermsook, Marek Cygan, Guy Kortsarz, Bundit Laekhanukit, Danupon Nanongkai, and Luca Trevisan [Cha+17]. These three chapters follow closely to the conference versions of the papers.

The ninth chapter is based on [DM18] from ITCS 2018 co-authored with Irit Dinur. The major addition from there is an application for the hardness of UNIQUE SET COVER (Section 9.7). Chapter 10 is based on a joint work with Arnab Bhattacharyya, Suprovat Ghoshal and Karthik C. S. published at ICALP 2018 [Bha+18]. The main difference between the two versions is that here we prove hardness of k-NCP and k-NVP simply by reducing from the hardness of UNIQUE SET COVER from the previous chapter. The conference version uses a more direct reduction from 2-CSP, which yields a worse inapproximability ratio and is arguably more complicated. The proofs of hardness for k-MDP and k-SVP also contain some simplifications from a journal version in preparation (which will be a merge between [Bha+18] and a manuscript [Bon+18] of Édouard Bonnet, László Egri, Bingkai Lin and Dániel Marx).

Chapter 11 closely follows a joint work with Karthik C.S. from ITCS 2019 [KM19].

Excluded Works. While this dissertation includes a large part of my work, several papers have to be (regretfully) left out of this thesis. These include works on “traditional” approximation algorithms and hardness of approximation [MNT16; Chl+17b; AMM17; Man17b; CM18; Man19a], distributed algorithms [Bec+18], subexponential and parameterized approximation algorithms [Man19b; Gup+19; MT18], and computational social choice [MS17a; MS17b; MS19a; MS19b; FSM19; Bei+19].

Chapter 2

Notation, Preliminaries and Tools

In this section, we provide necessary preliminaries and tools that will be used in this dissertation. Before we do so, let us first define several additional notations.

2.1 Notation

For any positive integer n, we use $[n]$ to denote the set $\{1, \dots, n\}$. For two sets X and S, define X^S to be the set of tuples $(x_s)_{s \in S}$ indexed by S with $x_S \in X$. We sometimes view each tuple $(x_s)_{s \in S}$ as a function from S to X. For a set S and an integer $n \leqslant |S|$, we use $\binom{S}{n}$ to denote the collection of all subsets of S of size n. For convenience, we let $\binom{S}{0} = \{\emptyset\}$. We use $\binom{S}{\leqslant n}$ to denote $\binom{S}{0} \cup \cdots \cup \binom{S}{n}$. Moreover, let $\mathscr{P}(S) := \binom{S}{\leqslant |S|}$ denotes the power set of S.

We use $\exp(x)$ and $\log(x)$ to denote 2^x and $\log_2(x)$ respectively. poly(n), polylog(n), $\operatorname{polyloglog}(n)$ are used as a shorthand for $O(n^c), O((\log n)^c)$ and $O((\log \log n)^c)$ for some constant c respectively. Finally, $\widetilde{\Omega}(f(n))$ and $\widetilde{O}(f(n))$ are used to denote $\bigcup_{c \in \mathbb{N}} \Omega(f(n)/\log^c f(n))$ and $\bigcup_{c \in \mathbb{N}} O(f(n) \log^c f(n))$ correspondingly.

2.1.1 Graph Theoretic Notation

Unless state explicitly otherwise, graphs are used to referred to undirected unweighted graphs. For any graph G, we denote by $V(G)$ and $E(G)$ the vertex and edge sets of G, respectively. For each vertex $u \in V(G)$, we denote the set of its neighbors by $N_G(v)$; when the graph G is clear from the context, we sometimes drop it from the notation. For a subset $S \subseteq V(G)$, we use $G[S]$ to denote the subgraph of G induced by S; for convenience, we sometimes use $E(S)$ to denote the set of edges in $G[S]$, instead of the more cumbersome notion $E(G[S])$. The density[1] of a graph G on $|V(G)| \geqslant 2$ vertices is $\frac{|E(G)|}{\binom{|V(G)|}{2}}$. We say that a graph is *α-dense* if its density is α.

[1] It is worth noting that sometimes density is defined as $|E(G)|/|V(G)|$. For the DENSEST-k-SUBGRAPH problem, both definitions of density result in the same objective since $|S| = k$ is fixed. However, our notion is more convenient to deal with as it always lies in $[0, 1]$.

A bipartite graph $G = (U, V, E)$ is said to be *bi-regular* if every left vertex (in U) has the same degree, and every right vertex (in V) has the same degree. For a parameter $\tau \geqslant 1$, a bipartite graph is said to be *τ-almost-biregular* if the ratios $\frac{\max_{u \in U} \deg(u)}{\min_{u \in U} \deg(u)}$ and $\frac{\max_{v \in V} \deg(v)}{\min_{v \in V} \deg(v)}$ are at most τ.

A *clique* of G is a complete subgraph of G; sometimes we also refer to a clique as a subset $S \subseteq V(G)$ such that $G[S]$ is a clique. A *biclique* of G is a balanced complete bipartite subgraph of G. By k-biclique, we mean the graph $K_{k,k}$, i.e., a biclique where the number of vertices in each partition is k. An *independent set* of G is a subset of vertices $S \subseteq V(G)$ such there is no edge joining any pair of vertices in S. A *dominating set* of G is a subset of vertices $S \subseteq V(G)$ such that every vertex in G is either in S or has a neighbor in S. The *clique number* (resp., *independent number*) of G is the size of the largest clique (resp., independent set) in G. The *biclique number* of G is the largest integer k such that G contains $K_{k,k}$ as a subgraph. The *domination number* of G is defined similarly as the size of the smallest dominating set in G. The clique, independent and domination numbers of G are usually denoted by $\omega(G)$, $\alpha(G)$ and $\gamma(G)$, respectively. However, in this dissertation, we will refer to these numbers by $\textsc{Clique}(G), \text{MIS}(G), \textsc{DomSet}(G)$. Additionally, we denote the biclique number of G by $\textsc{Biclique}(G)$.

Moreover, for every $t \in \mathbb{N}$, we view each element of V^t as a t-size ordered multiset of V. $(L, R) \in V^t \times V^t$ is said to be a *labelled copy of a t-biclique* (or $K_{t,t}$) in G if, for every $u \in L$ and $v \in R$, $u \neq v$ and $(u, v) \in E$. The number of labelled copies of $K_{t,t}$ in G is the number of all such (L, R)'s. It is important that we distinguish between a labelled copy of t-biclique as just defined, and a *copy of t-biclique*; the latter is pair of disjoint *subsets* $L, R \subseteq V$ each of size t such that, for every $u \in L$ and $v \in R$, we have $(u, v) \in E$. Finally, we say that a graph is t-biclique-free if it does not contain a copy of t-biclique (or alternatively $\textsc{Biclique}(G) < t$).

In one occasion (Section 3.2), we also use the notion of labelled copies for unbalanced biclique as well, which is defined similar to above. Specifically, for $s, t \in \mathbb{N}$, $(L, R) \in V^s \times V^t$ is said to be a *labelled copy of a (s, t)-biclique* (or $K_{s,t}$) in G if, for every $u \in L$ and $v \in R$, $u \neq v$ and $(u, v) \in E$.

2.1.2 Distance Measures

For any $\mathbf{a} \in \mathbb{R}^N$, its ℓ_p norm is defined by $\|\mathbf{a}\|_p := \left(\sum_{i \in [N]} |a_i|^p\right)^{1/p}$ for $1 \leqslant p < \infty$. Its ℓ_0 norm, denoted by $\|\mathbf{a}\|_0$ is defined as $|\{i \in [N] : a_i \neq 0\}|$. It ℓ_∞ norm $\|\mathbf{a}\|_\infty$ is $\max_{i \in [N]} |a_i|$.

The distance in the ℓ_p,ℓ_0,ℓ_∞ metric between two points $\mathbf{a}, \mathbf{b} \in \mathbb{R}^N$ is defined as the corresponding norm of $\mathbf{a} - \mathbf{b}$. Sometimes we refer to ℓ_0, ℓ_2 norms/metrics as the *Hamming* and *Euclidean* norms/metrics respectively. The Hamming norm is also refered to as the *Hamming weight*.

We also sometimes use $\Delta(\mathbf{a})$ and $\Delta(\mathbf{a}, \mathbf{b})$ to denote $\|\mathbf{a}\|_0$ and $\|\mathbf{a} - \mathbf{b}\|_0$ respectively. Furthermore, we define $\Delta(\mathbf{a}, S) := \min\limits_{\mathbf{b} \in S} \Delta(\mathbf{a}, \mathbf{b})$ for any $\mathbf{a} \in \mathbb{R}^N$ and $S \subseteq \mathbb{R}^N$. For $\mathbf{a} \in \mathbb{F}_q^N$ and $d \in \mathbb{N}$, we use $\mathcal{B}_N(\mathbf{a}, d)$ to denote the (closed) Hamming ball of radius d centered at $\mathbf{a}$, i.e., $\mathcal{B}_N(\mathbf{a}, d) := \{\mathbf{b} \in \mathbb{F}_q^N \mid \Delta(\mathbf{a}, \mathbf{b}) \leqslant d\}$; when the dimension is clear from context, we may simply write $\mathcal{B}(\mathbf{a}, d)$ instead of $\mathcal{B}_N(\mathbf{a}, d)$.

We denote the inner product (associated with the Euclidean space) of $\mathbf{a}$ and $\mathbf{b}$ by $\langle \mathbf{a}, \mathbf{b} \rangle := \sum\limits_{i \in [N]} a_i \cdot b_i$. Finally, for every positive integer N we define the edit metric over Σ to be the space Σ^N

endowed with distance function $\mathsf{ed}(\mathbf{a}, \mathbf{b})$, which is defined as the minimum number of character substitutions/insertions/deletions to transform $\mathbf{a}$ into $\mathbf{b}$.

2.1.3 Probabilistic Notation

Let $\mathcal{X}$ be a probability distribution over a finite probability space Θ. We use $x \sim \mathcal{X}$ to denote a random variable x sampled according to $\mathcal{X}$. Sometimes we use shorthand $x \sim \Theta$ to denote x being drawn uniformly at random from Θ. For each $\theta \in \Theta$, we denote $\Pr_{x \sim \mathcal{X}}[x = \theta]$ by $\mathcal{X}(\theta)$. The *support* of $\mathcal{X}$ or $\operatorname{supp}(\mathcal{X})$ is the set of all $\theta \in \Theta$ such that $\mathcal{X}(\theta) \neq 0$. For any event E, we use $\mathbb{1}[E]$ to denote the indicator variable for the event.

2.2 Problem Definitions

Since this thesis considers a number of computational problems some of which occur in multiple chapters, we provide a list of recurring problems below for convenience of the readers.

- **k-SAT.** In the k-SAT problem (abbreviated as kSAT or k-SAT), we are given a CNF formula Φ with at most k literals in each clause and the goal is to decide whether Φ is satisfiable.

- **Dominating Set.** In the k-DOMINATING SET problem (k-DOMSET), we are given a graph G, and the goal is to decide whether G has a dominating set of size k. In the minimization version, called *Minimum Dominating Set* (DOMSET, for short), the goal is to find a dominating set in G of minimum size.

- **Set Cover.** The k-DOMSET is a special case of the k-SET COVER problem (k-SETCOV) where we are given a ground set $\mathcal{U}$, a collection of subsets $\mathcal{S} \subseteq \mathscr{P}(\mathcal{U})$ and an integer k. The goal is to determine whether there are k subsets from $\mathcal{S}$ whose union is $\mathcal{U}$. The minimization version of the problem, called *Minimum Set Cover* (SETCOV, for short), asks to find as few subsets from $\mathcal{S}$ as possible whose union is $\mathcal{U}$. We use SETCOV$(\mathcal{U}, \mathcal{S})$ to denote the optimum of SETCOV on instance $(\mathcal{U}, \mathcal{S})$.

- **Clique.** In the k-CLIQUE problem, we are given a graph G, and the goal is to decide whether G has a clique of size k as a subgraph. In the maximization version, called *Maximum Clique* (CLIQUE, for short), the goal is to find a clique in G of maximum size.

- **Densest k-Subgraph.** In the DENSEST k-SUBGRAPH (DkS), we are given a graph G and an integer k, and the goal is to find $S \subseteq V(G)$ of size k that induces maximum number of edges. DENSEST k-SUBGRAPH *with perfect completeness* refers to the variant of the problem where we are further promised that the graph G contains a k-clique.

- **k-CSP.** An instance $\mathcal{G}$ of MAX k-CSP consists of a variable set V, a finite alphabet set Σ, a distribution $\mathcal{Q}$ over $\binom{V}{k}$ and a predicate $P : \binom{V}{k} \times \Sigma^k \to [0, 1]$. The goal is to find an assignment $\phi : V \to \Sigma$ that maximizes the expected output of the predicate, i.e., $\mathbb{E}_{S \sim \mathcal{Q}}[P(S, \phi|_S)]$.

> We note here that MAX k-CSP is the only problem for which we study in the context of parameterization and do not use "k" as the parameter. In particular, in our study in Chapter 9, we study the problem when $k = 2$ and instead the parameter is the number of variables $|V|$. Indeed, in that chapter, we use k to denote $|V|$ instead of the arity of CSPs; to avoid confusion, we refer to this parameterized version of 2-CSP as PARAMETERIZED 2-CSP.

Lastly, to make it clear to the readers, we use "Research Question" throughout this dissertation for questions that we *do* answer (at least partially) in this dissertation. On the other hand, we use "Open Question" for questions we do *not* know the answer and are hence still open.

2.3 Exponential Time Hypotheses

While computational tractabilities of NP-hard problems can be based on just the $\mathsf{P} \neq \mathsf{NP}$ assumption, fine-grained results often require stronger assumptions. The first type of such assumptions are Exponential Time Hypotheses.

2.3.1 Exponential Time Hypothesis

The Exponential Time Hypothesis (ETH), proposed by Impagliazzo and Paturi [IP01], asserts that 3SAT cannot be solved in subexponential time, as stated below.

Hypothesis 1 (Exponential Time Hypothesis (ETH) [IP01; IPZ01])**.** *There exists $\delta > 0$ such that no algorithm can solve 3-SAT in $O(2^{\delta n})$ time where n is the number of variables. Moreover, this holds even when restricted to formulae in which each variable appears in at most three clauses.*

Note that the original version of the hypothesis from [IP01] does not enforce the requirement that each variable appears in at most three clauses. To arrive at the above formulation, we first apply the Sparsification Lemma of [IPZ01], which implies that we can assume without loss of generality that the number of clauses m is $O(n)$. We then apply Tovey's reduction [Tov84] which produces a 3-CNF instance with at most $3m + n = O(n)$ variables and every variable occurs in at most three clauses. This means that the bounded occurrence restriction is also w.l.o.g.

ETH has numerous implications in running time lower bounds for exact algorithms, parameterized complexity theory[2], and, as we will see shortly, even hardness of approximation.

2.3.2 Strong Exponential Time Hypothesis

We will also use a stronger hypothesis called the Strong Exponential Time Hypothesis (SETH) which postulates that, even the constant in the exponent has to be $(1 - \varepsilon)$ for k-SAT when k is sufficiently large. This is formulated below.

[2] Please refer to a survey by Lokshtanov, Marx and Saurabh [LMS11] for more information on implications of ETH on lower bounds for exact algorithms and parameterized complexity theory.

Hypothesis 2 (Strong Exponential Time Hypothesis (SETH) [IP01; IPZ01]). *For every $\varepsilon > 0$, there exists $k = k(\varepsilon) \in \mathbb{N}$ such that no algorithm can solve k-SAT in $O(2^{(1-\varepsilon)n})$ time where n is the number of variables. Moreover, this holds even when the number of clauses m is at most $c(\varepsilon) \cdot n$ where $c(\varepsilon)$ denotes a constant that depends only on ε.*

Again, we note that, in the original form [IP01], the bound on the number of clauses is not enforced. However, the Sparsification Lemma [IPZ01] allows us to do so without loss of generality.

2.3.3 Gap Exponential Time Hypothesis

Another strengthening of ETH we use is the Gap Exponential Time Hypothesis (Gap-ETH), which essentially states that even approximating 3SAT to some constant ratio takes exponential time:

Hypothesis 3 (Gap Exponential Time Hypothesis (Gap-ETH) [Din16; MR17a]). *There exist constants $\delta, \varepsilon > 0$ such that no $O(2^{\delta n})$-time algorithm can, given a 3-CNF formula ϕ with n variables, distinguish between the case where ϕ is satisfiable and the case where $\mathrm{val}(\phi) \leqslant 1 - \varepsilon$. Here $\mathrm{val}(\phi)$ denote the maximum fraction of clauses of ϕ satisfied by any assignment.*

Moreover, this holds even when the number of clauses m is $O(n)$.

While both SETH and Gap-ETH imply ETH, no formal relationship is known between the two. We would also like to remark that, while Gap-ETH may sound like a very strong assumption, as pointed out in [Din16; MR17b], there are a few evidences supporting the conjecture:

- As will be explained in more details below, Dinur's PCP Theorem [Din07] implies a running time lower bound of $2^{o(n/\mathrm{polylog}(n))}$ for Gap-3SAT, assuming ETH. The $\mathrm{polylog}(n)$ loss in the exponent comes from the size blow-up of the PCP; if a linear-size PCP, one in which the size blow-up is constant, exists then Gap-ETH would follow from ETH.
- No subexponential-time algorithm is known even for the following (easier) problem, which is sometimes referred to as *refutation of random 3SAT*: for a constant density parameter Δ, given a 3-CNF formula Φ with n variables and $m = \Delta n$ clauses, devise an algorithm that outputs either SAT or UNSAT such that the following two conditions are satisfied:
  - If Φ is satisfiable, the algorithm always output SAT.
  - Over all possible 3-CNF formulae Φ with n clauses and m variables, the algorihtm outputs UNSAT on at least 0.5 fraction of them.

  Note here that, when Δ is a sufficiently large constant (say 1000), a random 3-CNF formula is, with high probability, not only unsatisfiable but also not even 0.9-satisfiable. Hence, if Gap-ETH fails, then the algorithm that refutes Gap-ETH will also be a subexponential time algorithm for refutation of random 3SAT with density Δ.

  Refutation of random 3SAT, and more generally random CSPs, is an important question that has connections to many other fields, including hardness of approximation, proof complexity, cryptography and learning theory. We refer the reader to [AOW15] for a more comprehensive review of known results about the problem and its applications in various areas.

Despite being intensely studied for almost three decades, no subexponential-time algorithm is known for the above regime of parameter. In fact, it is known that the Sum-of-Squares hierarchies cannot refute random 3-SAT with constant density in subexponential time [Gri01; Sch08]. Given how powerful SDP [Rag08], and more specifically Sum-of-Squares [LRS15], are for solving (and approximating) CSPs, this suggests that refutation of random 3-SAT with constant density, and hence Gap-3SAT, may indeed be exponentially hard or, at the very least, beyond our current techniques.

- Dinur speculated that Gap-ETH might follow as a consequence of some cryptographic assumption [Din16]. This was recently confirmed by Applebaum [App17] who showed that Gap-ETH follows from an existence of any exponentially-hard locally-computable one-way function. In fact, he proved an even stronger result that Gap-ETH follows from ETH for some CSPs that satisfy certain "smoothness" properties.

Lastly, we note that the assumption $m = O(n)$ made in the conjecture can be made without loss of generality. As pointed out in both [Din16] and [MR17b], this follows from the fact that, given a 3-SAT formula ϕ with m clauses and n variables, if we create another 3-SAT formula ϕ' by randomly selected $m' = \Delta n$ clauses, then, w.h.p., $|\text{SAT}(\phi)/m - \text{SAT}(\phi')/m'| \leqslant O(1/\Delta)$.

2.4 Fine-Grained Complexity Assumptions

In addition to the Exponential Time Hypotheses, we will use two assumptions regarding problems in P: the Orthogonal Vector Hypothesis and the k-SUM Hypothesis. There are many more such assumptions that are used in fine-grained complexity, but we choose not to discuss them here; for readers interested in learning about other assumptions and the state-of-the-art conditional lower bounds, please refer to a survey of Williams [Wil18b].

2.4.1 Orthogonal Vector Hypothesis

The first fine-grained complexity assumption we use is the Orthogonal Vector Hypothesis (OVH). In the Orthogonal Vector problem (OV), we are given two sets of n vectors $A, B \subseteq \{0, 1\}^d$ and the goal is to determine whether there exist $a \in A, b \in B$ that are orthogonal.

Clearly, the problem can be solved in $O(n^2 d)$ by trivial brute-force algorithm. OVH states that this algorithm is nearly optimal, in the sense that there is no truly subquadratic time algorithm for the problem, even when $d = O(\log n)$. This is stated formally below.

Hypothesis 4 (Orthogonal Vector Hypothesis, OVH)**.** *For every $\varepsilon > 0$, no algorithm can solve* OV *in $O(n^{2-\varepsilon})$ time. Moreover, this holds even when the dimension d is at most $c(\varepsilon) \log n$ where $c(\varepsilon)$ denotes a constant that depends only on ε.*

It is known that SETH implies OVH [Wil05], and therefore the results based on OVH (in Chapter 11) also hold under SETH.

2.4.2 k-SUM Hypothesis

Our final hypothesis is the k-SUM Hypothesis. In the k-SUM problem, we are given k sets $S_1, \dots, S_k$ each of n integers in the range $[-M, M]$, and we are asked to determine whether we can pick k integers, one from each set, so that the sum is equal to zero. This problem can be solved via a "meet-in-the-middle" approach in $O(n^{\lceil k/2 \rceil})$ time. The k-SUM Hypothesis states that this algorithm is essentially optimal:

Hypothesis 5 (k-SUM Hypothesis [AL13]). *For every integer $k \geqslant 3$ and every $\varepsilon > 0$, no $O(n^{\lceil k/2 \rceil - \varepsilon})$ time algorithm can solve k-SUM where n denotes the total number of input integers, i.e., $n = |S_1| + \cdots + |S_k|$. Moreover, this holds even when $M = n^{2k}$.*

The above hypothesis is a natural extension of the more well-known 3-SUM Hypothesis [GO95; Pat10], which states that 3-SUM cannot be solved in $O(n^{2-\varepsilon})$ time for any $\varepsilon > 0$. Moreover, the k-SUM Hypothesis is closely related to the question of whether SUBSET-SUM can be solved in $O(2^{(1/2-\varepsilon)n})$ time; if the answer to this question is negative, then k-SUM cannot be solved in $O(n^{k/2-\varepsilon})$ time for every $\varepsilon > 0, k \in \mathbb{N}$. We remark that, if one is only willing to assume this latter weaker lower bound of $O(n^{k/2-\varepsilon})$ instead of $O(n^{\lceil k/2 \rceil - \varepsilon})$, our reduction in Chapter 6 would give an $O(n^{k/2-\varepsilon})$ running time lower bound for approximating k-DOMSET. Finally, we note that the assumption that $M = n^{2k}$ can be made without loss of generality since there is a randomized reduction from the general version of the problem (where M is, say, 2^n) to this version of the problem and it can be derandomized under a certain circuit complexity assumption [ALW14].

2.5 Nearly-Linear Size PCPs and (Sub)exponential Time Reductions

The celebrated PCP Theorem [AS98; Aro+98], which lies at the heart of virtually all known NP-hardness of approximation results, can be viewed as a polynomial-time reduction from 3SAT to a gap version of 3SAT, as stated below. While this perspective is a rather narrow viewpoint of the theorem that leaves out the fascinating relations between parameters of PCPs, it will be the most convenient for our purpose.

Theorem 2.1 (PCP Theorem [AS98; Aro+98]). *For some constant $\varepsilon > 0$, there exists a polynomial-time reduction that takes a 3-CNF formula φ and produces a 3-CNF formula ϕ such that*

- (Completeness) *if φ is satisfiable, then ϕ is satisfiable, and,*
- (Soundness) *if φ is unsatisfiable, then* $\mathrm{val}(\phi) \leqslant 1 - \varepsilon$.

Following the first proofs of the PCP Theorem, considerable efforts have been made to improve the trade-offs between the parameters in the theorem. One such direction is to try to reduce the size of the PCP, which, in the above formulation, translates to reducing the size of ϕ relative to φ. On this front, it is known that the size of ϕ can be made nearly-linear in the size of φ [Din07; BS08;

MR10]. For our purpose, we will use Dinur's PCP Theorem [Din07], which has a blow-up of only polylogarithmic in the size of ϕ:

Theorem 2.2 (Dinur's PCP Theorem [Din07])**.** *For some constant $\varepsilon, d, c > 0$, there exists a polynomial-time reduction that takes a 3-CNF formula φ with m clauses and produces another 3-CNF formula ϕ with $m' = O(m \log^c m)$ clauses such that*

- (Completeness) *if φ is satisfiable, then ϕ is satisfiable, and,*
- (Soundness) *if φ is unsatisfiable, then* $\mathrm{val}(\phi) \leqslant 1 - \varepsilon$*, and,*
- (Bounded Degree) *each variable of ϕ appears in $\leqslant d$ clauses.*

Note that Dinur's PCP, combined with ETH, implies a lower bound of $2^{\Omega(m/\text{polylog } m)}$ on the running time of algorithms that solve the gap version of 3SAT, which is only a factor of $O(\text{polylog } m)$ in the exponent off from Gap-ETH. Putting it differently, Gap-ETH is closely related to the question of whether a linear size PCP, one where the size blow-up is only constant instead of polylogarithmic, exists; its existence would mean that Gap-ETH is implied by ETH.

Under the exponential time hypothesis, nearly-linear size PCPs allow us to start with an instance ϕ of the gap version of 3SAT and reduce, in subexponential time, to another problem. As long as the time spent in the reduction is $2^{o(m/\log^c m)}$, we arrive at a lower bound for the problem. Arguably, Aaronson et al. [AIM14] popularized this method, under the name *birthday repetition*, by using such a reduction of size $2^{\widetilde{\Omega}(\sqrt{m})}$ to prove ETH-hardness for free games and dense CSPs. Without going into any detail now, let us mention that the name birthday repetition comes from the use of the birthday paradox in their proof and, since its publication, their work has inspired many inapproximability results [BKW15; Rub16; Rub17a; Rub17b; DFS16; Bra+17]. Our results in Part I too are inspired by this line of work and, as we will see soon, part of our proof also contains a birthday-type paradox. In fact, Chapter 3 directly deals with the exact construction of [AIM14] and, in the process, resolves some open questions from that work.

While Dinur's PCP Theorem (Theorem 2.2) suffices for most of our results in Part I, our proofs in Chapter 5 require a PCP theorem with low soundness of Moshkovitz and Raz [MR10]. To state the theorem, we first need to define the LABEL COVER problem, a central problem in the area of hardness of approximation.

Definition 2.3 (Label Cover)**.** *A label cover instance $\mathcal{L}$ consists of $(G, \Sigma_U, \Sigma_V, \Pi)$, where*

- *$G = (U, V, E)$ is a bipartite graph between vertex sets U and V and an edge set E,*
- *Σ_U and Σ_V are sets of* alphabets *to be assigned to vertices in U and V, respectively, and*
- *$\Pi = \{\Pi_e\}_{e \in E}$ is a set of* constraints*, where $\Pi_e \subseteq \Sigma_U \times \Sigma_V$ denote the accepting assignments for the edge e.*

We said that a label cover instance $\mathcal{L}$ satisfies projection property *(or $\mathcal{L}$ is a* projection game*) if for every edge $e = (u, v) \in E$ and every $\alpha \in \Sigma_U$, there is exactly one $\beta \in \Sigma_V$ such that $(\alpha, \beta) \in \Pi_e$.*

In other words, we may represent our constraints as projections $\pi_e : \Sigma_U \to \Sigma_V$, where (α, β) satisfies the constraint iff $\pi_e(\alpha) = \beta$.

An assignment (aka labeling) for $\mathcal{L}$ is a pair $\sigma = (\sigma_U, \sigma_V)$ of functions $\sigma_U : U \to \Sigma_U$ and $\sigma_V : V \to \Sigma_V$. The value of σ, denoted by $\operatorname{val}_{\mathcal{L}}(\sigma)$ is defined as the fraction of edges $(u, v) \in E$ such that $(\sigma_u, \sigma_v) \in \Pi_{(u,v)}$; these edges are called satisfied *edges. The value of the instance $\mathcal{L}$, $\operatorname{val}(\mathcal{L})$, is defined as the maximum value among all assignments σ.*

For convenience, we sometimes use the notation $|\mathcal{L}|$ to denote the size of the label cover instance; in particular, $|\mathcal{L}| = |\Sigma_U| + |\Sigma_V| + |U| + |V|$.

Sometimes it will be convenient to think of a labeling σ as a function from $U \cup V$ to $\Sigma_U \cup \Sigma_V$ and we use the two notions interchangeably; whenever this is the case, we always assume that every vertex in U is mapped to Σ_U whereas every vertex in V is mapped to Σ_V. Furthermore, we occasionally work with assignments that only label a subset of $U \cup V$ but leaves the rest unlabeled. We refer to such an assignment as a *partial assignment* to an instance; more specifically, for any $S \subseteq U \cup V$, an S-partial assignment (or partial assignment on S) is a function $\sigma : S \to \Sigma_U \cup \Sigma_V$. For notational convenience, we also use Σ to denote $\Sigma_U \cup \Sigma_V$ and Σ^S to denote the set of all functions from S to Σ.

We also often work with multiple graphs in our reductions/proofs. To avoid confusion, we might refer to the graph G as the *super-graph* and the vertices of G as the *super-nodes*.

The PCP Theorem by Moshkovitz and Raz [MR10] is a reduction from 3SAT to the gap version of Label Cover that preserves the size to be almost linear and (importantly) achieves low soundness:

Theorem 2.4 (Moshkovitz-Raz PCP [MR10]). *For every $\nu = \nu(m) > 0$, there exists a polynomial-time reduction that takes a 3-CNF formula φ with m clauses and produces a bi-regular projection game $\mathcal{L}$ with $|U|, |V|, |E| = m^{1+o(1)} poly(1/\nu)$ and $|\Sigma_U|, |\Sigma_V| \leqslant 2^{poly(1/\nu)}$ such that*

- (Completeness) *if φ is satisfiable, then $\mathcal{L}$ is satisfiable, and,*
- (Soundness) *if φ is unsatisfiable, then* $\operatorname{val}(\mathcal{L}) \leqslant \nu$.

2.6 Parameterized Complexity

Over the years, many approaches have been devised to cope with NP-hardness of fundamental computational problems. Prominent among them is the area of *parameterized complexity*. In parameterized problems, part of the input is designated as the parameter, and the notion of "efficient algorithm" is relaxed to the notion of *fixed-parameter (FPT) algorithms*, which are algorithms with running time $T(k) \cdot \operatorname{poly}(n)$ where k is the parameter, n is the size of the input and T can be any computable function. For instance, an algorithm with running time $2^k \cdot \operatorname{poly}(n)$ or $2^{2^k} \cdot \operatorname{poly}(n)$ is considered FPT. The problems that admit FPT algorithms are said to be fixed-parameter tractable; the class of such problems is also denoted by FPT.

The area has led to numerous algorithmic tools and techniques that allow one to tackle NP-hard problems, especially when the parameter is chosen appropriately. Since we are only dealing with intractability here, we will not discuss algorithmic techniques any further. Interested readers may refer to the many books in the field; for instance, Cygan et al.'s book [Cyg+15] provides a relatively up-to-date review of basic principles used in parameterized algorithms.

For convenience, unless stated otherwise, we use "k" to denote the parameter throughout this thesis. Moreover, we often refer to the parameterized variant of a problem Π as k-Π.

2.6.1 The W Hierarchy

Unsurprisingly, many parameterized problems remain intractable even in the FPT notion. Over the years, there have been many parameterized complexity classes defined to capture such an intractability phenomenon. Arguably, the most widely used hierarchy today is the W hierarchy, and this will be the only class we discuss in this section. For interested readers, the book [FG06] of Flum and Grohe provides a rather comprehensive look on the different complexity classes/hierarchies (including para-NP and the A hierarchy) and how they relate to each other.

We now turn our focus back to the W hierarchy. The complexity classes of the hierarchy is defined based on the following notion of reduction: a *parameterized reduction* (aka *FPT reduction*) from a parameterized problem $\mathscr{A}$ to another parameterized problem $\mathscr{B}$ is an algorithm that takes in an instance (x, k) of $\mathscr{A}$, runs in $f(k) \cdot \text{poly}(|x|)$ time for some function k and produces an instance (x', k') where $k' \leqslant g(k)$ for some function g. As usual, it must be that if (x, k) is a YES (resp. NO) instance of $\mathscr{A}$, then (x', k') is a YES (resp. NO) instance of $\mathscr{B}$. (It is simple to see that, if $\mathscr{A}$ parameterized reduces to $\mathscr{B}$ and $\mathscr{B}$ is in FPT, then $\mathscr{A}$ is also in FPT.)

With the notion of parameterized reduction in mind, the class $\text{W}[t]$ for any positive integer $t \in \mathbb{N}$ is defined as all problems that can be reduced to the following problem: given a circuit of weft (at most) t, is there an assignment with Hamming weight k to the input that satisfies the circuit? Here weft refers to the number of unbounded fan-in gates from any input to output path.

A reason that makes the hierarchy popular is that many fundamental combinatorial problems turn out to be complete for the classes in the hierarchy. Specifically, k-Clique and k-DomSet are complete for the classes W[1] and W[2] respectively. In this dissertation, we in fact rarely use the W hierarchy; our only result that relies on the hierarchy is the W[1]-hardness of approximation k-DomSet (Theorem 6.1). To state this result in a consistent manner with other results in the respective chapter, we may view the W[1]-hardness as being conditional on the assumption that W[1] does not collapse to FPT:

Hypothesis 6 ($\text{W}[1] \neq$ FPT)**.** *For any computable function $T : \mathbb{N} \to \mathbb{N}$, no algorithm can solve k-*Clique *in $T(k) \cdot poly(n)$ time where n denotes the number of vertices in the input graph.*

2.6.2 FPT Approximation and Total Inapproximability

As this dissertation deals with FPT approximation algorithms, we have to define several notations regarding FPT approximation algorithms and inapproximability results.

To do so, let us start by formalizing the the notation of optimization problems; here we follow the notation due to Chen et al. [CGG06]. An *optimization problem* Π is defined by three components: (1) for each input instance I of Π, a set of valid solutions of I denoted by $\text{SOL}_{\Pi}(I)$,

(2) for each instance I of Π and each $y \in \mathrm{SOL}_\Pi(I)$, the cost of y with respect to I denoted by $\mathcal{C}_\Pi(I, y)$, and (3) the goal of the problem $\mathrm{GOAL}_\Pi \in \{\min, \max\}$ which specifies whether Π is a minimization or maximization problem. Throughout this work, we will assume that $\mathcal{C}_\Pi(I, y)$ can be computed in time $|I|^{O(1)}$. Finally, we denote by $\mathrm{OPT}_\Pi(I)$ the optimal value of each instance I, i.e., $\mathrm{OPT}_\Pi(I) = \mathrm{GOAL}_\Pi\, \mathcal{C}(I, y)$ where y is taken over $\mathrm{SOL}_\Pi(I)$.

We often (but not always) parametrize by the solution size. In this case, the most convenient definition is to consider the following "gap" version of these problems. It is rather straightforward to check that this notion is weaker (i.e. easier) than the other notion where the $\mathrm{OPT}_\Pi(I)$ is itself a parameter. That is, our impossibility results for gap versions translate to those versions as well. For a formal statements relating the two, please refer to Propositions 2.3 and 2.4 in [Cha+17].

Definition 2.5 (FPT gap approximation)**.** *For any optimization problem* Π *and any computable function* $f : \mathbb{N} \to [1, \infty)$*, an algorithm* $\mathbb{A}$*, which takes as input an instance* I *of* Π *and a positive integer* k*, is said to be an* f-FPT-approximation algorithm *for* Π *if the following conditions hold on every input* (I, k)*:*

- $\mathbb{A}$ *runs in time* $t(k) \cdot |I|^{O(1)}$ *for some computable function* $t : \mathbb{N} \to \mathbb{N}$.
- *If GOAL*$_\Pi = \max$*,* $\mathbb{A}$ *must output 1 if* $\mathrm{OPT}_\Pi(I) \geqslant k$ *and output 0 if* $\mathrm{OPT}_\Pi(I) < k/f(k)$.
  If GOAL$_\Pi = \min$*,* $\mathbb{A}$ *must output 1 if* $\mathrm{OPT}_\Pi(I) \leqslant k$ *and output 0 if* $\mathrm{OPT}_\Pi(I) > k \cdot f(k)$.

Π *is said to be* f-FPT-approximable *if there is an* f*-FPT-approximation algorithm for* Π.

Next, we formalize the concept of *totally FPT inapproximable*, which encapsulates the non-existence of non-trivial FPT approximations alluded to earlier in the introduction.

Definition 2.6. *A minimization problem* Π *is said to be* totally FPT inapproximable *if, for every computable function* $f : \mathbb{N} \to [1, \infty)$, Π *is not* f*-FPT-approximable.*

A maximization problem Π *is said to be* totally FPT inapproximable *if, for every computable function* $f : \mathbb{N} \to [1, \infty)$ *such that* $f(k) = o(k)$ *(i.e.,* $\lim_{k\to\infty} k/f(k) = \infty$*),* Π *is not* f*-FPT-approximable.*

As stated earlier, we do not always parametrize by the optimum. The exceptions are DENSEST k-SUBGRAPH (in Chapter 8), PARAMETERIZED 2-CSP and DIRECTED STEINER NETWORK (in Chapter 9). For these results, we will state more explicitly what our results rule out. Another point we note is that, in Chapter 6, we show that k-DOMSET is totally FPT inapproximable (under W[1] $\neq$ FPT), but we choose to state the results slightly differently (see Theorem 6.1), so that the results from different assumptions are more consistent.

2.6.3 FPT Inapproximability via Inherently Enumerative

Another notion that will be useful in proving FPT inapproximability is the concept of *inherently enumerative* problems, which will be formalized shortly.

To motivate the concept, note that many problems Π considered in this thesis admit exact algorithms that run in time $O^\star(|I|^{\mathrm{OPT}_\Pi(I)})$. For instance, to find a clique of size k in G, one can enumerate all $\binom{|V(G)|}{k} = |V(G)|^{O(k)}$ possibilities[3]. For many W[1]-hard problems, this running time is nearly the best possible assuming ETH: Any algorithm that finds a k-clique in time $|V(G)|^{o(k)}$ would break ETH. In the light of such result, it is natural to ask the following question.

Assume that say $\textsc{Clique}(G) \geqslant 2^{2^k}$, can we find a clique of size k in time $|V(G)|^{o(k)}$?

In other words, can we exploit a prior knowledge that there is a clique of size much larger than k to help us find a k-clique faster? Roughly speaking, we will show later that, assuming Gap-ETH, the answer of this question is also negative, even when 2^{2^k} is replaced by any constant independent of k. This is encapsulated in the concept of inherently enumerative as defined below.

Definition 2.7 (Inherently Enumerative). *A problem* Π *is said to be* inherently enumerative *if there exist constants* $\delta, r_0 > 0$ *such that, for any integers* $q \geqslant r \geqslant r_0$*, no algorithm can decide, on every input instance* I *of* Π*, whether (i)* $\mathrm{OPT}_\Pi(I) < r$ *or (ii)* $\mathrm{OPT}_\Pi(I) \geqslant q$ *in time*[4] $O_{q,r}(|I|^{\delta r})$.

While we will show that $\textsc{Clique}$ and $\textsc{DomSet}$ are inherently enumerative, we cannot do the same for some other problems, such as $\textsc{Biclique}$. Even for the exact version of $\textsc{Biclique}$, the best running time lower bound known is only $|V(G)|^{\Omega(\sqrt{k})}$ [Lin15] assuming ETH. In order to succinctly categorize such lower bounds, we define a similar but weaker notation of *weakly* inherently enumerative:

Definition 2.8 (Weakly Inherently Enumerative). *For any function* $\beta = \omega(1)$ *(i.e.,* $\lim_{r\to\infty} \beta(r) = \infty$*), a problem* Π *is said to be* β-weakly inherently enumerative *if there exists a constant* $r_0 > 0$ *such that, for any integers* $q \geqslant r \geqslant r_0$*, no algorithm can decide, on every input instance* I *of* Π*, whether (i)* $\mathrm{OPT}_\Pi(I) < r$ *or (ii)* $\mathrm{OPT}_\Pi(I) \geqslant q$ *in time* $O_{q,r}(|I|^{\beta(r)})$.

Π *is said to be* weakly inherently enumerative *if it is* β*-weakly inherently enumerative for some* $\beta = \omega(1)$.

It follows from the definitions that any inherently enumerative problem is also weakly inherently enumerative. As stated earlier, we will prove total FPT inapproximability through inherently enumerative; the proposition below formally establishes a connection between the two.

Proposition 2.9. *If* Π *is weakly inherently enumerative, then* Π *is totally FPT inapproximable.*

An important tool in almost any branch of complexity theory, including parameterized complexity, is a notion of reductions. For the purpose of facilitating proofs of totally FPT inapproximability, we define the following reduction, which we call *FPT gap reductions*.

Definition 2.10 (FPT gap reduction). *For any functions* $f, g = \omega(1)$*, a problem* Π_0 *is said to be* (f, g)-FPT gap reducible *to a problem* Π_1 *if there exists an algorithm* $\mathbb{A}$ *which takes in an instance* I_0 *of* Π_0 *and integers* q, r *and produce an instance* I_1 *of* Π_1 *such that the following conditions hold.*

[3] A faster algorithm runs in time $|V(G)|^{\omega k/3}$ can be done by a reduction to matrix multiplication.

[4] $O_{q,r}(\cdot)$ here and in Definition 2.8 hides any multiplicative term that is a function of q and r.

- $\mathbb{A}$ *runs in time* $t(q, r) \cdot |I_0|^{O(1)}$ *for some computable function* $t : \mathbb{N} \times \mathbb{N} \to \mathbb{N}$.
- *For every positive integer* q*, if* $\mathrm{OPT}_{\Pi_0}(I_0) \geqslant q$*, then* $\mathrm{OPT}_{\Pi_1}(I_1) \geqslant f(q)$.
- *For every positive integer* r*, if* $\mathrm{OPT}_{\Pi_0}(I_0) < g(r)$*, then* $\mathrm{OPT}_{\Pi_1}(I_1) < r$.

It is not hard to see that FPT gap reduction indeed preserves totally FPT inapproximability, as formalized in Proposition 2.11 below.

Proposition 2.11. *If a problem* Π_0 *is (i)* (f, g)*-FPT gap reducible to* Π_1 *for some computable non-decreasing functions* $f, g = \omega(1)$*, and (ii) totally FPT inapproximable, then* Π_1 *is also totally FPT inapproximable.*

As stated earlier, we sometimes work with inherently enumerative concepts instead of working directly with totally FPT inapproximability; fortunately, FPT gap reductions can also be used for (weakly) inherently enumerativeness, as stated below.

Proposition 2.12. *If a problem* Π_0 *is (i)* (f, g)*-FPT gap reducible to* Π_1 *and (ii)* β*-weakly inherently enumerative for some* $f, g, \beta = \omega(1)$*, then* Π_1 *is* $\Omega(\beta \circ g)$*-weakly inherently enumerative.*

The (straightforward) proofs of Propositions 2.9, 2.11 and 2.12 can be found in [Cha+17]. As a final remark, we note that our results in Chapter 6 for k-DOMSET, specifically Theorem 6.2, imply that it is inherently enumerative assuming ETH; however, we choose not to state it in this term, to highlight the dependency of approximation ratio on n.

2.7 Error-Correcting Codes

An error-correcting code (ECC) is a map $C : \Sigma^m \to \Sigma^d$ here m and d are positive integers which are referred to as the *message length* and *block length* of C respectively. Intuitively, the function C encodes an original message of length m to an encoded message of length d. Since we will also deal with communication protocols, for which "message length" has another meaning, we will sometimes refer to the message length of codes as *code message length* whenever there is an ambiguity. The *rate* of a code $\rho(C)$ is defined as the ratio between its message length and its block length, i.e., $\rho(C) = m/d$. The *distance* of a code, denoted by $\Delta(C)$, is defined as $\min\limits_{\mathbf{x} \neq \mathbf{y} \in \Sigma^m} \|C(\mathbf{x}) - C(\mathbf{y})\|_0$. (Recall here that $\| \cdot \|_0$ is used to the denote the hamming weight.) Its *relative distance* is defined as $\delta(C) := \Delta(C)/d$.

For an ECC C, we use the caligraphic letter $\mathcal{C}$ to denote the set of corresponding codewords, i.e., $\mathcal{C} := \{C(\mathbf{a}) \mid \mathbf{a} \in \Sigma^m\}$. We sometimes use $\mathcal{C}$ in place of C for the above notations, e.g. $\Delta(\mathcal{C})$.

When $\Sigma = \mathbb{F}_q$ is a finite field, we said that the ECC C is *linear* iff $C : \mathbb{F}_q^m \to \mathbb{F}_q^d$ is a linear function, i.e., there exists a matrix $\mathbf{C} \in \mathbb{F}_q^{d \times m}$ such that $C(\mathbf{a}) = \mathbf{C}\mathbf{a}$ for all $\mathbf{a} \in \mathbb{F}_q^m$. We often use the notion $[d, m, \Delta]_q$ to denote a linear code of block length d, message length m, and distance Δ over alphabet $\mathbb{F}_q$. Note also that, for a linear code C, $\Delta(C)$ is equal to the minimum weight of a non-zero codeword in $\mathcal{C}$. Finally, for any code $\mathcal{C}$, we use $A_w(\mathcal{C}) := |\{\mathbf{c} \in \mathcal{C} \mid \Delta(\mathbf{c}) = w\}|$ to denote the number of codewords of weight w.

In each of Chapters 6, 10 and 11, we will need different error-correcting codes with properties that are tailored towards the applications at hand. Due to this, we will not define these codes here, but rather at the respective chapters when they are needed.

2.8 Zarankiewicz Problem and Related Bounds

The Zarankiewicz problem [Zar51] is an old but yet unsolved problem in extremal combinatorics, which asks: what is the maximum number of edges can an N-vertex $K_{t,t}$-free graph[5] has? This question is similar to another classic problem which asks the exact same question, except $K_{t,t}$-free is replaced with K_t-free. Unlike the Zarankiewicz problem, the latter problem is completely understood: the N-vertex K_t-free graph with maximum number of edges is the graph whose N vertices are partitioned into $t-1$ groups each of size either $\lceil \frac{N}{t-1} \rceil$ or $\lfloor \frac{N}{t-1} \rfloor$, and two vertices are connected by an edge if and only if they are from different groups. This result is typically referred to as Turán's theorem [Tur41], and such a graph as a Turán's graph. Observe that, for any $t \geqslant 2$, the Turán's graph has $\Omega(N^2)$ edges.

While the Zarankiewicz problem is not yet fully resolved, several upper bounds and lower bounds are known. We will use the following well-known upper bound on the number of edges:

Theorem 2.13 (Kővári-Sós-Turán (KST) Theorem [KST54]). *For every positive integer N and $t \leqslant N$, every $K_{t,t}$-free graph on N vertices has at most $O(N^{2-1/t})$ edges (i.e., density $O(N^{-1/t})$).*

We remark here that KST Theorem demonstrates the stark contrast between K_t-free graphs and $K_{t,t}$-free graphs. As stated above, in the former, one can still have very dense graph (of density $\Omega(1)$); however, for the latter, the density can be at most $O(N^{-1/t})$.

In fact, Alon [Alo02] shows that the assumption that the graph is $K_{t,t}$-free can be relaxed: to conclude that a graph is sparse, it suffices for the graph to have few labelled copies of $K_{t,t}$, as stated below.

Lemma 2.14 ([Alo02, Lemma 2.1][6]). *For every positive integer $N \geqslant 2$ and $t \leqslant N$ and $\alpha \in \mathbb{R}^+$, any graph G on N vertices that has at most $(\alpha/2)^{t^2} N^{2t}$ labelled copies of $K_{t,t}$ has density at most α.*

It will also be convenient for us to use a bipartite version of Alon's lemma, stated below. We provide the proof here for completeness; the proof is essentially the same as Alon's original proof but just in the context of bipartite graph.

[5] Some versions of the problem consider bipartite graphs. However, it will be more convenient for us to consider general graphs. Note that it is simple to see that the bounds in both cases are within a constant factor of each other.

[6] The lemma is stated slightly differently in [Alo02]. Namely, it was stated there that any graph G with $\geqslant \varepsilon N^2$ edges contains at least $(2\varepsilon)^{t^2} N^{2t}$ labelled copies of $K_{t,t}$. The formulation here follows from the fact that α-dense graph on $N \geqslant 2$ vertices contains at least $(\alpha/4)N^2$ edges.

Lemma 2.15. *For every positive integer $N, M \geqslant 2$ and $s \leqslant N, t \leqslant M$ and any positive real number α, any bipartite graph $G = (A, B, E)$ with $|A| = N, |B| = M$ that contains at most $\alpha^{st} N^s M^t$ labelled copies of $K_{s,t}$ has at most αNM edges.*

Proof. We will prove the contrapositive; suppose that $G = (A, B, E)$ has $|A| = N, |B| = M$ and $|E| > \alpha NM$. Observe that the number of labelled copies of $K_{1,t}$ is exactly $\sum_{a \in A} \deg(a)^t$. From power-mean inequality, this is at least $N \left(\frac{1}{N} \sum_{a \in A} \deg(a)\right)^t > N(\alpha^t M^t)$.

Next, let us partition the labelled copies of $K_{1,t}$ based on the t-tuple of right vertices. More precisely, for every $R \in B^t$, let $\mathcal{K}_{1,t}(R) = \{(\{a\}, R) \mid \forall b \in B, (a, b) \in E\}$. Observe that the number of labelled copies of $K_{s,t}$ is exactly

$$\sum_{R \in B^t} |\mathcal{K}_{(1,t)}(R)|^s \geqslant M^t \left(\frac{1}{M^t} \sum_{R \in B^t} |\mathcal{K}_{(1,t)}(R)|\right)^s > \alpha^{st} N^s M^t,$$

where the first inequality is from power-mean inequality and the second follows from the number of labelled copies of $K_{1,t}$ is more than $N(\alpha^t M^t)$. □

KST Theorem and Alon's lemmas allow us to prove that a graph is sparse by showing that it is $K_{t,t}$-free, or it contains few labelled copies of $K_{t,t}$. This will become useful in Chapters 3, 4 and 8.

2.9 Well-Behaved Subsets

Throughout this thesis, we often want to construct subsets that "behave like random subsets" of a specific size. While we can of course just take the random subsets, they are not preferred as we would rather have deterministic constructions. Nonetheless, the parameter regimes for which we are interested in are the "easy" regimes where we can deterministically construct these subsets pretty simply. Here we provide such a construction; we start with the following definition:

Definition 2.16. *For any $k, q \in \mathbb{N}$, let $T_i = \{\mathbf{x} \in [q]^k \mid x_i = 1\}$ for all $i \in [k]$.*

From now on, we use $\alpha := 1/q$ to denote the fraction of total elements from $[q]^k$ contained in each subset T_i. A simple but crucial observation regarding the constructed sets is that the indicator variables are independent random variables with mean α, as stated more formally below.

Observation 2.17. *Let $k, q, T_1, \dots, T_k$ be as in the above definition. Let $X_1, \dots, X_k$ be boolean random variables on the uniform distribution over $[q]^k$ such that $X_i = 1$ on atom $\mathbf{x}$ iff $\mathbf{x} \in T_i$. Then, $X_1, \dots, X_k$ are independent random variables with $\Pr[X_i = 1] = \alpha$.*

Using the observation above, it is now easy to argue that these sets "behave like random sets". To be more precise, we need two types of properties as stated below.

2.9.1 Uniformity

The first property is what we call *uniformity*. Intuitively, it says that, if we pick some subcollection $\widetilde{\mathcal{T}} \subseteq \{T_1, \dots, T_k\}$ that is "not too small", then most of the elements still appear in "many subsets" $T \in \widetilde{\mathcal{T}}$. The formal definition is stated below.

Definition 2.18 (Uniformity)**.** *For any universe $\mathcal{U}$, a collection $\widetilde{\mathcal{S}}$ of subsets of $\mathcal{U}$ is (γ, μ)-uniform if, for at least $(1-\mu)$ fraction of elements $u \in \mathcal{U}$, u appears in at least γ fraction of the subsets in $\widetilde{\mathcal{S}}$. In other words, $\widetilde{\mathcal{S}}$ is (γ, μ)-uniform if and only if $|\{u \in \mathcal{U} \mid \Pr_{S \sim \widetilde{\mathcal{S}}}[u \in S] \geqslant \gamma\}| \geqslant (1-\mu)|\mathcal{U}|$.*

We can now compute the relation between parameters for the subsets in Definition 2.16:

Proposition 2.19. *For any $\mu > 0$, any subcollection $\widetilde{\mathcal{T}} \subseteq \{T_1, \dots, T_k\}$ of size at least $\lceil 8\ln(1/\mu)/\alpha \rceil$ is $(\alpha/2, \mu)$-uniform (with respect to the universe $[q]^k$).*

Proof. Consider a subcollection $\widetilde{\mathcal{T}} = \{T_{i_1}, \dots, T_{i_h}\}$ where $h \geqslant \lceil 8\ln(1/\mu)/\alpha \rceil$. Notice that the condition $\Pr_{T \sim \widetilde{\mathcal{T}}}[u \in T] \geqslant \alpha/2$ is exactly equivalent to $X_{i_1} + \cdots + X_{i_h} \geqslant (\alpha/2)h$. Since $X_{i_1}, \dots, X_{i_h}$ are i.i.d. boolean r.v.s with mean α, we can apply Chernoff bound, which gives:

$$\Pr[X_{i_1} + \cdots X_{i_h} \geqslant (\alpha/2)h] \geqslant 1 - e^{-\frac{\alpha h}{8}} \geqslant 1 - \mu,$$

as desired. □

2.9.2 Dispersers and Intersection Dispersers

We also need the definition of *dispersers*. Recall that[7], roughly speaking, a collection of subsets is a disperser if, when we pick "sufficiently many" subsets, then their union is almost the entire universe. This is formalized below.

Definition 2.20 (Disperser)**.** *For any universe $\mathcal{U}$, a collection $\mathcal{S}$ of subsets of $\mathcal{U}$ is an (r, η)-*disperser *if, for any r distinct subsets $S^1, \dots, S^r \subseteq \mathcal{S}$, we have*

$$\left| \bigcup_{i=1}^{r} S^i \right| \geqslant (1-\eta)|\mathcal{U}|.$$

In Chapter 9, we will in fact need a stronger property that, when we pick "sufficiently many" *subcollections*, then the union of the intersection of each subcollection is almost the entire universe. This is encapsulated in the definition of what we call an *intersection disperser*:

Definition 2.21 (Intersection Disperser)**.** *For any universe $\mathcal{U}$, a collection $\mathcal{S}$ of subsets of $\mathcal{U}$ is an (r, ℓ, η)-*intersection disperser *if, for any r disjoint subcollections $\mathcal{S}^1, \dots, \mathcal{S}^r \subseteq \mathcal{S}$ each of size at most ℓ, we have*

$$\left| \bigcup_{i=1}^{r} \left(\bigcap_{S \in \mathcal{S}^i} S \right) \right| \geqslant (1-\eta)|\mathcal{U}|.$$

[7] Even though dispersers are often described in terms of graphs or distributions in literatures (see, e.g., [Vad12]), it is more convenient for us to describe it in terms of subsets.

When $\ell = 1$, the definition of intersection dispersers is exactly the same as that of dispersers. Note also that in Definition 2.21 we require $\mathcal{S}^1, \dots, \mathcal{S}^r$ to be disjoint. This is necessary because otherwise we can include a common set $S \in \mathcal{S}$ into all the subcollections. In this case, the union will be contained in S and hence will not cover almost all the universe.

For the subsets in Definition 2.16, we get the following parameters for intersection dispersers:

Proposition 2.22. *For any $\eta > 0$ and $\ell \in \mathbb{N}$, the collection $\{T_1, \dots, T_k\}$ is a $(\lceil \ln(1/\eta)/\alpha^\ell \rceil, \ell, \eta)$-intersection disperser (with respect to the universe $[q]^k$).*

Proof. Consider any subcollections $\mathcal{T}^1, \cdots, \mathcal{T}^r$ each of size ℓ, where $r = \lceil \ln(1/\eta)/\alpha^\ell \rceil$. For every $i \in [r]$, let Y_i denote the indicator variable whether u belongs to $\bigcap_{T \in \mathcal{T}^i} T$ Notice that, since $\mathcal{T}^1, \dots, \mathcal{T}^r$ are disjoint ℓ-size subcollections, $Y_1, \dots, Y_r$ are i.i.d. with mean α^ℓ. Observe also that u belongs to $\bigcup_{i=1}^r \left(\bigcap_{T \in \mathcal{T}^i} T\right)$ iff $Y_1 + \cdots + Y_r \geqslant 1$. Hence, we have

$$\Pr[Y_1 + \cdots + Y_r \geqslant 1] = 1 - \Pr[Y_1 = 0, \dots, Y_r = 0] = 1 - (1 - \alpha^\ell)^r \geqslant 1 - e^{-\alpha^\ell \cdot r} \geqslant 1 - \eta.$$

As a result, we have $\left|\bigcup_{i=1}^r \left(\bigcap_{T \in \mathcal{T}^i} T\right)\right| > (1 - \eta) q^k$ as desired. □

2.10 Two Variants of Label Covers

While a standard version of label cover is to find an assignment with maximum value (as specified in Definition 2.3), other objectives will also be useful for us in order to prove hardness of approximation of various problems. In particular, we will be working with two additional versions of label cover: the Max-Cover variant and the Min-Label variant. For both problems, the input is still the usual label cover instance, but the objectives are different, as defined below.

Max-Cover Problem. Here we use similar notations as in Definition 2.3.

We say that a labeling σ *covers* a vertex u if every edge incident to u is satisfied by σ. Let $\textsc{MaxCov}(\mathcal{L})$ denote the maximum number of vertices in U that can be covered by a labeling, i.e.,

$$\textsc{MaxCov}(\mathcal{L}) := \max_{\sigma_U : U \to \Sigma_U, \sigma_V : V \to \Sigma_V} |\{u \in U \mid (\sigma_U, \sigma_V) \text{ covers } u\}|.$$

The goal of the Max-Cover problem is to compute $\textsc{MaxCov}(\mathcal{L})$.

Min-Label Problem. A *multi-labeling* of $\mathcal{L}$, is a pair of mappings $\sigma_U : U \to \Sigma_U$ and $\hat{\sigma}_V : V \to \mathscr{P}(\Sigma_V)$. We say that an edge (u, v) is satisfied by $(\sigma_U, \hat{\sigma}_V)$, if there exists $\beta \in \hat{\sigma}_V(v)$ such that $(\sigma(u), \beta) \in \Pi_{uv}$. Similar to before, We say that $(\sigma_U, \hat{\sigma}_V)$ covers a vertex u if it satisfies every edge incident to u. For any label cover instance $\mathcal{L}$, let $\textsc{MinLab}(\mathcal{L})$ denote the minimum number of labels needed to assign to vertices in V in order to cover *all* vertices in U, i.e.,

$$\textsc{MinLab}(\mathcal{L}) := \min_{\sigma_U : U \to \Sigma_U, \sigma_V : V \to \Sigma_V} \sum_{v \in V} |\hat{\sigma}_V(v)|$$

where the minimization is over multi-labelings $(\sigma_U, \hat{\sigma}_V)$ that covers every $u \in U$. For brevity, we sometimes refer to $\sum_{v \in V} |\hat{\sigma}_V(v)|$ as the *size* of the multi-labeling $(\sigma_U, \hat{\sigma}_V)$

In other words, the different between $\textsc{MaxCov}$ and $\textsc{MinLab}$ is that, for $\textsc{MaxCov}$ we are allowed to only pick one label for each vertex $v \in V$ and would like to cover any many vertices $u \in U$ as possible, whereas for $\textsc{MinLab}$ we have to cover every $u \in U$ but we would like to minimize the total number of labels assigned to the vertices in V.

Let us end this section by stating two facts that relate the different objectives of a label cover instance. The first is a relationship between $\textsc{MaxCov}(\mathcal{L})$ and $\textsc{MinLab}(\mathcal{L})$:

Lemma 2.23. *Let $\mathcal{L} = (G = (U, V, E), \Sigma_U, \Sigma_V, \Pi)$ be any label cover instance. Then, we have*

- *If* $\textsc{MaxCov}(\mathcal{L}) = |U|$, *then* $\textsc{MinLab}(\mathcal{L}) = |V|$.
- *If* $\textsc{MaxCov}(\mathcal{L}) \leqslant \varepsilon|U|$ *for some* $\varepsilon > 0$, *then* $\textsc{MinLab}(\mathcal{L}) \geqslant (1/\varepsilon)^{1/|V|} \cdot |V|$.
- *If* $\textsc{MaxCov}(\mathcal{L}) \leqslant \varepsilon|U|$ *for some* $\varepsilon > 0$ *and* (U, V, E) *is a bi-regular with left degree* d_U, *then* $\textsc{MinLab}(\mathcal{L}) \geqslant (1/\varepsilon)^{1/d_U} \cdot |V|$.

Proof.

- Suppose $\textsc{MaxCov}(\mathcal{L}) = |U|$, i.e., that some labeling (σ_U, σ_V) covers every node in U. Hence, (σ_U, σ_V) is also a multi-labeling that covers every node in U, which implies that $\textsc{MinLab}(\mathcal{L}) = |V|$.
- We prove by contrapositive. Assume that $\textsc{MinLab}(\mathcal{L}) < (1/\varepsilon)^{1/|V|}|V|$. Then there exists a multi-labeling $(\sigma_U, \hat{\sigma}_V)$ of size less than $(1/\varepsilon)^{1/|V|}|V|$ that covers every node in $|U|$. Let us construct a labeling (σ_U, σ_V) by uniformly and independently choosing one label from each $\hat{\sigma}_V(v)$ at random, for each $v \in V$.

  Thus, the expected number of left nodes covered by (σ_U, σ_V) is

$$\begin{aligned}
\mathbb{E}_{\sigma_V}\left[|\{u \in U : (\sigma_U, \sigma_V) \text{ covers } u\}|\right] &\geqslant \sum_{u \in U} \prod_{v \in N_G(u)} |\hat{\sigma}_V(v)|^{-1} \\
&\geqslant \sum_{u \in U} \prod_{v \in V} |\hat{\sigma}_V(v)|^{-1} \\
\text{(By AM-GM inequality)} &\geqslant \sum_{u \in U} \left(\frac{1}{|V|} \sum_{v \in V} |\hat{\sigma}_V(v)|\right)^{-|V|} \\
&> |U| \cdot \left(\frac{1}{|V|} \cdot \left(\left(\frac{1}{\varepsilon}\right)^{1/|V|} |V|\right)\right)^{-|V|} \\
&= |U| \cdot \varepsilon
\end{aligned}$$

  Hence, there is a labeling that covers $> \varepsilon|U|$ nodes in U, i.e., $\textsc{MaxCov}(\mathcal{L}) > \varepsilon|U|$.

- Similar to the previous item, we assume contrapositively that $\textsc{MinLab}(\mathcal{L}) < (1/\varepsilon)^{1/d_U}|V|$. Let d_V denote the right degree of (U, V, E). Again, let $(\sigma_U, \hat{\sigma}_V)$ be a multi-labeling of size less than $(1/\varepsilon)^{1/d_U}|V|$ that covers every node in $|U|$, and let (σ_U, σ_V) denote a labeling where $\sigma_V(v)$ is chosen uniformly at random from $\hat{\sigma}_V(v)$. The expected number of left nodes covered by (σ_U, σ_V) is

$$\begin{aligned}
\mathbb{E}_{\sigma_V}\left[|\{u \in U : (\sigma_U, \sigma_V) \text{ covers } u\}|\right] &\geqslant \sum_{u \in U} \prod_{v \in N_G(u)} |\hat{\sigma}_V(v)|^{-1} \\
\text{(By AM-GM inequality)} &\geqslant \sum_{u \in U} \left(\frac{1}{d_U} \sum_{v \in N_G(u)} |\hat{\sigma}_V(v)|\right)^{-d_U} \\
\text{(By Power Mean inequality)} &\geqslant |U| \cdot \left(\frac{1}{|U|} \sum_{u \in U} \left(\frac{1}{d_U} \sum_{v \in N_G(u)} |\hat{\sigma}_V(v)|\right)^{-1}\right)^{d_U} \\
\text{(By Cauchy-Schwarz inequality)} &\geqslant |U| \cdot \left(\frac{|U|}{\sum_{u \in U}\left(\frac{1}{d_U}\sum_{v \in N_G(u)} |\hat{\sigma}_V(v)|\right)}\right)^{d_U} \\
\text{(By bi-regularity of } (U, V, E)) &= |U| \cdot \left(\frac{|U|}{\frac{d_V}{d_U}\sum_{v \in V} |\hat{\sigma}_V(v)|}\right)^{d_U} \\
&> |U| \cdot \left(\frac{|U|}{\frac{d_V}{d_U} \cdot (1/\varepsilon)^{1/d_U} |V|}\right)^{d_U} \\
\text{(Since } d_U \cdot |U| = |E| = d_V \cdot |V|) &= |U| \cdot \left(\varepsilon^{1/d_U}\right)^{d_U} \\
&= |U| \cdot \varepsilon
\end{aligned}$$

Hence, $\text{MaxCov}(\mathcal{L}) > \varepsilon|U|$ as desired. □

The other fact is a simple observation relating $\text{MaxCov}(\mathcal{L})$ and $\text{val}(\mathcal{L})$.

Observation 2.24. *Let* $\mathcal{L} = (G = (U, V, E), \Sigma_U, \Sigma_V, \Pi)$ *be any label cover instance. Then, we have*

- *If* $\text{val}(\mathcal{L}) = 1$, *then* $\text{MaxCov}(\mathcal{L}) = |U|$.
- *If* $\text{val}(\mathcal{L}) \leqslant 1 - \varepsilon$ *for some* $\varepsilon > 0$ *and that* G *is left-regular, then* $\text{MaxCov}(\mathcal{L}) \leqslant 1 - \varepsilon$.

Proof.

- Suppose that $\text{val}(\mathcal{L}) = 1$; there exists a labeling (σ_U, σ_V) that satisfies all the edges, and hence covers all left vertices. Hence, $\text{MaxCov}(\mathcal{L}) = |U|$.
- Suppose that $\text{val}(\mathcal{L}) \leqslant 1 - \varepsilon$ and G is left-regular. Since $\text{val}(\mathcal{L}) \leqslant 1 - \varepsilon$, any labeling (σ_U, σ_V) satisfies at most $1 - \varepsilon$ fraction of edges in E. Moreover, because the graph G is left-regular, the unsatisfied edges must be adjacent to at least $\varepsilon|U|$ left vertices; these vertices are not covered by (σ_U, σ_V). Hence, we have $\text{MaxCov}(\mathcal{L}) \leqslant (1 - \varepsilon)|U|$ as desired. □

2.11 Feige's Reduction From Label Cover to Set Cover

In [Fei98], Feige shows the hardness of approximating SET COVER and MAXIMUM k-COVERAGE, by reductions from label cover instances. Here we observe that his reduction for MAXIMUM k-COVERAGE in fact gives a reduction from MINLAB to SET COVER, as stated below.

Lemma 2.25 (Reduction from MINLAB to SETCOV)**.** *There is an algorithm that, given a label cover instance $\mathcal{L} = (U, V, E, \Sigma_U, \Sigma_V, \{\Pi_e\}_{e\in E})$, outputs a* SETCOV *instance $(\mathcal{U}, \mathcal{S}, k = |V|)$ where*

- MINLABEL$(\mathcal{L})$ = SETCOV$(\mathcal{U}, \mathcal{S})$.
- $|\mathcal{S}| = |V| \cdot |\Sigma_V|$.
- $|\mathcal{U}| \leqslant |U| \cdot |V|^{|\Sigma_U|}$.
- *The reduction runs in time $O\left(|\mathcal{S}| \cdot |\mathcal{U}|\right)$.*

Proof. Our construction is based on a standard hypercube set system, as used by Feige [Fei98] in proving the hardness of the *k-Maximum Coverage* problem. We explain it here for completeness.

Hypercube set system: Let $z, k \in \mathbb{N}$ be parameters. The hypercube set system $H(z, k)$ is a set system $(\mathcal{U}, \mathcal{S})$ with the ground set $\mathcal{U} = [z]^k$. We view each element of $\mathcal{U}$ as a length-k vector $\mathbf{x}$ where each coordinate assumes a value in $[z]$. There is a collection of *canonical sets* $\mathcal{S} = \{X_{i,a}\}_{i\in[z],a\in[k]}$ defined as

$$X_{i,a} = \{\mathbf{x} : x_a = i\}$$

In other words, each set $X_{i,a}$ contains the vectors whose a^{th} coordinate is i. A nice property of this set system is that, it can only be covered completely if all canonical sets corresponding to some a^{th} coordinate are chosen.

Proposition 2.26. *Consider any sub-collection $\mathcal{S}' \subseteq \mathcal{S}$. We have $\bigcup \mathcal{S}' = \mathcal{U}$ if and only if there is a value $a \in [k]$ for which $X_{1,a}, X_{2,a}, \ldots, X_{z,a} \in \mathcal{S}'$.*

Proof. The if part is obvious. For the "only if" part, assume that for each $a \in [k]$, there is a value $i_a \in [z]$ for which $X_{i_a,a}$ is not in $\mathcal{S}'$. Define vector $\mathbf{x}$ by $x_a = i_a$. Notice that $\mathbf{x}$ does not belong to any set in $\mathcal{S}'$ (By definition, if $X_{i',a'}$ contains $\mathbf{x}$, then it must be the case that $x_{a'} = i' = i_{a'}$.) ⌟

The construction: We start from the MINLAB instance $\mathcal{L} = (U, V, E, \Sigma_U, \Sigma_V, \Pi)$. We will create the set system $\mathcal{I} = (\mathcal{U}, \mathcal{S})$. We make $|U|$ different copies of the hypercube set system: For each vertex $u \in U$, we have the hypercube set system $(\mathcal{U}^u, \mathcal{S}^u) = H(N_G(u), \Sigma_U)$, i.e., the ground set $\mathcal{U}^u$ is a copy of $N_G(u)^{\Sigma_U}$ and $\mathcal{S}^u$ contains $|N_G(u)||\Sigma_U|$ "virtual" sets, that we call $\{S^u_{v,a}\}_{v\in N_G(u),a\in\Sigma_U}$ where each such set corresponds to a canonical set of the hypercube. We remark that these virtual sets are not the eligible sets in our instance $\mathcal{I}$. For each vertex $v \in V$, for each label $b \in \Sigma_V$, we define a set

$$S_{v,b} = \bigcup_{u \in N_G(v),(a,b)\in\Pi_{uv}} S^u_{v,a}$$

The set system $(\mathcal{U}, \mathcal{S})$ in our instance is simply:

$$\mathcal{U} = \bigcup_{u \in U} \mathcal{U}^u \quad \text{and} \quad \mathcal{S} = \{S_{v,b} : v \in V, b \in \Sigma_V\}$$

Notice that the number of sets is $|V||\Sigma_V|$ and the number of elements in the ground set is $|\mathcal{U}| \leqslant |U||V|^{|\Sigma_U|}$. This completes the description of our instance.

Analysis: We argue that the optimal value of $\mathcal{L}$ is equal to the optimal of $(\mathcal{U}, \mathcal{S})$.

First, we will show that $\textsc{MinLab}(\mathcal{L}) \leqslant \textsc{SetCov}(\mathcal{U}, \mathcal{S})$. Let $(\sigma_U, \hat{\sigma}_V)$ be a feasible MinLab cover for $\mathcal{L}$ (recall that $\hat{\sigma}_V$ is a multi-labeling, while σ_U is a labeling.) For each $v \in V$, the SetCov solution chooses the set $S_{v,b}$ for all $b \in \hat{\sigma}_V(v)$. Denote this solution by $\mathcal{S}' \subseteq \mathcal{S}$. The total number of sets chosen is exactly $\sum_v |\hat{\sigma}(v)|$, exactly matching the cost of $\textsc{MinLab}(\mathcal{L})$. We argue that this is a feasible set cover: For each u, the fact that u is covered by $(\sigma_U, \hat{\sigma}_V)$ implies that, for all $v \in N_G(u)$, there is a label $b_v \in \hat{\sigma}_V(v)$ such that $(\sigma_U(u), b_v) \in \Pi_{uv}$. Notice that $S^u_{v,\sigma_U(u)} \subseteq S_{v,b_v} \in \mathcal{S}'$ for every $v \in N_G(u)$, so we have

$$\bigcup_{S \in \mathcal{S}'} S \supseteq \bigcup_{v \in N_G(u)} S_{v,b_v} \supseteq \bigcup_{v \in N_G(u)} S^u_{v,\sigma_U(u)} = \mathcal{U}^u$$

where the last equality comes from Chapter 2.26. In other words, $\mathcal{S}'$ covers all elements in $\mathcal{U}^u$. Hence, $\mathcal{S}'$ is indeed a valid SetCov solution for $(\mathcal{U}, \mathcal{S})$.

To prove the converse, consider a collection of sets $\{S_{v,b}\}_{(v,b)\in\Lambda}$ that covers the whole universe $\mathcal{U}$. We define the (multi-)labeling $\hat{\sigma}_V : V \to 2^{\Sigma_V}$ where $\hat{\sigma}_V(v) = \{b : (v,b) \in \Lambda\}$ for each $v \in V$. Clearly, $\sum_{v \in V} |\hat{\sigma}_V(v)| = |\Lambda|$, so the cost of $\hat{\sigma}_V$ as a solution for MinLab is exactly the cost of SetCov. We verify that all left vertices $u \in U$ of $\mathcal{L}$ are covered (and along the way will define $\sigma_U(u)$ for all $u \in U$.) Consider each vertex $u \in U$. The fact that the ground elements in $\mathcal{U}^u$ are covered implies that (from Proposition 2.26) there is a label $a_u \in \Sigma_U$ where all virtual sets $\{S^u_{v,a_u}\}_{v \in N_G(u)}$ are included in the solution. Therefore, for each $v \in N_G(u)$, there must be a label $b_v \in \hat{\sigma}_V(v)$ such that $a_u b_v \in \Pi_{uv}$. We simply define $\sigma_U(u) = a_u$. Therefore, the vertex u is covered by the assignment $(\sigma_U, \hat{\sigma}_V)$. □

It will also be convenient to also state the reduction in terms of MaxCov instead of MinLab. In particular, by combining Lemma 2.25 and Lemma 2.23, we have the following:

Lemma 2.27 (Reduction from MaxCov to SetCov). *There is an algorithm that, given a label cover instance $\mathcal{L} = (U, V, E, \Sigma_U, \Sigma_V, \{\Pi_e\}_{e\in E})$, outputs a* SetCov *instance $(\mathcal{U}, \mathcal{S}, k = |V|)$ where*

- *If* $\textsc{MaxCov}(\mathcal{L}) = |U|$, *then* $\textsc{SetCov}(\mathcal{U}, \mathcal{S}) = k$.

- *If* $\textsc{MaxCov}(\mathcal{L}) \leqslant \varepsilon \cdot |U|$*, then* $\textsc{SetCov}(\mathcal{U}, \mathcal{S}) \geqslant (1/\varepsilon)^{1/k} \cdot k$*.*
- $|\mathcal{S}| = |V| \cdot |\Sigma_V|$.
- $|\mathcal{U}| \leqslant |U| \cdot |V|^{|\Sigma_U|}$.
- *The reduction runs in time* $O\left(|\mathcal{S}| \cdot |\mathcal{U}|\right)$.

Finally, we note that it is a well-known fact that we can reduce SETCOV to DOMSET by constructing a graph whose vertices are $\mathcal{U} \cup \mathcal{S}$; there are edges between every pairs of subsets, and there is an edge between $S \in \mathcal{S}$ to $u \in \mathcal{U}$ iff u belongs to S. It is obvious to see that the optimum of the DOMSET instance is exactly the same as that of the original SETCOV instance.

Part I

Problems Between P and NP

Chapter 3

A Birthday Repetition Theorem and Its Applications

Polynomial-time reductions between computational problems are among the central tools in complexity theory. The rich and vast theory of hardness of approximation emerged out of the celebrated PCP Theorem [Aro+98; AS98] and the intricate web of polynomial-time reductions developed over the past two decades. During this period, an extensive set of reduction techniques such as parallel repetition and long-codes have been proposed and a variety of mathematical tools including discrete harmonic analysis, information theory and Gaussian isoperimetry have been applied towards analyzing these reductions. These developments have led to an almost complete understanding of the approximability of many fundamental combinatorial optimization problems like SET COVER and MAX 3SAT. Yet, there are a few central problems such as computing approximate Nash equlibria, the DENSEST k-SUBGRAPH problem and the SMALL SET EXPANSION problem, that remain out of reach of the web of polynomial-time reductions.

A promising new line of work proposes to understand the complexity of these problems through the lens of *sub-exponential time reductions*. Specifically, the idea is to construct a sub-exponential time reduction from 3SAT to the problem at hand, say, the Approximate Nash Equilibrium problem. Assuming that 3SAT does not admit sub-exponential time algorithms (also known as the Exponential Time Hypothesis (ETH) [IP01; IPZ01]), this would rule out polynomial time algorithms for the Approximate Nash Equilibrium problem.

At the heart of this line of works, lies the so-called *birthday repetition* of two-prover games. To elaborate on this, we begin by formally defining the notion of two-prover games[1].

Definition 3.1. *(Two-prover game) A two prover game $\mathcal{G}$ consists of*

- *A finite set of questions X, Y and corresponding answer sets Σ_X, Σ_Y.*
- *A distribution $\mathcal{Q}$ over pairs of questions $X \times Y$.*

[1] The definition of two-prover games is in fact equivalent to that of Label Cover in Definition 2.3. However, for the purpose of describing parallel and birthday repetitions, two-prover game interpretation is more natural.

- *A verification function* $P : X \times Y \times \Sigma_X \times \Sigma_Y \to \{0, 1\}$.

The value of the game is the maximum over all strategies $\phi : X \cup Y \to \Sigma_X \cup \Sigma_Y$ *of the output of the verification function, i.e.,* $val(\mathcal{G}) = \max_{\phi: X\cup Y \to \Sigma_X \cup \Sigma_Y} \mathbb{E}_{(x,y)\sim\mathcal{Q}}[P(x, y, \phi(x), \phi(y))]$.

Two prover games earn their name from the following interpretation of the above definition: The game $\mathcal{G}$ is played between a verifier V and two cooperating provers $Merlin_1$ and $Merlin_2$ who have agreed upon a common strategy, but cannot communicate with each other during the game. The verifier samples two questions $(x, y) \sim \mathcal{Q}$ and sends x to $Merlin_1$ and y to $Merlin_2$. The provers respond with answers $\phi(x)$ and $\phi(y)$, which the verifier accepts or rejects based on the value of the verifiaction function $P(x, y, \phi(x), \phi(y))$.

Two-prover games and, more specifically, a special class of two-prover games known as the LABEL COVER problem are the starting points for reductions in a large body of hardness of approximation results. The PCP theorem implies that for some absolute constant ε_0, approximating the value of a two prover game to within an additive ε_0 is $\mathbf{NP}$-hard. However, this hardness result on its own is inadequate to construct reductions to other combinatorial optimization problems. To this end, this hardness result can be strengthened to imply that it is $\mathbf{NP}$-hard to approximate the value of two-prover games to any constant factor, using the *parallel repetition theorem*.

For an integer k, the k-wise parallel repetition $\mathcal{G}^{\otimes k}$ of a game $\mathcal{G}$ can be described as follows. The question and answer sets in $\mathcal{G}^{\otimes k}$ consist of k-tuples of questions and answers from $\mathcal{G}$. The distribution over questions in $\mathcal{G}^{\otimes k}$ is given by the product distribution $\mathcal{Q}^k$. The verifier for $\mathcal{G}^{\otimes k}$ accepts the answers if and only if the verifier for $\mathcal{G}$ accepts each of the k individual answers.

Roughly speaking, the parallel repetition theorem asserts that the value of the repeated game $\mathcal{G}^k$ decays exponentially in k. Parallel repetition theorems form a key ingredient in obtaining tight hardness of approximation results, and have aptly received considerable attention in literature [Raz98; Hol09; Rao11; DS14; Mos14; BG15].

Birthday repetition, introduced by Aaronson et al. [AIM14], is an alternate transformation on two-prover games defined as follows.

Definition 3.2. *(Birthday Repetition) The* $(k \times \ell)$*-birthday repetition of a two-prover game* $\mathcal{G}$ *consists of*

- *The set of questions in* $\mathcal{G}^{k\times\ell}$ *are* $\binom{X}{k}$ *and* $\binom{Y}{\ell}$ *respectively, i.e., each question is a subset* $S \subseteq X$ *of size* k *and subset* $T \subseteq Y$ *of size* ℓ.
- *The distribution over questions is the uniform product distribution over* $\binom{X}{k} \times \binom{Y}{\ell}$.
- *The verifier accepts only if, for every pair of* $(x, y) \in S \times T$ *such that* (x, y) *form a valid pair of questions in* $\mathcal{G}$*, i.e.,* $(x, y) \in \operatorname{supp}(\mathcal{Q})$*, the answers to* x *and* y *are accepted in the original game* $\mathcal{G}$.

The basic idea of birthday repetition can be traced back to the work of Aaronson et al. [Aar+09] on quantum multiprover proof systems $\mathbf{QMA}(k)$ for 3SAT. Subsequent work by Aaronson et al. [AIM14] on the classical analogue of $\mathbf{QMA}(k)$, namely $\mathsf{AM}(k)$, formally defined birthday repetition for two-prover games, and set the stage for applications in hardness of approximation.

Unlike parallel repetition, birthday repetition is only effective for large values of k and ℓ. In particular, if $k, \ell < o(\sqrt{|X| + |Y|})$, then, for most pairs of S and T, there is no pair of questions $(x, y) \in S \times T$ such that (x, y) belongs to the support of the questions in the original game.

However, if we pick $k = \ell = \omega(\sqrt{n})$ where $n = |X| + |Y|$, then by the birthday paradox, with high probability the sets S, T contain an edge (x, y) from the original game $\mathcal{G}$. Hence, for this choice of k and ℓ, the game played by the provers is seemingly at least as difficult to succeed, as the original game $\mathcal{G}$. Aaronson et al. [AIM14] confirmed this intuition by proving the following theorem.

Theorem 3.3. *[AIM14] For any two-prover game $\mathcal{G}$ such that $\mathcal{Q}$ is uniform over its support, if the bipartite graph induced by $(X, Y, \mathrm{supp}(\mathcal{Q}))$ is biregular, then $val(\mathcal{G}^{k\times\ell}) \leqslant val(\mathcal{G}) + O(\sqrt{\frac{n}{k\ell}})$.*

On the one hand, birthday repetition is ineffective in that it has to incur a blowup of $2^{\sqrt{n}}$ in the size, to even simulate the original game $\mathcal{G}$. The distinct advantage of birthday repetition is that the resulting game $\mathcal{G}^{k,\ell}$ has a distinct structure – in that it is a *free game*.

Definition 3.4. *(Free game) A free game is a two-player game $\mathcal{G} = (X, Y, \mathcal{Q}, \Sigma_X, \Sigma_Y, P)$ such that $\mathcal{Q}$ is the uniform distribution over $X \times Y$.*

The birthday repetition theorem of Aaronson et al. [AIM14] immediately implies a hardness of approximation for the value of free games. Specifically, they show that it is ETH-hard to approximate free games to some constant ratio in almost quasi-polynomial time. Interestingly, this lower bound is nearly tight in that free games admit a quasipolynomial time approximation scheme (QPTAS) [Bar+11; AIM14].

Following Aaronson et al.'s work, birthday repetition has received numerous applications, which can be broadly classified in to two main themes. On the one hand, there are problems such as computing approximate Nash equilibria [BKW15; BPR16], approximating free games [AIM14], and approximate symmetric signaling in zero sum games [Rub17b], where the underlying problems admit quasipolynomial-time algorithms [Dug14; LMM03; FS97] and birthday repetition can be used to show that such a running time is necessary, assuming ETH. On the other hand, there are computational problems like Densest k-Subgraph [Bra+17], injective tensor norms [Aar+09; HM13; Bar+12], 2-to-4-norms [Aar+09; HM13; Bar+12] wherein an $\mathbf{NP}$-hardness of approximation result seems out of reach of current techniques. But the framework of birthday repetition can be employed to show a quasi-polynomial hardness assuming ETH[2].

Unlike the parallel repetition theorem, the birthday repetition theorem of [AIM14] does not achieve any reduction in the value of the game. It is thus natural to ask whether birthday repetition can be used to decrease the value of a game, just like parallel repetition. Aaronson et al. conjectured not only that the value of the game deteriorates with birthday repetition, but also that it decreases

[2] Although the hardness results for injective tensor norms and 2-to-4-norms build over quantum multiprover proof systems, the basic idea of birthday repetition [Aar+09] lies at the heart of these reductions.

exponentially in $\Omega(k\ell/n)$. Notice that the expected number of edges between S and T in birthday repetition is $\Theta(k\ell/n)$.

The main technical contribution of this chapter is that we resolve the conjecture positively by showing the following theorem.

Theorem 3.5. *(Birthday Repetition Theorem (informal); See Theorem 3.15) Let $\mathcal{G} = (X, Y, \mathcal{Q}, \Sigma_X, \Sigma_Y, P)$ be a two-prover game such that $\mathcal{Q}$ is uniform over its support, $(X, Y, \mathrm{supp}(\mathcal{Q}))$ is biregular. If $val(\mathcal{G}) \leqslant 1 - \varepsilon$ for some $\varepsilon \geqslant 1/n$, then*

$$val(\mathcal{G}^{k\times\ell}) \leqslant (1-\varepsilon/2)^{\Omega\left(\frac{k\ell}{n\log(1/\varepsilon)}\right)}.$$

Our theorem is, in fact, more general than stated above and can handle non-biregular graphs as well as the case $\varepsilon < 1/n$, albeit with some loss in parameters (see Theorem 3.15).

We remark that Theorem 3.15 contains quantitative improvements over the corresponding theorem in the conference version of this work [MR17a]. In particular, the birthday repetition theorem of [MR17a] has a factor of ε^5 in the exponent instead of $1/\log(1/\varepsilon)$ in Theorem 3.15. To achieve this improvement, we use a completely different technique compared to that in [MR17a] based on counting number of bicliques and Alon's lemmas (Lemmas 2.14 and 2.15).

We also note that quantitatively Theorem 3.15 matches that of a follow-up work of Ko [Ko18], with an advantage that Theorem 3.15 applies to any two-prover games whereas Ko's technique only applies to projection games.

By definition, the birthday repetition theorem almost immediately implies a hardness of approximation result for the value of a free game.

Corollary 3.6. *Unless ETH is false, no polynomial time algorithm can approximate the value of a free game to within a factor of $2^{\widetilde{\Omega}(\log(nq))}$ where n is the number of questions and q is the alphabet (answer set) size.*

The above hardness result improves upon polylog(nq) ratio achieved in [AIM14] and is tight up to a factor of $\mathrm{polyloglog}(nq)$ in the exponent since there exists a polynomial-time algorithm that achieves $O(q^\varepsilon)$ approximation for every constant $\varepsilon > 0$ [AIM14; MM15].

Dense CSPs

A free game can be considered an instance of 2-ary constraint satisfaction problems. From this perspective, free games are *dense*, in that there are constraints between a constant fraction of all pairs of variables. As an application of our birthday repetition theorem, we will show almost-tight hardness of approximation results for dense CSPs. To this end, we begin by defining MAX k-CSP and its density.

Definition 3.7. *(MAX k-CSP) A MAX k-CSP instance $\mathcal{G}$ consists of*

- *A finite set of variables V and a finite alphabet set Σ.*

- *A distribution $\mathcal{Q}$ over k-size subsets of variables $\binom{V}{k}$.*
- *A predicate $P : \binom{V}{k} \times \Sigma^k \to [0, 1]$.*

The value of the instance is the maximum over all assignments $\phi : V \to \Sigma$ of the expected output of the predicate, i.e., $\mathrm{val}(\mathcal{G}) = \max_{\phi : V \to \Sigma} \mathbb{E}_{S \sim \mathcal{Q}}[P(S, \phi|_S)]$ *where $\phi|_S$ is the restriction of ϕ to S.*

Finally, an instance is called Δ-dense if $\Delta \cdot \mathcal{Q}(S) \leqslant \binom{|V|}{k}$ for every $S \in \binom{V}{k}$. Fully-dense instances are defined to be simply the 1-dense instances.

There has been a long line of works on approximating dense CSPs. Arora, Karger and Karpinski were first to devise a polynomial-time approximation scheme for the problem when alphabet size is constant [AKK95]. Since then, numerous algorithms have been invented for approximating dense CSPs; these algorithms use wide ranges of techniques such as combinatorial algorithms with exhaustive sampling [AKK95; Veg+05; MS08; Yar14; MM15; FLP16], subsampling of instances [Alo+03; Bar+11], regularity lemmas [FK96; CCF10] and linear and semidefinite program hierarchies [VK07; BRS11; GS11; YZ14]. Among the known algorithms, the fastest is that of Yaroslavtsev [Yar14] that achieves approximation ratio $(1 + \varepsilon)$ in $q^{O_k(\log q/\varepsilon^2)} + (nq)^{O(1)}$ time[3] where n and q denote the number of variables and alphabet size respectively.

Unfortunately, when q is (almost-)polynomial in n, none of the mentioned algorithms run in polynomial time. CSPs in such regime of parameters have long been studied in hardness of approximation (e.g. [Bel+93; RS97; AS03; Din+11; MR10; Mos12]) and have recently received more attention from the approximation algorithm standpoint, both in the general case [Pel07; CHK11; MM17] and the dense case [MM15]. The approximabilities of these two cases are vastly different. In the general case, it is known that, for some constant $k > 0$, approximating Max k-CSP to within a factor of $2^{\log^{1-\varepsilon}(nq)}$ is $\mathbf{NP}$-hard for any constant $\varepsilon > 0$ [Din+11]. Moreover, the long-standing Sliding Scale Conjecture of Bellare et al. [Bel+93] states there are constants $k, \varepsilon > 0$ such that it is $\mathbf{NP}$-hard to approximate Max k-CSP to within a factor of $(nq)^{\varepsilon}$. On the other hand, aforementioned algorithms for dense CSPs rule out such hardnesses for the dense case.

While the gap between known approximation algorithms and inapproximability results in the general case is tiny ($2^{\log^{\varepsilon}(nq)}$ for any constant $\varepsilon > 0$), the story is different for the dense case, especially when we restrict ourselves to polynomial-time algorithms. Aaronson et al.'s result only rules out, assuming ETH, polylog(nq) factor approximation for such algorithms [AIM14]. However, for $k > 2$, no non-trivial polynomial time algorithm for dense Max k-CSP on large alphabet is even known. We settle down the complexity of approximating dense Max k-CSP almost completely by answering the following fine-grained question: "for each $i \in \mathbb{N}$, what is the best approximation for dense Max k-CSP, achievable by algorithms running in time $(nq)^i$?".

Manurangsi and Moshkovitz developed an algorithm for dense Max 2-CSP that, when the instance has value $1 - \varepsilon$, can approximate the value to within a factor of $O(q^{1/i}/(1 - \varepsilon)^i)$ in

[3] [Yar14] states that the algorithm takes $q^{O_k(1/\varepsilon^2)} + (nq)^{O(1)}$ time but it in fact takes $q^{O_k(\log q/\varepsilon^2)} + (nq)^{O(1)}$ time [Yar16].

$(nq)^{O(i)}$ time [MM15][4]. Due to the algorithm's combinatorial nature, it is unclear whether the algorithm can be extended to handle dense MAX k-CSPs when $k > 2$.

Using a conditioning-based rounding technique developed in [BRS11; RT12; YZ14], we show that the Sherali-Adams relaxation exhibits a similar approximation even when $k > 2$, as stated below.

Theorem 3.8. *(Informal; See Theorem 3.29) For every $i > 0$ and any dense* MAX k-CSP *instance of value $1-\varepsilon$, an $O_{k,\varepsilon}(i/\Delta)$-level of the Sherali-Adams relaxation yields an $O(q^{1/i})$-approximation for the instance.*

Using our birthday repetition theorem, we show that it is impossible to improve the above tradeoff between run-time and approximation ratio using the sum-of-squares SDP hierarchy (aka the Lasserre SDP hierarchy). Specifically, we use birthday repetition on the $\Omega(n)$-level Lasserre integrality gap for MAX 3XOR by Schoenebeck [Sch08] to show the following.

Lemma 3.9. *(Informal; See Lemma 3.23) For every sufficiently large $i > 0$, there is a fully-dense* MAX 2-CSP *instance of value $1/(nq)^{1/i}$ such that the value of $\widetilde{\Omega}(i)$-level Lasserre relaxation is one.*

Instead, if we assume that there exists a constant $\varepsilon > 0$ so that MAX 3SAT cannot be approximated to $1 - \varepsilon$ in sub-exponential time (which we call the Exponential Time Hypothesis for Approximating MAX 3SAT (Gap-ETH)), then we can arrive at the following hardness result.

Lemma 3.10. *(Informal; See Lemma 3.22) Assuming Gap-ETH, for every sufficiently large $i > 0$, no $(nq)^{\widetilde{O}(i)}$-time algorithm can approximate fully-dense* MAX 2-CSP *to within a factor of $(nq)^{1/i}$.*

Thus, assuming Gap-ETH, our hardness result and algorithm resolve complexity of approximating dense CSPs up to a factor of polylog(i) and a dependency on k in the exponent of the running time.

Almost Optimal $\mathsf{AM}(2)$ Protocol for 3SAT

Another interpretation of our improved hardness of approximation of free games is as an improved $\mathsf{AM}(2)$ protocol for 3SAT. The Arthur-Merlin (AM) protocol [Bab85] is a protocol where Arthur (verifier) tosses some random coins and sends the results to Merlin (prover). The prover sends back a proof to Arthur who then decides whether to accept it. Motivated by quantum complexity class $\mathbf{QMA}(k)$, Aaronson et al. [AIM14] proposes a multi-prover version of AM called $\mathsf{AM}(k)$ where there are k non-communicating Merlins[5]. Authur sends an independent random challenge to each Merlin who then sends an answer back to Arthur. Finally, Arthur decides to accept or reject based on the received answers. The protocol is formally defined below.

[4] Note that it is unclear whether Aaronson, Impagliazzo and Moshkovitz's algorithm [AIM14] that achieves a similar guarantee for free games can be extended to handle dense MAX 2-CSP.

[5] $\mathsf{AM}(k)$ is not to be confused with $\mathsf{AM}[k]$ defined in [Bab85]. In $\mathsf{AM}[k]$, there is only one Merlin but Arthur and Merlin are allowed to engage in k rounds of communication.

Definition 3.11. *[AIM14] An $\mathsf{AM}(k)$ protocol for a language $L \subseteq \{0,1\}^*$ of length $p(n) = kq(n)$, completeness $c(n)$, and soundness $s(n)$ consists of a probabilistic polynomial-time verifier V such that*

- *(Completeness) For every $x \in L$, there exists functions $m_1, \dots, m_k : \{0,1\}^{q(n)} \to \{0,1\}^{q(n)}$ such that $\Pr_{y_1,\dots,y_k \sim \{0,1\}^{q(n)}}[V(x, y_1, \dots, y_k, m(y_1), \dots, m(y_k))] \geqslant c(n)$, and,*
- *(Soundness) For every $x \notin L$ and for every function $m_1, \dots, m_k : \{0,1\}^{q(n)} \to \{0,1\}^{q(n)}$, we have $\Pr_{y_1,\dots,y_k \sim \{0,1\}^{q(n)}}[V(x, y_1, \dots, y_k, m(y_1), \dots, m(y_k))] \leqslant s(n)$*

The complexity class $\mathsf{AM}_{p(n)}(k)$ is a set of all languages L such that there exists an $\mathsf{AM}(k)$ protocol of length $p(n)$, completeness 1/3, and soundness 2/3. Finally, the class $\mathsf{AM}(k)$ is defined as $\bigcup_{c \in \mathbb{N}} \mathsf{AM}_{n^c}(k)$.

Similar to the interpretation of a two-prover game as a two-prover protocol, a free game can be viewed as an $\mathsf{AM}(2)$ protocol. Under this view, inapproximabilities of free games translate to $\mathsf{AM}(2)$ protocols whereas approximation algorithms for free games translate to lower bounds on the lengths of $\mathsf{AM}(2)$ protocols.

With this viewpoint, Aaronson et al. constructed, via birthday repetition, an $\mathsf{AM}(2)$ protocol of length $n^{1/2+o(1)}\text{poly}(1/\delta)$ for 3SAT with completeness 1 and soundness δ for every $\delta > 0$. They also showed a lower bound of $\Omega(\sqrt{n \log(1/\delta)})$ on the length of such protocol. Equipped with our birthday repetition theorem, we construct an $\mathsf{AM}(2)$ protocol whose length is optimal up to a factor of polylog$n$.

Lemma 3.12. *For any $\delta > 0$, there is an $\mathsf{AM}(2)$ protocol for 3SAT of length $\widetilde{O}(\sqrt{n \log(1/\delta)})$ with completeness 1 and soundness δ.*

We note that, by picking $\delta = 1/3$, Lemma 3.12 immediately imply 3SAT $\in \mathsf{AM}_{\widetilde{O}(\sqrt{n})}(2)$. Since every problem in $\mathbf{NTIME}(n)$ is reducible to a quasi-linear size 3SAT instance [Coo88], we arrive at the following corollary, resolving the first open question posted in [AIM14].

Corollary 3.13. $\mathbf{NTIME}(n) \subseteq \mathsf{AM}_{\widetilde{O}(\sqrt{n})}(2)$.

Organization of this chapter

The rest of the chapter is organized as follows. In the following section, we provide preliminaries and state notations that we use in this chapter. Then, in Section 3.2, we prove our main theorems. Next, Section 3.3 demonstrates applications of our birthday repetition theorem, including new hardnesses of approximation and Lasserre integrality gap for dense CSPs, and an almost optimal $\mathsf{AM}(2)$ protocol for 3SAT. The algorithm for dense Max k-CSP is described and its approximation guarantee is proved in Section 3.4. Finally, we conclude by proposing open questions and future research directions in Section 3.5.

3.1 Additional Preliminaries and Notations

In this section, we define notations and state some well-known facts that will be subsequently used.

3.1.1 Information Theory

Let us define some information theoretic notions that will be useful in the analysis of our algorithm. The *informational divergence* (aka *Kullback-Leibler divergence*) between two probability distributions $\mathcal{X}$ and $\mathcal{Y}$ is $D_{KL}(\mathcal{X}\|\mathcal{Y}) = \sum_{\theta\in\text{supp}(\mathcal{X})} \mathcal{X}(\theta)\log(\mathcal{X}(\theta)/\mathcal{Y}(\theta))$. Note that, when $\text{supp}(\mathcal{Y}) \nsubseteq \text{supp}(\mathcal{X})$, we let $D_{KL}(\mathcal{X}\|\mathcal{Y}) = \infty$. It is well-known that $D_{KL}(\mathcal{X}\|\mathcal{Y}) \geqslant 0$ for any distributions $\mathcal{X}$ and $\mathcal{Y}$.

The *entropy* of a random variable $x \sim \mathcal{X}$ is defined as $H(x) = -\sum_{\theta\in\text{supp}(\mathcal{X})} \mathcal{X}(\theta)\log\mathcal{X}(\theta)$. For jointly distributed random variables $x_1, \dots, x_n$, the entropy of $x_1, \dots, x_n$ is defined similarly as $H(x_1, \dots, x_n) = -\sum_{(\theta_1,\dots,\theta_n)\in\text{supp}(\mathcal{X}_{1,\dots,n})} \mathcal{X}_{1,\dots,n}(\theta)\log\mathcal{X}_{1,\dots,n}(\theta)$ where $\mathcal{X}_{1,\dots,n}$ is the joint distribution of $x_1, \dots, x_n$.

The conditional entropy $H(x_1, \dots, x_{n-1}|x_n)$ is defined as $\mathbb{E}_{\theta\sim\text{supp}(\mathcal{X}_n)}[H(x_1, \dots, x_{n-1})|x_n = \theta]$ where $\mathcal{X}_n$ is the marginal distribution of x_n.

Last information theoretic measure we will use is the *total correlation* defined as $C(x_1; \dots; x_n) = D_{KL}(\mathcal{X}_{1,\dots,n}\|\mathcal{X}_1\times\cdots\times\mathcal{X}_n)$ where $\mathcal{X}_{1,\dots,n}$ is the joint distribution of $x_1, \dots, x_n$ whereas $\mathcal{X}_1, \dots, \mathcal{X}_n$ are the marginal distributions of $x_1, \dots, x_n$ respectively. We note that the total correlation defined here is always non-negative whereas the mutual information can be negative.

The total correlation is related to entropies and mutual information as follows.

Lemma 3.14. *For any random variables $x_1, \dots, x_n$, $C(x_1; \dots; x_n) = \sum_{i\in[n]} H(x_i) - H(x_1; \dots; x_n)$.*

Finally, similar to conditional entropy and conditional mutual information, we define the conditional total correlation as $C(x_1; \dots; x_{n-1}|x_n) = \mathbb{E}_{\theta\sim\text{supp}(\mathcal{X}_n)}[C(x_1; \dots; x_{n-1})|x_n = \theta]$.

3.1.2 Two-prover Game, Free Game and MAX k-CSP

Two-prover games, free games, and MAX k-CSP are defined in similar manners as in the introduction. However, for convenience, we write the predicates as $P_S(\phi|_S)$ instead of $P(S, \phi|_S)$, and, when $\mathcal{Q}$ is the uniform distribution on Θ, we sometimes write the instance as $(V, \Theta, \{P_S\})$ instead of $(V, \mathcal{Q}, \{P_S\})$. Moreover, for an assignment ϕ of a MAX k-CSP instance $\mathcal{G} = (V, \mathcal{W}, \{P_S\})$, we define its value as $val_{\mathcal{G}}(\phi) = \mathbb{E}_{S\sim\mathcal{W}}[P_S(\phi|_S)]$. When it is clear from the context, we will drop $\mathcal{G}$ and write it simply as $val(\phi)$. Note that $val(\mathcal{G})$ is the maximum of $val_{\mathcal{G}}(\phi)$ among all possible assignments ϕ's. We say that $\mathcal{G}$ is *satisfiable* if its value is one.

We use n to denote the number of variables $|V|$, q to denote the alphabet size $|\Sigma|$ and N to denote the instance size $|\text{supp}(\mathcal{W})|q^k$, the number of bits needed to encode the input if each predicate is a boolean function. Note that, when the instance is fully dense, N is simply $(nq)^k$. Similar notations are also used for two-prover games and free games.

3.1.3 Sherali-Adams and Lasserre Hierarchies

We also consider two hierarchies of linear and semidefinite program relaxations of Max k-CSP. For compactness, we only write down the relaxations of Max k-CSP but do not describe the hierarchies in full details. For interested readers, we refer to Chlamtác and Tulsiani's survey on the topic [CT12].

The first hierarchy we consider is the Sherali-Adams (SA) hierarchy, introduced in [SA90]. An *r-level SA solution* of a Max k-CSP instance $\mathcal{G} = (V, \mathcal{W}, \{P_S\})$ is a collection $\mu = \{\mathcal{X}_S\}_{|S| \leqslant t}$ of distributions $\mathcal{X}_S$ on Σ^S for every subset S of V of size at most r such that, for every $S, T \subseteq V$ of size at most r, the marginal probability of $\mathcal{X}_S$ and $\mathcal{X}_T$ on $\Sigma^{S \cap T}$ agrees. The value of an r-level SA solution μ for $r \geqslant k$ is defined to be $val_{SA}(\mu) = \mathbb{E}_{S \sim \mathcal{W}}[\mathbb{E}_{x_S \sim \mu}[P_S(x_S)]]$ where $\mathbb{E}_{x_S \sim \mu}[P_S(x_S)]$ is a shorthand for $\mathbb{E}_{\phi_S \sim \mathcal{X}_{\{i_1, \dots, i_k\}}}[P_S(\phi_S)]$ when $S = (x_{i_1}, \dots, x_{i_k})$. The optimal of the r-level SA relaxation of $\mathcal{G}$, $opt^r_{SA}(\mathcal{G})$, is defined as the maximum value among all the r-level SA solutions. It is easy to see that finding $opt^r_{SA}(\mathcal{G})$ can be formulated as a linear program with at most $(nq)^{O(r)}$ variables and inequalities and, thus, can be solved in $(nq)^{O(r)}$ time.

Another hierarchy we consider is the Lasserre hierarchy [Las00]. Before stating the Lasserre relaxation for Max k-CSP, we define additional notations regarding assignments. Two assignments $\phi_1 \in \Sigma^{S_1}, \phi_2 \in \Sigma^{S_2}$ are said to be *consistent* if $\phi_1(x) = \phi_2(x)$ for all $x \in S_1 \cap S_2$. The two assignments are said to be *inconsistent* otherwise. More than two assignments are consistent if every pair of the assignments is consistent; otherwise, they are said to be inconsistent. Moreover, for two consistent assignments $\phi_1 \in \Sigma^{S_1}, \phi_2 \in \Sigma^{S_2}$, we define $\phi_1 \circ \phi_2 \in \Sigma^{S_1 \cap S_2}$ by $\phi_1 \circ \phi_2(x) = \phi_1(x)$ if $x \in S_1$ and $\phi_1 \circ \phi_2(x) = \phi_2(x)$ otherwise.

An *r-level Lasserre solution* of an instance $\mathcal{G} = (V, \mathcal{W}, \{P_S\})$ is a collection $\{U_{(S,\phi_S)}\}_{|S| \leqslant r, \phi_S \in \Sigma^S}$ of vectors $U_{(S,\phi_S)}$ for all $S \subseteq V$ of size at most r and assignments ϕ_S of S satisfying the following constraints.

$$\begin{aligned}
\langle U_{(S_1,\phi_1)}, U_{(S_2,\phi_2)} \rangle &\geqslant 0 && \forall S_1, S_2, \phi_1 \phi_2 \\
\langle U_{(S_1,\phi_1)}, U_{(S_2,\phi_2)} \rangle &= \langle U_{(S_3,\phi_3)}, U_{(S_4,\phi_4)} \rangle && \forall S_1 \cup S_2 = S_3 \cup S_4 \text{ and } \phi_1 \circ \phi_2 = \phi_3 \circ \phi_4 \\
\langle U_{(S_1,\phi_1)}, U_{(S_2,\phi_2)} \rangle &= 0 && \forall S_1, S_2, \phi_1 \phi_2 \text{ s.t. } \phi_1, \phi_2 \text{ are inconsistent} \\
\sum_{\sigma \in \Sigma} \|U_{(x,\sigma)}\|^2 &= 1 && \forall x \in V \\
\|U_{(\emptyset,\emptyset)}\| &= 1
\end{aligned}$$

where S_1, S_2, S_3, S_4 are over all subset of V of size at most r and $\phi_1, \phi_2, \phi_3, \phi_4$ are over all assignments of S_1, S_2, S_3, S_4 respectively. The value of an r-level Lasserre solution $\{U_{(S,\phi_S)}\}$ is defined as $val_{Las}(\{U_{(S,\phi_S)}\}) = \mathbb{E}_{S \sim \mathcal{W}}[\sum_{\phi_S \in \Sigma^S} \|U_{(S,\phi_S)}\|^2 P_S(\phi_S)]$. A Lasserre solution is called *complete* if its value is one.

Note that we abuse the notation here as S in $\{U_{(S,\phi_S)}\}$ is a set whereas S in $\mathcal{W}$ is a tuple. Here and elsewhere in this chapter, when we write $U_{(S,\phi_S)}$ for some tuple $S = (x_{i_1}, \dots, x_{i_m})$, this simply refers to $U_{\{x_{i_1}, \dots, x_{i_m}\}, \phi_S}$ if the assignment ϕ_S does not assign the same variable to different values and the all zero vector otherwise. Finally, we use $opt^r_{Las}(\mathcal{G})$ to denote the maximum value among all r-level Lasserre solutions $\{U_{(S,\phi_S)}\}$.

It is not hard to see that finding $opt^r_{Las}(\mathcal{G})$ can be formulated as SDP with $(nq)^{O(r)}$ variables and, hence, can be approximated up to arbitrarily small error within $(nq)^{O(r)}$ time. Moreover, it is known that the r-level Lasserre relaxation is stronger than the r-level SA relaxation [Lau03]. In the case of Max k-CSP, this can be easily seen since we can define an r-level SA solution $\mu = \{\mathcal{X}_S\}_{|S| \leqslant t}$ from an r-level Lasserre solution $\{U_{(S,\phi_S)}\}$ by $\mathcal{X}_S(\phi_S) = \|U_{(S,\phi_S)}\|^2$.

3.2 Birthday Repetition Theorem and Its Proof

In this section, we prove our birthday repetition theorem, stated formally below.

Theorem 3.15. *Let $\mathcal{G} = (X, Y, E, \Sigma_X, \Sigma_Y, \{P_{(x,y)}\})$ be any two-prover game. Suppose further that (X, Y, E) is τ-almost-biregular. If $val(\mathcal{G}) = 1 - \varepsilon$, then for all $0 \leqslant k \leqslant |X|, 0 \leqslant \ell \leqslant |Y|$, we have*

$$val(\mathcal{G}^{k \times \ell}) \leqslant \left(1 - \frac{\varepsilon}{2\tau}\right)^{\Omega\left(\frac{k\ell}{\log(\tau/\varepsilon)(n+\tau/\varepsilon)}\right)}.$$

To prove Theorem 3.15, we have to show that any strategy ψ of $\mathcal{G}^{k \times \ell}$ is accepted with probability at most $\left(1 - \frac{\varepsilon}{2\tau}\right)^{\Omega\left(\frac{k\ell}{\log(\tau/\varepsilon)(n+\tau/\varepsilon)}\right)}$. Equivalently, we would like to show that the graph $G^{\text{SAT}}_\psi := \left(\binom{X}{k}, \binom{Y}{\ell}, E^{\text{SAT}}_\psi\right)$, where E^{SAT}_ψ denote the set of all accepted pairs of questions, is sparse.

Let us recall Alon's lemmas (Lemmas 2.14 and 2.15), which roughly states that, if a graph contains few labelled copies of biclique $K_{s,t}$, then it must be sparse. Hence, to show that G^{SAT}_ψ is sparse, it suffices to bound the number of labelled copies of $K_{s,t}$ in G^{SAT}_ψ for appropriately chosen s, t (which will be specified later), and invoke Alon's lemmas. To do so, we first define additional notations.

- Let $\mathcal{A}_X : \binom{X}{k}^s \to \mathscr{P}(X)$ denote a "flattening" operation that, on input $L \in \binom{X}{k}^s$, outputs the sets of all elements that appears in at least one of the sets in L; more formally, $\mathcal{A}_X(L) := \cup_{u \in L} u$. We define $\mathcal{A}_Y : \binom{Y}{\ell}^t \to \mathscr{P}(Y)$ similarly by $\mathcal{A}_Y(R) := \cup_{v \in R} v$.
- Let $\mathcal{K}_{s,t}$ denote the set of labelled copies of $K_{s,t}$ in the graph G^{SAT}_ψ, i.e., $\mathcal{K}_{s,t} := \left\{(L, R) \in \binom{X}{k}^s \times \binom{Y}{\ell}^t \middle| \, \forall u \in L, \, \forall v \in R, u \neq v \wedge (u, v) \in E^{\text{SAT}}_\psi\right\}$.
- For every $A \subseteq X, B \subseteq Y$, let $\mathcal{K}^{\subseteq}_{s,t}(A, B)$ denote the set of labelled copies of $K_{s,t}$ whose flattenings are contained in A, B, i.e., $\mathcal{K}^{\subseteq}_{s,t}(A, B) := \{(L, R) \in \mathcal{K}_{s,t} \mid \mathcal{A}_X(L) \subseteq A \wedge \mathcal{A}_Y(R) \subseteq B\}$. We remark here that $\mathcal{K}_{s,t}$ is equal to $\mathcal{K}_{s,t}(X, Y)$.

Now, the birthday repetition theorem can be proved via two simple observations. The first observation is that, any labelled copy (L, R) of $K_{s,t}$ cannot have flattenings that are both large. This is because (L, R) in fact induces a (partial) strategy for the original game that is accepted for all edges whose endpoints are in $\mathcal{A}_X(L)$ and $\mathcal{A}_Y(R)$. Hence, if both $\mathcal{A}_X(L)$ and $\mathcal{A}_Y(R)$ are already large, then this strategy would violate the assumption that $\mathrm{val}(\mathcal{G}) = 1 - \varepsilon$. A more precise version of the statement is proved below.

Observation 3.16. *Let $(L, R) \in \binom{X}{k}^s \times \binom{Y}{\ell}^t$ be any labelled $K_{s,t}$ of G_ψ^{SAT}. Then, $|\mathcal{A}_X(L)| \leqslant \left(1 - \frac{\varepsilon}{2\tau}\right)|X|$ or $|\mathcal{A}_Y(R)| \leqslant \left(1 - \frac{\varepsilon}{2\tau}\right)|Y|$.*

Proof. Suppose for the sake of contradiction that there exists a labelled $K_{s,t}$ copy (L, R) of G_ψ^{SAT} such that $|\mathcal{A}_X(L)| > \left(1 - \frac{\varepsilon}{2\tau}\right)|X|$ and $|\mathcal{A}_Y(R)| > \left(1 - \frac{\varepsilon}{2\tau}\right)|Y|$. Let $f : \mathcal{A}_X(L) \cup \mathcal{A}_Y(R) \to \Sigma_X \cup \Sigma_Y$ be a partial strategy to $\mathcal{G}$ defined as follows.

- For every $x \in \mathcal{A}_X(L)$, let $f(x) = \psi(u)_x$ for $u \in L$ that contains x (when there are multiple such u's, pick one arbitrarily).
- Similarly, for every $y \in \mathcal{A}_Y(R)$, let $f(y) = \psi(v)_y$ for $v \in R$ that contains x.

Consider any edge $(x, y) \in E \cap (\mathcal{A}_X(L) \times \mathcal{A}_Y(R))$. Let $u \in L, v \in R$ be such that $x \in u, y \in v, f(x) = \psi(u)_x$ and $f(y) = \psi(v)_y$. Since $u \in L, v \in R$ and (L, R) is a labelled copy of a biclique of G_ψ^{SAT}, we must have $(u, v) \in E_\psi^{\mathrm{SAT}}$; equivalently, $(\psi(u), \psi(v))$ is an accepting answer of the birthday repetition game for the edge (u, v). This implies that the verifier of the original game $\mathcal{G}$ accepts $(\psi(u)_x, \psi(v)_y) = (f(x), f(y))$ for the edge (x, y).

As a result, the verifier of the original game $\mathcal{G}$ accepts all edges in $E \cap (\mathcal{A}_X(L) \times \mathcal{A}_Y(R))$ for partial strategy f. In other words, the number of edges it does not accept is at most

$$\begin{aligned}
\sum_{x \notin \mathcal{A}_X(L)} \deg_G(x) + \sum_{y \notin \mathcal{A}_Y(R)} \deg_G(y) &\leqslant \sum_{x \notin \mathcal{A}_X(L)} \left(\tau \cdot \min_{x' \in X} \deg_G(x')\right) + \sum_{y \notin \mathcal{A}_Y(R)} \left(\tau \cdot \min_{y' \in Y} \deg_G(y')\right) \\
&\leqslant \tau|E| \cdot \left(\frac{|X| - |A_X(L)|}{|X|}\right) + \tau|E| \cdot \left(\frac{|Y| - |A_Y(R)|}{|Y|}\right) \\
&< \varepsilon|E|,
\end{aligned}$$

where the first inequality comes from the τ-almost-biregularity of G and the last inequality comes from our assumptions on the size of $\mathcal{A}_X(L), \mathcal{A}_Y(R)$. This contradicts the assumption that $\mathrm{val}(G) = 1 - \varepsilon$, which concludes our proof. □

Another observation is that, if we fix $A \subseteq X, B \subseteq Y$ such that either A or B is small and count the number of labelled copies of $K_{s,t}$ whose flattenings are contained in A, B (i.e. $|\mathcal{K}_{s,t}(A, B)|$), then this number will be much smaller than $\binom{|X|}{k}^s \binom{|Y|}{\ell}^t$. Intuitively, this is just because we must choose every left vertex from $\binom{A}{k}$ instead of $\binom{X}{k}$ and every right vertex from $\binom{B}{\ell}$ instead of $\binom{Y}{\ell}$. This is formalized below.

Observation 3.17. *For every $A \subseteq X, B \subseteq Y$, we have $|\mathcal{K}_{s,t}^{\subseteq}(A, B)| \leqslant \left(\frac{|A|}{|X|}\right)^{sk} \left(\frac{|B|}{|Y|}\right)^{t\ell} \binom{|X|}{k}^s \binom{|Y|}{\ell}^t$.*

Proof. Consider any $(L, R) \in \mathcal{K}^{\subseteq}_{s,t}(A, B)$. Since $\mathcal{A}_X(L) \subseteq A$ and $\mathcal{A}_Y(R) \subseteq B$, we have $L \in \binom{A}{k}^s$ and $R \in \binom{B}{\ell}^t$. This implies that $\mathcal{K}^{\subseteq}_{s,t}(A, B) \subseteq \binom{A}{k}^s \times \binom{B}{\ell}^t$. Hence,

$$|\mathcal{K}^{\subseteq}_{s,t}(A, B)| \leqslant \binom{|A|}{k}^s \binom{|B|}{\ell}^t \leqslant \left(\frac{|A|}{|X|}\right)^{sk} \left(\frac{|B|}{|Y|}\right)^{t\ell} \binom{|X|}{k}^s \binom{|Y|}{\ell}^t,$$

as desired. □

With these two observations ready, the proof of the birthday repetition theorem is straightforward: we simply sum the bounds in Observation 3.17 over an appropriate range of (A, B) which is given by Observation 3.16.

Proof of Theorem 3.15. Let $m = 100(n + \tau/\varepsilon)\log(100\tau/\varepsilon)$. Pick $s := \lceil m/k \rceil$ and $t := \lceil m/\ell \rceil$. From Observation 3.16, we have

$$\mathcal{K}_{s,t} \subseteq \left(\bigcup_{A \subseteq X, |A| = \lfloor \left(1 - \frac{\varepsilon}{2\tau}\right)|X| \rfloor} \mathcal{K}^{\subseteq}_{s,t}(A, Y) \right) \cup \left(\bigcup_{B \subseteq Y, |B| = \lfloor \left(1 - \frac{\varepsilon}{2\tau}\right)|Y| \rfloor} \mathcal{K}^{\subseteq}_{s,t}(X, B) \right). \tag{3.1}$$

As a result, we can bound $|\mathcal{K}_{s,t}|$ by

$$\begin{aligned}
|\mathcal{K}_{s,t}| &\leqslant \left(\sum_{A \subseteq X, |A| = \lfloor \left(1 - \frac{\varepsilon}{2\tau}\right)|X| \rfloor} |\mathcal{K}^{\subseteq}_{s,t}(A, Y)| \right) + \left(\sum_{B \subseteq Y, |B| = \lfloor \left(1 - \frac{\varepsilon}{2\tau}\right)|Y| \rfloor} |\mathcal{K}^{\subseteq}_{s,t}(X, B)| \right) \\
\text{(Observation 3.17)} &\leqslant \left(\sum_{A \subseteq X, |A| = \lfloor \left(1 - \frac{\varepsilon}{2\tau}\right)|X| \rfloor} \left(1 - \frac{\varepsilon}{2\tau}\right)^{sk} \binom{|X|}{k}^s \binom{|Y|}{\ell}^t \right) + \\
&\quad \left(\sum_{B \subseteq Y, |B| = \lfloor \left(1 - \frac{\varepsilon}{2\tau}\right)|Y| \rfloor} \left(1 - \frac{\varepsilon}{2\tau}\right)^{t\ell} \binom{|X|}{k}^s \binom{|Y|}{\ell}^t \right) \\
&= \left(\binom{|X|}{\lceil \frac{\varepsilon|X|}{2\tau} \rceil} \left(1 - \frac{\varepsilon}{2\tau}\right)^{sk} + \binom{|Y|}{\lceil \frac{\varepsilon|Y|}{2\tau} \rceil} \left(1 - \frac{\varepsilon}{2\tau}\right)^{t\ell} \right) \binom{|X|}{k}^s \binom{|Y|}{\ell}^t \\
\text{(From our choice of } s, t) &\leqslant \left(\binom{|X|}{\lceil \frac{\varepsilon|X|}{2\tau} \rceil} + \binom{|Y|}{\lceil \frac{\varepsilon|Y|}{2\tau} \rceil} \right) \left(1 - \frac{\varepsilon}{2\tau}\right)^{m} \binom{|X|}{k}^s \binom{|Y|}{\ell}^t \\
&\leqslant \left((2e\tau/\varepsilon)^{\lceil \frac{\varepsilon|X|}{2\tau} \rceil} + (2e\tau/\varepsilon)^{\lceil \frac{\varepsilon|Y|}{2\tau} \rceil} \right) \left(1 - \frac{\varepsilon}{2\tau}\right)^{m} \binom{|X|}{k}^s \binom{|Y|}{\ell}^t \\
&\leqslant (2e\tau/\varepsilon)^{2 + \frac{\varepsilon n}{2\tau}} \left(1 - \frac{\varepsilon}{2\tau}\right)^{m} \binom{|X|}{k}^s \binom{|Y|}{\ell}^t \\
\text{(From our choice of } m) &\leqslant 2^{0.1m\varepsilon/\tau} \left(1 - \frac{\varepsilon}{2\tau}\right)^{m} \binom{|X|}{k}^s \binom{|Y|}{\ell}^t
\end{aligned}$$

$$\leqslant \left(1 - \frac{\varepsilon}{2\tau}\right)^{0.5m} \binom{|X|}{k}^s \binom{|Y|}{\ell}^t.$$

Now, Lemma 2.15 implies that

$$|E_\psi^{\text{SAT}}| \leqslant \left(\left(1 - \frac{\varepsilon}{2\tau}\right)^{0.5m}\right)^{1/(st)} \binom{|X|}{k}\binom{|Y|}{\ell} = \left(1 - \frac{\varepsilon}{2\tau}\right)^{\Omega\left(\frac{k\ell}{\log(\tau/\varepsilon)(n+\tau/\varepsilon)}\right)} \binom{|X|}{k}\binom{|Y|}{\ell}.$$

In other words, every strategy ψ is accepted with probability at most $\left(1 - \frac{\varepsilon}{2\tau}\right)^{\Omega\left(\frac{k\ell}{\log(\tau/\varepsilon)(n+\tau/\varepsilon)}\right)}$. Hence, we have $\text{val}(\mathcal{G}^{k\times\ell}) \leqslant \left(1 - \frac{\varepsilon}{2\tau}\right)^{\Omega\left(\frac{k\ell}{\log(\tau/\varepsilon)(n+\tau/\varepsilon)}\right)}$ as desired. □

3.3 Applications of the Birthday Repetition Theorem

We next provide several implications of our birthday repetition theorem, including hardness of approximation results and integrality gaps for dense CSPs and improved $\mathsf{AM}(2)$ protocol for 3SAT.

3.3.1 Lower Bounds for Fully-Dense CSPs

We start with inapproximability and integrality gaps for fully dense MAX 2-CSP. Note that these bounds also carry over to fully dense MAX k-CSP for any $k \geqslant 2$, since we can always construct an instance of the latter from the former by making each predicate ignore the last $k-2$ variables.

As the birthday repetition theorem is stated in terms of free games, it will be convenient to first show the connection from free games to dense CSPs. Specifically, given a free games, one can easily creates a "symmetrized" version of the game that is a fully-dense MAX 2-CSP and has the same value of the original game:

Lemma 3.18 (Symmetrization of Free Games)**.** *Given a free game $\mathcal{G} = (X, Y, \Sigma_X, \Sigma_Y, P)$, we can, in polynomial time, create a* MAX 2-CSP *instance $\mathcal{G}_{sym}$ with alphabet $\Sigma_X \times \Sigma_Y$ such that (i)* $\text{val}(\mathcal{G}_{sym}) \leqslant \text{val}(\mathcal{G})$ *and (ii) if* $\text{val}(\mathcal{G}) = 1$*, then* $\text{val}(\mathcal{G}_{sym}) = 1$.

Proof. We define an instance $\mathcal{G}_{\text{sym}} = (X \times Y, \Sigma_X \times \Sigma_Y, P')$ where the variables are $X \times Y$, the alphabet is $\Sigma_X \times \Sigma_Y$ and $P'_{\{(x_1,y_1),(x_2,y_2)\}}((\sigma_{x_1}, \sigma_{y_1}), (\sigma_{x_2}, \sigma_{y_2})) = \prod_{i,j\in\{1,2\}} P_{(x_i,y_j)}(\sigma_{x_i}, \sigma_{y_j})$.

Now, we will show that $\text{val}(\mathcal{G}_{\text{sym}}) \leqslant \text{val}(\mathcal{G})$. Let ϕ be an optimal assignment of $\mathcal{G}_{\text{sym}}$, i.e., where $\text{val}(\phi) = \text{val}(\mathcal{G}_{\text{sym}})$. Let f be a strategy for $\mathcal{G}$ defined randomly as follows: for every $x \in X$, randomly select $y \in Y$ and set $f(x) = \phi(x,y)_1$, and, for every $y \in Y$, randomly select $x \in X$ and set $f(x) = \phi(x,y)_2$. We have

$$\begin{aligned}\mathbb{E}_f[\text{val}(f)] &= \mathbb{E}_{f,(x,y)\sim(X\times Y)}[P_{(x,y)}(f(x), f(y))] \\ &= \mathbb{E}_{(x,y),(x',y')\sim(X\times Y)}[P_{(x,y)}(\phi((x,y'))_1, \phi((x',y))_2)] \\ &= \frac{1}{|X||Y|}\mathbb{E}_{(x,y)\sim(X\times Y)}[P_{(x,y)}(\phi(x,y)_1, \phi(x,y)_2)] +\end{aligned}$$

$$\left(1 - \frac{1}{|X||Y|}\right) \mathbb{E}_{(x,y),(x',y') \sim (X \times Y)}[P_{(x,y)}(\phi(x,y')_1, \phi(x',y)_2) \mid (x,y) \neq (x',y')] \quad (3.2)$$

We can bound the first term by

$$\begin{aligned}
&\mathbb{E}_{(x,y) \sim (X \times Y)}[P_{(x,y)}(\phi(x,y)_1, \phi(x,y)_2)] \\
&= \mathbb{E}_{(x,y),(x',y') \sim (X \times Y)}[P_{(x,y)}(\phi(x,y)_1, \phi(x,y)_2) \mid (x',y') \neq (x,y)] \\
&\leqslant \mathbb{E}_{(x,y),(x',y') \sim (X \times Y)}[P'_{\{(x,y),(x',y')\}}(\phi(x,y), \phi(x',y')) \mid (x',y') \neq (x,y)] \\
&= \text{val}(\phi).
\end{aligned}$$

The second term can be bounded similarly:

$$\begin{aligned}
&\mathbb{E}_{(x,y),(x',y') \sim (X \times Y)}[P_{(x,y)}(\phi(x,y')_1, \phi(x',y)_2) \mid (x,y) \neq (x',y')] \\
&\leqslant \mathbb{E}_{(x,y),(x',y') \sim (X \times Y)}[P'_{\{(x',y),(x,y')\}}(\phi(x',y), \phi(x,y')) \mid (x,y) \neq (x',y')] \\
&= \text{val}(\phi).
\end{aligned}$$

Plugging both back to (3.2), we get $\mathbb{E}_f[\text{val}(f)] \geqslant \text{val}(\phi)$. Hence, we have $\text{val}(\mathcal{G}) \geqslant \text{val}(\phi) = \text{val}(\mathcal{G}_{\text{sym}})$, concluding the first part of the claim.

Finally, suppose that $\text{val}(\mathcal{G}) = 1$; let f be an optimal strategy for $\mathcal{G}$. The let ϕ be the assignment of $\mathcal{G}_{\text{sym}}$ defined by $\phi(x,y) := (f(x), f(y))$. It is obvious that ϕ satisfies all constraints. □

ETH-Based Hardness of Approximation of Fully-Dense Max 2-CSP

The first application of the birthday repetition theorem we present is an ETH-based almost-polynomial ratio hardness for fully-dense Max k-CSP, as stated formally below.

Lemma 3.19. *Assuming ETH, for any $k \geqslant 2$, no polynomial-time algorithm can, given any fully-dense* Max k-CSP *instance $\mathcal{G}$ of size N, distinguish* $\text{val}(\mathcal{G}) = 1$ *from* $\text{val}(\mathcal{G}) \leqslant 2^{-\widetilde{\Omega}(\log N)}$.

We prove this by essentially applying the birthday repetition theorem with $k = \ell = \widetilde{\Omega}(n)$ to a two-prover game produced by Dinur's PCP Theorem [Din07] (Theorem 2.2). Note, however, that Theorem 2.2 produces a Gap-3SAT instance instead of a two-prover game. Nevertheless, reductions between the two are standard. Here we use the so-called clause/variable reduction. This reduction is well-known and has appeared in literatures before (in e.g. [AIM14]). It is stated formally below.

Definition 3.20. *(Clause/variable game) For any* Max k-CSP *instance $\mathcal{G} = (V, E, \{P_S\})$, its clause/variable game is a projection game $\mathcal{G}' = (X', Y', \Sigma'_X, \Sigma'_Y, E', \{P'_{(x,y)}\})$ defined as follows. X' is the set of constraints of $\mathcal{G}$, i.e., $X' = E$. Y' is V, the set of variables of $\mathcal{G}$. Σ'_X is Σ^k; for each constraint S, Σ'_X is identified with the assignments of S in $\mathcal{G}$. Σ'_Y is simply Σ. Finally, E' contains all (S, x) such that $x \in S$ and $P_{(S,x)}(\phi, \sigma) = 1$ iff $P_S(\phi) = 1$ and $\phi(x) = \sigma$.*

It is easy to see that, when $val(\mathcal{G})$ is bounded away from one, then so is $val(\mathcal{G}')$:

Proposition 3.21. *Let $\mathcal{G}$ and $\mathcal{G}'$ be as in Definition 3.20. If $val(\mathcal{G}) \leqslant 1-\varepsilon$, then $val(\mathcal{G}') \leqslant 1-\varepsilon/k$.*

Proof. Suppose for contrapositively that there is an assignment ϕ' of $\mathcal{G}'$ with $val(\phi') > 1-\varepsilon/k$. Define $\phi : V \to \Sigma$ by $\phi(x) := \phi'(x)$ for every $x \in V$. Since less than ε/k fraction of the edges are not satisfied by ϕ' in $\mathcal{G}'$ and each $S \in X'$ has degree k, more than $1-\varepsilon$ fraction of $S \in X'$ touches only satisfied edges. These clauses are satisfied by ϕ in $\mathcal{G}$. Hence, $val(\phi) > 1-\varepsilon$. □

We can now prove Lemma 3.19.

Proof of Lemma 3.19. Given 3SAT instance ϕ of m clauses. We first use Dinur's PCP Theorem (Theorem 2.2) to reduce ϕ to ϕ' with $m' = m \log^c m$ clauses. Let $\mathcal{G}$ be the clause-variable game of ϕ'. Consider the fully-dense Max 2-CSP instance $\mathcal{G}_{\text{sym}}^{k\times\ell}$ which is the symmetrization of the $(k \times \ell)$-birthday repetition game, where $k = \ell = m/\log^2 m$.

Let $\tilde{n}$ and $\tilde{q}$ be the number of variables and the alphabet size of $\mathcal{G}_{\text{sym}}^{k\times\ell}$. We have $\tilde{n} \leqslant \binom{m'}{k}^2 \leqslant 2(m')^{2k} \leqslant 2^{O\left(\frac{m}{\log m}\right)}$ and $\tilde{q} \leqslant 2^{O(k)} \leqslant 2^{O\left(\frac{m}{\log m}\right)}$. Hence, the size of $\mathcal{G}_{\text{sym}}^{k\times\ell}$ is $\widetilde{N} = (\tilde{n}\tilde{q})^O(1) \leqslant 2^{O(m/\log m)}$. We next analyze the completeness and soundness of the reduction.

When $val(\phi) = 1$, from the PCP theorem, we have $val(\phi') = 1$. It is also obvious from the reduction that $val(\mathcal{G}_{\text{sym}}^{k\times\ell})$ is one. On the other hand, when $val(\phi) < 1$, we have $val(\phi') \leqslant 1-\varepsilon$, meaning that $\text{val}(\mathcal{G}) \leqslant 1-\varepsilon/3$. Hence, by Theorem 3.15 and Lemma 3.18, we have

$$\text{val}(\mathcal{G}_{\text{sym}}^{k\times\ell}) \leqslant (1-\Omega(\varepsilon))^{\Omega\left(\frac{k^2}{m'^2}\right)} \leqslant 2^{-\widetilde{\Omega}(m)} = 2^{-\widetilde{\Omega}(\log \widetilde{N})}.$$

Thus, if a algorithm can distinguish $\text{val}(\mathcal{G}_{\text{sym}}^{k\times\ell}) = 1$ from $\text{val}(\mathcal{G}_{\text{sym}}^{k\times\ell}) \leqslant 2^{-\widetilde{\Omega}(\log \widetilde{N})}$ in time polynomial in $\widetilde{N}$, then it can also solve 3SAT in time $2^{O(n/\log n)}$ time, violating ETH. □

Improved Hardness of Approximation Result Based on Gap-ETH

The $\text{polyloglog}\, N$ loss in the exponent of Lemma 3.19 is due to the quasi-linear size of the PCP and can be eliminated if we instead assume the stronger Gap-ETH:

Lemma 3.22. *Assuming Gap-ETH, for any sufficiently large i, no algorithm can, given any fully-dense* Max 2-CSP *$\mathcal{G}$ of size N, distinguish $val(\mathcal{G}) = 1$ from $val(\mathcal{G}) \leqslant N^{-1/i}$ in time $N^{\widetilde{O}(i)}$.*

The proof is essentially the same as that of Lemma 3.19 except that, since the size of our starting game is linear, we can now use birthday repetition for $k = \ell = \Theta_i(n)$ instead of $n/\text{polylog}(n)$.

Proof of Lemma 3.22. Given Gap-3SAT instance ϕ of m variables. Let $\mathcal{G}$ be its clause/variable game. Observe that $\mathcal{G}$ has $m' = O(m)$ variables, $O(1)$ alphabet size and maximum degree $O(1)$. Consider $\mathcal{G}_{\text{sym}}^{k\times\ell}$, the symmetrized $(k \times \ell)$-birthday repetition with $k = \ell = \frac{\beta n \log i}{i}$ where β is a small constant which to be chosen later.

Let $\widetilde{n}$ and $\widetilde{q}$ be the number of variables and the alphabet size of $\mathcal{G}_{\text{sym}}^{k\times\ell}$. We have $\widetilde{q} \leqslant 2^{O(k)} \leqslant 2^{O\left(\frac{\beta n \log i}{i}\right)}$. Moreover, when β is sufficiently small, we have

$$\widetilde{n} \leqslant \binom{m'}{k}^2 \leqslant \left(\frac{em'}{k}\right)^{2k} = \left(O\left(\frac{i}{\beta \log i}\right)\right)^{2l} \leqslant 2^{O\left(\frac{\beta m \log^2 i \log(1/\beta)}{i}\right)} \leqslant 2^{O\left(\frac{\sqrt{\beta} m \log^2 i}{i}\right)}.$$

As for the completeness and soundness of the reduction, first, it is obvious that $\mathrm{val}(\phi) = 1$ implies $\mathrm{val}(\mathcal{G}_{\text{sym}}^{k\times\ell}) = 1$. Otherwise, from Proposition 3.21, if $\mathrm{val}(\phi) \leqslant 1-\varepsilon$, then $\mathrm{val}(\mathcal{G}) \leqslant 1-\varepsilon/3$. By by Theorem 3.15 and Lemma 3.18, we have

$$\mathrm{val}(\mathcal{G}_{\text{sym}}^{k\times\ell}) \leqslant (1-\Omega(\varepsilon))^{\Omega\left(\frac{k^2}{m'}\right)} \leqslant 2^{-\Omega(\beta^2 n \log^2 i/i^2)} \leqslant (\widetilde{n}\widetilde{q})^{-\Omega(\beta^2/i)} = \widetilde{N}^{-\Omega(\beta^2/i)}$$

where $\widetilde{N} = (\widetilde{n}\widetilde{q})^O(1) \leqslant 2^{O(\sqrt{\beta} n \log^2 i/i)}$ is the size of $\mathcal{G}_{\text{sym}}^{k\times\ell}$.

Pick β to be sufficiently small so that $\widetilde{N} \leqslant O(2^{\delta m \log^2 i/i})$ where δ is the constant from Gap-ETH. If an algorithm distinguishes $\mathrm{val}(\mathcal{G}_{\text{sym}}^{k\times\ell}) = 1$ from $\mathrm{val}(\mathcal{G}_{\text{sym}}^{k\times\ell}) \leqslant (\widetilde{N})^{-\Omega(1/i)}$ in $O(\widetilde{N}^{\frac{i}{\log^2 i}})$ time, it also distinguishes $val(\phi) = 1$ from $val(\phi) \leqslant 1-\varepsilon$ in time $O(2^{\delta m})$, violating Gap-ETH. □

Lasserre Integrality Gap for Fully-Dense Max 2-CSP

We will now show how to get a polynomial integrality gap for the Lasserre relaxation for dense CSPs. In particular, even for $\widetilde{\Omega}(i)$-level of Lasserre hierarchy, the integrality gap remains $N^{1/i}$ for fully-dense Max 2-CSP, as stated formally below.

Lemma 3.23. *For any sufficiently large N and any sufficiently large i, there exists a fully-dense* Max 2-CSP *instance $\mathcal{G}$ of size N such that $opt_{Las}^{\widetilde{\Omega}(i)}(\mathcal{G}) = 1$ and* $\mathrm{val}(\mathcal{G}) \leqslant N^{-1/i}$.

One way to interpret Lemma 3.23 is as a lower bound for SDP or LP hierarchies algorithm for dense Max 2-CSP. From this perspective, our result indicates that one cannot hope to use $\widetilde{O}(i)$-level Lasserre relaxation to approximate fully-dense Max 2-CSP to within a factor of $N^{1/i}$. Since the Lasserre hierarchy is stronger than the SA and the Lovász-Schrijver hierarchies [Lau03], such lower bound holds for those hierarchies as well. Interestingly, this lower bound essentially matches, up to a factor of polylog(i) in the number of levels, our algorithmic result presented in the next section, justifying the running time of our algorithm.

On the other hand, Lemma 3.23 can be viewed as an unconditional analogue of Lemma 3.22. In this sense, we get rid of Gap-ETH assumption at the expense of restricting our computational model to only Lasserre relaxation. Other than those differences, the two lemmas are essentially the same. In fact, to prove Lemma 3.23, we use an unconditional analogue of Gap-ETH under the Lasserre hierarchy model, which is stated below.

Lemma 3.24. *For sufficiently large N, there exists a projection game $\mathcal{G}$ of size N with the following properties.*

- *(Vector Completeness) $opt_{Las}^{\Omega(N)} = 1$.*

- *(Soundness)* $val(\mathcal{G}) = 1 - \varepsilon$ *for some constant* $\varepsilon > 0$.
- *(Bounded Degree) Each variable has constant degree.*
- *(Bounded Alphabet Size) The alphabet size is constant.*

Results similar to Lemma 3.24 have been proven before in [Bha+12] and [Man15] by applying the clause/variable reduction to integrality gap instances of MAX k-CSP from [Sch08; Tul09]. For a detailed proof of Lemma 3.24, please refer to Appendix D.2 in the full version of [MR17a].

With the help of Lemma 3.24, the proof of Lemma 3.23 proceeds in a similar fashion as that of Lemma 3.22. However, while the soundness argument remains unchanged, we need to argue completeness for Lasserre solution instead. On this front, several works (including [Sch08; Tul09; Bha+12; Man15]) have argued similar statements before. Roughly speaking, it holds that, if the reduction produces each new constraint by a "composition of at most t constraints", then the Lasserre solution carries over to the new instance, albeit at the multiplicative loss of t in the number of levels. In the context of birthday repetition games, this implies the following.

Observation 3.25. *For any two-prover game* $\mathcal{G}$, *if* $opt^{r}_{Las}(\mathcal{G}) = 1$ *for some* $r \geqslant 2(k+\ell)$, *then* $opt^{\frac{r}{k+\ell}}_{Las}(\mathcal{G}^{k\times\ell}_{sym}) = 1$.

We will now prove the above observation in this dissertation; interested readers can refer to Appendix D.1 in the full version of [MR17a], which contains a full proof of the statement.

We now move on to prove Lemma 3.23.

Proof of Lemma 3.23. We start with a projection game $\mathcal{G}$ from Lemma 3.24 of size N with $n \leqslant N$ variables, $q = O(1)$ alphabet size and maximum degree $d = O(1)$. Consider the fully-dense MAX k-CSP $\mathcal{G}^{k\times\ell}_{\text{sym}}$, the symmetrized $(k \times \ell)$-birthday repetition of $\mathcal{G}$, with $k = \ell = \frac{n \log i}{i}$.

Let $\widetilde{n}$ and $\widetilde{q}$ be the number of variables and the alphabet size of $\mathcal{G}^{k\times\ell}_{\text{sym}}$. We have $\widetilde{q} \leqslant q^{2l} \leqslant 2^{O\left(\frac{N \log i}{i}\right)}$. Moreover, we have $\widetilde{n} \leqslant \binom{n}{l}^2 \leqslant 2^{O\left(\frac{N \log^2 i}{i}\right)}$.

Furthermore, from Observation 3.25 and from $opt^{\Omega(N)}_{Las}(\mathcal{G}) = 1$, we have $opt^{\Omega(N/k)}_{Las}(\mathcal{G}^{k\times\ell}_{\text{sym}}) = opt^{\widetilde{\Omega}(i)}_{Las}(\mathcal{G}^{k\times\ell}_{\text{sym}}) = 1$. Finally, by Theorem 3.15 and Lemma 3.18, we have $\text{val}(\mathcal{G}^{k\times\ell}_{\text{sym}}) \leqslant (1 - \Omega(\varepsilon))^{\Omega\left(k^2/n\right)} \leqslant (\widetilde{n}\widetilde{q})^{-\Omega(1/i)} = (\widetilde{N})^{-\Omega(1/i)}$ where $\widetilde{N} = (\widetilde{n}\widetilde{q})^{O(1)}$ is the size of $\mathcal{G}^{k\times\ell}_{\text{sym}}$. This completes our proof. □

3.3.2 Almost Optimal $\mathsf{AM}(2)$ Protocol for 3SAT

In [AIM14], Aaronson et al. provided an $\mathsf{AM}(2)$ protocol of length $\widetilde{O}(\sqrt{n})$ for 3SAT with completeness 1 and soundness δ for *some* constant $\delta < 1$. However, since they did not prove that birthday repetition can amplify soundness, they could not get a similar result for arbitrarily small δ. In that case, they invoke Moshkovitz-Raz PCP [MR10], which, incontrast to Dinur's PCP, gives arbitrarily small soundness. However, due to the length of Moshkovitz-Raz PCP, their protocol length is $n^{1/2+o(1)}\text{poly}(1/\delta)$. Since we have proved that the birthday repetition amplifies the soundness, we overcome this obstacle and we can prove Lemma 3.12 easily as follows.

Proof of Lemma 3.12. Given a 3SAT instance ϕ of n clauses, the protocol works as follows. Arthur uses Dinur's PCP Theorem and the clause/variable reduction to reduce ϕ to $\mathcal{G}$ with $n' = n\log^c n$ variables, constant alphabet size and constant maximum degree. He then produces a free game $G^{k\times\ell} = (X, Y, X \times Y, \Sigma_X, \Sigma_Y, \{P_{(x,y)}\})$, the $(k \times \ell)$-birthday repetition of $\mathcal{G}$, with $k = \ell = d(\log^{c/2} n)\sqrt{n\log(1/\delta)}$ for some large constant d to be chosen later.

Arthur then sends independent random questions to the Merlins where the questions for first and second Merlins are drawn from X and Y respectively. The proof of each Merlin is an assignment to the variable he is given. Finally, if the two Merlins receive questions $x \in X, y \in Y$, Arthur uses the predicate $P_{(x,y)}$ to check whether the assignments he received satisfy the predicate. If so, Arthur accepts. Otherwise, he rejects.

It is obvious that, when $\phi \in$ 3SAT, i.e., ϕ is satisfiable, $\mathcal{G}^{k\times\ell}$ is satisfiable and Arthur always accepts if Merlins answer according to a satisfying assignment of $\mathcal{G}^{k\times\ell}$. On the other hand, if $\phi \notin$ 3SAT, $\text{val}(\mathcal{G}^{k\times\ell}) \leqslant (1 - \Omega(\varepsilon))^{\Omega(k^2/n')}$, which is at most δ for sufficiently large d. Hence, the soundness of the protocol is at most δ. Finally, observe that the protocol has length $2k\log n = \widetilde{O}(\sqrt{n\log(1/\delta)})$ as desired. □

3.4 Improved Approximation Algorithm for Dense CSPs

Before describing our algorithm, we first explain ingredients central in conditioning-based algorithms: a conditioning operator and a rounding procedure.

Conditioning Sherali-Adams Solution. Let $\mu = \{\mathcal{X}_S\}$ be a solution of an r-level SA relaxation of a Max k-CSP instance. For any set $T \subseteq V$ of size at most $r - k$ and for any $\phi_T \subseteq \Sigma^T$ such that $\mathcal{X}_T(\phi_T) > 0$, μ conditioned on ϕ_T is $\mu|\phi_T = \{\widetilde{\mathcal{X}_S}\}_{|S|\leqslant r-|T|}$ defined as

$$\widetilde{\mathcal{X}_S}(\phi_S) = \begin{cases} \mathcal{X}_{S\cup T}(\phi_S \circ \phi_T)/\mathcal{X}_T(\phi_T) & \text{if } \phi_S \text{ is consistent with } \phi_T, \\ 0 & \text{otherwise.} \end{cases}$$

It is not hard to see that $\mu|\phi_T$ is an $(r - |T|)$-level SA solution.

(Derandomized) Independent Rounding. A naive way to arrive at an actual solution to the Max k-CSP instance from a SA relaxation solution $\{\mathcal{X}_S\}_{|S|\leqslant r}$ is to independently sample each variable x based on the distribution $\mathcal{X}_x$. Observe that the rounded solution expected value is $\mathbb{E}_{S=\{x_{i_1},\dots,x_{i_k}\}\sim\mathcal{W}}\left[\mathbb{E}_{\phi_S\sim\mathcal{X}_{i_1}\times\cdots\times\mathcal{X}_{i_k}}\left[P_S(\phi_S)\right]\right]$. Note that such rounding can be easily derandomized via a standard conditional expectation argument.

Without going into too much detail, conditioning-based algorithms typically proceed as follows. First, solve a LP/SDP relaxation of the problem. As long as the solution has large "total correlation", try conditioning it on an assignment to a random variable. Once the solution has small total correlation, use independent rounding on the solution to get the desired assignment.

The intuition behind such algorithms is that, if the solution has large total correlation, conditioning on one variable substantially reduces the total correlation. Hence, after a certain number of rounds of conditioning, the total correlation becomes small. At this point, the solution is quite independent and independent rounding gives a good approximation.

Our algorithm will also follow this framework. In fact, our algorithm remains largely unchanged from [YZ14] with the exception that we will use a stronger relaxation to reduce our work in arguing about the value of conditioned solutions. However, our main contribution lies in the analysis: we will show that independent rounding does well even when the total correlation is large (super-constant). This is in contrast to the previously known conditioning-based algorithms [BRS11; RT12; YZ14], all of which require their measures of correlation to be small constants to get any meaningful result.

The new relaxation that we will used is the following. For convenience, we call this the r-level relaxation Sherali-Adams with Conditioning (SAC) relaxation of MAX k-CSP.

$$\begin{aligned}
&\text{maximize } \lambda \\
&\text{subject to } \{\mathcal{X}_S\}_{|S| \leqslant r} \text{ is a valid } r\text{-level SA solution} \\
&\qquad \mathbb{E}_{S \sim \mathcal{W}}[\mathbb{E}_{\phi_S \sim (\mu|\phi_T)}[P_S(\phi_S)]] \geqslant \lambda \qquad \forall T, \phi_T \text{ s.t. } |T| \leqslant r-k, \mathcal{X}_T(\phi_T) > 0.
\end{aligned}$$

At a glance, the program above may not look like a linear program. Fortunately for us, $\mathbb{E}_{S \sim \mathcal{W}}[\mathbb{E}_{\phi_S \sim (\mu|\phi_T)}[P_S(\phi_S)]] \geqslant \lambda$ can be written as $\mathbb{E}_{S \sim \mathcal{W}}[\sum_{\phi_S \in \Sigma^S} \mathcal{X}_{S \cup T}(\phi_S \circ \phi_T) P_{S \cup T}(\phi_S \circ \phi_T)] \geqslant \lambda \mathcal{X}_T(\phi_T)$, which is linear when λ is a constant rather than a variable. As a result, we can solve the optimization problem above by binary search on λ: for a fixed λ, we can check whether the inequalities is feasible using a typical polynomial-time algorithm for LP. Hence, we can approximate λ to within arbitrarily small additive error in polynomial time. To compute λ exactly, observe that $\mathcal{W}$ is part of the input and is expressible in polynomial number of bits. This means that there are only exponentially many choices for λ; in particular, if all probabilities in $\mathcal{W}$ has only b digits after decimal point, then so does λ. Hence, the described binary search can find λ in $(nq)^{O(r)}$ time.

We now state our algorithm. In summary, we first solve an $O(\frac{k^2 i}{\Delta} + k)$-level SAC relaxation for the instance. We then try every possible conditioning (i.e., on every set $T \subseteq V$ of size at most $k^2 i/\Delta$ and every assignment to T). For each conditioned solution, we use independent rounding to arrive at an assignment. Finally, output the best such assignment. The pseudo-code for the full algorithm is shown below in Figure 3.1.

The rest of the section is organized as follows. In Subsection 3.4.1, we formally define total correlation and state a bound from [YZ14; MR17a; JKR19] on the total correlation of conditioned solutions. Next, in Subsection 3.4.2, we state and prove our main contribution of this section, i.e., that even when the total correlation is super-constant, we can still get a non-trivial approximation from independent rouding. Finally, in Subsection 3.4.3, we put these together and prove the approximation guarantee for our algorithm.

```
Algorithm 1 Approximation Algorithm for Dense CSPs
Input: a $\Delta$-dense MAX $k$-CSP instance $\mathcal{G} = (V, \mathcal{W}, \{P_S\})$, an integer $i$
Output: An assignment $\phi : V \to \Sigma$
  $r \leftarrow (k^2 i/\Delta + k)$
  do
      $r \leftarrow r + 1$
      $\mu \leftarrow$ solution of $r$-level of SAC relaxation for $\mathcal{G}$.
      $\lambda \leftarrow$ value of $\mu$
  while $(r - k)\lambda < k^2 i/\Delta$ and $r < n$
  $\phi \leftarrow \emptyset$
  for $T \subseteq V$ of size at most $r - k$ do
      for $\phi_T \in \Sigma^T$ do
          $\phi' \leftarrow$ independent rounding of $\mu|\phi_T$
          if $val(\phi') > val(\phi)$ then
              $\phi \leftarrow \phi'$.
  return $\phi$
```

Figure 3.1: Pseudo-code of Our Approximation Algorithm for Dense CSPs. The only difference between this pseudo-code and the above summary of our algorithm is that we need to iteratively increase the number of levels of the hierarchy. This is due to the fact that, as we will see in Lemma 3.28, the number of levels needed depends on the value of the solution. More specifically, we want $r \geqslant k^2 i/(\Delta\lambda) + k$

3.4.1 Total Correlation of Conditioned Sherali-Adams Relaxation Solution

We start by defining the total correlation of a SA solution. For a k-level SA solution $\mu = \{\mathcal{X}_S\}$ and for a k-size set $S = \{x_{i_1}, \dots, x_{i_k}\} \in \binom{V}{k}$, the total correlation among $x_{i_1}, \dots, x_{i_j}$ is $C_\mu(x_S) := C(\sigma_{i_1}; \dots; \sigma_{i_k})$ where $\sigma_{i_1}, \dots, \sigma_{i_k}$ are jointly sampled from $\mathcal{X}_{\{x_{i_1}, \dots, x_{i_k}\}}$. The total correlation of μ is then defined as $C(\mu) = \mathbb{E}_{S \sim \mathcal{W}}[C_\mu(x_S)]$. We call μ *a κ-independent solution* if $C(\mu) \leqslant \kappa$.

Yoshida and Zhou [YZ14] show that, for any $l > 0$ and any $(l+k)$-level SA solution μ, there exists a subset T of size at most l and an assignment $\phi_T \in \Sigma^T$ such that the total correlation of $(\mu|\phi_T)$ is at most $\frac{3^k \log q}{l\Delta}$ where Δ is the density of the instance. In [MR17a], this bound was improved to $\frac{k^2 \log q}{l\Delta}$ via a slightly sharper analysis. Minor mistakes in those proofs were later found and corrected in [JKR19]; we state the corrected version of the statement below.

Lemma 3.26 ([YZ14; MR17a; JKR19]). *Let μ be any r-level SA solution of a Δ-dense* MAX k-CSP *instance $\mathcal{G} = (V, \mathcal{W}, \{P_S\})$ with alphabet size q. Then, for any $0 < l \leqslant r - k$, there exists $t \leqslant l$ and $\phi_T \in \Sigma^t$ such that $\mathbb{E}_{T \sim \binom{V}{t}}[C(\mu|\phi_T)] \leqslant \frac{k^2 \log q}{l\Delta}$.*

3.4.2 New Bound on Rounding κ-independent Solution

In this subsection, we prove our main lemma for this section. For the known conditioning-based algorithms, once the solution is fairly independent, it is easy to show that independent rounding gives a good solution. In particular, Raghavendra-Tan [RT12] and Yoshida-Zhou[YZ14] proofs, whose measures of correlation are the same as ours[6], conclude this step by using the Pinsker's inequality, which states that, for any distributions $\mathcal{X}$ and $\mathcal{Y}$, $D_{KL}(\mathcal{X}\|\mathcal{Y}) \geqslant (2\log 2)\|\mathcal{X}-\mathcal{Y}\|_1^2$ where $\|\mathcal{X}-\mathcal{Y}\|_1 = \sum_{\theta\in\Theta}|\mathcal{X}(\theta)-\mathcal{Y}(\theta)|$ is the L^1-distance between $\mathcal{X}$ and $\mathcal{Y}$. Roughly speaking, $\mathcal{X}$ is going to be the distribution in the LP solution whereas $\mathcal{Y}$ is the distribution resulting from independent rounding. Hence, when they bound $D_{KL}(\mathcal{X}\|\mathcal{Y})$ to be at most a small constant ε, it follows immediately that any predicate f with domain $\operatorname{supp}(\mathcal{X})$ in $[0,1]$ satisfies $|\mathbb{E}_{x\sim\mathcal{X}}[f(x)] - \mathbb{E}_{y\sim\mathcal{Y}}[f(y)]| \leqslant \sqrt{\varepsilon/(2\log 2)}$. Thus, if $\mathbb{E}_{x\sim\mathcal{X}}[f(x)]$, the value of the LP solution, is large, then $\mathbb{E}_{y\sim\mathcal{Y}}[f(y)]$, the expected value of a solution from independent rouding, is also large.

While this works great for small constant ε, it does not yield any meaningful bound when ε is larger than a certain constant. A natural question is whether one can prove any non-trivial bound for super-constant ε. In this regard, we prove the following lemma, which positively answers the question. For convenience, 0^0 is defined to be 1 throughout this and next subsections and, whenever we write the expression $(\delta^\delta e^{-\kappa})^{\frac{1}{1-\delta}}(1-\delta)$ with $\delta = 1$, we define it to be 0.

Lemma 3.27. *Let $\mathcal{X}$ and $\mathcal{Y}$ be any two probability distributions over a finite domain Θ such that $D_{KL}(\mathcal{X}\|\mathcal{Y}) \leqslant \kappa$ and let $f : \Theta \to [0,1]$ be any function. If $\mathbb{E}_{x\sim\mathcal{X}}[f(x)] = 1-\delta$, then $\mathbb{E}_{y\sim\mathcal{Y}}[f(y)] \geqslant \left(\delta^\delta e^{-\kappa}\right)^{\frac{1}{1-\delta}}(1-\delta)$.*

Proof of Lemma 3.27. We assume without loss of generality that $\delta \notin \{0,1\}$ since, when $\delta = 0$, we can modify f infinitesimally small and take the limit of the bound and, when $\delta = 1$, the bound is trivial.

Let $\mathcal{Z}$ and $\mathcal{T}$ be two probability distributions on Θ such that $\mathcal{Z}(\theta) = \frac{\mathcal{X}(\theta)f(\theta)}{1-\delta}$ and $\mathcal{T}(\theta) = \frac{\mathcal{X}(\theta)(1-f(\theta))}{\delta}$. Observe that $\mathcal{Z}$ and $\mathcal{T}$ are indeed valid distributions on Θ since $\mathbb{E}_{\theta\sim\mathcal{X}}[f(\theta)] = 1-\delta$. Observe that $\operatorname{supp}(\mathcal{Z}), \operatorname{supp}(\mathcal{T}) \subseteq \operatorname{supp}(\mathcal{X})$, which is in turn contained in $\operatorname{supp}(\mathcal{Y})$ since $D_{KL}(\mathcal{X}\|\mathcal{Y}) \neq \infty$.

From Weighted A.M.-G.M. inequality, we have

$$\begin{aligned}
\mathbb{E}_{y\sim\mathcal{Y}}[f(y)] = \sum_{\theta\in\Theta}\mathcal{Y}(\theta)f(\theta) &\geqslant \sum_{\theta\in\operatorname{supp}(\mathcal{Z})}\mathcal{Z}(\theta)\left(\frac{\mathcal{Y}(\theta)f(\theta)}{\mathcal{Z}(\theta)}\right) \\
\text{(Weighted A.M.-G.M. inequality)} &\geqslant \prod_{\theta\in\operatorname{supp}(\mathcal{Z})}\left(\frac{\mathcal{Y}(\theta)f(\theta)}{\mathcal{Z}(\theta)}\right)^{\mathcal{Z}(\theta)} \\
&= (1-\delta)\left(\prod_{\theta\in\operatorname{supp}(\mathcal{Z})}\left(\frac{\mathcal{Y}(\theta)}{\mathcal{X}(\theta)}\right)^{\mathcal{X}(\theta)f(\theta)}\right)^{\frac{1}{1-\delta}}.
\end{aligned}$$

[6] In [RT12], only 2-CSPs were studied and they measure correlation by mutual information of the variables in the constraints.

We will next bound $\prod_{\theta \in \mathrm{supp}(\mathcal{Z})} \left(\frac{\mathcal{Y}(\theta)}{\mathcal{X}(\theta)}\right)^{\mathcal{X}(\theta)f(\theta)}$ by writing it in term of $D_{KL}(\mathcal{X}\|\mathcal{Y})$ and a small term which will be bounded later.

$$\begin{aligned}
\prod_{\theta \in \mathrm{supp}(\mathcal{Z})} \left(\frac{\mathcal{Y}(\theta)}{\mathcal{X}(\theta)}\right)^{\mathcal{X}(\theta)f(\theta)} &= \left(\prod_{\theta \in \mathrm{supp}(\mathcal{X})} \left(\frac{\mathcal{Y}(\theta)}{\mathcal{X}(\theta)}\right)^{\mathcal{X}(\theta)}\right)\left(\prod_{\theta \in \mathrm{supp}(\mathcal{T})} \left(\frac{\mathcal{X}(\theta)}{\mathcal{Y}(\theta)}\right)^{\mathcal{X}(\theta)(1-f(\theta))}\right) \\
&= \frac{1}{e^{D_{KL}(\mathcal{X}\|\mathcal{Y})}}\left(\prod_{\theta \in \mathrm{supp}(\mathcal{T})} \left(\frac{\mathcal{X}(\theta)}{\mathcal{Y}(\theta)}\right)^{\mathcal{X}(\theta)(1-f(\theta))}\right) \\
(\text{Since } D_{KL}(\mathcal{X}\|\mathcal{Y}) \leqslant \kappa) &\geqslant e^{-\kappa}\left(\prod_{\theta \in \mathrm{supp}(\mathcal{T})} \left(\frac{\mathcal{X}(\theta)}{\mathcal{Y}(\theta)}\right)^{\mathcal{X}(\theta)(1-f(\theta))}\right)
\end{aligned}$$

Intuitively, the term $\prod_{\theta \in \mathrm{supp}(\mathcal{T})} \left(\frac{\mathcal{X}(\theta)}{\mathcal{Y}(\theta)}\right)^{\mathcal{X}(\theta)(1-f(\theta))}$ should not be much smaller than one since the sum of the exponent is just $\sum_{\theta \in \mathrm{supp}(\mathcal{T})} \mathcal{X}(\theta)(1-f(\theta)) = \delta$. Indeed, this term is small as we can bound it as follows:

$$\begin{aligned}
\prod_{\theta \in \mathrm{supp}(\mathcal{T})} \left(\frac{\mathcal{X}(\theta)}{\mathcal{Y}(\theta)}\right)^{\mathcal{X}(\theta)(1-f(\theta))} &= \left(\prod_{\theta \in \mathrm{supp}(\mathcal{T})} \left(\frac{\delta}{1-f(\theta)} \cdot \frac{\mathcal{T}(\theta)}{\mathcal{Y}(\theta)}\right)^{\mathcal{T}(\theta)}\right)^{\delta} \\
&\geqslant \left(\prod_{\theta \in \mathrm{supp}(\mathcal{T})} \left(\delta \cdot \frac{\mathcal{T}(\theta)}{\mathcal{Y}(\theta)}\right)^{\mathcal{T}(\theta)}\right)^{\delta} \\
&= \delta^{\delta}\left(e^{D_{KL}(\mathcal{T}\|\mathcal{Y})}\right)^{\delta} \\
&\geqslant \delta^{\delta}
\end{aligned}$$

The last inequality comes from the fact that the informational divergence of any two distributions is no less than zero.

Combining the three inequalities, we have $\mathbb{E}_{\theta\sim\mathcal{Y}}[f(\theta)] \geqslant (1-\delta)\left(e^{-\kappa}\delta^{\delta}\right)^{\frac{1}{1-\delta}}$, as desired. □

Now, we will use Lemma 3.27 to give a new bound for the value of the output from independent rounding on a k-level κ-independent solution of the Sherali-Adams Hierarchy.

Lemma 3.28. *If $\{\mathcal{X}_S\}$ is a k-level κ-independent SA solution of value $1-\delta$ for a* Max k-CSP *instance $(V, \mathcal{W}, \{P_S\})$, independent rounding gives an assignment of value $\geqslant (\delta^{\delta}e^{-\kappa})^{\frac{1}{1-\delta}}(1-\delta)$.*

Proof. Again, we assume without loss of generality that $\delta \notin \{0,1\}$.

For each k-size set $S = \{x_{i_1}, \dots, x_{i_k}\}$, let $\kappa_S = D_{KL}(\mathcal{X}_S\|\mathcal{X}_{i_1} \times \cdots \times \mathcal{X}_{i_k})$ and $\delta_S = 1 - \mathbb{E}_{\phi_S\sim\mathcal{X}_S}[P_S(\phi_S)]$. Recall that the value of $\{\mathcal{X}_S\}$ in the SA relaxation is $\mathbb{E}_{S\sim\mathcal{W}}[\mathbb{E}_{\phi_S\sim\mathcal{X}_S}[P_S(\phi_S)]] = (1-\delta)$. Hence, we have $\mathbb{E}_{S\sim\mathcal{W}}[\delta_S] = \delta$. Moreover, since $\{\mathcal{X}_S\}$ is κ-independent, we have $\mathbb{E}_{S\sim\mathcal{W}}[\kappa_S] \leqslant \kappa$.

As stated earlier, the independent rounding algorithm gives an assignment of expected value

$$\mathbb{E}_{S=\{x_{i_1},\dots,x_{i_k}\}\sim\mathcal{W}}\left[\mathbb{E}_{\phi_S\sim\mathcal{X}_{i_1}\times\cdots\times\mathcal{X}_{i_k}}\left[P_S(\phi_S)\right]\right].$$

From Lemma 3.27, we have $\mathbb{E}_{\phi_S\sim\mathcal{X}_{i_1}\times\cdots\times\mathcal{X}_{i_k}}\left[P_S(\phi_S)\right] \geqslant (\delta_S^{\delta_S}e^{-\kappa_S})^{\frac{1}{1-\delta_S}}(1-\delta_S)$. Thus, the assignment from the rounding procedure has value at least $\mathbb{E}_{S\sim\mathcal{W}}[(\delta_S^{\delta_S}e^{-\kappa_S})^{\frac{1}{1-\delta_S}}(1-\delta_S)]$.

Next, let $\mathcal{Y}$ and $\mathcal{Z}$ be distributions on $\binom{V}{k}$ defined by $\mathcal{Y}(S) = \frac{\mathcal{W}(S)(1-\delta_S)}{(1-\delta)}$ and $\mathcal{Z}(S) = \frac{\mathcal{W}(S)\delta_S}{\delta}$. $\mathcal{Y}$ and $\mathcal{Z}$ are valid distributions since $\mathbb{E}_{S\sim\mathcal{W}}[\delta_S] = \delta$.

We can now bound $\mathbb{E}_{S\sim\mathcal{W}}[(\delta_S^{\delta_S}e^{-\kappa_S})^{\frac{1}{1-\delta_S}}(1-\delta_S)]$ as follows:

$$\begin{aligned}
\mathbb{E}_{S\sim\mathcal{W}}[(\delta_S^{\delta_S}e^{-\kappa_S})^{\frac{1}{1-\delta_S}}(1-\delta_S)] &= \sum_{S\in\binom{V}{k}} \mathcal{W}(S)(\delta_S^{\delta_S}e^{-\kappa_S})^{\frac{1}{1-\delta_S}}(1-\delta_S) \\
&= (1-\delta)\sum_{S\in\operatorname{supp}(\mathcal{Y})} \mathcal{Y}(S)(\delta_S^{\delta_S}e^{-\kappa_S})^{\frac{1}{1-\delta_S}} \\
(\text{Weighted A.M.-G.M. inequality}) &\geqslant (1-\delta)\prod_{S\in\operatorname{supp}(\mathcal{Y})} \left(\delta_S^{\delta_S}e^{-\kappa_S}\right)^{\frac{\mathcal{Y}(S)}{1-\delta_S}} \\
&= (1-\delta)\left(\prod_{S\in\operatorname{supp}(\mathcal{Y})} \left(\delta_S^{\delta_S}e^{-\kappa_S}\right)^{\mathcal{W}(S)}\right)^{\frac{1}{1-\delta}} \\
(\text{Since } \mathbb{E}_{S\sim\mathcal{W}}[\kappa_S]=\kappa \text{ and } \operatorname{supp}(\mathcal{Y})\subseteq\operatorname{supp}(\mathcal{W})) &\geqslant (1-\delta)\left(e^{-\kappa}\prod_{S\in\operatorname{supp}(\mathcal{Y})}\delta_S^{\mathcal{W}(S)\delta_S}\right)^{\frac{1}{1-\delta}} \\
&= (1-\delta)\left(e^{-\kappa}\prod_{S\in\operatorname{supp}(\mathcal{Z})}\delta_S^{\mathcal{W}(S)\delta_S}\right)^{\frac{1}{1-\delta}}
\end{aligned}$$

The last equality is true because $\delta_S = 1$ for every $S \in \operatorname{supp}(\mathcal{Z}) - \operatorname{supp}(\mathcal{Y})$ and $\delta_S = 0$ for every $S \in \operatorname{supp}(\mathcal{Y}) - \operatorname{supp}(\mathcal{Z})$.

We can now write $\prod_{S\in\operatorname{supp}(\mathcal{Z})}\delta_S^{\mathcal{W}(S)\delta_S}$ as

$$\begin{aligned}
\prod_{S\in\operatorname{supp}(\mathcal{Z})}\delta_S^{\mathcal{W}(S)\delta_S} &= \left(\prod_{S\in\operatorname{supp}(\mathcal{Z})}\left(\delta\cdot\frac{\mathcal{Z}(S)}{\mathcal{W}(S)}\right)^{\mathcal{Z}(S)}\right)^{\delta} \\
&= \delta^{\delta}(e^{D_{KL}(\mathcal{Z}\|\mathcal{X})})^{\delta} \\
(\text{Since } D_{KL}(\mathcal{Z}\|\mathcal{X})\geqslant 0) &\geqslant \delta^{\delta}.
\end{aligned}$$

Combining the two inequality yields $\mathbb{E}_{S\sim\mathcal{W}}[(\delta_S^{\delta_S}e^{-\kappa_S})^{\frac{1}{1-\delta_S}}(1-\delta_S)] \geqslant (1-\delta)(e^{-\kappa}\delta^{\delta})^{\frac{1}{1-\delta}}$, which completes the proof of the lemma. □

3.4.3 New Approximation Guarantee for the Algorithm

With Lemma 3.26 and Lemma 3.28 set up, we now prove the algorithmic guarantee for Algorithm 1.

Theorem 3.29. *For any* Max k-CSP *instance $\mathcal{G}$ of value $1-\delta > 0$ and density $\Delta > 0$, Algorithm 1 runs in time $N^{O\left(\frac{ki}{(1-\delta)\Delta}\right)}$ and outputs an assignment of value at least $(1-\delta)\delta^{\frac{\delta}{1-\delta}}/q^{1/i}$.*

Proof. Observe that the running time is $(nq)^{O(r)}$ where r is the maximum level of the SAC relaxation solved by the algorithm. Since the program is a relaxation of Max k-CSP, λ is always at least $1-\delta$. By the condition of the loop, r is at most $1 + k + \frac{k^2 i}{(1-\delta)\Delta}$. Hence, the running time of the algorithm is $N^{O\left(\frac{ki}{(1-\delta)\Delta}\right)}$.

Next, we will argue about the value of the output assignment. From Lemma 3.26, there exists a set $T \subseteq V$ of size at most $\frac{k^2 i}{\lambda\Delta}$ and an assignment $\phi_T \in \Sigma^T$ such that $\mu|\phi_T$ is an $(\lambda \log q/i)$-independent solution. Moreover, from how SAC program is defined, we know that $\mathrm{val}_{SA}(\mu|\phi_T) \geqslant \lambda$. As a result, from Lemma 3.28, independent rounding on $\mu|\phi_T$ gives an assignment of value at least

$$((1-\lambda)^{1-\lambda} e^{-\lambda \log q/i})^{\frac{1}{\lambda}} \lambda = \lambda(1-\lambda)^{\frac{1-\lambda}{\lambda}}/q^{1/i}.$$

Finally, since $|T| \leqslant \frac{k^2 i}{\lambda\Delta} \leqslant r - k$, it is considered in the conditioning step of the algorithm. Thus, the output assignment is of value at least $\lambda(1-\lambda)^{\frac{1-\lambda}{\lambda}}/q^{1/i} \geqslant (1-\delta)\delta^{\frac{\delta}{1-\delta}}/q^{1/i}$. □

Observe that, when the instance is satisfiable, $\delta = 0$ and the value of the output assignment is at least $1/q^{1/i}$. By taking i to be large enough, one arrives at a quasi-polynomial time approximation scheme (QPTAS) for dense Max k-CSP, as stated below. We note that our algorithm unfortunately does not give a QPTAS for the nonsatisfiable case since we also lose an additional factor of $\delta^{\frac{\delta}{1-\delta}}$ in the value of the output solution.

Corollary 3.30. *There is an algorithm that, given a satisfiable Δ-dense* Max k-CSP *instance $\mathcal{G}$ and any $1/2 > \varepsilon > 0$, runs in $N^{O\left(\frac{k \log q}{\varepsilon\Delta}\right)}$ time and output an assignment to $\mathcal{G}$ of value at least $1-\varepsilon$.*

Proof. Run Algorithm 1 with $i = \log q/\log(1+\varepsilon)$. From Theorem 3.29, the output assignment has value at least $q^{1/i} = 1/(1+\varepsilon) \geqslant 1-\varepsilon$ while the running time is $N^{O\left(\frac{ki}{\Delta}\right)}$. Finally, we conclude by observing that $i = \log q/\log(1+\varepsilon) \leqslant O(\log q/\varepsilon)$, which follows from the Bernoulli's inequality. □

3.5 Discussion and Open Problems

We prove that birthday repetition can amplify gap in hardness of approximation. This has several interesting consequences to the approximability of dense Max k-CSP. First, we prove almost-polynomial ratio polynomial-time ETH-hardness for the problem. Second, we show, assuming the stronger Gap-ETH, that it is impossible to approximate dense Max 2-CSP to within factor $N^{1/i}$

in time $N^{\widetilde{O}(i)}$. Third, we prove a similar integrality gap for Lasserre relaxation of the problem. Moreover, we provide an approximation algorithm that almost matches our lower bound based on Gap-ETH and the Lasserre integrality gap.

While our results settle down the approximability of dense MAX k-CSP up to the dependency on k and a factor of polylog$i$ in the exponent, our work also raises many interesting questions, which we list below.

- *What is the right dependency on ε in the birthday repetition theorem?* It is unclear whether the dependency of $1/\log(1/\varepsilon)$ is needed in the exponent. We remark here that, in the case that $\mathcal{Q}$ is uniform over a regular graph, Ko [Ko18] gives a lower bound that is tight up to the factor of $1/\log(1/\varepsilon)$ in the exponent.

- *Can our approximation algorithm for dense k-*CSP *be made to run in $q^{O_k(i)}+N^{O(1)}$ time?* As stated earlier, Yaroslavtev's algorithm [Yar14] runs in $q^{O_k(\log q/\varepsilon^2)} + N^{O(1)}$ time and provides an ε additive approximation to the problem. As for our algorithm, we can, in fact, turn the condioning step into a randomized algorithm where we just randomly pick a set and an assignment to condition[7], which takes only linear time. The bottleneck, however, is solving the linear program (SAC relaxation), which takes $N^{\Omega(r)}$ time where r is the number of rounds. Related to this, Barak et al. [BRS11] showed that their Lasserre hierarchy-based algorithm runs in $2^r N^{O(1)}$ instead of $N^{O(r)}$ time[8]. It is an interesting question to ask whether our algorithm can also be sped up using their technique.

- *Can Lemma 3.27 be used to prove new approximation guarantees for other problems?* Lemma 3.27 is a generic bound on the (multiplicative) difference of expectations of a function on two distributions based on their informational divergence. Hence, it may yield new approximation guarantees for other correlation-based algorithms as well.

- *Is it possible to prove a result similar to Lemma 3.28 without losing a constant factor?* Lemma 3.28 at the heart of our approximatin algorithm has one drawback: when δ is not zero, we always lose a factor of $\delta^{\frac{\delta}{1-\delta}}$. While the loss here is only constant (since it is minimized when $\delta \to 1$ which gives $\delta^{\frac{\delta}{1-\delta}} \geqslant 0.367$), it prevents us from getting a QPTAS for non-satisfiable dense MAX k-CSP. If this factor can be removed, we can establish the number of levels needed for any approximation ratio from as large as polynomial in q to as small as any constant.

[7] This is because the bound in Lemma 3.26 on total correlation of conditioned solution actually holds (in expectation) for random T and ϕ_T sampled according to the marginal distribution $\mathcal{X}_T$.

[8] Note here that the number of rounds r used in Barak et al.'s algorithm is polynomial in the alphabet size q.

Chapter 4

Densest k-Subgraph with Perfect Completeness

In the DENSEST k-SUBGRAPH (DkS) problem, we are given an undirected graph G on n vertices and a positive integer $k \leqslant n$. The goal is to find a set S of k vertices such that the induced subgraph on S has maximum number of edges. Since the size of S is fixed, the problem can be equivalently stated as finding a k-subgraph (i.e. subgraph on k vertices) with maximum density.

DENSEST k-SUBGRAPH, a natural generalization of k-CLIQUE [Kar72], was first formulated and studied by Kortsarz and Peleg [KP93] in the early 90s. Since then, it has been the subject of intense study in the context of approximation algorithm and hardness of approximation [FS97; SW98; FL01; FKP01; AHI02; Fei02; Kho06; GL09; RS10; Bha+10; Alo+11; Bha+12; Bar15; Bra+17]. Despite this, its approximability still remains wide open and is considered by some to be an important open question in approximation algorithms [Bha+10; Bha+12; Bra+17].

On the algorithmic front, Kortsarz and Peleg [KP93], in the same work that introduced the problem, gave a polynomial-time $\widetilde{O}(n^{0.3885})$-approximation algorithm for DkS. Feige, Kortsarz and Peleg [FKP01] later provided an $O(n^{1/3-\delta})$-approximation for the problem for some constant $\delta \approx 1/60$. This approximation ratio was the best known for almost a decade [1] until Bhaskara et al. [Bha+10] invented a log-density based approach which yielded an $O(n^{1/4+\varepsilon})$-approximation for any constant $\varepsilon > 0$. This remains the state-of-the-art approximation algorithm for DkS.

While the above algorithms demonstrate the main progresses of approximations of DkS in general case over the years, many special cases have also been studied. Most relevant to our work is the case where the optimal k-subgraph has high density (e.g. is a k-Clique), in which better approximations are known [FS97; ST08; MM15; Bar15]. The first and most representative algorithm of this kind is that of Feige and Seltser [FS97], which provides the following guarantee: when the input graph contains a k-clique, the algorithm can find an $(1-\varepsilon)$-dense k-subgraph in $n^{O(\log n/\varepsilon)}$ time. We will refer to this problem of finding densest k-subgraph when the input graph is promised to have a k-clique DENSEST k-SUBGRAPH *with perfect completeness*. DkS with perfect

[1] Around the same time as Bhaskara et al.'s work [Bha+10], Goldstein and Langberg [GL09] presented an algorithm with approximation ratio $O(n^{0.3159})$, which is slightly better than [FKP01] but is worse than [Bha+10].

completeness should of course reminds us of dense CSPs from the previous section, as both have quasi-polynomial time algorithm and hence are unlikely to be NP-hard to approximate. Moreover, Feige and Seltser's algorithm [FS97] can[2] achieve approximation ratio n^{ε} in time $n^{O(1/\varepsilon)}$, further suggesting the similarity between dense CSPs and DkS with perfect completeness.

Although many algorithms have been devised for DkS, relatively little is known regarding its hardness of approximation. While it is commonly believed that the problem is hard to approximate to within some polynomial ratio [Alo+11; Bha+12], not even a constant factor NP-hardness of approximation is known. To circumvent this, Feige [Fei02] came up with a hypothesis that a random 3SAT formula is hard to refute in polynomial time and proved that, assuming this hypothesis, DkS is hard to approximate to within some constant factor.

Alon et al. [Alo+11] later used a similar conjecture regarding random k-AND to rule out polynomial-time algorithms for DkS with any constant approximation ratio. Moreover, they proved hardnesses of approximation of DkS under the following *Planted Clique Hypothesis* [Jer92; Kuč95]: there is no polynomial-time algorithm that can distinguish between a typical Erdős-Rényi random graph $\mathcal{G}(n, 1/2)$ and one in which a clique of size polynomial in n (e.g. $n^{1/3}$) is planted. Assuming this hypothesis, Alon et al. proved that no polynomial-time algorithm approximates DkS to within any constant factor. They also showed that, when the hypothesis is strengthened to rule out not only polynomial-time but also super-polynomial time algorithms for the Planted Clique problem, their inapproximability guarantee for DkS can be improved. In particular, if no $n^{O(\sqrt{\log n})}$-time algorithm solves the Planted Clique problem, then $2^{O(\log^{2/3} n)}$-approximation for DkS cannot be achieved in polynomial time.

There are also several inapproximability results of DkS based on worst-case assumptions. Khot [Kho06] showed, assuming NP $\nsubseteq$ BPTIME($2^{n^{\varepsilon}}$) for some constant $\varepsilon > 0$, that no polynomial-time algorithm can approximate DkS to within $(1 + \delta)$ factor where $\delta > 0$ is a constant depending only on ε; the proof is based on a construction of a "quasi-random" PCP, which is then used in place of a random 3SAT in a reduction similar to that from [Fei02].

While no inapproximability of DkS is known under the Unique Games Conjecture, Raghavendra and Steurer [RS10] showed that a strengthened version of it, in which the constraint graph is required to satisfy a "small-set expansion" property, implies that DkS is hard to approximate to within any constant ratio.

Recently, Braverman et al. [Bra+17], showed, assuming ETH, that, for some constant $\varepsilon > 0$, no $n^{\widetilde{O}(\log n)}$-time algorithm can approximate DENSEST k-SUBGRAPH with perfect completeness to within $(1 + \varepsilon)$ factor. Their result matches almost exactly with the previously mentioned Feige-Seltser algorithm [FS97]. In fact, their construction uses the "label-extended graph" of the birthday repetition game described in the previous section.

Since none of these inapproximability results achieve a polynomial ratio, there have been efforts to prove better lower bounds for restricted classes of algorithms. For example, Bhaskara et al. [Bha+12] provided polynomial ratio lower bounds against SDP relaxations of DkS. Specifically, for the Sum-of-Squares hierarchy, they showed integrality gaps of $n^{2/53-\varepsilon}$ and n^{ε} against

[2] This guarantee was not stated explicitly in [FS97] but it can be easily achieved by changing the degree threshold in their algorithm **DenseSubgraph** from $(1 - \varepsilon)n$ to n^{ε}.

$n^{\Omega(\varepsilon)}$ and $n^{1-O(\varepsilon)}$ levels of the hierarchy respectively. (See also [Man15; Chl+17b] in which $2/53$ in the exponent was improved to $1/14$.) Unfortunately, it is unlikely that these lower bounds can be translated to inapproximability results and the question of whether any polynomial-time algorithm can achieve subpolynomial approximation ratio for DkS remains an intriguing open question.

Our Results

We rule out, under the exponential time hypothesis (Hypothesis 1), polynomial-time approximation algorithms for DkS (even with perfect completeness) with slightly subpolynomial ratio:

Theorem 4.1. *There exists $c > 0$ such that, assuming ETH, no polynomial-time algorithm can, given a graph G on n vertices and a positive integer $k \leqslant n$, distinguish between the following two cases:*

- *There exist k vertices of G that induce a k-clique.*
- *Every k-subgraph of G has density at most $n^{-1/(\log\log n)^c}$.*

If we assume the stronger gap exponential time hypothesis (Hypothesis 3), the ratio can be improved to $n^{g(n)}$ for any [3] $g \in o(1)$. In fact, we can get an even finer-grained trade-off: to achieve n^{ε}-approximation, the running time of the algorithm has to be $n^{\widetilde{\Omega}(1/\varepsilon^{1/3})}$, as formalized below.

Theorem 4.2. *For any $\nu > 0$, assuming Gap-ETH, there is no algorithm that, given any graph G and any $\varepsilon > 0$, runs in $n^{O\left(\frac{1}{\varepsilon^{1/3}\cdot\log^{4/3+\nu}(1/\varepsilon)}\right)}$ time and can distinguish between the following two cases:*

- *There exist k vertices of G that induce a k-clique.*
- *Every k-subgraph of G has density at most $n^{-\varepsilon}$.*

Recall that, for DkS with perfect completeness, the aforementioned Feige-Seltser algorithm achieves an n^{ε}-approximation in time $n^{O(1/\varepsilon)}$ for every $\varepsilon > 0$ [FS97]. Our above lower bound indeed confirms that, to get better approximation ratio, more running time is needed. Nonetheless, it does not yet resolve the correct trade-off between these parameters, and it remains an interesting open question to bridge the gap of the running time between $n^{O(1/\varepsilon)}$ in the upper bound of [FS97] and $n^{\widetilde{\Omega}(1/\varepsilon^{1/3})}$ in our above lower bound.

Comparison to Previous Results. In terms of inapproximability ratios, the ratios ruled out in this work are almost polynomial and provides a vast improvement over previous results. Prior to our result, the best known ratio ruled out under any worst case assumption is only any constant [RS10] and the best ratio ruled out under any average case assumption is only $2^{O(\log^{2/3} n)}$ [Alo+11]. In addition, our results also have perfect completeness, which was only achieved in [Bra+17] under ETH and in [Alo+11] under the Planted Clique Hypothesis but not in [Kho06; Fei02; RS10].

[3] Recall that $g \in o(1)$ if and only if $\lim_{n\to\infty} g(n) = 0$.

Regarding the assumptions our results are based upon, the average case assumptions used in [Fei02; Alo+11] are incomparable to ours. The assumption NP $\not\subseteq$ BPTIME($2^{n^{\varepsilon}}$) used in [Kho06] is also incomparable to ours since, while not stated explicitly, ETH and Gap-ETH by default focus only on deterministic algorithms and our reductions are also deterministic. The strengthened Unique Games Conjecture used in [RS10] is again incomparable to ours as it is a statement that a specific problem is NP-hard. Finally, although Braverman et al.'s result [Bra+17] also relies on ETH, its relation to our results is more subtle. Specifically, their reduction time is only $2^{\widetilde{\Theta}(\sqrt{m})}$ where m is the number of clauses, meaning that they only need to assume that 3SAT $\notin$ DTIME($2^{\widetilde{\Theta}(\sqrt{m})}$) to rule out a constant ratio polynomial-time approximation for DkS. However, as we will see in Theorem 4.4, even to achieve a constant gap, our reduction time is $2^{\widetilde{\Omega}(m^{3/4})}$. Hence, if 3SAT somehow ends up in DTIME($2^{\widetilde{\Theta}(m^{3/4})}$) but outside of DTIME($2^{\widetilde{\Theta}(\sqrt{m})}$), their result will still hold whereas ours will not even imply constant ratio inapproximability for DkS.

Implications of Our Results. One of the reasons that DkS has received significant attention in the approximation algorithm community is due to its connections to many other problems. Most relevant to our work are the problems to which there are reductions from DkS that preserve approximation ratios to within some polynomial[4]. These problems include DENSEST AT-MOST-k-SUBGRAPH [AC09], SMALLEST m-EDGE SUBGRAPH [CDK12], STEINER k-FOREST [HJ06] and QUADRATIC KNAPSACK [Pis07]. For brevity, we do not define these problems here. We refer interested readers to cited sources for their definitions and reductions from DkS to respective problems. We also note that this list is by no means exhaustive and there are indeed numerous other problems with similar known connections to DkS (see e.g. [Haj+06; KS07; Kor+11; CHK11; HIM11; LNV14; Che+15a; CL15; CZ15; SFL15; TV15; Chl+16; Chu+15; Lee16]). Our results also imply hardness of approximation results with similar ratios to DkS for such problems:

Corollary 4.3. *For some $c > 0$, assuming ETH, there is no polynomial-time $n^{1/(\log\log n)^c}$-approximation algorithm for* DENSEST AT-MOST-k-SUBGRAPH, SMALLEST m-EDGE SUBGRAPH, STEINER k-FOREST, QUADRATIC KNAPSACK. *Moreover, for any function $f \in o(1)$, there is no polynomial-time $n^{f(n)}$-approximation algorithm for any of these problems, unless Gap-ETH is false.*

4.1 The Reduction and Proofs of The Main Theorems

The reduction from Gap-3SAT to DkS is simple. Given a 3SAT formula ϕ on n variables $x_1, \dots, x_n$ and an integer $1 \leqslant \ell \leqslant n$, we construct a graph[5] $G_{\phi,\ell} = (V_{\phi,\ell}, E_{\phi,\ell})$ as follows:

[4] These are problems whose $O(\rho)$-approximation gives an $O(\rho^c)$-approximation for DkS for some constant c.

[5] For interested readers, we note that our graph is *not* the same as the FGLSS graph [Fei+91] of the PCP in which the verifier reads ℓ random variables and accepts if no clause is violated; while this graph has the same vertex set as ours, the edges are different since we check that no clause between the two vertices is violated, which is not checked in the FGLSS graph. It is possible to modify our proof to make it work for this FGLSS graph. However, the soundness guarantee for the FGLSS graph is worse.

- Its vertex set $V_{\phi,\ell}$ contains all partial assignments to ℓ variables. That is, each vertex is $\{(x_{i_1}, b_{i_1}), \dots, (x_{i_\ell}, b_{i_\ell})\}$ where $x_{i_1}, \dots, x_{i_\ell}$ are ℓ distinct variables and $b_{i_1}, \dots, b_{i_\ell} \in \{0, 1\}$ are the bits assigned to them.
- We connect two vertices $\{(x_{i_1}, b_{i_1}), \dots, (x_{i_\ell}, b_{i_\ell})\}$ and $\{(x_{i'_1}, b_{i'_1}), \dots, (x_{i'_\ell}, b_{i'_\ell})\}$ by an edge iff the two partial assignments are consistent (i.e. no variable is assigned 0 in one vertex and 1 in another), and, every clause in ϕ all of whose variables are from $x_{i_1}, \dots, x_{i_\ell}, x_{i'_1}, \dots, x_{i'_\ell}$ is satisfied by the partial assignment induced by the two vertices.

Clearly, if $\text{val}(\phi) = 1$, the $\binom{n}{\ell}$ vertices corresponding to a satisfying assignment induce a clique. Our main technical contribution is proving that, when $\text{val}(\phi) \leqslant 1 - \varepsilon$, every $\binom{n}{\ell}$-subgraph is sparse:

Theorem 4.4. *For any $d, \varepsilon > 0$, there exists $\tau > 0$ such that, for any 3SAT formula ϕ on n variables such that $\text{val}(\phi) \leqslant 1 - \varepsilon$ and each variable appears in at most d clauses and for any integer $\ell \in [n^{3/4}/\tau, n/2]$, any $\binom{n}{\ell}$-subgraph of $G_{\phi,\ell}$ has density $\leqslant 2^{-\tau\ell^4/n^3}$.*

We remark that there is nothing special about 3SAT; we can start with any boolean CSP and end up with a similar result, albeit the soundness deteriorates as the arity of the CSP grows. However, it is crucial that the variables are boolean; in fact, Braverman et al. [Bra+17] considered a graph similar to ours for 2CSPs but they were unable to achieve subconstant soundness since their variables were not boolean[6]. Specifically, there is a non-boolean 2CSP with low value which results in the graph having a biclique of size $\geqslant \binom{n}{\ell}$ (see Appendix A of [Man17a]), i.e., one cannot get an inapproximability ratio more than two starting from a non-boolean CSP.

Once we have Theorem 4.4, the inapproximability results of DkS (Theorem 4.1 and 4.2) can be easily proved by applying the theorem with appropriate choices of ℓ. We defer these proofs to Subsection 4.1.2. For now, let us turn our attention to the proof of Theorem 4.4. Recall that Alon's lemma [Alo02] (Lemma 2.14) states that every dense graph contains many labelled copies of bicliques. Equipped with this lemma, our proof strategy is to bound the number of labelled copies of $K_{t,t}$ in $G_{\phi,\ell}$ where t is to be chosen later.

Before we proceed with the proof, we note here that, while the overall proof strategy is similar to that of the previous chapter, here we have to bound the number of copies of bicliques in the constructed graph whereas we previously only had to bound the number of copies of bicliques in the constraint graph.

To bound the number of labelled copies of $K_{t,t}$ in $G_{\phi,\ell}$, we will need some additional notations:

- First, let $A_\phi := \{(x_1, 0), (x_1, 1), \dots, (x_n, 0), (x_n, 1)\}$ be the set of all single-variable partial assignments. Observe that $V_{\phi,\ell} \subseteq \binom{A_\phi}{\ell}$, i.e., each $u \in V_{\phi,\ell}$ is a subset of A_ϕ of size ℓ.
- Let $\mathcal{A} : (V_{\phi,\ell})^t \to \mathscr{P}(A_\phi)$ be a "flattening" function that, on input $T \in (V_{\phi,\ell})^t$, outputs the set of all single-variable partial assignments that appear in at least one vertex in T. In other words, when each vertex u is viewed as a subset of A_ϕ, we can write $\mathcal{A}(T)$ simply as $\bigcup_{u \in T} u$.

[6] Any satisfiable boolean 2CSP is solvable in polynomial time so one cannot start with a boolean 2CSP either.

- Let $\mathcal{K}_{t,t} := \{(L, R) \in (V_{\phi,\ell})^t \times (V_{\phi,\ell})^t \mid \forall u \in L, \forall v \in R, u \neq v \wedge (u, v) \in E_{\phi,\ell}\}$ denote the set of all labelled copies of $K_{t,t}$ in $G_{\phi,\ell}$ and, for each $A, B \subseteq A_\phi$, let $\mathcal{K}_{t,t}(A, B) := \{(L, R) \in \mathcal{K}_{t,t} \mid \mathcal{A}(L) = A, \mathcal{A}(R) = B\}$ denote the set of all $(L, R) \in \mathcal{K}_{t,t}$ with $\mathcal{A}(L) = A$ and $\mathcal{A}(R) = B$.

The number of labelled copies of $K_{t,t}$ in $G_{\phi,\ell}$ can be written as

$$|\mathcal{K}_{t,t}| = \sum_{A,B \subseteq A_\phi} |\mathcal{K}_{t,t}(A, B)|. \tag{4.1}$$

To bound $|\mathcal{K}_{t,t}|$, we will prove the following bound on $|\mathcal{K}_{t,t}(A, B)|$.

Lemma 4.5. *Let ϕ, n, ℓ, d and ε be as in Theorem 4.4. There exists $\lambda > 0$ depending only on d and ε such that, for any $t \in \mathbb{N}$ and any $A, B \subseteq A_\phi$, $|\mathcal{K}_{t,t}(A, B)| \leqslant \left(2^{-\lambda\ell^2/n}\binom{n}{\ell}\right)^{2t}$.*

Before we prove the above lemma, let us see how Lemma 2.14 and Lemma 4.5 imply Theorem 4.4.

Proof of Theorem 4.4. Assume w.l.o.g. that $\lambda \leqslant 1$. Pick $\tau = \lambda^2/8$ and $t = (4/\lambda)(n^2/\ell^2)$. From Lemma 4.5 and (4.1), we have

$$|\mathcal{K}_{t,t}| \leqslant 2^{4n} \cdot \left(2^{-\lambda\ell^2/n}\binom{n}{\ell}\right)^{2t} \leqslant (2^{-\lambda\ell^2/n})^t \cdot \binom{n}{\ell}^{2t}$$

where the second inequality comes from our choice of t; note that t is chosen so that the 2^{4n} factor is consumed by $2^{-\lambda\ell^2/n}$ from Lemma 4.5. Finally, consider any $\binom{n}{\ell}$-subgraph of $G_{\phi,\ell}$. By the above bound, it contains at most $(2^{-\lambda\ell^2/n})^t \cdot \binom{n}{\ell}^{2t}$ labelled copies of $K_{t,t}$. Thus, from Lemma 2.14 and from $\ell \geqslant n^{3/4}/\delta$, its density is at most $2 \cdot 2^{-\lambda\ell^2/(nt)} = 2 \cdot 2^{-2\tau\ell^4/n^3} \leqslant 2^{-\tau\ell^4/n^3}$ as desired. □

We now move on to the proof of Lemma 4.5.

Proof of Lemma 4.5. First, notice that if (x, b) appears in A and $(x, \neg b)$ appears in B for some variable x and bit b, then $\mathcal{K}_{t,t}(A, B) = \emptyset$; this is because, for any L with $\mathcal{A}(L) = A$ and R with $\mathcal{A}(R) = B$, there exist $u \in L$ and $v \in R$ that contain (x, b) and $(x, \neg b)$ respectively, meaning that there is no edge between u and v and, thus, $(L, R) \notin \mathcal{K}_{t,t}(A, B)$. Hence, from now on, we can assume that, if (x, b) appears in one of A, B, then the other does not contain $(x, \neg b)$. Observe that this implies that, for each variable x, its assignments can appear in A and B at most two times[7] in total. This in turn implies that $|A| + |B| \leqslant 2n$.

Let us now argue that $|\mathcal{K}_{t,t}(A, B)| \leqslant \binom{n}{\ell}^{2t}$; while this is not the bound we are looking for yet, it will serve as a basis for our argument later. For every $(L, R) \in \mathcal{K}_{t,t}(A, B)$, observe that, since

[7] This is where we use the fact that the variables are boolean. For non-boolean CSPs, each variable x can appear more than two times in one of A or B alone, which can indeed be problematic (see Appendix A of [Man17a]).

$\mathcal{A}(L) = A$ and $\mathcal{A}(R) = B$, we have $L \in \binom{A}{\ell}^t$ and $R \in \binom{B}{\ell}^t$. This implies that $\mathcal{K}_{t,t}(A, B) \subseteq \binom{A}{\ell}^t \times \binom{B}{\ell}^t$. Hence,

$$|\mathcal{K}_{t,t}(A, B)| \leqslant \binom{|A|}{\ell}^t \binom{|B|}{\ell}^t. \tag{4.2}$$

Moreover, $\binom{|A|}{\ell}\binom{|B|}{\ell}$ can be further bounded as

$$\binom{|A|}{\ell}\binom{|B|}{\ell} = \frac{1}{(\ell!)^2}\prod_{i=0}^{\ell-1}(|A| - i)(|B| - i) \leqslant \frac{1}{(\ell!)^2}\prod_{i=0}^{\ell-1}\left(\frac{|A| + |B|}{2} - i\right)^2 \leqslant \binom{n}{\ell}^2 \tag{4.3}$$

where the inequalities come from the AM-GM Inequality and from $|A| + |B| \leqslant 2n$ respectively. Combining (4.2) and (4.3) indeed yields $|\mathcal{K}_{t,t}(A, B)| \leqslant \binom{n}{\ell}^{2t}$.

Inequality (4.2) is very crude; we include all elements of $\binom{A}{\ell}$ and $\binom{B}{\ell}$ as candidates for vertices in L and R respectively. However, as we will see soon, only tiny fraction of elements of $\binom{A}{\ell}$, $\binom{B}{\ell}$ can actually appear in L, R when $(L, R) \in \mathcal{K}_{t,t}(A, B)$. To argue this, let us categorize the variables into three groups:

- x is *terrible* iff its assignments appear at most once in total in A and B (i.e. $|\{(x, 0), (x, 1)\} \cap A| + |\{(x, 0), (x, 1)\} \cap B| \leqslant 1$).
- x is *good* iff, for some $b \in \{0, 1\}$, $(x, b) \in A \cap B$. Note that this implies that $(x, \neg b) \notin A \cup B$.
- x is *bad* iff either $\{(x, 0), (x, 1)\} \subseteq A$ or $\{(x, 0), (x, 1)\} \subseteq B$.

The next and last step of the proof is where birthday-type paradoxes come in. Before we continue, let us briefly demonstrate the ideas behind this step by considering the following extreme cases:

- If all variables are terrible, then $|A| + |B| \leqslant n$ and (4.3) can be immediately tightened.
- If all variables are bad, assume w.l.o.g. that, for at least half of variables x's, $\{(x, 0), (x, 1)\} \subseteq A$. Consider a random element u of $\binom{A}{\ell}$. Since u is a set of random ℓ distinct elements of A, there will, in expectation, be $\Omega(\ell^2/n)$ variables x's with $(x, 0), (x, 1) \in u$. However, the presence of such x's means that u is not a valid vertex. Moreover, it is not hard to turn this into the following probabilistic statement: with probability at most $2^{-\Omega(\ell^2/n)}$, u contains at most one of $(x, 0), (x, 1)$ for every variable x. In other words, only $2^{-\Omega(\ell^2/n)}$ fraction of elements of $\binom{A}{\ell}$ are valid vertices, which yields the desired bound on $|\mathcal{K}_{t,t}(A, B)|$.
- If all variables are good, then $A = B$ is simply an assignment to all the variables. Since $\text{val}(\phi) \leqslant 1 - \varepsilon$, at least εm clauses are unsatisfied by this assignment. As we will argue below, every element of $\binom{A}{\ell}$ that contains two variables from some unsatisfied clause cannot

be in L for any $(L, R) \in \mathcal{K}_{t,t}(A, B)$. This means that there are $\Theta_\varepsilon(m) \geqslant \Omega_\varepsilon(n)$ prohibited pairs of variables that cannot appear together. Again, similar to the previous case, it is not hard to argue that only $2^{-\Omega_{\varepsilon,d}(\ell^2/n)}$ fraction of elements of $\binom{A}{\ell}$ can be candidates for vertices of L.

To turn this intuition into a bound on $|\mathcal{K}_{t,t}(A, B)|$, we need the following inequality. Its proof is straightforward and is deferred to Subsection 4.1.1.

Proposition 4.6. *Let U be any set and $P \subseteq \binom{U}{2}$ be any set of pairs of elements of U such that each element of U appears in at most q pairs. For any positive integer $2 \leqslant r \leqslant |U|$, the probability that a random element of $\binom{U}{r}$ does not contain both elements of any pair in P is at most* $\exp\left(-\frac{|P|r^2}{4q|U|^2}\right)$.

We are now ready to formalize the above intuition and finish the proof of Lemma 4.5. For the sake of convenience, denote the sets of good, bad and terrible variables by X_g, X_b and X_t respectively. Moreover, let $\beta := \varepsilon/(100d)$ and pick $\lambda = \min\{-\log(1-\beta/2), \beta/64, \varepsilon/(384d)\}$. To refine the bound on the size of $\mathcal{K}_{t,t}(A, B)$, consider the following three cases:

1. $|X_t| \geqslant \beta n$. Since each $x \in X_t$ contributes at most one to $|A| + |B|$, $|A| + |B| \leqslant (1 - \beta/2)(2n)$. Hence, we can improve (4.3) to $\binom{|A|}{\ell}\binom{|B|}{\ell} \leqslant \binom{(1-\beta/2)n}{\ell}^2$. Thus, we have

$$|\mathcal{K}_{t,t}(A, B)| \overset{(4.2)}{\leqslant} \binom{|A|}{\ell}^t \binom{|B|}{\ell}^t \leqslant \binom{(1-\beta/2)n}{\ell}^{2t} \leqslant \left((1-\beta/2)^\ell \binom{n}{\ell}\right)^{2t} \leqslant \left(2^{-\lambda \ell^2/n}\binom{n}{\ell}\right)^{2t}$$

where the last inequality comes from $\lambda \leqslant -\log(1-\beta/2)$ and $\ell > \ell^2/n$.

2. $|X_b| \geqslant \beta n$. Since each $x \in X_b$ appears either in A or B, one of A and B must contain assignments to at least $(\beta/2)n$ variables in X_b. Assume w.l.o.g. that A satisfies this property. Let X_b^L be the set of all $x \in X_b$ whose assignments appear in A. Note that $|X_b^L| \geqslant (\beta/2)n$. Observe that an element $u \in \binom{A}{\ell}$ is not a valid vertex if it contains both $(x, 0)$ and $(x, 1)$ for some $x \in X_b^L$. We invoke Proposition 4.6 with $U = A$, $P = \{\{(x, 0), (x, 1)\} \mid x \in X_b^L\}, q = 1$ and $r = \ell$, which implies that a random element of $\binom{A}{\ell}$ does not contain any prohibited pairs in P with probability at most $\exp\left(-\frac{|X_b^L|\ell^2}{4|A|^2}\right) \leqslant \exp\left(-\frac{(\beta/2)n\ell^2}{4(2n)^2}\right)$, which is at most $2^{-2\lambda\ell^2/n}$ because $\lambda \leqslant \beta/64$. In other words, at most $2^{-2\lambda\ell^2/n}$ fraction of elements of $\binom{A}{\ell}$ are valid vertices. This gives us the following bound:

$$|\mathcal{K}_{t,t}(A, B)| \leqslant \left(2^{-2\lambda\ell^2/n} \cdot \binom{|A|}{\ell}\right)^t \cdot \binom{|B|}{\ell}^t \overset{(4.3)}{\leqslant} \left(2^{-\lambda\ell^2/n}\binom{n}{\ell}\right)^{2t}.$$

3. $|X_t| < \beta n$ and $|X_b| < \beta n$. In this case, $|X_g| > (1 - 2\beta)n$. Let S denote the set of clauses whose variables all lie in X_g. Since each variable appears in at most d clauses, $|S| > m - (2\beta n)d \geqslant (1 - \varepsilon/2)m$ where the second inequality comes from our choice of β and from $m \geqslant n/3$.

Consider the partial assignment $f : X_g \to \{0,1\}$ induced by A and B, i.e., $f(x) = b$ iff $(x, b) \in A, B$. Since $\operatorname{val}(\phi) \leqslant 1 - \varepsilon$, the number of clauses in S satisfied by f is at most $(1-\varepsilon)m$. Hence, at least $\varepsilon m/2$ clauses in S are unsatisfied by f. Denote the set of such clauses by S_{UNSAT}.

Fix a clause $C \in S_{\text{UNSAT}}$ and let x, y be two different variables in C. We claim that x, y cannot appear together in any vertex of L for any $(L, R) \in \mathcal{K}_{t,t}(A, B)$. Suppose for the sake of contradiction that $(x, f(x))$ and $(y, f(y))$ both appear in $u \in L$ for some $(L, R) \in \mathcal{K}_{t,t}(A, B)$. Let $z \in X_g$ be another variable[8] in C. Since $(z, f(z)) \in B$, some vertex $v \in R$ contains $(z, f(z))$. Thus, there is no edge between u and v in $G_{\phi,\ell}$, which contradicts with $(L, R) \in \mathcal{K}_{t,t}$.

We can now appeal to Proposition 4.6 with $U = A$, $q = 2d$, $r = \ell$ and P be the prohibited pairs described above. This implies that with probability at most $\exp\left(-\frac{|P|\ell^2}{8d|A|^2}\right) \leqslant \exp\left(-\frac{\varepsilon\ell^2}{192dn}\right)$, a random element of $\binom{A}{\ell}$ contains no prohibited pair from P. In other words, at most $\exp\left(-\frac{\varepsilon\ell^2}{192dn}\right)$ fraction of elements of $\binom{A}{\ell}$ can be candidates for each element of L for $(L, R) \in \mathcal{K}_{t,t}(A, B)$. This gives the following bound:

$$|\mathcal{K}_{t,t}(A, B)| \leqslant \left(\exp\left(-\frac{\varepsilon\ell^2}{192dn}\right) \cdot \binom{|A|}{\ell}\right)^t \cdot \binom{|B|}{\ell}^t \overset{(4.3)}{\leqslant} \left(2^{-\varepsilon\ell^2/(384dn)} \cdot \binom{n}{\ell}\right)^{2t}.$$

Since we picked $\lambda \leqslant \varepsilon/(384d)$, $|\mathcal{K}_{t,t}(A, B)|$ is once again bounded above by $\left(2^{-\lambda\ell^2/n}\binom{n}{\ell}\right)^{2t}$.

In all three cases, we have $|\mathcal{K}_{t,t}(A, B)| \leqslant \left(2^{-\lambda\ell^2/n}\binom{n}{\ell}\right)^{2t}$, completing the proof of Lemma 4.5. □

4.1.1 Proof of Proposition 4.6

Proof of Proposition 4.6. We first construct $P' \subseteq P$ such that each element of U appears in at most one pair in P' as follows. Start out by marking every pair in P as active and, as long as there are active pairs left, include one in P' and mark every pair that shares an element of U with this pair as inactive. Since each element of U appears in at most q pairs in P, we mark at most $2q$ pairs as inactive per each inclusion. This implies that $|P'| \geqslant |P|/(2q)$.

Suppose that $P' = \{\{a_1, b_1\}, \ldots, \{a_{|P'|}, b_{|P'|}\}\}$ where $a_1, b_1, \ldots, a_{|P'|}, b_{|P'|}$ are distinct elements of U. Let u be a random element of $\binom{U}{r}$. For each $i = 1, \ldots, |P'|$, we have

$$\begin{aligned}\Pr[\{a_i, b_i\} \not\subseteq u] &= 1 - \frac{\binom{|U|-2}{r-2}}{\binom{|U|}{r}} \\ &= 1 - \frac{r(r-1)}{|U|(|U|-1)}\end{aligned}$$

[8] If C contains two variables, let $z = x$. Note that we can assume w.l.o.g. that C contains at least two variables.

$$\begin{aligned}(\text{Since } r-1 \geqslant r/2 \text{ for all } r \geqslant 2) &\leqslant 1 - \frac{r^2}{2|U|^2}\\ (\text{Since } 1-z \leqslant \exp(-z) \text{ for all } z \in \mathbb{R}) &\leqslant \exp\left(-\frac{r^2}{2|U|^2}\right).\end{aligned}$$

If u does not contain both elements of any pairs in P, it does not contain both elements of any pairs in P'. The probability of the latter can be written as

$$\Pr\left[\bigwedge_{i=1}^{|P'|} \{a_i, b_i\} \not\subseteq u\right] = \prod_{i=1}^{|P'|} \Pr\left[\{a_i, b_i\} \not\subseteq u \,\middle|\, \bigwedge_{j=1}^{i-1} \{a_j, b_j\} \not\subseteq u\right].$$

In addition, since $a_1, b_1, \dots, a_{|P'|}, b_{|P'|}$ are distinct, it is not hard to see that $\Pr[\{a_i, b_i\} \not\subseteq u] \geqslant \Pr\left[\{a_i, b_i\} \not\subseteq u \middle| \bigwedge_{j=1}^{i-1} \{a_j, b_j\} \not\subseteq u\right]$. Hence, we have

$$\begin{aligned}\Pr\left[\bigwedge_{i=1}^{|P'|} \{a_i, b_i\} \not\subseteq u\right] = \prod_{i=1}^{|P'|} \Pr[\{a_i, b_i\} \not\subseteq u] \leqslant \left(\exp\left(-\frac{r^2}{2|U|^2}\right)\right)^{|P'|} &= \exp\left(-\frac{|P'|r^2}{2|U|^2}\right)\\ &\leqslant \exp\left(-\frac{|P|r^2}{4q|U|^2}\right),\end{aligned}$$

completing the proof of Proposition 4.6. □

4.1.2 Proofs of Inapproximability Results of DkS

In this subsection, we prove Theorem 4.1 and 4.2. The proof of Theorem 4.1 is simply by combining Dinur's PCP Theorem and Theorem 4.4 with $\ell = m/\text{polylog} m$, as stated below.

Proof of Theorem 4.1. For any 3SAT formula φ with m clauses, use Theorem 2.2 to produce ϕ with $m' = O(m\text{polylog}m)$ clauses such that each variable appears in at most d clauses. Let ζ be a constant such that $m' = O(m \log^{\zeta} m)$ and let $\ell = m/\log^2 m$. Let us consider the graph $G_{\phi,\ell}$ with $k = \binom{n}{\ell}$ where n is the number of variables of ϕ. Let N be the number of vertices of $G_{\phi,\ell}$. Observe that $N = 2^{\ell}\binom{n}{\ell} \leqslant n^{2\ell} \leqslant (m')^{O(\ell)} = 2^{O(\ell \log m')} = 2^{o(m)}$.

If φ is satisfiable, ϕ is also satisfiable and it is obvious that $G_{\phi,\ell}$ contains an induced k-clique. Otherwise, If φ is unsatisfiable, $\text{val}(\phi) \leqslant 1 - \varepsilon$. From Theorem 4.4, any k-subgraph of $G_{\phi,\ell}$ has density at most $2^{-\Omega(\ell^4/n^3)} \leqslant 2^{-\Omega(m/\log^{3\zeta+8} m)} = N^{-\Omega(1/(\log\log N)^{3\zeta+8})}$, which is at most $N^{-1/(\log\log N)^{3\zeta+9}}$ when m is sufficiently large. Hence, if there is a polynomial-time algorithm that can distinguish between the two cases in Theorem 4.1 when $c = 3\zeta + 9$, then there also exists an algorithm that solves 3SAT in time $2^{o(m)}$, contradicting with ETH. □

The proof of Theorem 4.2 is even simpler since, under Gap-ETH, we have the gap version of 3SAT to begin with. Hence, we can directly apply Theorem 4.4 without going through Dinur's PCP:

Proof of Theorem 4.2. Let ϕ be any 3SAT formula with m clauses such that each variable appears in $O(1)$ clauses. For any constant $1 > \gamma, \nu > 0$, let $\ell = m\left(\gamma^{1/3} \cdot \log^{1/3+\nu/2}(1/\gamma)\right)$ and consider the graph $G_{\phi,\ell}$ with $k = \binom{n}{\ell}$ where n is the number of variables of ϕ. The number of vertices N of $G_{\phi,\ell}$ is $2^\ell\binom{n}{\ell} \leqslant 2^\ell\left(\frac{en}{\ell}\right)^\ell \leqslant 2^{O(\ell\log(m/\ell))} = 2^{O\left(m\left(\gamma^{1/3}\cdot\log^{4/3+\nu/2}(1/\gamma)\right)\right)}$.

The completeness is again obvious. For the soundness, if $\mathrm{val}(\phi) \leqslant 1-\varepsilon$, from Theorem 4.4, any k-subgraph of $G_{\phi,\ell}$ has density at most $2^{-\Omega(\ell^4/n^3)} \leqslant 2^{-\Omega(m\gamma^{4/3}\log^{4/3+2\nu}(1/\gamma))} \leqslant N^{-\Omega(\gamma\cdot\log^{3\nu/2}(1/\gamma))}$, which is at most $N^{-\gamma}$ when γ is sufficiently small. Now suppose contrapositively that, for some $\nu > 0$, there is an algorithm that, given a graph G and γ, runs in time $N^{O\left(\frac{1}{\gamma^{1/3}\log^{4/3+\nu}(\gamma)}\right)}$ and can distinguish between the two cases in Theorem 4.2. By running this algorithm on $G_{\phi,\ell}$, we can solve Gap-3SAT in time $2^{O\left(\frac{m}{\log^{\nu/2}(1/\gamma)}\right)}$; by picking a sufficiently small γ, this contradicts with Gap-ETH. □

4.2 Discussion and Open Questions

In this chapter, we provide a subexponential time reduction from the gap version of 3SAT to DkS and prove that it establishes an almost-polynomial ratio hardness of approximation of the latter under ETH and Gap-ETH. Even with our results, however, approximability of DkS still remains wide open. Namely, it is still not known whether it is NP-hard to approximate DkS to within some constant factor, and, no polynomial ratio hardness of approximation is yet known.

Although our results appear to almost resolve the second question, it still seems out of reach with our current knowledge of hardness of approximation. In particular, to achieve a polynomial ratio hardness for DkS, it is plausible that one has to prove a long-standing conjecture called *the sliding scale conjecture (SSC)* [Bel+93]. In short, SSC essentially states that LABEL COVER is NP-hard to approximate to within some polynomial ratio. Note here that polynomial ratio hardness for LABEL COVER is not even known under stronger assumptions such as ETH or Gap-ETH; we refer the readers to [Din16] and Chapter 9 for more detailed discussions on the topic.

There is in fact an approximation preserving reduction from DkS to LABEL COVER [CHK11] but it does not provide perfect completeness, which is required in SSC; this leaves a possibility that a polynomial ratio hardness of approximation of DkS can be achieved without resolving SSC.

Apart from the approximability of DkS, our results also prompt the following natural question: since previous techniques, such as Feige's Random 3SAT Hypothesis [Fei02], Khot's Quasi-Random PCP [Kho06], Unique Games with Small Set Expansion Conjecture [RS10] and the Planted Clique Hypothesis [Jer92; Kuč95], that were successful in showing inapproximability of DkS also gave rise to hardnesses of approximation of many problems that are not known to be APX-hard including SPARSEST CUT, MIN BISECTION, BALANCED SEPARATOR, MINIMUM LINEAR ARRANGEMENT and 2-CATALOG SEGMENTATION [AMS07; Sak10; RST12], is it possible to modify our construction to prove inapproximability for these problems as well? An evidence suggesting that this may be possible is the case of ε-approximate Nash Equilibrium with ε-optimal welfare, which was first proved to be hard under the Planted Clique Hypothesis by Hazan and

Krauthgamer [HK11] before Braverman, Ko and Weinstein proved that the problem was also hard under ETH using the birthday repetition framework [BKW15].

Chapter 5

VC Dimension and Littlestone's Dimension

A common and essential assumption in learning theory is that the concepts we want to learn come from a nice, simple concept class, or (in the agnostic case) they can at least be approximated by a concept from a simple class. When the concept class is sufficiently simple, there is hope for good (i.e. sample-efficient and low-error) learning algorithms.

There are many different ways to measure the *simplicity* of a concept class. The most influential measure of simplicity is the VC Dimension, which captures learning in the PAC model. We also consider Littlestone's Dimension [Lit88], which corresponds to minimizing mistakes in online learning (see Section 5.2 for definitions). When either dimension is small, there are algorithms that exploit the simplicity of the class, to obtain good learning guarantees.

We consider the most optimistic setting where the entire universe and concept class are given as explicit input (a binary matrix whose (x, C)-th entry is 1 iff element x belongs to concept C). In this setting, both VC Dimension and Littlestone's Dimension can be computed (exactly[1]) in $n^{O(\log n)}$ time[2]. Hence, similar to problems considered earlier in this thesis, the problem of computing these dimensions are unlikely to be NP-hard.

Nonetheless, two decades ago, it was shown (under appropriate computational complexity assumptions) that neither dimension can be computed in polynomial time [PY96; FL98]. Under ETH, their reduction also yields a tight running time lower bound of $n^{\Omega(\log n)}$ to compute the two dimensions. Such computational intractability of computing the (VC, Littlestone's) dimension of a concept class suggests that even in cases where a simple structure exists, it may be inaccessible to computationally bounded algorithms (see Discussion below).

We prove a (almost) tight running time lower bound, similar to those implied by [PY96; FL98], that hold even against approximation algorithms, as stated below.

Theorem 5.1 (Hardness of Approximating VC Dimension)**.** *Assuming Randomized ETH, approximating VC Dimension to within a $(1/2 + o(1))$-factor requires $n^{\log^{1-o(1)} n}$ time.*

[1] As discussed in Section 1.1.3, this is unlike dense CSPs and DkS with perfect completeness, for which the exact versions are NP-hard but QPTASs exist.

[2] For VC Dimension, this is because the dimension itself is at most $\log n$ and hence we can simply enumerate all sets $S \subseteq \mathcal{U}$ or size at most $\log n$ and check whether it is shattered. For Littlestone's Dimension, there is a simple divide-and-conquer algorithm for it (see Section 5.4.4).

Theorem 5.2 (Hardness of Approximating Littlestone's Dimension)**.** *There exists an absolute constant $\varepsilon > 0$ such that, assuming Randomized ETH, approximating Littlestone's Dimension to within a $(1-\varepsilon)$-factor requires $n^{\log^{1-o(1)} n}$ time.*

5.1 Interpretation of the Results

As we mentioned before, the computational intractability of computing the (VC, Littlestone's) dimension of a concept class suggests that even in cases where a simple structure exists, it may be inaccessible to computationally bounded algorithms. We note however that it is not at all clear that any particular algorithmic applications are immediately intractable as a consequence of our results.

Consider for example the adversarial online learning zero-sum game corresponding to Littlestone's Dimension: At each iteration, Nature presents the learner with an element from the universe; the learner attempts to classify the element, and loses a point for every wrong classification; at the end of the iteration, the correct (binary) classification is revealed. The Littlestone's Dimension is equal to the worst case loss of the Learner before learning the exact concept. (see Section 5.2 for a more detailed definition.)

What can we learn from the fact that the Littlestone's Dimension is hard to compute? The first observation is that there is no efficient learner that can commit to a concrete mistake bound. But this does not rule out a computationally-efficient learner that plays optimal strategy and makes at most as many mistakes as the unbounded learner. We can, however, conclude that Nature's task is computationally intractable! Otherwise, we could efficiently construct an entire worst-case mistake tree (for a concept class $\mathcal{C}$, any mistake tree has at most $|\mathcal{C}|$ leaves, requiring $|\mathcal{C}|-1$ oracle calls to Nature).

On a philosophical level, we think it is interesting to understand the implications of an intractable, adversarial Nature. Perhaps this is another evidence that the mistake bound model is too pessimistic?

Also, the only algorithm we know for computing the optimal learner's decision requires computing the Littlestone's Dimension. We think that it is an interesting open question whether an approximately optimal computationally-efficient learner exists.

In addition, let us note that in the other direction, computing Littlestone's Dimension exactly implies an exactly optimal learner. However, since the learner has to compute Littlestone's Dimension many times, we have no evidence that an approximation algorithm for Littlestone's Dimension would imply any guarantee for the learner.

Finally, we remark that for either problem (VC or Littlestone's Dimension), we are not aware of any non-trivial approximation algorithms.

5.1.1 Techniques

As with our previous two chapters, we once again follow the "birthday repetition" framework. However, there are multiple unique challenges we have to overcome for both VC Dimension and Littlestone's Dimension, as described below.

VC Dimension The first challenge we have to overcome in order to adapt this framework to hardness of approximation of VC Dimension is that the number of concepts involved in shattering a subset S is $2^{|S|}$. Therefore any inapproximability factor we prove on the size of the shattered set of elements, "goes in the exponent" of the size of the shattering set of concepts. Even a small constant factor gap in the VC Dimension requires proving a polynomial factor gap in the number of shattering concepts (obtaining polynomial gaps via "birthday repetition" for simpler problems is an interesting open problem [MR17a; Man17a]). Fortunately, having a large number of concepts is also an advantage: we use each concept to test a different set of label cover constraints chosen independently at random; if the original instance is far from satisfied, the probability of passing all $2^{\Theta(|S|)}$ tests should now be doubly-exponentially small ($2^{-2^{\Theta(|S|)}}$)! More concretely, we think of half of the elements in the shattered set as encoding an assignment, and the other half as encoding which tests to run on the assignments.

Littlestone's Dimension Our starting point is the reduction for VC Dimension outlined in the previous paragraph. While we haven't yet formally introduced Littlestone's Dimension, recall that it corresponds to an online learning model. If the test-selection elements arrive before the assignment-encoding elements, the adversary can adaptively tailor his assignment to pass the specific test selected in the previous steps. To overcome this obstacle, we introduce a special gadget that forces the assignment-encoding elements to arrive first; this makes the reduction to Littlestone's Dimension somewhat more involved. Note that there is a reduction by [FL98] from VC Dimension to Littlestone's Dimension. Unfortunately, their reduction is not (approximately) gap-preserving, so we cannot use it directly to obtain Theorem 5.2 from Theorem 5.1.

5.1.2 Related Work

The study of the computational complexity of the VC Dimension was initiated by Linial, Mansour, and Rivest [LMR91], who observed that it can be computed in quasi-polynomial time. [PY96] proved that it is complete for the class LOGNP which they define in the same paper. [FL98] reduced the problem of computing the VC dimension to that of computing Littlestone's Dimension, hence the latter is also LOGNP-hard. (It follows as a corollary of our Theorem 5.1 that, assuming ETH, solving any LOGNP-hard problem requires quasi-polynomial time.)

Both problems were also studied in an implicit model, where the concept class is given in the form of a Boolean circuit that takes as input an element x and a concept c and returns 1 iff $x \in c$. Observe that in this model even computing whether either dimension is 0 or not is already NP-hard. Schafer proved that the VC Dimension is Σ_3^{P}-complete [Sch99], while the Littlestone's Dimension is PSPACE-complete [Sch00]. [MU02] proved that VC Dimension is Σ_3^{P}-hard to approximate to within a factor of almost 2; can be approximated to within a factor slightly better than 2 in AM; and is AM-hard to approximate to within $n^{1-\varepsilon}$.

Another line of related work in the implicit model proves computational intractability of PAC learning (which corresponds to the VC Dimension). Such intractability has been proved either from cryptographic assumptions, e.g. [KV94; Kha93; Kha95; Fel+06; Kal+08; KS09; Kli16] or from average case assumptions, e.g. [DS16; Dan16]. [Blu94] showed a "computational" separation between PAC learning and online mistake bound (which correspond to the VC Dimension and Littlestone's Dimension, respectively): if one-way function exist, then there is a concept class that can be learned by a computationally-bounded learner in the PAC model, but not in the mistake-bound model.

Recently, [BFS16] introduced a generalization of VC Dimension which they call Partial VC Dimension, and proved that it is NP-hard to approximate (even when given an explicit description of the universe and concept class).

5.2 Additional Notations and Preliminaries

For a universe (or ground set) $\mathcal{U}$, a concept C is simply a subset of $\mathcal{U}$ and a concept class $\mathcal{C}$ is a collection of concepts. For convenience, we sometimes relax the definition and allow the concepts to not be subsets of $\mathcal{U}$; all definitions here extend naturally to this case.

The VC and Littlestone's Dimensions can be defined as follows.

Definition 5.3 (VC Dimension [VC71]). *A subset $S \subseteq \mathcal{U}$ is said to be* shattered *by a concept class $\mathcal{C}$ if, for every $T \subseteq S$, there exists a concept $C \in \mathcal{C}$ such that $T = S \cap C$.*

The VC Dimension $\text{VC-dim}(\mathcal{C}, \mathcal{U})$ *of a concept class $\mathcal{C}$ with respect to the universe $\mathcal{U}$ is the largest d such that there exists a subset $S \subseteq \mathcal{U}$ of size d that is shattered by $\mathcal{C}$.*

Definition 5.4 (Mistake Tree and Littlestone's Dimension [Lit88]). *A depth-d instance-labeled tree of $\mathcal{U}$ is a full binary tree of depth d such that every internal node of the tree is assigned an element of $\mathcal{U}$. For convenience, we will identify each node in the tree canonically by a binary string s of length at most d.*

A depth-d mistake tree (aka shattered tree [BPS09]) for a universe $\mathcal{U}$ and a concept class $\mathcal{C}$ is a depth-d instance-labeled tree of $\mathcal{U}$ such that, if we let $v_s \in \mathcal{U}$ denote the element assigned to the vertex s for every $s \in \{0,1\}^{<d}$, then, for every leaf $\ell \in \{0,1\}^d$, there exists a concept $C \in \mathcal{C}$ that agrees with the path from root to it, i.e., that, for every $i < d$, $v_{\ell_{\leqslant i}} \in C$ iff $\ell_{i+1} = 1$ where $\ell_{\leqslant i}$ denote the prefix of ℓ of length i.

The Littlestone's Dimension $\text{L-dim}(\mathcal{C}, \mathcal{U})$ *of a concept class $\mathcal{C}$ with respect to the universe $\mathcal{U}$ is defined as the maximum d such that there exists a depth-d mistake tree for $\mathcal{U}, \mathcal{C}$.*

An equivalent formulation of Littlestone's Dimension is through mistakes made in online learning, as stated below. This interpretation will be useful in our proof.

Definition 5.5 (Mistake Bound). *An online algorithm $\mathcal{A}$ is an algorithm that, at time step i, is given an element $x_i \in \mathcal{U}$ and the algorithm outputs a prediction $p_i \in \{0,1\}$ whether x is in the class. After the prediction, the algorithm is told the correct answer $h_i \in \{0,1\}$. For a sequence $(x_1, h_1), \dots, (x_n, h_n)$,* prediction mistake *of $\mathcal{A}$ is defined as the number of incorect predictions, i.e., $\sum_{i \in n} \mathbb{1}[p_i \neq h_i]$. The* mistake bound *of $\mathcal{A}$ for a concept class $\mathcal{C}$ is defined as the maximum prediction mistake of $\mathcal{A}$ over all the sequences $(x_1, h_1), \dots, (x_n, h_n)$ which corresponds to a concept $C \in \mathcal{C}$ (i.e. $h_i = \mathbb{1}[x_i \in C]$ for all $i \in [n]$).*

Theorem 5.6 ([Lit88]). *For any universe $\mathcal{U}$ and any concept class $\mathcal{C}$,* $\text{L-dim}(\mathcal{C},\mathcal{U})$ *is equal to the minimum mistake bound of $\mathcal{C},\mathcal{U}$ over all online algorithms.*

The following facts are well-know and follow easily from the above definitions.

Fact 5.7. *For any universe $\mathcal{U}$ and concept class $\mathcal{C}$, we have*

$$\text{VC-dim}(\mathcal{C},\mathcal{U}) \leqslant \text{L-dim}(\mathcal{C},\mathcal{U}) \leqslant \log|\mathcal{C}|.$$

Fact 5.8. *For any two universes $\mathcal{U}_1,\mathcal{U}_2$ and any concept class $\mathcal{C}$,*

$$\text{L-dim}(\mathcal{C},\mathcal{U}_1 \cup \mathcal{U}_2) \leqslant \text{L-dim}(\mathcal{C},\mathcal{U}_1) + \text{L-dim}(\mathcal{C},\mathcal{U}_2).$$

5.2.1 Useful Lemmata

We end this section by listing a couple of lemmata that will be useful in our proofs.

Lemma 5.9 (Chernoff Bound). *Let $X_1,\dots,X_n$ be i.i.d. random variables taking value from $\{0,1\}$ and let p be the probability that $X_i = 1$, then, for any $\delta > 0$, we have*

$$\Pr\left[\sum_{i=1}^{n} X_i \geqslant (1+\delta)np\right] \leqslant \begin{cases} 2^{-\delta^2 np/3} & \text{if } \delta < 1, \\ 2^{-\delta np/3} & \text{otherwise.} \end{cases}$$

Lemma 5.10 (Partitioning Lemma [Rub17a, Lemma 2.5]). *For any bi-regular bipartite graph $G = (A,B,E)$, let $n = |A| + |B|$ and $r = \sqrt{n}/\log n$. When n is sufficiently large, there exists a partition of $A \cup B$ into $U_1,\dots,U_r$ such that*

$$\forall i \in [r], \frac{n}{2r} \leqslant |U_i| \leqslant \frac{2n}{r}$$

and

$$\forall i,j \in [r], \frac{|E|}{2r^2} \leqslant |(U_i \times U_j) \cap E|, |(U_j \times U_i) \cap E| \leqslant \frac{2|E|}{r^2}.$$

Moreover, such partition can be found in randomized linear time (alternatively, deterministic $n^{O(\log n)}$ time).

5.3 VC Dimension

In this section, we present our reduction from Label Cover to VC Dimension, stated more formally below. We note that this reduction, together with Moshkovitz-Raz PCP (Theorem 2.4), with parameter $\delta = 1/\log n$ gives a reduction from 3SAT on n variables to VC Dimension of size $2^{n^{1/2+o(1)}}$ with gap $1/2 + o(1)$, which immediately implies Theorem 5.1.

Theorem 5.11. *For every $\delta > 0$, there exists a randomized reduction from a bi-regular Label Cover instance $\mathcal{L} = (A, B, E, \Sigma, \{\pi_e\}_{e \in E})$ such that $|\Sigma| = O_\delta(1)$ to a ground set $\mathcal{U}$ and a concept class $\mathcal{C}$ such that, if $n := |A| + |B|$ and $r := \sqrt{n}/\log n$, then the following conditions hold for every sufficiently large n.*

- *(Size) The reduction runs in time $|\Sigma|^{O(|E| poly(1/\delta)/r)}$ and $|\mathcal{C}|, |\mathcal{U}| \leqslant |\Sigma|^{O(|E| poly(1/\delta)/r)}$.*
- *(Completeness) If $\mathcal{L}$ is satisfiable, then* $\text{VC-dim}(\mathcal{C},\mathcal{U}) \geqslant 2r$.
- *(Soundness) If* $\text{val}(\mathcal{L}) \leqslant \delta^2/100$*, then* $\text{VC-dim}(\mathcal{C},\mathcal{U}) \leqslant (1 + \delta)r$ *with high probability.*

In fact, the above properties hold with high probability even when δ and $|\Sigma|$ are not constants, as long as $\delta \geqslant \log(1000n \log |\Sigma|)/r$.

We remark here that when $\delta = 1/\log n$, Moshkovitz-Raz PCP produces a Label Cover instance with $|A| = n^{1+o(1)}, |B| = n^{1+o(1)}$ and $|\Sigma| = 2^{\text{polylog}(n)}$. For such parameters, the condition $\delta \geqslant \log(1000n \log |\Sigma|)/r$ holds for every sufficiently large n.

5.3.1 A Candidate Reduction (and Why It Fails)

To best understand the intuition behind our reduction, we first describe a simpler candidate reduction and explain why it fails, which will lead us to the eventual construction. In this candidate reduction, we start by evoking Lemma 5.10 to partition the vertices $A \cup B$ of the Label Cover instance $\mathcal{L} = (A, B, E, \Sigma, \{\pi_e\}_{e \in E})$ into $U_1, \dots, U_r$ where $r = \sqrt{n}/\log n$. We then create the universe $\mathcal{U}$ and the concept class $\mathcal{C}$ as follows:

- We make each element in $\mathcal{U}$ correspond to a partial assignment to U_i for some $i \in [r]$, i.e., we let $\mathcal{U} = \{x_{i,\sigma_i} \mid i \in [r], \sigma_i \in \Sigma^{U_i}\}$. In the completeness case, we expect to shatter the set of size r that corresponds to a satisfying assignment $\sigma^* \in \Sigma^{A \cup B}$ of the Label Cover instance $\mathcal{L}$, i.e., $\{x_{i,\sigma^*|_{U_i}} \mid i \in [r]\}$. As for the soundness, our hope is that, if a large set $S \subseteq \mathcal{U}$ gets shattered, then we will be able to decode an assignment for $\mathcal{L}$ that satisfies many constraints, which contradicts with our assumption that $\text{val}(\mathcal{L})$ is small. Note that the number of elements of $\mathcal{U}$ in this candidate reduction is at most $r \cdot |\Sigma|^{O(|E| \text{poly}(1/\delta) r)} = 2^{\widetilde{O}(\sqrt{n})}$ as desired.
- As stated above, the intended solution for the completeness case is $\{x_{i,\sigma^*|_{U_i}} \mid i \in [r]\}$, meaning that we must have at least one concept corresponding to each subset $I \subseteq [r]$. We will try to make our concepts "test" the assignment; for each $I \subseteq [r]$, we will choose a set $T_I \subseteq A \cup B$ of $\widetilde{O}(\sqrt{n})$ vertices and "test" all the constraints within T_I. Before we specify how T_I is picked, let us elaborate what "test" means: for each T_I-partial assignment ϕ_I that does not violate any constraints within T_I, we create a concept C_{I,ϕ_I}. This concept contains x_{i,σ_i} if and only if $i \in I$ and σ_i agrees with ϕ_I (i.e. $\phi_I|_{T_I \cap U_i} = \sigma_i|_{T_I \cap U_i}$). Recall that, if a set $S \subseteq \mathcal{U}$ is shattered, then each $\widetilde{S} \subseteq S$ is an intersection between S and C_{I,ϕ_I} for some

I, ϕ_I. We hope that the I's are different for different $\widetilde{S}$ so that many different tests have been performed on S.

Finally, let us specify how we pick T_I. Assume without loss of generality that r is even. We randomly pick a perfect matching between r, i.e., we pick a random permutation $\pi_I : [r] \to [r]$ and let $\Big(\pi_I(1), \pi_I(2)\Big), \dots, \Big(\pi_I(r-1), \pi_I(r)\Big)$ be the chosen matching. We pick T_I such that all the constraints in the matchings, i.e., constraints between $U_{\pi_I(2i-1)}$ and $U_{\pi_I(2i)}$ for every $i \in [r/2]$, are included. More specifically, for every $i \in [r]$, we include each vertex $v \in U_{\pi_I(2i-1)}$ if at least one of its neighbors lie in $U_{\pi_I(2i)}$ and we include each vertex $u \in U_{\pi_I(2i)}$ if at least one of its neighbors lie in $U_{\pi_I(2i-1)}$. By Lemma 5.10, for every pair in the matching the size of the intersection is at most $\frac{2|E|}{r^2}$, so each concept contains assignments to at most $\frac{2|E|}{r}$ variables; so the total size of the concept class is at most $2^r \cdot |\Sigma|^{\frac{2|E|}{r}}$.

Even though the above reduction has the desired size and completeness, it unfortunately fails in the soundness. Let us now sketch a counterexample. For simplicity, let us assume that each vertex in $T_{[r]}$ has a unique neighbor in $T_{[r]}$. Note that, since $T_{[r]}$ has quite small size (only $\widetilde{O}(\sqrt{n})$), almost all the vertices in $T_{[r]}$ satisfy this property w.h.p., but assuming that all of them satisfy this property makes our life easier.

Pick an assignment $\widetilde{\sigma} \in \Sigma^V$ such that none of the constraints in $T_{[r]}$ is violated. From our unique neighbor assumption, there is always such an assignment. Now, we claim that the set $S_{\widetilde{\sigma}} := \{x_{i,\widetilde{\sigma}|_{U_i}} \mid i \in [r]\}$ gets shattered. This is because, for every subset $I \subseteq [r]$, we can pick another assignment σ' such that σ' does not violate any constraint in $T_{[r]}$ and $\sigma'|_{U_i} = \widetilde{\sigma}|_{U_i}$ if and only if $i \in I$. This implies that $\{x_{i,\widetilde{\sigma}|_{U_i}} \mid i \in I\} = S \cap C_{[r],\sigma'}$ as desired. Note here that such σ' exists because, for every $i \notin I$, if there is a constraint from a vertex $a \in U_i \cap A$ to another vertex $b \in T_{[r]} \cap B$, then we can change the assignment to a in such a way that the constraint is not violated[3]; by doing this for every $i \notin I$, we have created the desired σ'. As a result, VC-dim$(\mathcal{C},\mathcal{U})$ can still be as large as r even when the value of $\mathcal{L}$ is small.

5.3.2 The Final Reduction

In this subsection, we will describe the actual reduction. To do so, let us first take a closer look at the issue with the above candidate reduction. In the candidate reduction, we can view each $I \subseteq [r]$ as being a seed used to pick a matching. Our hope was that many seeds participate in shattering some set S, and that this means that S corresponds to an assignment of high value. However, the counterexample showed that in fact only one seed ($I = [r]$) is enough to shatter a set. To circumvent this issue, we will not use the subset I as our seed anymore. Instead, we create r new elements $y_1, \dots, y_r$, which we will call *test selection elements* to act as seeds; namely, each subset $H \subseteq \mathcal{Y}$ will now be a seed. The benefit of this is that, if $S \subseteq \mathcal{Y}$ is shattered and contains test selection elements $y_{i_1}, \dots, y_{i_t}$, then at least 2^t seeds must participate in the shattering of S. This

[3] Here we assume that $|\pi^{-1}_{(a,b)}(\widetilde{\sigma}(b))| > 1$; note that this always holds for Label Cover instances produced by Moshkovitz-Raz construction.

is because, for each $H \subseteq \mathcal{Y}$, the intersection of S with any concept corresponding to H, when restricted to $\mathcal{Y}$, is always $H \cap \{y_{i_1}, \dots, y_{i_t}\}$. Hence, each subset of $\{y_{i_1}, \dots, y_{i_t}\}$ must come a from different seed.

The only other change from the candidate reduction is that each H will test multiple matchings rather than one matching. This is due to a technical reason: we need the number of matchings, ℓ, to be large in order get the approximation ratio down to $1/2 + o(1)$; in our proof, if $\ell = 1$, then we can only achieve a factor of $1 - \varepsilon$ to some $\varepsilon > 0$. The full details of the reduction are shown in Figure 5.1.

Before we proceed to the proof, let us define some additional notation that will be used throughout.

- Every assignment element of the form x_{i,σ_i} is called an *i-assignment element*; we denote the set of all i-assignment elements by X_i, i.e., $\mathrm{X}_i = \{x_{i,\sigma_i} \mid \sigma_i \in \Sigma^{U_i}\}$. Let X denote all the assignment elements, i.e., $\mathrm{X} = \bigcup_i \mathrm{X}_i$.

- For every $S \subseteq \mathcal{U}$, let $I(S)$ denote the set of all $i \in [r]$ such that S contains an i-assignment element, i.e., $I(S) = \{i \in [r] \mid S \cap \mathrm{X}_i \neq \emptyset\}$.

- We call a set $S \subseteq \mathrm{X}$ *non-repetitive* if, for each $i \in [r]$, S contains at most one i-assignment element, i.e., $|S \cap \mathrm{X}_i| \leqslant 1$. Each non-repetitive set S canonically induces a partial assignment $\phi(S) : \bigcup_{i \in I(S)} U_i \to \Sigma$. This is the unique partial assignment that satisfies $\phi(S)|_{U_i} = \sigma_i$ for every $x_{i,\sigma_i} \in S$

- Even though we define each concept as C_{I,H,σ_H} where σ_H is a partial assignment to a subset $T_H \subseteq A \cup B$, it will be more convenient to view each concept as $C_{I,H,\sigma}$ where $\sigma \in \Sigma^V$ is the assignment to the entire Label Cover instance. This is just a notational change: the actual definition of the concept does not depend on the assignment outside T_H.

- For each $I \subseteq [r]$, let U_I denote $\bigcup_{i \in I} U_i$. For each $\sigma_I \in \Sigma^{U_I}$, we say that (I, σ_I) *passes* $H \subseteq \mathcal{Y}$ if σ_I does not violate any constraint within T_H. Denote the collection of H's that (I, σ_I) passes by $\mathcal{H}(I, \sigma_I)$.

- Finally, for any non-repetitive set $S \subseteq \mathrm{X}$ and any $H \subseteq \mathcal{Y}$, we say that S *passes* H if $(I(S), \phi(S))$ passes H. We write $\mathcal{H}(S)$ as a shorthand for $\mathcal{H}(I(S), \phi(S))$.

The output size of the reduction and the completeness follow almost immediately from definition.

Output Size of the Reduction. Clearly, the size of $\mathcal{U}$ is $\sum_{i \in [r]} |\Sigma|^{|U_i|} \leqslant r \cdot |\Sigma|^{n/r} \leqslant |\Sigma|^{O(|E| \text{poly}(1/\delta)/r)}$. As for $|\mathcal{C}|$, note first that the number of choices for I and H are both 2^r. For fixed I and H, Lemma 5.10 implies that, for each matching $\pi_H^{(t)}$, the number of vertices from each U_i with at least one constraint to the matched partition in $\pi_H^{(t)}$ is at most $O(|E|/r^2)$. Since there are ℓ matchings, the number of vertices in $T_H = \mathcal{N}_1(M_H(1)) \cup \cdots \cup \mathcal{N}_r(M_H(r))$ is at most $O(|E|\ell/r)$. Hence, the number of choices for the partial assignment σ_H is at most $|\Sigma|^{O(|E| \text{poly}(1/\delta)/r)}$. In total, we can conclude that $\mathcal{C}$ contains at most $|\Sigma|^{O(|E| \text{poly}(1/\delta)/r)}$ concepts.

Input: A bi-regular Label Cover instance $\mathcal{L} = (A, B, E, \Sigma, \{\pi_e\}_{e \in E})$ and a parameter $\delta > 0$.
Output: A ground set $\mathcal{U}$ and a concept class $\mathcal{C}$.
The procedure to generate $(\mathcal{U}, \mathcal{C})$ works as follows:

- Let r be $\sqrt{n}/\log n$ where $n = |A| + |B|$. Use Lemma 5.10 to partition $A \cup B$ into r blocks $U_1, \ldots, U_r$.
- For convenience, we assume that r is even. Moreover, for $i \neq j \in [r]$, let $\mathcal{N}_i(j) \subseteq U_i$ denote the set of all vertices in U_i with at least one neighbor in U_j (w.r.t. the graph (A, B, E)). We also extend this notation naturally to a set of j's; for $J \subseteq [r]$, $\mathcal{N}_i(J)$ denotes $\bigcup_{j \in J} \mathcal{N}_i(j)$.
- The universe $\mathcal{U}$ consists of two types of elements, as described below.
  - *Assignment elements*: for every $i \in [r]$ and every partial assignment $\sigma_i \in \Sigma^{U_i}$, there is an assignment element x_{i,σ_i} corresponding to it. Let X denote all the assignment elements, i.e., $\mathrm{X} = \{x_{i,\sigma_i} \mid i \in [r], \sigma_i \in \Sigma^{U_i}\}$.
  - *Test selection elements*: there are r test selection elements, which we will call $y_1, \ldots, y_r$. Let $\mathcal{Y}$ denote the set of all test selection elements.
- The concepts in $\mathcal{C}$ are defined by the following procedure.
  - Let $\ell := 80/\delta^3$ be the number of matchings to be tested.
  - For each $H \subseteq \mathcal{Y}$, we randomly select ℓ permutations $\pi_H^{(1)}, \ldots, \pi_H^{(\ell)} : [r] \to [r]$; this gives us ℓ matchings (i.e. the t-th matching is $\left(\pi_H^{(t)}(1), \pi_H^{(t)}(2)\right), \ldots, \left(\pi_H^{(t)}(r-1), \pi_H^{(t)}(r)\right)$). For brevity, let us denote the set of (up to ℓ) elements that i is matched with in the matchings by $M_H(i)$. Let $T_H = \bigcup_i \mathcal{N}_i(M_H(i))$
  - For every $I \subseteq [r], H \subseteq \mathcal{Y}$ and for every partial assignment $\sigma_H \in \Sigma^{T_H}$ that does not violate any constraints, we create a concept C_{I,H,σ_H} such that each $x_{i,\sigma_i} \in \mathrm{X}$ is included in C_{I,H,σ_H} if and only if $i \in I$ and σ_i is consistent with σ_H, i.e., $\sigma_i|_{\mathcal{N}_i(M_H(i))} = \sigma_H|_{\mathcal{N}_i(M_H(i))}$ whereas $y_i \in \mathcal{Y}$ in included in C_{I,H,σ_H} if and only if $y \in H$.

Figure 5.1: Reduction from Label Cover to VC Dimension

Completeness. If $\mathcal{L}$ has a satisfying assignment $\sigma^* \in \Sigma^V$, then the set $S_{\sigma^*} = \{x_{i,\sigma^*|_{U_i}} \mid i \in [r]\} \cup \mathcal{Y}$ is shattered because, for any $S \subseteq S_{\sigma^*}$, we have $S = S_{\sigma^*} \cap C_{I(S), S \cap \mathcal{Y}, \sigma^*}$. Hence, VC-$\dim(\mathcal{C}, \mathcal{U}) \geqslant 2r$.

The rest of this section is devoted to the soundness analysis.

5.3.3 Soundness

In this subsection, we will prove the following lemma, which, combined with the completeness and output size arguments above, imply Theorem 5.11.

Lemma 5.12. *Let $(\mathcal{C},\mathcal{U})$ be the output from the reduction in Figure 5.1 on input $\mathcal{L}$. If $\operatorname{val}(\mathcal{L}) \leqslant \delta^2/100$ and $\delta \geqslant \log(1000n\log|\Sigma|)/r$, then $\text{VC-dim}(\mathcal{C},\mathcal{U}) \leqslant (1+\delta)r$ w.h.p.*

At a high level, the proof of Lemma 5.12 has two steps:

1. Given a shattered set $S \subseteq \mathcal{U}$, we extract a maximal non-repetitive set $S^{\text{NO-REP}} \subseteq S$ such that $S^{\text{NO-REP}}$ passes many ($\geqslant 2^{|S|-|S^{\text{NO-REP}}|}$) H's. If $|S^{\text{NO-REP}}|$ is small, the trivial upper bound of 2^r on the number of different H's implies that $|S|$ is also small. As a result, we are left to deal with the case that $|S^{\text{NO-REP}}|$ is large.

2. When $|S^{\text{NO-REP}}|$ is large, $S^{\text{NO-REP}}$ induces a partial assignment on a large fraction of vertices of $\mathcal{L}$. Since we assume that $\operatorname{val}(\mathcal{L})$ is small, this partial assignment must violate many constraints. We will use this fact to argue that, with high probability, $S^{\text{NO-REP}}$ only passes very few H's, which implies that $|S|$ must be small.

The two parts of the proof are presented in Subsection 5.3.3 and 5.3.3 respectively. We then combine them in Subsection 5.3.3 to prove Lemma 5.12.

Part I: Finding a Non-Repetitive Set That Passes Many Tests

The goal of this subsection is to prove the following lemma, which allows us to, given a shattered set $S \subseteq \mathcal{U}$, find a non-repetitive set $S^{\text{NO-REP}}$ that passes many H's.

Lemma 5.13. *For any shattered $S \subseteq \mathcal{U}$, there is a non-repetitive set $S^{\text{NO-REP}}$ of size $|I(S)|$ s.t. $|\mathcal{H}(S^{\text{NO-REP}})| \geqslant 2^{|S|-|I(S)|}$.*

We will start by proving the following lemma, which will be a basis for the proof of Lemma 5.13.

Lemma 5.14. *Let $C, C' \in \mathcal{C}$ correspond to the same H (i.e. $C = C_{I,H,\sigma}$ and $C' = C_{I',H,\sigma'}$ for some $H \subseteq \mathcal{Y}, I, I' \subseteq [r], \sigma, \sigma' \in \Sigma^V$).*

For any subset $S \subseteq \mathcal{U}$ and any maximal non-repetitive subset $S^{\text{NO-REP}} \subseteq S$, if $S^{\text{NO-REP}} \subseteq C$ and $S^{\text{NO-REP}} \subseteq C'$, then $S \cap C = S \cap C'$.

The most intuitive interpretation of this lemma is as follows. Recall that if S is shattered, then, for each $\widetilde{S} \subseteq S$, there must be a concept $C_{I_{\widetilde{S}},H_{\widetilde{S}},\sigma_{\widetilde{S}}}$ such that $\widetilde{S} = S \cap C_{I_{\widetilde{S}},H_{\widetilde{S}},\sigma_{\widetilde{S}}}$. The above lemma implies that, for each $\widetilde{S} \supseteq S^{\text{NO-REP}}$, $H_{\widetilde{S}}$ must be different. This means that at least $2^{|S|-|S^{\text{NO-REP}}|}$ different H's must be involved in shattering S. Indeed, this will be the argument we use when we prove Lemma 5.13.

Proof of Lemma 5.14. Let $S, S^{\text{NO-REP}}$ be as in the lemma statement. Suppose for the sake of contradiction that there exists $H \subseteq \mathcal{Y}, I, I' \subseteq [r], \sigma, \sigma' \in \Sigma^V$ such that $S^{\text{NO-REP}} \subseteq C_{I,H,\sigma}, S^{\text{NO-REP}} \subseteq C_{I',H,\sigma'}$ and $S \cap C_{I,H,\sigma} \neq S \cap C_{I',H,\sigma'}$.

First, note that $S \cap C_{I,H,\sigma} \cap \mathcal{Y} = S \cap H \cap \mathcal{Y} = S \cap C_{I',H,\sigma'} \cap \mathcal{Y}$. Since $S \cap C_{I,H,\sigma} \neq S \cap C_{I',H,\sigma'}$, we must have $S \cap C_{I,H,\sigma} \cap \mathrm{X} \neq S \cap C_{I',H,\sigma'} \cap \mathrm{X}$. Assume w.l.o.g. that there exists $x_{i,\sigma_i} \in (S \cap C_{I,H,\sigma}) \setminus (S \cap C_{I',H,\sigma'})$.

Note that $i \in I(S) = I(S^{\text{NO-REP}})$ (where the equality follows from maximality of $S^{\text{NO-REP}}$). Thus there exists $\sigma'_i \in \Sigma^{U_i}$ such that $x_{i,\sigma'_i} \in S^{\text{NO-REP}} \subseteq C_{I,H,\sigma} \cap C_{I',H,\sigma'}$. Since x_{i,σ'_i} is in both $C_{I,H,\sigma}$ and $C_{I',H,\sigma'}$, we have $i \in I \cap I'$ and

$$\sigma|_{\mathcal{N}_i(M_H(i))} = \sigma'_i|_{\mathcal{N}_i(M_H(i))} = \sigma'|_{\mathcal{N}_i(M_H(i))}. \tag{5.1}$$

However, since $x_{i,\sigma_i} \in (S \cap C_{I,H,\sigma}) \setminus (S \cap C_{I',H,\sigma'})$, we have $x_{i,\sigma_i} \in C_{I,H,\sigma} \setminus C_{I',H,\sigma'}$. This implies that

$$\sigma|_{\mathcal{N}_i(M_H(i))} = \sigma_i|_{\mathcal{N}_i(M_H(i))} \neq \sigma'|_{\mathcal{N}_i(M_H(i))},$$

which contradicts to (5.1). □

In addition to the above lemma, we will also need the following observation, which states that, if a non-repetitive $S^{\text{NO-REP}}$ is contained in a concept C_{I,H,σ_H}, then $S^{\text{NO-REP}}$ must pass H. This observation follows definitions.

Observation 5.15. *If a non-repetitive set $S^{\text{NO-REP}}$ is a subset of some concept C_{I,H,σ_H}, then $H \in \mathcal{H}(S^{\text{NO-REP}})$.*

With Lemma 5.14 and Observation 5.15 ready, it is now easy to prove Lemma 5.13.

Proof of Lemma 5.13. Pick $S^{\text{NO-REP}}$ to be any maximal non-repetitive subset of S. Clearly, $|S^{\text{NO-REP}}| = |I(S)|$. To see that $|\mathcal{H}(S^{\text{NO-REP}})| \geqslant 2^{|S|-|I(S)|}$, consider any $\widetilde{S}$ such that $S^{\text{NO-REP}} \subseteq \widetilde{S} \subseteq S$. Since S is shattered, there exists $I_{\widetilde{S}}, H_{\widetilde{S}}, \sigma_{\widetilde{S}}$ such that $S \cap C_{I_{\widetilde{S}},H_{\widetilde{S}},\sigma_{\widetilde{S}}} = \widetilde{S}$. Since $\widetilde{S} \supseteq S^{\text{NO-REP}}$, Observation 5.15 implies that $H_{\widetilde{S}} \in \mathcal{H}(S^{\text{NO-REP}})$. Moreover, from Lemma 5.14, $H_{\widetilde{S}}$ is distinct for every $\widetilde{S}$. As a result, $|\mathcal{H}(S^{\text{NO-REP}})| \geqslant 2^{|S|-|I(S)|}$ as desired. □

Part II: No Large Non-Repetitive Set Passes Many Tests

The goal of this subsection is to show that, if $\mathrm{val}(\mathcal{L})$ is small, then w.h.p. (over the randomness in the construction) every large non-repetitive set passes only few H's. This is formalized as Lemma 5.16 below.

Lemma 5.16. *If $\mathrm{val}(\mathcal{L}) \leqslant \delta^2/100$ and $\delta \geqslant 8/r$, then, with high probability, for every non-repetitive set $S^{\text{NO-REP}}$ of size at least δr, $|\mathcal{H}(S^{\text{NO-REP}})| \leqslant 100n \log |\Sigma|$.*

Note that the mapping $S^{\text{NO-REP}} \mapsto (I(S^{\text{NO-REP}}), \phi(S^{\text{NO-REP}}))$ is a bijection from the collection of all non-repetitive sets to $\{(I, \sigma_I) \mid I \subseteq [r], \sigma_I \in \Sigma^{U_I}\}$. Hence, the above lemma is equivalent to the following.

Lemma 5.17. *If* $\mathrm{val}(\mathcal{L}) \leqslant \delta^2/100$ *and* $\delta \geqslant 8/r$*, then, with high probability, for every* $I \subseteq [r]$ *of size at least* δr *and every* $\sigma_I \in \Sigma^{U_I}$*,* $|\mathcal{H}(I, \sigma_I)| \leqslant 100n \log |\Sigma|$*.*

Here we use the language in Lemma 5.17 instead of Lemma 5.16 as it will be easier for us to reuse this lemma later. To prove the lemma, we first need to bound the probability that each assignment σ_I does not violate any constraint induced by a random matching. More precisely, we will prove the following lemma.

Lemma 5.18. *For any* $I \subseteq [r]$ *of size at least* δr *and any* $\sigma_I \in \Sigma^{U_I}$*, if* $\pi : [r] \to [r]$ *is a random permutation of* $[r]$*, then the probability that* σ_I *does not violate any constraint in* $\bigcup_{i \in [r]} \mathcal{N}_i(M(i))$ *is at most* $(1 - 0.1\delta^2)^{\delta r/8}$ *where* $M(i)$ *denote the index that* i *is matched with in the matching* $\Big(\pi(1), \pi(2)\Big), \dots, \Big(\pi(r-1), \pi(r)\Big)$*.*

Proof. Let p be any positive odd integer such that $p \leqslant \delta r/2$ and let $i_1, \dots, i_{p-1} \in [r]$ be any $p-1$ distinct elements of $[r]$. We will first show that conditioned on $\pi(1) = i_1, \dots, \pi(p-1) = i_{p-1}$, the probability that σ_I violates a constraint induced by $\pi(p), \pi(p+1)$ (i.e. in $\mathcal{N}_{\pi(p)}(\pi(p+1)) \cup \mathcal{N}_{\pi(p+1)}(\pi(p))$) is at least $0.1\delta^2$.

To see that this is true, let $I_{\geqslant p} = I \backslash \{i_1, \dots, i_{p-1}\}$. Since $|I| \geqslant \delta r$, we have $|I_{\geqslant p}| = |I| - p + 1 \geqslant \delta r/2 + 1$. Consider the partial assignment $\sigma_{\geqslant p} = \sigma_I|_{U_{I_{\geqslant p}}}$. Since $\mathrm{val}(\mathcal{L}) \leqslant 0.01\delta^2$, $\sigma_{\geqslant p}$ can satisfy at most $0.01\delta^2|E|$ constraints. From Lemma 5.10, we have, for every $i \neq j \in I_{\geqslant p}$, the number of constraints between U_i and U_j are at least $|E|/r^2$. Hence, there are at most $0.01\delta^2 r^2$ pairs of $i < j \in I_{\geqslant p}$ such that $\sigma_{\geqslant p}$ does not violate any constraint between U_i and U_j. In other words, there are at least $\binom{|I_{\geqslant p}|}{2} - 0.01\delta^2 r^2 \geqslant 0.1\delta^2 r^2$ pairs $i < j \in I_{\geqslant p}$ such that $\sigma_{\geqslant p}$ violates some constraints between U_i and U_j. Now, if $\pi(p) = i$ and $\pi(p+1) = j$ for some such pair i, j, then $\phi(S^{\text{NO-REP}})$ violates a constraint induced by $\pi(p), \pi(p+1)$. Thus, we have

$$\Pr\left[\sigma_I \text{ does not violate a constraint induced by } \pi(p), \pi(p+1) \;\middle|\; \bigwedge_{t=1}^{p-1} \pi(t) = i_t\right] \leqslant 1 - 0.1\delta^2. \tag{5.2}$$

Let E_p denote the event that σ_I does not violate any constraints induced by $\pi(p)$ and $\pi(p+1)$. We can now bound the desired probability as follows.

$$\begin{aligned}
\Pr\left[\sigma_I \text{ does not violate any constraint in } \bigcup_{i \in [r]} \mathcal{N}_i(M(i))\right] &\leqslant \Pr\left[\bigwedge_{\text{odd } p \in [\delta r/2+1]} E_p\right] \\
&= \prod_{\text{odd } p \in [\delta r/2+1]} \Pr\left[E_p \;\middle|\; \bigwedge_{\text{odd } t \in [p-1]} E_t\right] \\
(\text{From (5.2)}) &\leqslant \prod_{\text{odd } p \in [\delta r/2+1]} (1 - 0.1\delta^2) \\
&\leqslant (1 - 0.1\delta^2)^{\delta r/4 - 1},
\end{aligned}$$

which is at most $(1 - 0.1\delta^2)^{\delta r/8}$ since $\delta \geqslant 8/r$. □

We can now prove our main lemma.

Proof of Lemma 5.17. For a fixed $I \subseteq [r]$ of size at least δr and a fixed $\sigma_I \in \Sigma^{U_I}$, Lemma 5.18 tells us that the probability that σ_I does not violate any constraint induced by a single matching is at most $(1-0.1\delta^2)^{\delta r/8}$. Since for each $H \subseteq \mathcal{Y}$ the construction picks ℓ matchings at random, the probability that (I, σ_I) passes each H is at most $(1-0.1\delta^2)^{\delta \ell r/8}$. Recall that we pick $\ell = 80/\delta^3$; this gives the following upper bound on the probability:

$$\Pr[(I,\sigma_I) \text{ passes } H] \leqslant (1-0.1\delta^2)^{\delta\ell r/8} = (1-0.1\delta^2)^{10r/\delta^2} \leqslant \left(\frac{1}{1+0.1\delta^2}\right)^{10r/\delta^2} \leqslant 2^{-r} \quad (5.3)$$

where the last inequality comes from Bernoulli's inequality.

Inequality (5.3) implies that the expected number of H's that (I, σ_I) passes is less than 1. Since the matchings M_H are independent for all H's, we can apply Chernoff bound which implies that

$$\Pr[|\mathcal{H}(I,\sigma_I)| \geqslant 100n \log|\Sigma|] \leqslant 2^{-10n\log|\Sigma|} = |\Sigma|^{-10n}.$$

Finally, note that there are at most $2^r|\Sigma|^n$ different (I, σ_I)'s. By union bound, we have

$$\begin{aligned}\Pr\left[\exists I \subseteq [r], \sigma_I \in \Sigma^{U_I} \text{ s.t. } |I| \geqslant \delta r \text{ AND } |\mathcal{H}(I,\sigma_I)| \geqslant 100n\log|\Sigma|\right] &\leqslant (2^r|\Sigma|^n)\left(|\Sigma|^{-10n}\right) \\ &\leqslant |\Sigma|^{-8n},\end{aligned}$$

which concludes the proof. □

Putting Things Together

Proof of Lemma 5.12. From Lemma 5.16, every non-repetitive set $S^{\text{NO-REP}}$ of size at least δr, $|\mathcal{H}(S^{\text{NO-REP}})| \leqslant 100n\log|\Sigma|$. Conditioned on this event happening, we will show that $\text{VC-dim}(\mathcal{U}, \mathcal{C}) \leqslant (1+\delta)r$.

Consider any shattered set $S \subseteq \mathcal{U}$. Lemma 5.13 implies that there is a non-repetitive set $S^{\text{NO-REP}}$ of size $|I(S)|$ such that $|\mathcal{H}(S^{\text{NO-REP}})| \geqslant 2^{|S|-|I(S)|}$. Let us consider two cases:

1. $|I(S)| \leqslant \delta r$. Since $\mathcal{H}(S^{\text{NO-REP}}) \subseteq \mathcal{P}(\mathcal{Y})$, we have $|S| - |I(S)| \leqslant |\mathcal{Y}| = r$. This implies that $|S| \leqslant (1+\delta)r$.

2. $|I(S)| > \delta r$. From our assumption, $|\mathcal{H}(S^{\text{NO-REP}})| \leqslant 100n\log|\Sigma|$. Thus, $|S| \leqslant |I(S)| + \log(100n\log|\Sigma|) \leqslant (1+\delta)r$ where the second inequality comes from our assumption that $\delta \geqslant \log(1000n\log|\Sigma|)/r$.

Hence, $\text{VC-dim}(\mathcal{U}, \mathcal{C}) \leqslant (1+\delta)r$ with high probability. □

5.4 Littlestone's Dimension

We next proceed to Littlestone's Dimension. The main theorem of this section is stated below. Again, note that this theorem and Theorem 2.4 implies Theorem 5.2.

Theorem 5.19. *There exists $\varepsilon > 0$ such that there is a randomized reduction from any bi-regular Label Cover instance $\mathcal{L} = (A, B, E, \Sigma, \{\pi_e\}_{e \in E})$ with $|\Sigma| = O(1)$ to a ground set $\mathcal{U}$ and a concept classes $\mathcal{C}$ such that, if $n := |A| + |B|, r := \sqrt{n}/\log n$ and $k := 10^{10}|E|\log|\Sigma|/r^2$, then the following conditions hold for every sufficiently large n.*

- *(Size) The reduction runs in time $2^{rk} \cdot |\Sigma|^{O(|E|/r)}$ and $|\mathcal{C}|, |\mathcal{U}| \leqslant 2^{rk} \cdot |\Sigma|^{O(|E|/r)}$.*
- *(Completeness) If $\mathcal{L}$ is satisfiable, then $\text{L-dim}(\mathcal{C}, \mathcal{U}) \geqslant 2rk$.*
- *(Soundness) If $\text{val}(\mathcal{L}) \leqslant 0.001$, then $\text{L-dim}(\mathcal{C}, \mathcal{U}) \leqslant (2 - \varepsilon)rk$ with high probability.*

5.4.1 Why the VC Dimension Reduction Fails for Littlestone's Dimension

It is tempting to think that, since our reduction from the previous section works for VC Dimension, it may also work for Littlestone's Dimension. In fact, thanks to Fact 5.7, completeness for that reduction even translates for free to Littlestone's Dimension. Alas, the soundness property does not hold. To see this, let us build a depth-$2r$ mistake tree for $\mathcal{C}, \mathcal{U}$, even when $\text{val}(\mathcal{L})$ is small, as follows.

- We assign the test-selection elements to the first r levels of the tree, one element per level. More specifically, for each $s \in \{0,1\}^{<r}$, we assign $y_{|s|+1}$ to s.
- For every string $s \in \{0,1\}^r$, the previous step of the construction gives us a subset of $\mathcal{Y}$ corresponding to the path from root to s; this subset is simply $H_s = \{y_i \in \mathcal{Y} \mid s_i = 1\}$. Let T_{H_s} denote the set of vertices tested by this seed H_s. Let $\phi_s \in \Sigma^V$ denote an assignment that satisfies all the constraints in T_{H_s}. Note that, since T_{H_s} is of small size (only $\widetilde{O}(\sqrt{n})$), even if $\text{val}(\mathcal{L})$ is small, ϕ_s is still likely to exist (and we can decide whether it exists or not in time $2^{\widetilde{O}(\sqrt{n})}$).

  We then construct the subtree rooted at s that corresponds to ϕ_s by assigning each level of the subtree $x_{i,\phi_s|_{U_i}}$. Specifically, for each $t \in \{0,1\}^{\geqslant r}$, we assign $x_{|t|-r+1, \phi_{t_{\leqslant r}}|_{U_{|t|-r+1}}}$ to node t of the tree.

It is not hard to see that the constructed tree is indeed a valid mistake tree. This is because the path from root to each leaf $l \in \{0,1\}^{2r}$ agrees with $C_{I(l), H_{l_{\leqslant r}}, \phi_{l_{\leqslant r}}}$ (where $I(l) = \{i \in [r] \mid l_i = 1\}$).

5.4.2 The Final Reduction

The above counterexample demonstrates the main difference between the two dimensions: order does not matter in VC Dimension, but it does in Littlestone's Dimension. By moving the test-selection elements up the tree, the tests are chosen before the assignments, which allows an adversary to "cheat" by picking different assignments for different tests. We would like to prevent this, i.e., we would like to make sure that, in the mistake tree, the upper levels of the tree are occupied with the assignment elements whereas the lower levels are assigned test-selection elements. As in the VC Dimension argument, our hope here is that, given such a tree, we should be able to decode an assignment that passes tests on many different tests. Indeed we will tailor our construction to achieve such property.

Recall that, if we use the same reduction as VC Dimension, then, in the completeness case, we can construct a mistake tree in which the first r layers consist solely of assignment elements and the rest of the layers consist of only test-selection elements. Observe that there is no need for different nodes on the r-th layer to have subtrees composed of the same set of elements; the tree would still be valid if we make each test-selection element only work with a specific $s \in \{0,1\}^r$ and create concepts accordingly. In other words, we can modify our construction so that our test-selection elements are $\mathcal{Y} = \{y_{I,i} \mid I \subseteq [r], i \in [r]\}$ and the concept class is $\{C_{I,H,\sigma_H} \mid I \subseteq [r], H \subseteq \mathcal{Y}, \sigma_H \in \Sigma^{T_H}\}$ where the condition that an assignment element lies in C_{I,H,σ_H} is the same as in the VC Dimension reduction, whereas for $y_{I',i}$ to be in C_{I,H,σ_H}, we require not only that $i \in H$ but also that $I = I'$. Intuitively, this should help us, since each $y_{I,i}$ is now only in a small fraction ($\leqslant 2^{-r}$) of concepts; hence, one would hope that any subtree rooted at any $y_{I,i}$ cannot be too deep, which would indeed implies that the test-selection elements cannot appear in the first few layers of the tree.

Alas, for this modified reduction, it is not true that a subtree rooted at any $y_{I,i}$ has small depth; specifically, we can bound the depth of a subtree $y_{I,i}$ by the log of the number of concepts containing $y_{I,i}$ plus one (for the first layer). Now, note that $y_{I,i} \in C_{I',H,\sigma_H}$ means that $I' = I$ and $i \in H$, but there can be still as many as $2^{r-1} \cdot |\Sigma|^{|T_H|} = |\Sigma|^{O(|E|/r)}$ such concepts. This gives an upper bound of $r + O(|E| \log|\Sigma|/r)$ on the depth of the subtree rooted at $y_{I,i}$. However, $|E| \log|\Sigma|/r = \Theta(\sqrt{n} \log n) = \omega(r)$; this bound is meaningless here since, even in the completeness case, the depth of the mistake tree is only $2r$.

Fortunately, this bound is not useless after all: if we can keep this bound but make the intended tree depth much larger than $|E| \log|\Sigma|/r$, then the bound will indeed imply that no $y_{I,i}$-rooted tree is deep. To this end, our reduction will have one more parameter $k = \Theta(|E| \log|\Sigma|/r)$ where $\Theta(\cdot)$ hides a large constant and the intended tree will have depth $2rk$ in the completeness case; the top half of the tree (first rk layers) will again consist of assignment elements and the rest of the tree composes of the test-selection elements. The rough idea is to make k "copies" of each element: the assignment elements will now be $\{x_{i,\sigma_i,j} \mid i \in [r], \sigma_i \in \Sigma^{U_i}, j \in [k]\}$ and the test-selection elements will be $\{y_{I,i,j} \mid I \subseteq [r] \times [k], j \in [k]\}$. The concept class can then be defined as $\{C_{I,H,\sigma_H} \mid I \subseteq [r] \times [k], H \subseteq [r] \times [k], \sigma_H \in \Sigma^{T_H}\}$ naturally, i.e., H is used as the seed to pick the test set T_H, $y_{I',i,j} \in C_{I,H,\sigma_H}$ iff $I' = I$ and $(i,j) \in H$ whereas $x_{i,\sigma_i,j} \in C_{I,H,\sigma_H}$ iff $(i,j) \in I$ and $\sigma_i|_{(I,\sigma_I)} = \sigma_H|_{(I,\sigma_I)}$. For this concept class, we can again bound the depth of $y_{I,i}$-rooted tree to be $rk + O(|E| \log |\Sigma|/r)$; this time, however, rk is much larger than $|E| \log |\Sigma|/r$, so this bound is no more than, say, $1.001rk$. This is indeed the desired bound, since this means that, for any depth-$1.999rk$ mistake tree, the first $0.998rk$ layers must consist solely of assignment elements.

Unfortunately, the introduction of copies in turn introduces another technical challenge: it is not true any more that a partial assignment to a large set only passes a few tests w.h.p. (i.e. an analogue of Lemma 5.17 does not hold). By Inequality (5.3), each H is passed with probability at most 2^{-r}, but now we want to take a union bound there are $2^{rk} \gg 2^r$ different H's. To circumvent this, we will define a map $\tau : \mathcal{P}([r] \times [k]) \to \mathcal{P}([r])$ and use $\tau(H)$ to select the test instead of H itself. The map τ we use in the construction is the *threshold projection* where i is included in H if and only if, for at least half of $j \in [k]$, H contains (i, j). To motivate our choice of τ, recall that our overall proof approach is to first find a node that corresponds to an assignment to a large subset of the Label Cover instance; then argue that it can pass only a few tests, which we hope would imply that the subtree rooted there cannot be too deep. For this implication to be true, we need the following to also hold: for any small subset $\mathcal{H} \subseteq \mathcal{P}([r])$ of $\tau(H)$'s, we have that $\text{L-dim}(\tau^{-1}(\mathcal{H}), [r] \times [k])$ is small. This property indeed holds for our choice of τ (see Lemma 5.27).

With all the moving parts explained, we state the full reduction formally in Figure 5.2.

Similar to our VC Dimension proof, we will use the following notation:

- For every $i \in [r]$, let $\mathrm{X}_i := \{x_{i,\sigma_i,j} \mid \sigma_i \in \Sigma^{U_i}, j \in [k]\}$; we refer to these elements as the i-assignment elements. Moreover, for every $(i, j) \in [r] \times [k]$, let $\mathrm{X}_{i,j} := \{x_{i,\sigma_i,j} \mid \sigma_i \in \Sigma^{U_i}\}$; we refer to these elements as the (i, j)-assignment elements.

- For every $S \subseteq \mathcal{U}$, let $I(S) = \{i \in [r] \mid S \cap \mathrm{X}_i \neq \emptyset\}$ and $IJ(S) = \{(i, j) \in [r] \times [k] \mid S \cap \mathrm{X}_{i,j} \neq \emptyset\}$.

- A set $S \subseteq \mathrm{X}$ is *non-repetitive* if $|S \cap \mathrm{X}_{i,j}| \leqslant 1$ for all $(i, j) \in [r] \times [k]$.

- We say that S *passes* $\widetilde{H}$ if the following two conditions hold:
  - For every $i \in [r]$ such that $S \cap \mathrm{X}_i \neq \emptyset$, all i-assignment elements of S are consistent on $T_{\widetilde{H}}|_{U_i}$, i.e., for every $(i, \sigma_i, j), (i, \sigma'_i, j') \in S$, we have $\sigma_i|_{U_i} = \sigma'_i|_{U_i}$.
  - The canonically induced assignment on $T_{\widetilde{H}}$ does not violate any constraint (note that the previous condition implies that such assignment is unique).

  We use $\mathcal{H}(S)$ to denote the collection of all seeds $\widetilde{H} \subseteq [r]$ that S passes.

We also use the following notation for mistake trees:

- For any subset $S \subseteq \mathcal{U}$ and any function $\rho : S \to \{0, 1\}$, let $\mathcal{C}[\rho] := \{C \in \mathcal{C} \mid \forall a \in S, a \in C \Leftrightarrow \rho(a) = 1\}$ be the collections of all concept that agree with ρ on S. We sometimes abuse the notation and write $\mathcal{C}[S]$ to denote the collection of all the concepts that contain S, i.e., $\mathcal{C}[S] = \{C \in \mathcal{C} \mid S \subseteq C\}$.

- For any binary string s, let $\mathrm{pre}(s) := \{\emptyset, s_{\leqslant 1}, \dots, s_{\leqslant |s|-1}\}$ denote the set of all proper prefixes of s.

Input: A bi-regular Label Cover instance $\mathcal{L} = (A, B, E, \Sigma, \{\pi_e\}_{e \in E})$.
Output: A ground set $\mathcal{U}$ and a concept class $\mathcal{C}$.
The procedure to generate $(\mathcal{U}, \mathcal{C})$ works as follows:

- Let $r, U_1, \dots, U_r, \mathcal{N}$ be defined in the same manner as in Reduction 5.1 and let $k := 10^{10}|E| \log |\Sigma| / r^2$.
- The universe $\mathcal{U}$ consists of two types of elements, as described below.
  - *Assignment elements*: for every $i \in [r]$, every partial assignment $\sigma_i \in \Sigma^{U_i}$ and every $j \in [k]$, there is an assignment element $x_{i,\sigma_i,j}$ corresponding to it. Let X denote all the assignment elements, i.e., $\mathrm{X} = \{x_{i,\sigma_i,j} \mid i \in [r], \sigma_i \in \Sigma^{U_i}, j \in [k]\}$.
  - *Test-selection elements*: there are $rk(2^{rk})$ test-selection elements, which we will call $y_{I,i,j}$ for every $i \in [r], j \in [k], I \subseteq [r] \times [k]$. Let $\mathcal{Y}$ denote the set of all test-selection elements. Let $\mathcal{Y}_i$ denote $\{y_{I,i,j} \mid I \subseteq [r] \times [k], j \in [k]\}$. We call the elements of $\mathcal{Y}_i$ *i-test-selection elements*.
- The concepts in $\mathcal{C}$ are defined by the following procedure.
  - Let $\ell := 1000$ be the number of matchings to be tested.
  - For each $\widetilde{H} \subseteq [r]$, we randomly select ℓ permutations $\pi^{(1)}_{\widetilde{H}}, \dots, \pi^{(\ell)}_{\widetilde{H}} : [r] \to [r]$; this gives us ℓ matchings (i.e. the t-th matching is $\left(\pi^{(t)}_{\widetilde{H}}(1), \pi^{(t)}_{\widetilde{H}}(2)\right), \dots, \left(\pi^{(t)}_{\widetilde{H}}(r-1), \pi^{(t)}_{\widetilde{H}}(r)\right)$). Denote the set of elements that i is matched with in the matchings by $M_{\widetilde{H}}(i)$. Let $T_{\widetilde{H}} = \bigcup_i \mathcal{N}_i(M_{\widetilde{H}}(i))$
  - Let $\tau : \mathcal{P}([r] \times [k]) \to \mathcal{P}([r])$ denote the *threshold projection* operation where each $i \in [r]$ is included in $\tau(H)$ if and only if H contains at least half of the i-test-selection elements, i.e., $\tau(H) = \{i \in [r] \mid |H \cap \mathcal{Y}_i| \geqslant k/2\}$.
  - For every $I \subseteq [r] \times [k], H \subseteq [r] \times [k]$ and for every partial assignment $\sigma_{\tau(H)} \in \Sigma^{T_{\tau(H)}}$ that does not violate any constraints, we create a concept $C_{I,H,\sigma_{\tau(H)}}$ such that each $x_{i,\sigma_i,j} \in \mathrm{X}$ is included in $C_{I,H,\sigma_{\tau(H)}}$ if and only if $(i,j) \in I$ and σ_i is consistent with $\sigma_{\tau(H)}$, i.e., $\sigma_i|_{\mathcal{N}_i(M_{\tau(H)}(i))} = \sigma_{\tau(H)}|_{\mathcal{N}_i(M_{\tau(H)}(i))}$ whereas each $y_{I',i,j} \in \mathcal{Y}$ in included in $C_{I,H,\sigma_{\tau(H)}}$ if and only if $(i,j) \in H$ and $I' = I$.

Figure 5.2: Reduction from Label Cover to Littlestone's Dimension

- For any depth-d mistake tree $\mathcal{T}$, let $v_{\mathcal{T},s}$ denote the element assigned to the node $s \in \{0,1\}^{\leqslant d}$, and let $P_{\mathcal{T},s} := \{v_{\mathcal{T},s'} \mid s' \in \mathrm{pre}(s)\}$ denote the set of all elements appearing from the path from root to s (excluding s itself). Moreover, let $\rho_{\mathcal{T},s} : P_{\mathcal{T},s} \to \{0,1\}$ be the function corresponding to the path from root to s, i.e., $\rho_{\mathcal{T},s}(v_{\mathcal{T},s'}) = s_{|s'|+1}$ for every $s' \in \operatorname{pre}(s)$.

Output Size of the Reduction The output size of the reduction follows immediately from a similar argument as in the VC Dimension reduction. The only different here is that there are 2^{rk} choices for I and H, instead of 2^r choices as in the previous construction.

Completeness. If $\mathcal{L}$ has a satisfying assignment $\sigma^* \in \Sigma^V$, we can construct a depth-rk mistake tree $\mathcal{T}$ as follows. For $i \in [r], j \in [k]$, we assign $x_{i,\sigma^*|_{U_i},j}$ to every node in the $((i-1)k+j)$-th layer of $\mathcal{T}$. Note that we have so far assigned every node in the first rk layers. For the rest of the vertices s's, if s lies in layer $rk + (i-1)k + j$, then we assign $y_{I(\rho^{-1}_{\mathcal{T},s}(1)),i,j}$ to it. It is clear that, for a leaf $s \in \{0,1\}^{rk}$, the concept $C_{I(\rho^{-1}_{\mathcal{T},s}(1)),H_{\mathcal{T},s},\sigma^*}$ agrees with the path from root to s where $H_{\mathcal{T},s}$ is defined as $\{(i,j) \in [r] \times [k] \mid y_{I(\rho^{-1}_{\mathcal{T},s}(1)),i,j} \in \rho^{-1}_{\mathcal{T},s}(1)\}$. Hence, $\text{L-dim}(\mathcal{C},\mathcal{U}) \geqslant 2rk$.

5.4.3 Soundness

Next, we will prove the soundness of our reduction, stated more precisely below. For brevity, we will assume throughout this subsection that r is sufficiently large, and leave it out of the lemmas' statements. Note that this lemma, together with completeness and output size properties we argue above, implies Theorem 5.19 with $\varepsilon = 0.001$.

Lemma 5.20. *Let $(\mathcal{C},\mathcal{U})$ be the output from the reduction in Figure 5.2 on input $\mathcal{L}$. If $\operatorname{val}(\mathcal{L}) \leqslant 0.001$, then $\text{L-dim}(\mathcal{C},\mathcal{U}) \leqslant 1.999rk$ with high probability.*

Roughly speaking, the overall strategy of our proof of Lemma 5.20 is as follows:

1. First, we will argue that any subtree rooted at any test-selection element must be shallow (of depth $\leqslant 1.001rk$). This means that, if we have a depth-$1.999rk$ mistake tree, then the first $0.998rk$ levels must be assigned solely assignment elements.

2. We then argue that, in this $0.998rk$-level mistake tree of assignment elements, we can always extract a leaf s such that the path from root to s indicates inclusion of a large non-repetitive set. In other words, the path to s can be decoded into a (partial) assignment for the Label Cover instance $\mathcal{L}$.

3. Let the leaf from the previous step be s and the non-repetitive set be $S^{\text{NO-REP}}$. Our goal now is to show that the subtree rooted as s must have small depth. We start working towards this by showing that, with high probability, there are few tests that agree with $S^{\text{NO-REP}}$. This is analogous to Part II of the VC Dimension proof.

4. With the previous steps in mind, we only need to argue that, when $|\mathcal{H}(S^{\text{NO-REP}})|$ is small, the Littlestone's dimension of all the concepts that contains $S^{\text{NO-REP}}$ (i.e. $\text{L-dim}(\mathcal{C}[S^{\text{NO-REP}}],\mathcal{U})$) is small. Thanks to Fact 5.8, it is enough for us to bound $\text{L-dim}(\mathcal{C}[S^{\text{NO-REP}}],\mathrm{X})$ and $\text{L-dim}(\mathcal{C}[S^{\text{NO-REP}}],\mathcal{Y})$ separately. For the former, our technique from the second step also gives us the desired

bound; for the latter, we prove that $\text{L-dim}(\mathcal{C}[S^{\text{NO-REP}}], \mathcal{Y})$ is small by designing an algorithm that provides correct predictions on a constant fraction of the elements in $\mathcal{Y}$.

Let us now proceed to the details of the proofs.

Part I: Subtree of a Test-Selection Assignment is Shallow

Lemma 5.21. *For any $y_{I,i,j} \in \mathcal{Y}$, $\text{L-dim}(\mathcal{C}[\{y_{I,i,j}\}], \mathcal{U}) \leqslant rk + (4|E|\ell/r)\log|\Sigma| \leqslant 1.001rk$.*

Note that the above lemma implies that, in any mistake tree, the depth of the subtree rooted at any vertex s assigned to some $y_{I,i,j} \in \mathcal{Y}$ is at most $1 + 1.001rk$. This is because every concept that agrees with the path from the root to s must be in $\mathcal{C}[\{y_{I,i,j}\}]$, which has depth at most $1.001rk$.

Proof of Lemma 5.21. Consider any $C_{I',H,\sigma_{\tau(H)}} \in \mathcal{C}[\{y_{I,i,j}\}], \mathcal{U})$. Since $y_{I,i,j} \in C_{I',H,\sigma_{\tau(H)}}$, we have $I = I'$. Moreover, from Lemma 5.10, we know that $\left|\mathcal{N}_i\left(M_{\tau(H)}(i)\right)\right| \leqslant 4|E|\ell/r^2$, which implies that $|T_{\tau(H)}| \leqslant 4|E|\ell/r$. This means that there are only at most $|\Sigma|^{4|E|\ell/r}$ choices of $\sigma_{\tau(H)}$. Combined with the fact that there are only 2^{rk} choices of H, we have $|\mathcal{C}[\{y_{I,i,j}\}]| \leqslant 2^{rk} \cdot |\Sigma|^{4|E|\ell/r}$. Fact 5.7 then implies the lemma. □

Part II: Deep Mistake Tree Contains a Large Non-Repetitive Set

The goal of this part of the proof is to show that, for mistake tree of $\mathrm{X}, \mathcal{C}$ of depth slightly less than rk, there exists a leaf s such that the corresponding path from root to s indicates an inclusion of a large non-repetitive set; in our notation, this means that we would like to identify a leaf s such that $IJ(\rho^{-1}_{\mathcal{T},s}(1))$ is large. Since we will also need a similar bound later in the proof, we will prove the following lemma, which is a generalization of the stated goal that works even for the concept class $\mathcal{C}[S^{\text{NO-REP}}]$ for any non-repetitive $S^{\text{NO-REP}}$. To get back the desired bound, we can simply set $S^{\text{NO-REP}} = \emptyset$.

Lemma 5.22. *For any non-repetitive set $S^{\text{NO-REP}}$ and any depth-d mistake tree $\mathcal{T}$ of $\mathrm{X}, \mathcal{C}[S^{\text{NO-REP}}]$, there exists a leaf $s \in \{0,1\}^d$ such that $|IJ(\rho^{-1}_{\mathcal{T},s}(1)) \setminus IJ(S^{\text{NO-REP}})| \geqslant d - r$.*

The proof of this lemma is a double counting argument where we count a specific class of leaves in two ways, which ultimately leads to the above bound. The leaves that we focus on are the leaves $s \in \{0,1\}^d$ such that, for every (i,j) such that an (i,j)-assignment element appears in the path from root to s but not in $S^{\text{NO-REP}}$, the first appearance of (i,j)-assignment element in the path is included. In other words, for every $(i,j) \in IJ(P_{\mathcal{T},s}) \setminus IJ(S^{\text{NO-REP}})$, if we define $u_{i,j} := \inf_{s' \in \text{pre}(s), v_{\mathcal{T},s'} \in \mathrm{X}_{i,j}} |s'|$, then $s_{u_{i,j}+1}$ must be equal to 1. We call these leaves the *good* leaves. Denote the set of good leaves of $\mathcal{T}$ by $\mathcal{G}_{\mathcal{T},S^{\text{NO-REP}}}$.

Our first way of counting is the following lemma. Informally, it asserts that different good leaves agree with different sets $\widetilde{H} \subseteq [r]$. This can be thought of as an analogue of Lemma 5.14 in our proof for VC Dimension. Note that this lemma immediately gives an upper bound of 2^r on $|\mathcal{G}_{\mathcal{T},S^{\text{NO-REP}}}|$.

Lemma 5.23. *For any depth-d mistake tree $\mathcal{T}$ of $\mathrm{X}, \mathcal{C}[S^{\text{NO-REP}}]$ and any different good leaves $s_1, s_2 \in \mathcal{G}_{\mathcal{T},S^{\text{NO-REP}}}$, if C_{I_1,H_1,σ_1} agrees with s_1 and C_{I_2,H_2,σ_2} agrees with s_2 for some $I_1, I_2, H_1, H_2, \sigma_1, \sigma_2$, then $\tau(H_1) \neq \tau(H_2)$.*

Proof. Suppose for the sake of contradiction that there exist $s_1 \neq s_2 \in \mathcal{G}_{\mathcal{T},S^{\text{NO-REP}}}$, $H_1, H_2, I_1, I_2, \sigma_1, \sigma_2$ such that C_{I_1,H_1,σ_1} and C_{I_2,H_2,σ_2} agree with s_1 and s_2 respectively, and $\tau(H_1) = \tau(H_2)$. Let s be the common ancestor of s_1, s_2, i.e., s is the longest string in $\mathrm{pre}(s_1) \cap \mathrm{pre}(s_2)$. Assume w.l.o.g. that $(s_1)_{|s|+1} = 0$ and $(s_2)_{|s|+1} = 1$. Consider the node $v_{\mathcal{T},s}$ in tree $\mathcal{T}$ where the paths to s_1, s_2 split; suppose that this is $x_{i,\sigma_i,j}$. Therefore $x_{i,\sigma_i,j} \in C_{I_2,H_2,\sigma_2} \setminus C_{I_1,H_1,\sigma_1}$.

We now argue that there is some $x_{i,\sigma'_i,j}$ (with the same i, j but a different assignment σ'_i) that is in both concepts, i.e. $x_{i,\sigma'_i,j} \in C_{I_2,H_2,\sigma_2} \cap C_{I_1,H_1,\sigma_1}$. We do this by considering two cases:

- If $(i,j) \in IJ(S^{\text{NO-REP}})$, then there is $x_{i,\sigma'_i,j} \in S^{\text{NO-REP}} \subseteq C_{I_1,H_1,\sigma_1}, C_{I_2,H_2,\sigma_2}$ for some $\sigma'_i \in \Sigma^{U_i}$.

- Suppose that $(i,j) \notin IJ(S^{\text{NO-REP}})$. Since s_1 is a good leaf, there is some $t \in \mathrm{pre}(s)$ such that $v_{\mathcal{T},t} = x_{i,\sigma'_i,j}$ for some $\sigma'_i \in \Sigma^{U_i}$ and t is included by the path (i.e. $s_{|t|+1} = 1$). This also implies that $x_{i,\sigma'_i,j}$ is in both C_{I_1,H_1,σ_1} and C_{I_2,H_2,σ_2}.

Now, since both $x_{i,\sigma_i,j}$ and $x_{i,\sigma'_i,j}$ are in the concept C_{I_2,H_2,σ_2}, we have $(i,j) \in I_2$ and

$$\sigma_i|_{\mathcal{N}_i(M_{\tau(H_1)})} = \sigma_2|_{\mathcal{N}_i(M_{\tau(H_1)})} = \sigma'_i|_{\mathcal{N}_i(M_{\tau(H_1)})}. \tag{5.4}$$

On the other hand, since C_{I_1,H_1,σ_1} contains $x_{i,\sigma'_i,j}$ but not $x_{i,\sigma_i,j}$, we have $(i,j) \in I_1$ and

$$\sigma_i|_{\mathcal{N}_i(M_{\tau(H_2)})} \neq \sigma_1|_{\mathcal{N}_i(M_{\tau(H_2)})} = \sigma'_i|_{\mathcal{N}_i(M_{\tau(H_2)})}. \tag{5.5}$$

which contradicts (5.4) since $\tau(H_1) = \tau(H_2)$. □

Next, we will present another counting argument which gives a lower bound on the number of good leaves, which, together with Lemma 5.23, yields the desired bound.

Proof of Lemma 5.22. For any depth-d mistake tree $\mathcal{T}$ of $\mathcal{C}[S^{\text{NO-REP}}]$, X, let us consider the following procedure which recursively assigns a weight λ_s to each node s in the tree. At the end of the procedure, all the weight will be propagated from the root to good leaves.

1. For every non-root node $s \in \{0,1\}^{\geqslant 1}$, set $\lambda_s \leftarrow 0$. For root $s = \emptyset$, let $\lambda_\emptyset \leftarrow 2^d$.
2. While there is an internal node $s \in \{0,1\}^{<d}$ such that $\lambda_s > 0$, do the following:
    a) Suppose that $v_s = x_{i,\sigma_i,j}$ for some $i \in [r], \sigma_i \in \Sigma^{U_i}$ and $j \in [k]$.
    b) If so far no (i,j)-element has appeared in the path or in $S^{\text{NO-REP}}$, i.e., $(i,j) \notin IJ(P_{\mathcal{T},s}) \cup IJ(S^{\text{NO-REP}})$, then $\lambda_{s1} \leftarrow \lambda_s$. Otherwise, set $\lambda_{s0} = \lambda_{s1} = \lambda_s/2$.
    c) Set $\lambda_s \leftarrow 0$.

The following observations are immediate from the construction:

- The total of λ's over all the tree, $\sum_{s \in \{0,1\}^{\leqslant d}} \lambda_d$ always remain 2^d.
- At the end of the procedure, for every $s \in \{0,1\}^{\leqslant d}$, $\lambda_s \neq 0$ if and only if $s \in \mathcal{G}_{\mathcal{T}, S^{\text{NO-REP}}}$.
- If $s \in \mathcal{G}_{\mathcal{T}, S^{\text{NO-REP}}}$, then $\lambda_s = 2^{|IJ(\rho_{\mathcal{T},s}^{-1}(1)) \setminus IJ(S^{\text{NO-REP}})|}$ at the end of the execution.

Note that the last observation comes from the fact that λ always get divides in half when moving down one level of the tree unless we encounter an (i,j)-assignment element for some i,j that never appears in the path or in $S^{\text{NO-REP}}$ before. For any good leaf s, the set of such (i,j) is exactly the set $IJ(\rho_{\mathcal{T},s}^{-1}(1)) \setminus IJ(S^{\text{NO-REP}})$.

As a result, we have $2^d = \sum_{s \in \mathcal{G}_{\mathcal{T}, S^{\text{NO-REP}}}} 2^{|IJ(\rho_{\mathcal{T},s}^{-1}(1)) \setminus IJ(S^{\text{NO-REP}})|}$. Since Lemma 5.23 implies that $|\mathcal{G}_{\mathcal{T}, S^{\text{NO-REP}}}| \leqslant 2^r$, we can conclude that there exists $s \in \mathcal{G}_{\mathcal{T}, S^{\text{NO-REP}}}$ such that $|IJ(\rho_{\mathcal{T},s}^{-1}(1)) \setminus IJ(S^{\text{NO-REP}})| \geqslant d - r$ as desired. □

Part III: No Large Non-Repetitive Set Passes Many Test

The main lemma of this subsection is the following, which is analogous to Lemma 5.16

Lemma 5.24. *If* $\text{val}(\mathcal{L}) \leqslant 0.001$*, then, with high probability, for every non-repetitive set* $S^{\text{NO-REP}}$ *of size at least* $0.99rk$*,* $|\mathcal{H}(S^{\text{NO-REP}})| \leqslant 100n \log |\Sigma|$.

Proof. For every $I \subseteq [r]$, let $U_I := \bigcup_{i \in I} U_i$. For every $\sigma_I \in \Sigma^{U_I}$ and every $\widetilde{H} \subseteq \mathcal{Y}$, we say that (I, σ_I) *passes* $\widetilde{H}$ if σ_I does not violate any constraint in $T_{\widetilde{H}}$. Note that this definition and the way the test is generated in the reduction is the same as that of the VC Dimension reduction. Hence, we can apply Lemma 5.17 with $\delta = 0.99$, which implies the following: with high probability, for every $I \subseteq [r]$ of size at least $0.99r$ and every $\sigma_I \in \Sigma^{U_I}$, $|\mathcal{H}(I, \sigma_I)| \leqslant 100n \log |\Sigma|$ where $\mathcal{H}(I, \sigma_I)$ denote the set of all $\mathcal{H}$'s passed by (I, σ_I). Conditioned on this event happening, we will show that, for every non-repetitive set $S^{\text{NO-REP}}$ of size at least $0.99rk$, $|\mathcal{H}(S^{\text{NO-REP}})| \leqslant 100n \log |\Sigma|$.

Consider any non-repetitive set $S^{\text{NO-REP}}$ of size $0.99rk$. Let $\sigma_{I(S^{\text{NO-REP}})}$ be an assignment on $U_{I(S^{\text{NO-REP}})}$ such that, for each $i \in I(S^{\text{NO-REP}})$, we pick one $x_{i,\sigma_i,j} \in S^{\text{NO-REP}}$ (if there are more than one such x's, pick one arbitrarily) and let $\sigma_{I(S^{\text{NO-REP}})}|_{U_i} = \sigma_i$. It is obvious that $\mathcal{H}(S^{\text{NO-REP}}) \subseteq \mathcal{H}(I(S^{\text{NO-REP}}), \sigma_{I(S^{\text{NO-REP}})})$. Since $S^{\text{NO-REP}}$ is non-repetitive and of size at least $0.99rk$, we have $|I(S^{\text{NO-REP}})| \geqslant 0.99r$, which means that $|\mathcal{H}(I(S^{\text{NO-REP}}), \sigma_{I(S^{\text{NO-REP}})})| \leqslant 100n \log |\Sigma|$ as desired. □

Part IV: A Subtree Containing $S^{\text{NO-REP}}$ Must be Shallow

In this part, we will show that, if we restrict ourselves to only concepts that contain some non-repetitive set $S^{\text{NO-REP}}$ that passes few tests, then the Littlestone's Dimension of this restrictied concept class is small. Therefore when we build a tree for the whole concept class $\mathcal{C}$, if a path from root to some node indicates an inclusion of a non-repetitive set that passes few tests, then the subtree rooted at this node must be shallow.

Lemma 5.25. *For every non-repetitive set $S^{\text{NO-REP}}$,*

$$\text{L-dim}(\mathcal{C}[S^{\text{NO-REP}}], \mathcal{U}) \leqslant 1.75rk - |S^{\text{NO-REP}}| + r + 1000k\sqrt{r}\log(|\mathcal{H}(S^{\text{NO-REP}})| + 1).$$

We prove the above lemma by bounding $\text{L-dim}(\mathcal{C}[S^{\text{NO-REP}}], \mathrm{X})$ and $\text{L-dim}(\mathcal{C}[S^{\text{NO-REP}}], \mathcal{Y})$ separately, and combining them via Fact 5.8. First, we can bound $\text{L-dim}(\mathcal{C}[S^{\text{NO-REP}}], \mathrm{X})$ easily by applying Lemma 5.22 coupled with the fact that $|IJ(S^{\text{NO-REP}})| = |S^{\text{NO-REP}}|$ for every non-repetitive $S^{\text{NO-REP}}$. This immediately gives the following corollary.

Corollary 5.26. *For every non-repetitive set $S^{\text{NO-REP}}$,*

$$\text{L-dim}(\mathcal{C}[S^{\text{NO-REP}}], \mathrm{X}) \leqslant rk - |S^{\text{NO-REP}}| + r.$$

We will next prove the following bound on $\text{L-dim}(\mathcal{C}[S^{\text{NO-REP}}], \mathcal{Y})$. Note that Corollary 5.26, Lemma 5.27, and Fact 5.8 immediately imply Lemma 5.25.

Lemma 5.27. *For every non-repetitive set $S^{\text{NO-REP}}$,*

$$\text{L-dim}(\mathcal{C}[S^{\text{NO-REP}}], \mathcal{Y}) \leqslant 0.75rk + 500k\sqrt{r}\log(|\mathcal{H}(S^{\text{NO-REP}})| + 1).$$

The overall outline of the proof of Lemma 5.27 is that we will design a prediction algorithm whose mistake bound is at most $0.75rk + 1000k\sqrt{r}\log|\mathcal{H}(S^{\text{NO-REP}})|$. Once we design this algorithm, Lemma 5.6 immediately implies Lemma 5.27. To define our algorithm, we will need the following lemma, which is a general statement that says that, for a small collection of H's, there is a some $\widetilde{H}^* \subseteq [r]$ that agrees with almost half of every H in the collection.

Lemma 5.28. *Let $\mathcal{H} \subseteq \mathcal{P}([r])$ be any collections of subsets of $[r]$, there exists $\widetilde{H}^* \subseteq [r]$ such that, for every $\widetilde{H} \in \mathcal{H}$, $|\widetilde{H}^* \Delta \widetilde{H}| \leqslant 0.5r + 1000\sqrt{r}\log(|\mathcal{H}| + 1)$ where Δ denotes the symmetric difference between two sets.*

Proof. We use a simple probabilistic method to prove this lemma. Let $\widetilde{H}^r$ be a random subset of $[r]$ (i.e. each $i \in [r]$ is included independently with probability 0.5). We will show that, with non-zero probability, $|\widetilde{H}^r \Delta \widetilde{H}| \leqslant 0.5r + 1000\sqrt{r}\log(|\mathcal{H}| + 1)$ for all $\widetilde{H} \in \mathcal{H}$, which immediately implies that a desired $\widetilde{H}^*$ exists.

Fix $\widetilde{H} \in \mathcal{H}$. Observe that $|\widetilde{H}^r \Delta \widetilde{H}|$ can be written as $\sum_{i \in [r]} \mathbb{1}[i \in (\widetilde{H}^r \Delta \widetilde{H})]$. For each i, $\mathbb{1}[i \in (\widetilde{H}^r \Delta \widetilde{H})]$ is a $0, 1$ random variable with mean 0.5 independent of other $i' \in [r]$. Applying Chernoff bound here yields

$$\Pr[|\widetilde{H}^r \Delta \widetilde{H}| > 0.5r + 1000\sqrt{r}\log(|\mathcal{H}| + 1)] \leqslant 2^{-\log^2(|\mathcal{H}|+1)} \leqslant \frac{1}{|\mathcal{H}| + 1}.$$

Hence, by union bound, we have

$$\Pr[\exists \widetilde{H} \in \mathcal{H}, |\widetilde{H}^r \Delta \widetilde{H}| > 0.5r + 1000\sqrt{r}\log(|\mathcal{H}| + 1)] \leqslant \frac{|\mathcal{H}|}{|\mathcal{H}| + 1} < 1.$$

In other words, $|\widetilde{H}^r \Delta \widetilde{H}| \leqslant 0.5r + 1000\sqrt{r}\log(|\mathcal{H}| + 1)$ for all $\widetilde{H} \in \mathcal{H}$ with non-zero probability as desired. □

We also need the following observation, which is an analogue of Observation 5.15 in the VC Dimension proof; it follows immediately from definition of $\mathcal{H}(S)$.

Observation 5.29. *If a non-repetitive set $S^{\text{NO-REP}}$ is a subset of some concept $C_{I,H,\sigma_{\tau(H)}}$, then $\tau(H) \in \mathcal{H}(S^{\text{NO-REP}})$.*

With Lemma 5.28 and Observation 5.29 in place, we are now ready to prove Lemma 5.27.

Proof of Lemma 5.27. Let $\widetilde{H}^* \subseteq [r]$ be the set guaranteed by applying Lemma 5.28 with $\mathcal{H} = \mathcal{H}(S^{\text{NO-REP}})$. Let $H^* := \widetilde{H}^* \times [k]$.

Our prediction algorithm will be very simple: it always predicts according to H^*; i.e., on an input[4] $y \in \mathcal{Y}$, it outputs $\mathbb{1}[y \in H^*]$. Consider any sequence $(y_1, h_1), \dots, (y_w, h_w)$ that agrees with a concept $C_{I,H,\sigma_{\tau(H)}} \in \mathcal{C}[S^{\text{NO-REP}}]$. Observe that the number of incorrect predictions of our algorithm is at most $|H^* \Delta H|$.

Since $C_{I,H,\sigma_{\tau(H)}} \in \mathcal{C}[S^{\text{NO-REP}}]$, Observation 5.29 implies that $\tau(H) \in \mathcal{H}(S^{\text{NO-REP}})$. This means that $|\tau(H) \Delta \widetilde{H}^*| \leqslant 0.5r + 1000\sqrt{r}\log(|\mathcal{H}|+1)$. Now, let us consider each $i \in [r] \setminus (\tau(H) \Delta \widetilde{H}^*)$. Suppose that $i \in \tau(H) \cap \widetilde{H}^*$. Since $i \in \tau(H)$, at least $k/2$ elements of $\mathcal{Y}_i$ are in H and, since $i \in \widetilde{H}^*$, we have $\mathcal{Y}_i \subseteq H^*$. This implies that $|(H^* \Delta H) \cap Y_i| \leqslant k/2$. A similar bound can also be derived when $i \notin \tau(H) \cap \widetilde{H}^*$. As a result, we have

$$\begin{aligned}
|H^* \Delta H| &= \sum_{i \in [r]} |(H^* \Delta H) \cap Y_i| \\
&= \sum_{i \in \tau(H) \Delta \widetilde{H}^*} |(H^* \Delta H) \cap Y_i| + \sum_{i \in [r] \setminus (\tau(H) \Delta \widetilde{H}^*)} |(H^* \Delta H) \cap Y_i| \\
&\leqslant (|\tau(H) \Delta \widetilde{H}^*|)(k) + (r - |\tau(H) \Delta \widetilde{H}^*|)(k/2) \\
&\leqslant 0.75rk + 500k\sqrt{r}\log(|\mathcal{H}|+1),
\end{aligned}$$

concluding our proof of Lemma 5.27. □

Putting Things Together

Proof of Lemma 5.20. Assume that $\operatorname{val}(\mathcal{L}) \leqslant 0.001$. From Lemma 5.24, we know that, with high probability, $|\mathcal{H}(S^{\text{NO-REP}})| \leqslant 100n \log |\Sigma|$ for every non-repetitive set $S^{\text{NO-REP}}$ of size at least $0.99rk$. Conditioned on this event, we will show that $\text{L-dim}(\mathcal{C}, \mathcal{U}) \leqslant 1.999rk$.

Suppose for the sake of contradiction that $\text{L-dim}(\mathcal{C}, \mathcal{U}) > 1.999rk$. Consider any depth-$1.999rk$ mistake tree $\mathcal{T}$ of $\mathcal{C}, \mathcal{U}$. From Lemma 5.21, no test-selection element is assigned to any node in the first $1.999rk - 1.001rk - 1 \geqslant 0.997rk$ levels. In other words, the tree induced by the first $0.997rk$ levels is simply a mistake tree of $\mathcal{C}$, X. By Lemma 5.22 with $S^{\text{NO-REP}} = \emptyset$, there exists $s \in \{0,1\}^{0.997rk}$ such that $|IJ(\rho^{-1}_{\mathcal{T},s}(1))| \geqslant 0.997rk - r \geqslant 0.996rk$.

[4] We assume w.l.o.g. that input elements are distinct; if an element appears multiple times, we know the correct answer from its first appearance and can always correctly predict it afterwards.

Since $|IJ(\rho_{\mathcal{T},s}^{-1}(1))| \geqslant 0.996rk$, there exists a non-repetitive set $S^{\text{NO-REP}} \subseteq \rho_{\mathcal{T},s}^{-1}(1)$ of size $0.996rk$. Consider the subtree rooted at s. This is a mistake tree of $\mathcal{C}[\rho_{\mathcal{T},s}],\mathcal{U}$ of depth $1.002rk$. Since $S^{\text{NO-REP}} \subseteq \rho_{\mathcal{T},s}^{-1}(1)$, we have $\mathcal{C}[\rho_{\mathcal{T},s}] \subseteq \mathcal{C}[S^{\text{NO-REP}}]$. However, this implies

$$\begin{aligned} 1.002rk &\leqslant \text{L-dim}(\mathcal{C}[\rho_{\mathcal{T},s}],\mathcal{U}) \\ &\leqslant \text{L-dim}(\mathcal{C}[S^{\text{NO-REP}}],\mathcal{U}) \\ \text{(From Lemma 5.25)} &\leqslant 1.75rk - 0.996rk + r + 100k\sqrt{r}\log(|\mathcal{H}(S^{\text{NO-REP}})| + 1) \\ \text{(From Lemma 5.24)} &\leqslant 0.754rk + r + 100k\sqrt{r}\log(100n\log|\Sigma| + 1) \\ &= 0.754rk + o(rk), \end{aligned}$$

which is a contradiction when r is sufficiently large. □

5.4.4 Quasi-polynomial Algorithm for Littlestone's Dimension

In this section, we provides the following algorithm which decides whether $\text{L-dim}(\mathcal{C},\mathcal{U}) \leqslant d$ in time $O(|\mathcal{C}| \cdot (2|\mathcal{U}|)^d)$. Since we know that $\text{L-dim}(\mathcal{C},\mathcal{U}) \leqslant \log|\mathcal{C}|$, we can run this algorithm for all $d \leqslant \log|\mathcal{C}|$ and compute Littlestone's Dimension of $\mathcal{C},\mathcal{U}$ in quasi-polynomial time.

Theorem 5.30 (Quasi-polynomial Time Algorithm for Littlestone's Dimension)**.** *There is an algorithm that, given a universe $\mathcal{U}$, a concept class $\mathcal{C}$ and a non-negative integer d, decides whether $\text{L-dim}(\mathcal{C},\mathcal{U}) \leqslant d$ in time $O(|\mathcal{C}| \cdot (2|\mathcal{U}|)^d)$.*

Proof. Our algorithm is based on a simple observation: if an element x belongs to at least one concept and does not belong to at least one concept, the maximum depth of mistake trees rooted at x is exactly $1 + \min\{\text{L-dim}(\mathcal{C}[x \to 0],\mathcal{U}), \text{L-dim}(\mathcal{C}[x \to 1],\mathcal{U})\}$. Recall from Section 5.4.2 that $\mathcal{C}[x \to 0]$ and $\mathcal{C}[x \to 1]$ denote the collection of concepts that exclude x and the collection of concepts that include x respectively.

This yields the following natural recursive algorithm. For each $x \in \mathcal{U}$ such that $\mathcal{C}[x \to 0], \mathcal{C}[x \to 1] \neq \emptyset$, recursively run the algorithm on $(\mathcal{C}[x \to 0],\mathcal{U}, d-1)$ and $(\mathcal{C}[x \to 1],\mathcal{U}, d-1)$. If both executions return NO for some x, then output NO. Otherwise, output YES. When $d = 0$, there is no need for recursion as we can just check whether $|\mathcal{C}| \leqslant 1$.

Finally, we note that the running time can be easily proved by induction on d. □

5.5 Discussion and Open Questions

In this work, we prove inapproximability results for VC Dimension and Littlestone's Dimension based on the randomized exponential time hypothesis. Our results provide an almost matching running time lower bound of $n^{\log^{1-o(1)} n}$ for both problems while ruling out approximation ratios of $1/2 + o(1)$ and $1 - \varepsilon$ for some $\varepsilon > 0$ for VC Dimension and Littlestone's Dimension respectively. Even though our results help us gain more insights on approximability of both problems, it is not yet completely resolved. More specifically, we are not aware of any constant factor $n^{o(\log n)}$-time approximation algorithm for either problem; it is an intriguing open question whether such algorithm exists and, if not, whether our reduction can be extended to rule out such algorithm. Another potentially interesting research direction is to derandomize our construction; note that the only place in the proof in which the randomness is used is in Lemma 5.17.

A related question which remains open, originally posed by Ben-David and Eiron [BE98], is that of computing the *self-directed learning*[5] mistake bound. Similarly, it may be interesting to understand the complexity of computing (approximating) the recursive teaching dimension [Dol+14; Mor+15].

[5] Roughly, self-directed learning is similar to the online learning model corresponding to Littlestone's dimension, but where the learner chooses the order elements; see [BE98] for details.

Part II

Parameterized Problems

Chapter 6

Inapproximability of k-Dominating Set

In the *dominating set* problem (DOMSET), we are given an undirected graph G on n vertices and an integer k, and the goal is to decide whether there is a subset of vertices $S \subseteq V(G)$ of size k such that every vertex outside S has a neighbor in S (i.e., S *dominates* every vertex in G and is thus called a *dominating set*). Often regarded as one of the classical problems in computational complexity, DOMSET was first shown to be NP-complete in the seminal work of Karp [Kar72][1]. Thus, its optimization variant, namely the *minimum dominating set*, where the goal is to find a dominating set of smallest possible size, is also NP-hard. To circumvent this apparent intractability of the problem, the study of an approximate version was initiated. The quality of an approximation algorithm is measured by the *approximation ratio*, which is the ratio between the size of the solution output by an algorithm and the size of the minimum dominating set. A simple greedy heuristic for the problem, which has by now become one of the first approximation algorithms taught in undergraduate and graduate algorithm courses, was intensively studied and was shown to yield a $(\ln n - \ln \ln n + \Theta(1))$-approximation for the problem [Joh74; Chv79; Lov75; Sri95; Sla96]. On the opposite side, a long line of works in hardness of approximation [LY94; RS97; Fei98; AMS06; Mos12] culminated in the work of Dinur and Steurer [DS14], who showed that obtaining a $(1 - \varepsilon) \ln n$-approximation for the problem is NP-hard for every $\varepsilon > 0$. This essentially settles the approximability of the problem.

The parameterized version of DOMSET, which we will refer to simply as k-DOMSET, turns out to also be intractable: in the same work that introduced the W-hierarchy, Downey and Fellows [DF95b] showed that k-DOMSET is complete for the class W[2], which is generally believed to not be contained in FPT. In the ensuing years, stronger running time lower bounds have been shown for k-DOMSET under strengthened assumptions. Specifically, Chen et al. [Che+06] ruled out $T(k) \cdot n^{o(k)}$-time algorithm for k-DOMSET assuming ETH. Furthermore, Pătrascu and Williams [PW10] proved, for every $k \geqslant 2$, that, under SETH, not even $O(n^{k-\varepsilon})$ algorithm exists for k-DOMSET for any $\varepsilon > 0$. Note that the trivial algorithm that enumerates through every k-size subset and checks whether it forms a dominating set runs in $O(n^{k+1})$ time. It is possible to speed up this running time using fast matrix multiplication [EG04; PW10]. In particular, Pătrascu and

[1] To be precise, Karp showed NP-completeness of Set Cover, which is well-known to be equivalent to DOMSET.

Williams [PW10] themselves also gave an $n^{k+o(1)}$-time algorithm for every $k \geqslant 7$, painting an almost complete picture of the complexity of the problem.

Given the strong negative results for k-DOMSET discussed in the previous paragraph, it is natural to ask whether we can somehow incorporate the ideas from the area of approximation algorithms to come up with an FPT approximation algorithm for k-DOMSET. This brings us to the main question addressed in this chapter: *Is there an $F(k)$-FPT-approximation algorithm for* k-DOMSET *for some computable function F?* This question, which dates back to late 1990s (see, e.g., [DFM06]), has attracted significant attention in literature [DFM06; CGG06; Dow+08; CH10; DF13; HKK13; CHK13; Bon+15; CL16; Cha+17]. In fact, it is even listed in the seminal textbook of Downey and Fellows [DF13] as one of the six "most infamous" open questions[2] in the area of Parameterized Complexity. While earlier attempts fell short of ruling out either $F(k)$ that is super constant or all FPT algorithms (see Section 6 for more details), the last couple of years have seen significant progresses on the problem. In a remarkable result of Chen and Lin [CL16], it was shown that no FPT-approximation for k-DOMSET exists for any constant ratio unless $\mathrm{W}[1] = \mathrm{FPT}$. They also proved that, assuming ETH, the inapproximability ratio can be improved to $\log^{1/4-\varepsilon} k$ for any constant $\varepsilon > 0$. Very recently, Chalermsook et al. [Cha+17] proved, under GAP-ETH, that k-DOMSET is totally FPT inapproximable, i.e., that no $F(k)$-approximation algorithm for k-DOMSET exists for any computable function F.

Although Chalermsook et al.'s result on the surface seems to settle the parameterized complexity of approximating dominating set, several aspects of the result are somewhat unsatisfactory. First, while GAP-ETH may be plausible, it is quite strong and, in a sense, does much of the work in the proof. Specifically, GAP-ETH itself already gives the gap in hardness of approximation; once there is such a gap, it is not hard[3] to build on it and prove other inapproximability results. As an example, in the analogous situation in NP-hardness of approximation, once one inapproximability result can be shown, others follow via relatively simple gap-preserving reductions (see, e.g., [PY91]). On the other hand, creating a gap in the first place requires the PCP Theorem [AS98; Aro+98], which involves several new technical ideas such as local checkability of codes and proof composition[4]. Hence, it is desirable to bypass GAP-ETH and prove total FPT inapproximability under assumptions that do not involve hardness of approximation in the first place. Drawing a parallel to the theory of NP-hardness of approximation once again, it is imaginable that a success in bypassing GAP-ETH may also reveal a "PCP-like Theorem" for parameterized complexity.

An additional reason one may wish to bypass GAP-ETH for the total FPT inapproximability of k-DOMSET is that the latter is a statement purely about parameterized complexity, so one expects it to hold under a standard parameterized complexity assumption. Given that Chen and

[2] Since its publication, two of the questions, the parameterized complexity of k-BICLIQUE [Lin15] and that of k-EVEN SET (Chapter 10 and [Bon+18]) have been resolved.

[3] One issue grossed over in this discussion is that of *gap amplification*. While GAP-ETH gives some constant gap, Chalermsook et al. still needed to amplify the gap to arrive at total FPT inapproximability. Fortunately, unlike the NP-hardness regime that requires Raz's parallel repetition theorem [Raz98], the gap amplification step in [Cha+17], while non-trivial, only involved relatively simple combinatorial arguments. (See [Cha+17, Theorem 4.3].)

[4] Even in the "combinatorial proof" of the PCP Theorem [Din07], many of these tools still remain in use, specifically in the alphabet reduction step of the proof.

Lin [CL16] proved $\mathrm{W}[1]$-hardness of approximating k-DomSet to within any constant factor, a concrete question here is whether we can show $\mathrm{W}[1]$-hardness of approximation for every function $F(k)$:

Research Question 1. *Can we base the total FPT inapproximability of k-*DomSet *on* $\mathrm{W}[1] \neq$ FPT*?*

Another issue not completely resolved by [Cha+17] is the running time lower bound. While the work gives a quite strong running time lower bound that rules out any $T(k) \cdot n^{o(k)}$-time $F(k)$-approximation algorithm for any computable functions T and F, it is still possible that, say, an $O(n^{0.5k})$-time algorithm can provide a very good (even constant ratio) approximation for k-DomSet. Given the aforementioned $O(n^{k-\varepsilon})$ running time lower bound for exact algorithms of k-DomSet by Pătrascu and Williams [PW10], it seems reasonable to ask whether such a lower bound can also be established for approximation algorithms:

Research Question 2. *Is it hard to approximate k-*DomSet *in $O(n^{k-\varepsilon})$-time?*

This question has perplexed researchers, as even with the running time of, say, $O(n^{k-0.1})$, no $F(k)$-approximation algorithm is known for k-DomSet for any computable function F.

Our Contributions

Our contributions are twofold. Firstly, at a higher level, we prove parameterized inapproximabilty results for k-DomSet, answering the two aforementioned open questions (and more). Secondly, at a lower level, we demonstrate a connection between communication complexity and parameterized inapproximability, allowing us to translate running time lower bounds for parameterized problems into parameterized hardness of approximation. This latter part of the contribution extends ideas from a recent breakthrough of Abboud et al. [ARW17a], who discovered similar connections and used them to establish inapproximability for problems in P. In this subsection, we only focus on the first part of our contributions. The second part will be discussed in detail in Section 6.1.

Parameterized Inapproximability of Dominating Set

Our first batch of results are the inapproximability results for k-DomSet under various standard assumptions in parameterized complexity and fine-grained complexity: $\mathrm{W}[1] \neq$ FPT, ETH, SETH and the k-SUM Hypothesis. First, we show total inapproximability of k-DomSet under $\mathrm{W}[1] \neq$ FPT. In fact, we show an even stronger[5] inapproximation ratio of $(\log n)^{1/\mathrm{poly}(k)}$:

[5] Note that the factor of the form $(\log n)^{1/\mathrm{poly}(k)}$ is stronger than that of the form $F(k)$. To see this, assume that we have an $F(k)$-FPT-approximation algorithm for some computable function F. We can turn this into a $(\log n)^{1/\mathrm{poly}(k)}$-approximation algorithm by first checking which of the two ratios is smaller. If $F(k)$ is smaller, then just run the $F(k)$-FPT-approximation algorithm. Otherwise, use brute force search to solve the problem. Since the latter case can only occur when $n \leqslant \exp(F(k)^{\mathrm{poly}(k)})$, we have that the running time remains FPT.

Theorem 6.1. *Assuming* $\mathrm{W}[1] \neq \mathrm{FPT}$*, no FPT time algorithm can approximate* k-DOMSET *to within a factor of* $(\log n)^{1/\mathit{poly}(k)}$.

Our result above improves upon the constant factor inapproximability result of Chen and Lin [CL16] and resolves the question of whether we can base total FPT inapproximability of k-DOMSET on a purely parameterized complexity assumption. Furthermore, if we are willing to assume the stronger ETH, we can even rule out all $T(k) \cdot n^{o(k)}$-time algorithms:

Theorem 6.2. *Assuming* ETH*, no* $T(k) \cdot n^{o(k)}$*-time algorithm can approximate* k-DOMSET *to within a factor of* $(\log n)^{1/\mathit{poly}(k)}$.

Note that the running time lower bound and approximation ratio ruled out by the above theorem are exactly the same as those of Charlermsook et al.'s result based on GAP-ETH [Cha+17]. In other words, we successfully bypass GAP-ETH from their result completely. Prior to this, the best known ETH-based inapproximability result for k-DOMSET due to Chen and Lin [CL16] ruled out only $(\log^{1/4+\varepsilon} k)$-approximation for $T(k) \cdot n^{o(\sqrt{k})}$-time algorithms.

Assuming the even stronger hypothesis, SETH, we can rule out $O(n^{k-\varepsilon})$-time approximation algorithms for k-DOMSET, matching the running time lower bound from [PW10] while excluding not only exact but also approximation algorithms. We note, however, that the approximation ratio we get in this case is not $(\log n)^{1/\mathrm{poly}(k)}$ anymore, but rather $(\log n)^{1/\mathrm{poly}(k, e(\varepsilon))}$ for some function e, which arises from SETH and the Sparsification Lemma [IPZ01].

Theorem 6.3. *There is a function* $e : \mathbb{R}^+ \to \mathbb{N}$ *such that, assuming* SETH*, for every integer* $k \geqslant 2$ *and for every* $\varepsilon > 0$*, no* $O(n^{k-\varepsilon})$*-time algorithm can approximate* k-DOMSET *to within a factor of* $(\log n)^{1/\mathit{poly}(k, e(\varepsilon))}$.

Finally, to demonstrate the flexibility of our proof techniques (which will be discussed at length in the next section), we apply the framework to the k-SUM Hypothesis (Hypothesis 5) which yields an $n^{\lceil k/2 \rceil - \varepsilon}$ running time lower bound for approximating k-DOMSET as stated below.

Theorem 6.4. *Assuming the* k-SUM *Hypothesis, for every integer* $k \geqslant 3$ *and for every* $\varepsilon > 0$*, no* $O(n^{\lceil k/2 \rceil - \varepsilon})$*-time algorithm can approximate* k-DOMSET *to within a factor of* $(\log n)^{1/\mathit{poly}(k)}$.

We remark here that the k-SUM problem is known to be W[1]-hard [DF95b; ALW14] and our proof of Theorem 6.4 indeed yields an alternative proof of W[1]-hardness of approximating k-DOMSET (Theorem 6.1). Nevertheless, we provide a different self-contained W[1]-hardness reduction directly from CLIQUE since the ideas there are also useful for our ETH-hardness result (Theorem 6.2).

The summary of our results and those from previous works are shown in Table 6.1.

Comparison to Previous Works

In addition to the lower bounds previously mentioned, the parameterized inapproximability of k-DOMSET has also been investigated in several other works [Dow+08; CHK13; HKK13; Bon+15].

Summary of Previous Works and The Results in This Chapter

Complexity Assumption	Inapproximability Ratio	Running Time Lower Bound	Reference
W[1] $\neq$ FPT	Any constant	$T(k) \cdot \text{poly}(n)$	[CL16]
	$(\log n)^{1/\text{poly}(k)}$	$T(k) \cdot \text{poly}(n)$	This chapter
ETH	$(\log k)^{1/4+\varepsilon}$	$T(k) \cdot n^{o(\sqrt{k})}$	[CL16]
	$(\log n)^{1/\text{poly}(k)}$	$T(k) \cdot n^{o(k)}$	This chapter
GAP-ETH	$(\log n)^{1/\text{poly}(k)}$	$T(k) \cdot n^{o(k)}$	[Cha+17]
SETH	Exact	$O(n^{k-\varepsilon})$	[PW10]
	$(\log n)^{1/\text{poly}(k,e(\varepsilon))}$	$O(n^{k-\varepsilon})$	This chapter
k-SUM Hypothesis	$(\log n)^{1/\text{poly}(k)}$	$O(n^{\lceil k/2 \rceil-\varepsilon})$	This chapter

Table 6.1: Summary of our and previous results on k-DOMSET. We only show those whose inapproximability ratios are at least some constant greater than one (i.e., we exclude additive inapproximability results). Here $e : \mathbb{R}^+ \to \mathbb{N}$ is some function, $T : \mathbb{N} \to \mathbb{N}$ can be any computable function and ε can be any positive constant.

Specifically, Downey et al. [Dow+08] showed that obtaining an *additive* constant approximation for k-DOMSET is W[2]-hard. On the other hand, in [HKK13; CHK13], the authors ruled out $(\log k)^{1+\varepsilon}$-approximation in time $\exp(\exp((\log k)^{1+\varepsilon})) \cdot \text{poly}(n)$ for some fixed constant $\varepsilon > 0$ by assuming ETH and the *projection game conjecture* proposed in [Mos12]. Further, Bonnet et al. [Bon+15] ruled out $(1+\varepsilon)$-FPT-approximation, for some fixed constant $\varepsilon > 0$, assuming GAP-ETH[6]. We note that, with the exception of W[2]-hardness results [DF95b; Dow+08], our results subsume all other aforementioned lower bounds regarding k-DOMSET, both for approximation [CHK13; HKK13; Bon+15; CL16; Cha+17] and exact algorithms [Che+06; PW10].

While our techniques will be discussed at a much greater length in the next section (in particular we compare our technique with [ARW17a] in Section 6.1.2), we note that our general approach is to first show inapproximability of a parameterized variant of MAXCOV and then reduce MAXCOV to k-DOMSET. The first step employs the connection between communication complexity and inapproximability of MAXCOV, whereas the second step follows directly from the reduction in [Cha+17] (which is in turn based on [Fei98]). While MAXCOV was not explicitly defined until [Cha+17], its connection to k-DOMSET had been implicitly used both in the work of Pătrascu and Williams [PW10] and that of Chen and Lin [CL16].

From this perspective, the main difference between our work and [PW10; CL16; Cha+17] is

[6] The authors assume the same statement as GAP-ETH (albeit, with imperfect completeness) but have an additional assertion that it is implied by ETH (see Hypothesis 1 in [Bon+15]). It is not hard to see that their assumption can be replaced by GAP-ETH.

the source of hardness for MAXCOV. Recall that Pătrascu and Williams [PW10] ruled out only exact algorithms; in this case, a relatively simple reduction gave hardness for the exact version of MAXCOV. On the other hand, both Chalermsook et al. [Cha+17] and Chen and Lin [CL16] ruled out approximation algorithms, meaning that they needed gaps in their hardness results for MAXCOV. Chalermsook et al. obtained their initial gap from their assumption (GAP-ETH), after which they amplified it to arrive at an arbitrarily large gap for MAXCOV. On the other hand, [CL16] derived their gap from the hardness of approximating Maximum k-Intersection shown in Lin's earlier breakthrough work [Lin15]. Lin's proof [Lin15] made use of certain combinatorial objects called *threshold graphs* to prove inapproximability of Maximum k-Intersection. Unfortunately, this construction was not very flexible, in the sense that it produced MAXCOV instances with parameters that were not sufficient for proving total-FPT inapproximability for k-DOMSET. Moreover, his technique (i.e., threshold graphs) was limited to reductions from k-CLIQUE and was unable to provide a tight running time lower bound under ETH. By resorting to the connection between MAXCOV and communication complexity, we can generate MAXCOV instances with wider ranges of parameters from much more general assumptions, allowing us to overcome the aforementioned barriers.

Comparison to subsequent work of Lin. In [Lin19], the author provides a one-step reduction from an instance of k-set cover[7] on a universe of size $O(\log n)$ (where n is the number of subsets given in the collection) to an instance of k-set cover on a universe of size poly(n) with a gap of $\left(\frac{\log n}{\log\log n}\right)^{1/k}$. The author then uses this gap producing self-reduction to provide running time lower bounds (under different time hypotheses) for approximating k-set cover to a factor of $(1-o(1))\cdot\left(\frac{\log n}{\log\log n}\right)^{1/k}$, improving on the results in Table 6.1 with a better dependence on k in the exponent.

At a high level, the NP-hardness of gap set cover, proceeds by combining the gap label cover instance generated from the PCP theorem with the hypercube partition gadget (described in Section 2.11). In this article, we proceed in a similar way by combining the gap MAXCOV instance generated from the (generalized) Distributed PCP framework (see Figure 6.1), with the hypercube partition gadget. In [Lin19], the author seems to first combine the hypercube partition gadget with a derandomizing combinatorial object called *universal set*, to obtain a gap gadget, and then combines the gap gadget with the input k-set cover instance (on small universe but with no gap) to obtain a gap k-set cover instance.

Finally, we remark that unlike our proof framework, Lin's technique seems to be specifically tailored for the parameterized set cover problem; case in point, his technique does not give the inapproximability of the MAXCOV problem, which we believe to be a canonical parameterized gap problem.

Organization. In the next section, we give an overview of our lower level contributions; for readers interested in the general ideas without too much notational overhead, this section covers most

[7] Recall that there is a pair of polynomial-time L-reductions between the minimum dominating set problem and the set cover problem [Kan92].

of the main ideas from this chapter through a proof sketch of our W[1]-hardness of approximation result (Theorem 6.1). After that, in Section 6.2, we define additional notations and preliminaries needed to formalize our proofs. Section 6.3 provides a definition for Product Space Problems (PSP) and rewrites the hypotheses in these terms. Next, in Section 6.4, we establish a general theorem converting communication protocols to a reduction from PSP to MAXCOV. Sections 6.5, 6.6 and 6.7 provide communication protocols for our problems of interest: Set Disjointness, MULTI-EQUALITY and SUM-ZERO. Section 6.8 highlights the relation between parameterizd perspective of the distributed PCP framework and the hardness in P. Finally, in Section 6.9, we conclude with a few open questions and research directions.

6.1 Connecting Communication Complexity and Parameterized Inapproximability: An Overview

This section is devoted to presenting our connection between communication complexity and parameterized inapproximability (which is one of our main contributions as discussed in the introduction) and serves as an overview for all the proofs in this chapter. As mentioned previously, our discovery of this connection is inspired by the work of Abboud et al. [ARW17a] who showed the connection between the communication protocols and hardness of approximation for problems in P. More specifically, they showed how a Merlin-Arthur protocol for *Set Disjointness* with certain parameters implies the SETH-hardness of approximation for a problem called PCP-*Vectors* and used it as the starting point to prove inapproximability of other problems in P. We extend this idea by identifying a communication problem associated with each of the complexity assumptions (W[1] $\neq$ FPT, ETH, SETH and k-SUM Hypothesis) and then prove a generic theorem that translates communication protocols for these problems to conditional hardness of approximation for a parameterized variant of the *Label Cover* problem called MAXCOV [Cha+17]. Since the hardness of MAXCOV is known to imply the hardness of k-DOMSET [Cha+17] (see Section 2.11), we have arrived at our inapproximability results for k-DOMSET. As the latter part is not the contribution of this chapter, we will focus on explaining the connection between communication complexity and the hardness of approximating MAXCOV.

For concreteness, we focus on the W[1]-hardness result (Theorem 6.1); at the end of this section, we will discuss how this fits into a larger framework that encapsulates other hypotheses too.

For the purpose of our current discussion, it suffices to think of MAXCOV as being parameterized by $|V|$, the number of right super-nodes; from this viewpoint, we would like to show that it is W[1]-hard to approximate MAXCOV to within $(\log n)^{1/\mathrm{poly}(h)}$ factor. For simplicity, we shall be somewhat imprecise in our overview below, all proofs will be formalized later in the chapter.

We reduce from the k-CLIQUE problem, which is well-known to be W[1]-hard [DF95b]. The input to k-CLIQUE is an integer k and a graph which we will call $G' = (V', E')$ to avoid confusion with the label cover graph. The goal is to determine whether G' contains a clique of size k. Recall that, to prove the desired W[1]-hardness, it suffices to provide an *FPT-reduction* from any k-CLIQUE instance (G', k) to approximate MAXCOV instance $\mathcal{L} = (U, V, E, \Sigma_U, \Sigma_V, \{\Pi_e\}_{e \in E})$;

this is an FPT-time reduction such that the new parameter $|V|$ is bounded by a function of the original parameter k. Furthermore, since we want a hardness of approximation result for the latter, we will also show that, when (G', k) is a YES instance of k-CLIQUE, there is a labeling of G that covers all the left super-nodes. On the other hand, when (G', k) is a NO instance of k-CLIQUE, we wish to show that every labeling of G will cover at most $1/(\log n)^{1/\text{poly}(h)}$ fraction of the left super-nodes. If we had such a reduction, then we would have arrived at the total FPT-inapproximability of MAXCOV under $\mathrm{W}[1] \neq \mathrm{FPT}$. But, how would we come up with such a reduction? We will do this by devising a specific kind of protocol for a communication problem!

6.1.1 A Communication Problem for k-CLIQUE

The communication problem related to k-CLIQUE we consider is a multi-party problem where there are $h = \binom{k}{2}$ players, each associated with a two-element subset $\{i, j\}$ of $[k]$. The players cannot communicate with each other. Rather, there is a referee that they can send messages to. Each player $\{i, j\}$ is given two vertices $u_i^{\{i,j\}}$ and $u_j^{\{i,j\}}$ such that $\{u_i^{\{i,j\}}, u_j^{\{i,j\}}\}$ forms an edge in G'. The vertices $u_i^{\{i,j\}}$ and $u_j^{\{i,j\}}$ are allegedly the i-th and j-th vertices of a clique respectively. The goal is to determine whether there is indeed a k-clique in G' such that, for every $\{i, j\} \subseteq [k]$, $u_i^{\{i,j\}}$ and $u_j^{\{i,j\}}$ are the i-th and j-th vertices of the clique.

The communication protocol that we are looking for is a one-round protocol with public randomness and by the end of which the referee is the one who outputs the answer. Specifically, the protocol proceeds as follows. First, the players and the referee together toss r random coins. Then, each player sends an ℓ-bit message to the referee. Finally, the referee decides, based on the messages received and the randomness, either to accept or reject. The protocol is said to have perfect completeness and soundness s if (1) when there is a desired clique, the referee always accepts and (2) when there is no such clique, the referee accepts with probability at most s. The model described here is referred to in the literature as the multi-party *Simultaneous Message Passing* (SMP) model [Yao79; Bab+03; FOZ16]. We refer to a protocol in the SMP model as an SMP protocol.

From Communication Protocol to MAXCOV. Before providing a protocol for the previously described communication problem, let us describe how to turn the protocol into a label cover instance $\mathcal{L} = (U, V, E, \Sigma_U, \Sigma_V, \{\Pi_e\}_{e \in E})$.

- Throughout this chapter, the super graph (U, V, E) is always a complete bipartite graph, i.e., $E = U \times V$. From this point on, we will drop E from the notation for convenience.

- Let $h = \binom{k}{2}$. Again, we associate elements in $[h]$ with two-element subsets of $[k]$. Each right super-node represents Player $\{i, j\}$, i.e., $V = \binom{[k]}{2}$. We view each alphabet for super-node $\{i, j\}$ as a possible input to the player, i.e., an edge $\{u, v\} \in E'$ in the graph G'. Assume w.l.o.g. that $i < j$ and $u < v$. This label represents player $\{i, j\}$ receiving u and v as the alleged i-th and j-th vertices of the clique respectively.

- Let $U = \{0,1\}^r$; that is, we associate left super-node with an r-bit string. For each $\gamma \in \{0,1\}^r$, we view each label for the left super-node γ as an *accepting configuration* on randomness γ, i.e., an h-tuple of ℓ-bit strings $(m_{\{1,2\}}, \dots, m_{\{k-1,k\}}) \in (\{0,1\}^\ell)^h$ such that the referee accepts on randomness γ and message $m_{\{1,2\}}, \dots, m_{\{k-1,k\}}$.
- The constraint Π_e where $e = (\gamma, \{i,j\})$ is defined as follows. Recall that each label a of the right super-node $\{i,j\}$ corresponds to an input that the player $\{i,j\}$ receives in the protocol. For each $\gamma \in \{0,1\}^r$, suppose that the message produced on this randomness by the $\{i,j\}$-th player on the input corresponding to a is $m^{a,\gamma}$. We include in Π_e the pair between a and every accepting configuration on randomness γ that agrees with the message $m^{a,\gamma}$. More specifically, $((m_{\{1,2\}}, \dots, m_{\{k-1,k\}}), a) \in \Pi_e$ iff $m_{\{i,j\}} = m^{a,\gamma}$.

Consider any right labeling σ_V. It is not hard to see that, if we run the protocol where the $\{i,j\}$-th player is given the edge corresponding $\sigma_V(\{i,j\})$ as an input, then the referee accepts a random string $\gamma \in \{0,1\}^r$ if and only if the left super-node γ is covered by the labeling S. In other words, the fraction of the left super-nodes covered by S is exactly equal to the acceptance probability of the protocol. This means that if (G', k) is a YES-instance of k-CLIQUE, then we can select σ_V corresponding to the edges of a k-clique and every left super-node will be covered. On the other hand, if (G', k) is a NO-instance of k-CLIQUE, there is no labeling σ_V that corresponds to a valid k-clique, meaning that every labeling σ_V covers at most s fraction of the edges. Hence, we have indeed arrived at hardness of approximation for MAXCOV. Before we move on to describe the protocol, let us note that the running time of the reduction is poly$(2^{r+\ell h}, |E'|)$, which also gives an upper bound on the size of the label cover instance.

SMP Protocol. Observe first that the trivial protocol, one where every player sends the whole input to the referee, does not suffice for us; this is because the message length ℓ is $\Omega(\log n)$, meaning that the running time of the reduction is $n^{\Omega(h)} = n^{\Omega(k^2)}$ which is not FPT time.

Nevertheless, there still is a simple protocol that does the job. Notice that the input vertices $u_i^{\{i,j\}}$ and $u_j^{\{i,j\}}$ given to Player $\{i,j\}$ are already promised to form an edge. Hence, the only thing the referee needs to check is whether each alleged vertex of the clique sent to different players are the same; namely, he only needs to verify that, for every $i \in [k]$, we have $u_i^{\{i,1\}} = u_i^{\{i,2\}} = \cdots = u_i^{\{i,i-1\}} = u_i^{\{i,i+1\}} = \cdots = u_i^{\{i,k\}}$. In other words, he only needs to check equalities for each of the k unknowns. The equality problem and its variants are extensively studied in communication complexity (see, e.g., [Yao79; KN97]). In our case, the protocol can be easily derived using any error-correcting code. Specifically, for an outcome $\gamma \in \{0,1\}^r$ of the random coin tosses, every Player $\{i,j\}$ encodes each of his input ($u_i^{\{i,j\}}$ and $u_j^{\{i,j\}}$) using a binary error-correcting code and sends only the γ-th bit of each encoded word to the referee. The referee then checks whether, for every $i \in [k]$, the received γ-th bits of the encodings of $u_i^{\{i,1\}}, u_i^{\{i,2\}}, \dots, u_i^{\{i,k\}}$ are equal.

In the protocol described above, the message length ℓ is now only two bits (one bit per vertex), the randomness r used is logarithmic in the block length of the code, the soundness s is one minus the relative distance of the code. If we use a binary code with constant rate and constant relative distance (aka *good codes*), then r will be simply $O(\log \log n)$; this means that the running time of the reduction is $\text{poly}(n, \exp(O(k^2)))$ as desired. While the soundness in this case will just be some constant less than one, we can amplify the soundness by repeating the protocol multiple times independently; this increases the randomness and message length, but it is still not hard to see that, with the right number of repetitions, all parameters lie within the desired ranges. With this, we have completed our sketch for the proof of W[1]-hardness of approximating MAXCOV.

6.1.2 A Framework for Parameterized Hardness of Approximation

The W[1]-hardness proof sketched above is an example of a much more general connection between communication protocol and the hardness of approximating MAXCOV. To gain insight on this, consider any function $f : X_1 \times \cdots \times X_k \to \{0, 1\}$. This function naturally induces both a communication problem and a computational problem. The communication problem for f is one where there are k players, each player i receives an input $a_i \in X_i$, and they together wish to compute $f(a_1, \dots, a_k)$. The computational problem for f, which we call the ***Product Space Problem***[8] of f (abbreviated as PSP(f)), is one where the input consists of subsets $A_1 \subseteq X_1, \dots, A_k \subseteq X_k$ and the goal is to determine whether there exists $(a_1, \dots, a_k) \in A_1 \times \cdots \times A_k$ such that $f(a_1, \dots, a_k) = 1$. The sketch reduction to MAXCOV above in fact not only applies to the specific communication problem of k-CLIQUE: the analogous construction is a generic way to translate any SMP protocol for the communication problem of any function f to a reduction from PSP(f) to MAXCOV. To phrase it somewhat differently, if we have an SMP protocol for f with certain parameters and PSP(f) is hard to solve, then MAXCOV is hard to approximate.

This brings us to the framework we use in this chapter. It consists of only two steps. First, we rewrite the problem in the hypotheses as Product Space Problems of some family of functions $\mathcal{F}$. This gives us the conditional hardness for solving PSP($\mathcal{F}$). Second, we devise an SMP protocol for every function $f \in \mathcal{F}$. Given the connection outlined in the previous paragraph, this automatically yields the parameterized hardness of approximating MAXCOV.

To gain more intuition into the framework, note that in the case of k-CLIQUE above, the function $f \in \mathcal{F}$ we consider is just the function $f : X_{\{1,2\}} \times \cdots \times X_{\{k,k-1\}}$ where each of $X_{\{1,2\}}, \cdots, X_{\{k,k-1\}}$ is a copy of the edge set. The function f "checks" that the edges selected form a clique, i.e., that, for every $i \in [k]$, the alleged i-th vertex of the clique specified in the $\{i, j\}$-coordinate is equal for every $j \neq i$. Since this is a generalization of the equality function, we call such a class of functions "multi-equality". It turns out that 3-SAT can also be written as PSP of multi-equality; each X_i contains assignments to $1/k$ fraction of the clauses and the function f checks that each variable is assigned the same value across all X_i's they appear in. A protocol essentially the same as the one given above also works in this setting and immediately gives our ETH-hardness result (Theorem 6.2)! Unfortunately, this does not suffice for our SETH-hardness. In that case, the function used is the k-way set disjointness; this interpretation of SETH is well-known (see, e.g., [Wil05]) and is also used in [ARW17a]. Lastly, the k-SUM problem is already written in PSP form where f is just the SUM-ZERO function that checks whether the sum of k specified numbers equals to zero.

[8] The naming comes from the product structure of the domain of f.

Let us note that in the actual proof, we have to be more careful about the parameters than in the above sketch. Specifically, the reduction from MAXCOV to k-DOMSET from [Fei98; Cha+17] incurs a blow-up in size that is exponential in terms of the number of vertices in each left super-node (i.e., exponential in $|U_\gamma|$). This means that we need $|U_1|, \dots, |U_r| = o(\log n)$. In the context of communication protocol, this translates to keeping the message length $O(\log \log n)$ where $O(\cdot)$ hides a sufficiently small constant. Nevertheless, for the protocol for k-CLIQUE reduction (and more generally for multi-equality), this does not pose a problem for us since the message length before repetitions is $O(1)$ bits; we can make sure that we apply only $O(\log \log n)$ repetitions to the protocol.

For SUM-ZERO, known protocols either violate the above requirement on message length [Nis94] or use too much randomness [Vio15]. Nonetheless, a simple observation allows us to compose Nisan's protocol [Nis94] and Viola's protocol [Vio15] and arrive at a protocol with the best of both parameters. This new protocol may also be of independent interest beyond the scope of our work.

On the other hand, well-known communication complexity lower bounds on set disjointness [Raz92; KS92; Bar+04] rule out the existence of protocols with parameters we wish to have! [ARW17a] also ran into this issue; in our language, they got around this problem by allowing the referee to receive an advice. This will also be the route we take. Even with advice, however, devising a protocol with the desired parameters is a technically challenging issue. In particular, until very recently, no protocol for set disjointness with $O(\log \log n)$ message length (and $o(n)$ advice length) was known. This was overcome in the work of Rubinstein [Rub18] who used algebraic geometric codes to give such a protocol for the two-player case. We extend his protocol in a straightforward manner to the k-player case; this extension was also suggested to us by Rubinstein [Rub17a].

A diagram illustrating the overview of our approach can be found in Figure 6.1.

Comparison to Abboud et al. The main result of Abboud et al. [ARW17a] is their SETH-hardness of the gap label cover problem which they refer to as the PCP-Vectors problem. In fact, PCP-Vectors is equivalent to MAXCOV when $h = 2$ (i.e., the number of right super nodes is two). However, formulating the label cover problem as MAXCOV instead of PCP-Vectors is beneficial for us, as our goal it to reduce to graph problems.

In their work, they merge the roles of the referee and the first player as it is necessary to achieve the goal of proving hardness of approximation for important problems in P (which are usually defined on one or two sets of vectors). However, by doing this the details of the proof become a little convoluted. On the contrary, our framework with the SMP model is arguably a cleaner framework to work with and it works well for our goal of proving hardness of approximation for parameterized problems.

Finally, we note that our observation that the hardness of approximating MAXCOV can be obtained from any arbitrary hypothesis as long as there is an underlying product structure (as formalized via PSPs) is a new contribution of this chapter.

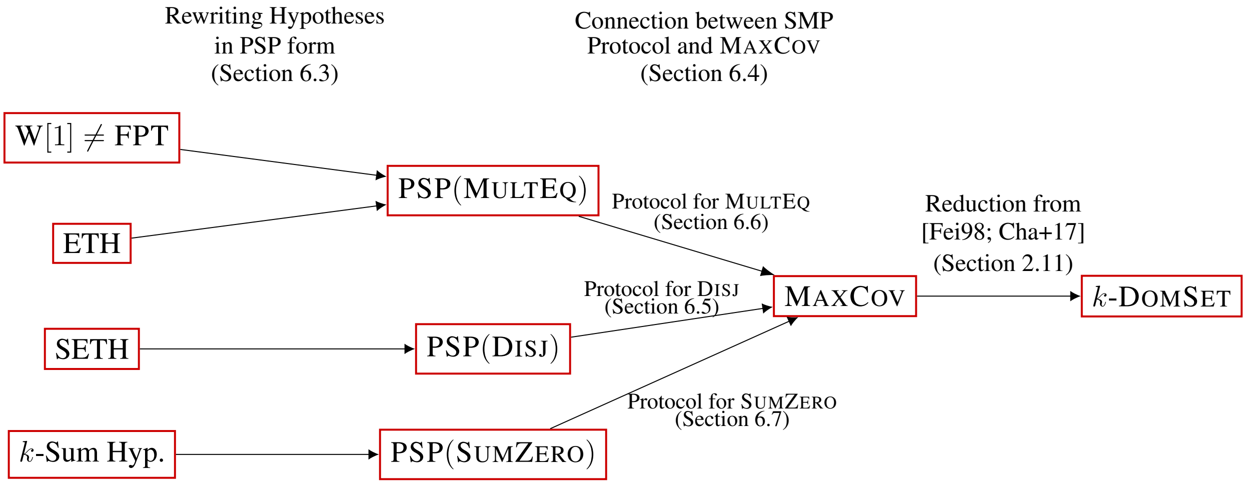

Rewriting Hypotheses in PSP form (Section 6.3)
Connection between SMP Protocol and MAXCOV (Section 6.4)
W[1] ≠ FPT
ETH
SETH
k-Sum Hyp.
PSP(MULTEQ)
PSP(DISJ)
PSP(SUMZERO)
Protocol for MULTEQ (Section 6.6)
Protocol for DISJ (Section 6.5)
Protocol for SUMZERO (Section 6.7)
MAXCOV
Reduction from [Fei98; Cha+17] (Section 2.11)
k-DOMSET

Figure 6.1: Overview of Our Framework. The first step is to reformulate each hypothesis in terms of hardness of a PSP problem, which is done in Section 6.3. Using the connection between SMP protocols and MAXCOV outlined earlier (and formalized in Section 6.4), our task is now to devise SMP protocols with certain parameters for the corresponding communication problems; these are taken care of in Sections 6.5, 6.6 and 6.7. For completeness, the final reduction from MAXCOV to k-DOMSET which was shown in [Cha+17] is included in Section 2.11.

6.2 Additional Preliminaries

We need error-correcting codes with specific properties, which are described below.

6.2.1 Good Codes

In the construction of our communication protocol in Section 6.6, we require our codes to have constant rate and constant relative distance (referred to as *good codes*). It is not hard to see that random codes, ones where each codeword $C(x)$ is randomly selected from Σ^d independently from each other, satisfy these properties. For binary codes (i.e., $|\Sigma| = 2$), one can explicitly construct such codes using expander graphs (so called *Expander Codes* [SS96]); alternatively *Justesen Code* [Jus72] also have the same property (see Appendix E.1.2.5 from [Gol08] for an excellent exposition).

Fact 6.5. *For some absolute constant $\delta, \rho > 0$, there exists a family of codes $\mathfrak{C} := \{C_m : \{0,1\}^m \to \{0,1\}^{d(m)}\}_{m\in\mathbb{N}}$ such that for every $m \in \mathbb{N}$ the rate of C_m is at least ρ and the relative distance of C_m is at least δ. Moreover, any codeword of C_m can be computed in time poly(m).*

6.2.2 Algebraic Geometric Codes

In the construction of our communication protocol in Section 6.5, we require our codes to have some special algebraic properties which have been shown to be present in algebraic geometric codes [GS96]. First, we will introduce a couple of additional definitions.

Definition 6.6 (Systematicity)**.** *Given $s \in \mathbb{N}$, a code $C : \Sigma^m \to \Sigma^d$ is s-*systematic *if there exists a size-s subset of $[d]$, which for convenience we identify with $[s]$, such that for every $x \in \Sigma^s$ there exists $w \in \Sigma^m$ in which $x = C(w)\mid_{[s]}$.*

Definition 6.7 (Degree-t Closure)**.** *Let Σ be a finite field. Given two codes $C : \Sigma^m \to \Sigma^d, C' : \Sigma^{m'} \to \Sigma^d$ and positive integer t, we say that C' is a degree-t closure of C if, for every $w_1, \dots, w_r \in \Sigma^m$ and $P \in \mathbb{F}[X_1, \dots, X_r]$ of total degree at most t, it holds that $\omega := P(C(w_1), \dots, C(w_r))$ is in the range of C', where $\omega \in \Sigma^d$ is defined coordinate-wise by the equation $\omega_i := P(C(w_1)_i, \dots, C(w_r)_i)$.*

Below we provide a self-contained statement of the result we rely on in Section 6.5; it follows from Theorem 7 of [Shu+01], which gives an efficient construction of the algebraic geometric codes based on [GS96]'s explicit towers of function fields.

Theorem 6.8 ([GS96; Shu+01])**.** *There are two polynomial functions $\hat{r}, \hat{q} : \mathbb{N} \to \mathbb{N}$ such that for every $k \in \mathbb{N}$ and any prime $q > \hat{q}(k)$, there are two code families $\mathfrak{A} = \{A_n\}_{n\in\mathbb{N}}$, $\mathfrak{B} = \{B_n\}_{n\in\mathbb{N}}$ such that the following holds for all $n \in \mathbb{N}$,*

- *A_n and B_n are n-systematic code with alphabet $\mathbb{F}_{q^2}$,*
- *A_n and B_n have block length less than $n \cdot \hat{r}(k)$.*
- *B_n has relative distance $\geqslant 1/2$,*
- *B_n is a degree-k closure of A_n, and,*
- *Any codeword in A_n or B_n can be computed in poly(n) time .*

We remark here that variants of the above theorem have previously found applications in the construction of special kinds of PCPs [Ben+16a; Ben+16b]. In these works, the theorems are also stated in a language similar to Theorem 6.8 above.

6.3 Product Space Problems and Popular Hypotheses

In this section, we define a class of computational problems called Product Space Problems (PSP). As the name suggests, a problem in this class is defined on a class of functions whose domain is a k-ary Cartesian Product, i.e., $f : X_1 \times \cdots \times X_k \to \{0, 1\}$. The input of the problem are subsets[9] $A_1 \subseteq X_1, \dots, A_k \subseteq X_k$, and the goal is to determine whether there exists $(a_1, \dots, a_k) \in A_1 \times \cdots \times A_k$ such that $f(a_1, \dots, a_k) = 1$. The size of the problem is determined by $\max\limits_{i\in[k]} |A_i|$. A formal definition of PSP can be found below.

[9] Each A_i will be explicitly given as part of the input through the elements that it contains.

Definition 6.9 (Product Space Problem). *Let $m : \mathbb{N} \times \mathbb{N} \to \mathbb{N}$ be any function and $\mathcal{F} := \{f_{N,k} : \{0,1\}^{m(N,k)\times k} \to \{0,1\}\}_{N,k\in\mathbb{N}}$ be a family of Boolean functions indexed by N and k. For each $k \in \mathbb{N}$, the* product space problem $\mathrm{PSP}(k, \mathcal{F})$ *of order N is defined as follows: given k subsets $A_1, \ldots, A_k$ of $\{0,1\}^{m(N,k)}$ each of cardinality at most N as input, determine if there exists $(a_1, \ldots, a_k) \in A_1 \times \cdots \times A_k$ such that $f_{N,k}(a_1, \ldots, a_k) = 1$. We use the following shorthand* $\mathrm{PSP}(k, \mathcal{F}, N)$ *to describe* $\mathrm{PSP}(k, \mathcal{F})$ *of order N.*

In all the PSPs considered in this chapter, the input length $m(N, k)$ is always at most $\mathrm{poly}(k) \cdot \log N$ and $f_{N,k}$ is always computable in time $\mathrm{poly}(m(N, k))$. In such a case, there is a trivial $N^{k+o_k(1)}$-time algorithm to solve $\mathrm{PSP}(k, \mathcal{F}, N)$: enumerating all $(a_1, \ldots, a_k) \in A_1 \times \cdots \times A_k$ and check whether $f_{N,k}(a_1, \ldots, a_k) = 1$. The rest of this section is devoted to rephrasing the hypotheses (SETH, ETH, W[1] $\neq$ FPT and the k-SUM Hypothesis) in terms of lower bounds for PSPs. The function families $\mathcal{F}$'s, and running time lower bounds will depend on the hypotheses. For example, SETH will corresponds to set disjointness whereas W[1] $\neq$ FPT will correspond to a generalization of equality called "multi-equality"; the former will give an $N^{k(1-o(1))}$ running time lower bound whereas the latter only rules out FPT time algorithms.

We would like to remark that the class of problems called 'locally-characterizable sets' introduced by Goldreich and Rothblum [GR18] are closely related to PSPs. Elaborating, we may interpret locally-characterizable sets as the negation of PSPs, i.e., for any $\mathrm{PSP}(k, \mathcal{F}, N)$, we may define the corresponding locally-characterizable set $\mathcal{S}$ as follows:

$$\mathcal{S} = \{(A_1, \ldots, A_k) \mid \text{for all } (a_1, \ldots, a_k) \in A_1 \times \cdots \times A_k \text{ we have } f_{N,k}(a_1, \ldots, a_k) = 0\}.$$

Finally, we note that the class of problems called 'counting local patterns' introduced in [GR18] are the counting counterpart of PSPs, i.e., for any instance $(A_1, \ldots, A_k)$ of $\mathrm{PSP}(k, \mathcal{F}, N)$, we may define the corresponding counting local pattern solution to be the number of distinct $(a_1, \ldots, a_k) \in A_1 \times \cdots \times A_k$ such that $f_{N,k}(a_1, \ldots, a_k) = 1$.

6.3.1 k-SUM Hypothesis

To familiarize the readers with our notations, we will start with the k-SUM Hypothesis, which is readily in the PSP form. Namely, the functions in the family are the SUM-ZERO functions that checks if the sum of k integers is zero:

Definition 6.10 (SUM-ZERO)**.** *Let $k, m \in \mathbb{N}$.* $\textsc{SumZero}_{m,k} : (\{0,1\}^m)^k \to \{0,1\}$ *is defined by*

$$\textsc{SumZero}_{m,k}(x_1, \ldots, x_k) = \begin{cases} 1 \text{ if } \sum\limits_{i\in[k]} x_i = 0, \\ 0 \text{ otherwise}, \end{cases}$$

where we think of each x_i as a number in $[-2^{m-1}, 2^{m-1} - 1]$, and the addition is over $\mathbb{Z}$.

The function family $\mathcal{F}^{\textsc{SumZero}}$ can now be defined as follows.

Definition 6.11 (Sum-Zero Function Family)**.** *Let $m : \mathbb{N} \times \mathbb{N} \to \mathbb{N}$ be a function defined by $m(N, k) = 2k\lceil \log N \rceil$. $\mathcal{F}^{\textsc{SumZero}}$ is defined as $\{\textsc{SumZero}_{m(N,k),k}\}_{N\in\mathbb{N},k\in N}$.*

The following proposition is immediate from the definition of the k-SUM Hypothesis.

Proposition 6.12. *Assuming the k-SUM Hypothesis, for every integer $k \geqslant 3$ and every $\varepsilon > 0$, no $O(N^{\lceil k/2 \rceil - \varepsilon})$-time algorithm can solve $\mathrm{PSP}(k, \mathcal{F}^{\textsc{SumZero}}, N)$ for all $N \in \mathbb{N}$.*

6.3.2 Set Disjointness and SETH

We recall the k-way disjointness function, which has been studied extensively in literature (see, e.g., [LS09] and references therein).

Definition 6.13 (Set Disjointness)**.** *Let $k, m \in \mathbb{N}$. $\textsc{Disj}_{m,k} : (\{0,1\}^m)^k \to \{0,1\}$ is defined by*

$$\textsc{Disj}_{m,k}(x_1, \ldots, x_k) = \neg \left(\bigvee_{i\in[m]} \left(\bigwedge_{j\in[k]} (x_j)_i \right) \right).$$

The function family $\mathcal{F}_c^{\textsc{Disj}}$ can now be defined as follows.

Definition 6.14 (Set Disjointness Function Family)**.** *For every $c \in \mathbb{N}$, let $m_c : \mathbb{N} \times \mathbb{N} \to \mathbb{N}$ be a function defined by $m_c(N, k) = c\lceil k \log N \rceil$. $\mathcal{F}_c^{\textsc{Disj}}$ is defined as $\{\textsc{Disj}_{m_c(N,k),k}\}_{N\in\mathbb{N},k\in N}$.*

We have the following proposition which follows easily from the definition of SETH and its well-known connection to the Orthogonal Vectors Hypothesis [Wil05].

Proposition 6.15. *Let $k \in \mathbb{N}$ such that $k > 1$. Assuming SETH, for every $\varepsilon > 0$ there exists $c := c_\varepsilon \in \mathbb{N}$ such that no $O(N^{k(1-\varepsilon)})$-time algorithm can solve $\mathrm{PSP}(k, \mathcal{F}_c^{\textsc{Disj}}, N)$ for all $N \in \mathbb{N}$.*

Proof. Fix $\varepsilon > 0$ and $k > 1$. By SETH, there exists $w := w(\varepsilon) \in \mathbb{N}$ and $c := c(\varepsilon) \in \mathbb{N}$ such that no algorithm can solve w-SAT in $O(2^{(1-\varepsilon)n})$ time where n is the number of variables and $m \leqslant cn$ is the number of clauses. For every w-SAT formula ϕ, we will build $A_1^\phi, \ldots, A_k^\phi \subseteq \{0,1\}^m$ each of cardinality $N := 2^{n/k}$ such that there exists $(a_1, \ldots, a_k) \in A_1^\phi \times \cdots \times A_k^\phi$ such that $\textsc{Disj}_{m,k}(a_1, \ldots, a_k) = 1$ if and only if ϕ is satisfiable. Thus, if there was an $O(N^{k(1-\varepsilon)})$-time algorithm that can solve $\mathrm{PSP}(k, \mathcal{F}_c^{\textsc{Disj}}, N)$ for all $N \in \mathbb{N}$, then it would violate SETH.

All that remains is to show the construction of $A_1^\phi, \ldots, A_k^\phi$ from ϕ. Fix $i \in [k]$. For every partial assignment σ to the variables $x_{(i-1)*(n/k)+1}, \ldots, x_{i*(n/k)}$ we build an m-bit vector $a_\sigma \in A_i^\phi$ as follows: $\forall j \in [m]$, we have $a_\sigma(j) = 0$ is σ satisfies the j^{th} clause, and $a_\sigma(j) = 1$ otherwise (i.e., the clause is not satisfied, or its satisfiability is indeterminate). It is easy to verify that there exists $(a_1, \ldots, a_k) \in A_1^\phi \times \cdots \times A_k^\phi$ such that $\textsc{Disj}_{m,k}(a_1, \ldots, a_k) = 1$ if and only if ϕ is satisfiable. □

We remark that we can prove a similar statement as that of Proposition 6.15 for ETH: assuming ETH, there exists k_0 such that for every $k > k_0$ there exists $c := c_{k_0} \in \mathbb{N}$ such that no $O(N^{o(k)})$-time algorithm can solve $\mathrm{PSP}(k, \mathcal{F}_c^{\textsc{Disj}}, N)$ for all $N \in \mathbb{N}$. However, instead of associating ETH with DISJ, we will associate with the Boolean function MULTEQ (which will be defined in the next subsection) and its corresponding PSP. This is because, associating ETH with MULTEQ provides a more elementary proof of Theorem 6.2 (in particular we will not need to use algebraic geometric codes – which are essentially inevitable if we associate ETH with DISJ).

6.3.3 W$[1] \neq$ FPT Hypothesis and ETH

Again, we recall the k-way EQUALITY function which has been studied extensively in literature (see, e.g., [AMS12; ABC09; CRR14; CMY08; LV11; PVZ12] and references therein).

Definition 6.16 (EQUALITY). *Let $k, m \in \mathbb{N}$. $\text{EQ}_{m,k} : (\{0,1\}^m)^k \to \{0,1\}$ is defined by*

$$\text{EQ}_{m,k}(x_1, \dots, x_k) = \bigwedge_{i,j \in [k]} (x_i = x_j)$$

where $x_i = x_j$ is a shorthand for $\bigwedge_{p \in [m]} (x_i)_p = (x_j)_p$.

Unfortunately, the PSP associated with EQ is in fact not hard: given sets $A_1, \dots, A_k$, it is easy to find whether they share an element by just sorting the combined list of $A_1 \cup \cdots \cup A_k$. Hence, we will need a generalization of the equality function to state our hard problem. Before we do so, let us first state an intermediate helper function, which is a variant of the usual equality function where some of the k inputs may be designed as "null" and the function only checks the equality over the non-null inputs. We call this function the SELECTIVE-EQUALITY (SELEQ) function. For notational convenience, in the definition below, each of the k inputs is now viewed as $(x_i, y_i) \in \{0,1\}^{m-1} \times \{\perp, \top\}$; if $y_i = \perp$, then (x_i, y_i) represents the "null" input.

Definition 6.17 (SELECTIVE-EQUALITY). *Let $k, m \in \mathbb{N}$. $\text{SELEQ}_{m,k} : (\{0,1\}^{m-1} \times \{\perp, \top\})^k \to \{0,1\}$ is defined by*

$$\text{SELEQ}_{m,k}((x_1, y_1), \dots, (x_k, y_k)) = \bigwedge_{i,j \in [k]} ((y_i = \perp) \vee (y_j = \perp) \vee (x_i = x_j)) .$$

Next, we introduce the variant of EQ whose associated PSP is hard under W$[1] \neq$ FPT and ETH. In the settings of both EQUALITY and SELECTIVE-EQUALITY defined above, there is only one unknown that is given in each of the k inputs $a_1 \in A_1, \dots, a_k \in A_k$ and the functions check whether they are equal. The following function, which we name MULTI-EQUALITY, is the t-unknown version of SELECTIVE-EQUALITY. Specifically, the i^{th} part of the input is now a tuple $((x_{i,1}, y_{i,1}), \dots, (x_{i,t}, y_{i,t}))$ where $x_{i,1}, \dots, x_{i,t}$ are bit strings representing the supposed values of the t unknowns while, similar to SELECTIVE-EQUALITY, each $y_{i,q} \in \{\perp, \top\}$ is a symbol indicating whether $(x_{i,q}, y_{i,q})$ is the "null" input. Below is the formal definition of MULTEQ; note that for convenience, we use $(x_{i,q}, y_{i,q})_{q \in [t]}$ as a shorthand for $((x_{i,1}, y_{i,1}), \dots, (x_{i,t}, y_{i,t}))$, i.e., the i^{th} part of the input.

Definition 6.18 (MULTI-EQUALITY)**.** *Let $k, t \in \mathbb{N}$ and let $m \in \mathbb{N}$ be any positive integer such that m is divisible by t. Let $m' = m/t$.* $\textsc{MultEq}_{m,k,t} : ((\{0,1\}^{m'-1} \times \{\bot, \top\})^t)^k \to \{0,1\}$ *is defined by*

$$\textsc{MultEq}_{m,k,t}((x_{1,q}, y_{1,q})_{q\in[t]}, \dots, (x_{k,q}, y_{k,q})_{q\in[t]}) = \bigwedge_{q\in[t]} \textsc{SelEq}_{m',k}((x_{1,q}, y_{1,q}), \dots, (x_{k,q}, y_{k,q})).$$

Next, we define the family $\mathcal{F}^{\textsc{MultEq}}$; note that in the definition below, we simply choose $t(k)$, the number of unknowns, to be $k + \binom{k}{2} + \binom{k}{3}$. As we will see later, this is needed for ETH-hardness. For W[1]-hardness, it suffices to use a smaller number of variables. However, we choose to define $t(k)$ in such a way so that we can conveniently use one family for both ETH and W[1]-hardness.

Definition 6.19. *Let $t : \mathbb{N} \to \mathbb{N}$ be defined by $t(k) = k + \binom{k}{2} + \binom{k}{3}$. Let $m : \mathbb{N} \times \mathbb{N} \to \mathbb{N}$ be defined by $m(N, k) = t(k)\,(1 + k\lceil \log N \rceil)$. We define $\mathcal{F}^{\textsc{MultEq}}$ as* $\{\textsc{MultEq}_{m(N,k),k,t(k)}\}_{N\in\mathbb{N},k\in\mathbb{N}}$.

We next show a reduction from k-CLIQUE to PSP$(k', \mathcal{F}^{\textsc{MultEq}})$ where $k' = \binom{k}{2}$. The overall idea of the reduction is simple. First, we associate the integers in $[k']$ naturally with the elements of $\binom{[k]}{2}$. We then create the sets $\left(A_{\{i,j\}}\right)_{\{i,j\}\subseteq[k], i\neq j}$ in such a way that each element of the set $A_{\{i,j\}}$ corresponds to picking an edge between the i-th and the j-th vertices in the supposed k-clique. Then, MULTEQ is used to check that these edges are consistent, i.e., that, for every $i \in [k]$, $a_{\{i,j\}}$ and $a_{\{i,j'\}}$ pick the same vertex to be the i^{th} vertex in the clique for all $j, j' \in [k] \setminus \{i\}$. This idea is formalized in the following proposition and its proof.

Proposition 6.20. *Let $k \in \mathbb{N}$ and $k' = \binom{k}{2}$. There exists a poly(N, k)-time reduction from any instance (G, k) of* CLIQUE *to an instance $(A_1, \dots, A_{k'})$ of the* PSP$(k', \mathcal{F}^{\textsc{MultEq}}, N')$ *where N denotes the number of vertices of G and $N' = \binom{N}{2}$.*

Proof. Given a CLIQUE instance[10] (G, k), the reduction proceeds as follows. For convenience, we assume that the vertex set $V(G)$ is $[N]$. Furthermore, we associate the elements of $[k']$ naturally with the elements of $\binom{[k]}{2}$. For the sake of conciseness, we sometimes abuse notation and think of $\{i, j\}$ as an ordered pair (i, j) where $i < j$. For every $\{i, j\} \in \binom{[k]}{2}$ such that $i < j$, the set $A_{\{i,j\}}$ contains one element $a^{\{u,v\}}_{\{i,j\}} = \left(a^{\{u,v\}}_{\{i,j\},1}, \dots, a^{\{u,v\}}_{\{i,j\},t(k')}\right)$ for each edge $\{u, v\} \in E(G)$ such that $u < v$, where

$$a^{\{u,v\}}_{\{i,j\},q} = \begin{cases} (u, \top) & \text{if } q = i, \\ (v, \top) & \text{if } q = j, \\ (0, \bot) & \text{otherwise.} \end{cases}$$

Note that in the definition above, we view u, v and 0 as $({}^{m(N',k')}/_{t(k')} - 1)$-bit strings, where $m : \mathbb{N} \times \mathbb{N} \to \mathbb{N}$ is as in Definition 6.19. Also note that each set $A_{\{i,j\}}$ has size at most $\binom{N}{2} = N'$,

[10] We assume without loss of generality that G does not contain any self-loop.

meaning that $(A_{\{i,j\}})_{\{i,j\} \subseteq [k]}$ is indeed a valid instance of PSP$(k', \mathcal{F}^{\text{MULTEQ}}, N')$. For brevity, below we will use f as a shorthand for MULTEQ$_{m(N',k'),k',t(k')}$.

($\Rightarrow$) Suppose that (G, k) is a YES instance for CLIQUE, i.e., there exists a k-clique $\{u_1, \dots, u_k\}$ in G. Assume without loss of generality that $u_1 < \cdots < u_k$. We claim that,

$$f\left(\left(a^{\{u_i,u_j\}}_{\{i,j\}}\right)_{i,j \in [k], i<j}\right) = 1.$$

To see that this is the case, observe that for every $q \in [t(k')]$ and for every $\{i, j\} \subseteq [k]$ such that $i < j$, we have either $a^{\{u_i,u_j\}}_{\{i,j\},q} = (0, \perp)$ or $a^{\{u_i,u_j\}}_{\{i,j\},q} = (u_q, \top)$. This means that, SELEQ $\left(\left(a^{\{u_i,u_j\}}_{\{i,j\},q}\right)_{i,j \in [k], i<j}\right) = 1$ for every $q \in [t(k')]$.

($\Leftarrow$) Suppose that $(A_{\{i,j\}})_{\{i,j\} \subseteq [k]}$ is a YES instance for PSP$(k', \mathcal{F}^{\text{MULTEQ}}, N')$, i.e., there exists $a^*_{\{i,j\}} \in A_{\{i,j\}}$ for every $\{i, j\} \subseteq [k]$ such that $f((a^*_{\{i,j\}})_{\{i,j\} \subseteq [k]}) = 1$. Suppose that $a^*_{\{i,j\}} = (x^*_{\{i,j\},1}, y^*_{\{i,j\},1}, \dots, x^*_{\{i,j\},t(k')}, y^*_{\{i,j\},t(k')})$. From this solution $\{a^*_{\{i,j\}}\}_{\{i,j\} \subseteq [k]}$, we can recover the k-clique as follows. For each $i \in [k]$, pick an arbitrary $j(i) \in [k]$ that is not equal to i. Let u_i be $x^*_{\{i,j\},i}$. We claim that $u_1, \dots, u_k$ forms a k-clique in G. To show this, it suffices to argue that, for every distinct $i, i' \in [k]$, there is an edge between u_i and $u_{i'}$ in G. To see that this holds, consider $a^*_{\{i,i'\}}$. Since $y^*_{\{i,i'\},i} = y^*_{\{i,j(i)\},i} = \top$, we have $x^*_{\{i,i'\},i} = x^*_{\{i,j(i)\},i} = u_i$. Similarly, we have $x^*_{\{i,i'\},i'} = u_{i'}$. Since $a^*_{\{i,i'\}} \in A_{\{i,i'\}}$ and from how the set $A_{\{i,i'\}}$ is defined, we have $\{u_i, u_{i'}\} \in E(G)$, which concludes our proof. □

Lemma 6.21. *Assuming* W[1] $\neq$ FPT*, for any computable function $T : \mathbb{N} \to \mathbb{N}$, there is no $T(k) \cdot$ poly(N) time algorithm that can solve* PSP$(k, \mathcal{F}^{\text{MULTEQ}}, N)$ *for every $N, k \in \mathbb{N}$.*

Proof. Suppose for the sake of contradiction that, for some computable function $T : \mathbb{N} \to \mathbb{N}$, there is a $T(k) \cdot \text{poly}(N)$ time algorithm $\mathcal{A}$ that can solve PSP$(k, \mathcal{F}^{\text{MULTEQ}}, N)$ for every $N, k \in \mathbb{N}$. We will show that this algorithm can also be used to solve k-CLIQUE parameterized by k in FPT time.

Given an instance (G, k) of k-CLIQUE, we first run the reduction from Proposition 6.20 to produce an instance $(A_1, \dots, A_{k'})$ of PSP$(k', \mathcal{F}^{\text{MULTEQ}}, N')$ in poly(N, k) time where $N = |V(G)|$, $N' = \binom{N}{2}$ and $k' = \binom{k}{2}$. We then run $\mathcal{A}$ on $(A_1, \dots, A_{k'})$, which takes time $T(k') \cdot \text{poly}(N')$. This means that we can also solve our k-CLIQUE instance (G, k) in time poly$(N, k) + T(k') \cdot \text{poly}(N') = \text{poly}(N, k) + T\left(\binom{k}{2}\right) \cdot \text{poly}(N)$, which is FPT time. Since k-CLIQUE is W[1]-complete, this contradicts with W[1] $\neq$ FPT. □

Next, we will prove ETH-hardness of PSP$(k, \mathcal{F}^{\text{MULTEQ}})$. Specifically, we will reduce a 3-SAT instance ϕ where each variable appears in at most three clauses to an instance of PSP$(k, \mathcal{F}^{\text{MULTEQ}}, N)$ where $N = 2^{O(n/k)}$ and n denotes the number of variables in ϕ. The overall idea is to partition the set of clauses into k parts of equal size and use each element in A_j to represent a partial assignment that satisfies all the clauses in the j^{th} partition. This indeed means that each group has size $2^{O(n/k)}$ as intended. However, choosing the unknowns are not as straightforward as in the reduction from k-CLIQUE above; in particular, if we view each variable by itself as an unknown, then we would have n unknowns, which is much more than the designated $t(k) = k + \binom{k}{2} + \binom{k}{3}$ unknowns! This is where we use the fact that each variable appears in at most three clauses: we group the variables of ϕ together based on which partitions they appear in and view each group as a single variable. Since each variable appears in at most three clauses, the number of ways they can appear in the k partitions is $k + \binom{k}{2} + \binom{k}{3}$ which is indeed equal to $t(k)$. The ideas are formalized below.

Proposition 6.22. *Let $k \in \mathbb{N}$. There exists a poly(N, k)-time reduction from any instance ϕ of 3-*SAT *such that each variable appears in at most three clauses in an instance $(A_1, \dots, A_k)$ of the* PSP$(k, \mathcal{F}^{\text{MULTEQ}}, N)$ *where $N = 2^{3\lceil m/k \rceil}$ and m denotes the number of clauses in ϕ.*

Proof. Given a 3-SAT formula ϕ such that each variable appears in at most three clauses. Let the variable set of ϕ be $\mathcal{Z} = \{z_1, \dots, z_n\}$ and the clauses of ϕ be $\mathcal{C} = \{C_1, \dots, C_m\}$. Then for every $k \in \mathbb{N}$, we produce an instance $(A_1, \dots, A_k)$ of PSP$(k, \mathcal{F}^{\text{MULTEQ}}, N)$ where $N = 2^{3\lceil m/k \rceil}$ as follows.

First, we partition the clause set $\mathcal{C}$ into k parts $\mathcal{C}_1, \dots, \mathcal{C}_k$ each of size at most $\lceil m/k \rceil$. For each variable z_i, let S_i denote $\{j \in [k] \mid \exists C_h \in \mathcal{C}_j \text{ such that } z_i \in C_h \text{ or } \overline{z}_i \in C_h\}$. Since every z_i appears in at most three clauses, we have $S_i \in \binom{[k]}{\leqslant 3}$. For each $S \in \binom{[k]}{\leqslant 3}$, let $\nu(S)$ denote the set of all variables z_i's such that $S_i = S$ (i.e. S is exactly equal to the set of all partitions that z_i appears in). The general idea of the reduction is that we will view a partial assignment to the variables in $\nu(S)$ as an unknown for MULTEQ; let us call this unknown X_S (hence there are $k + \binom{k}{2} + \binom{k}{3} = t(k)$ unknowns). For each $j \in [k]$, A_j contains one element for each partial assignment to the variables that appear in the clauses in $\mathcal{C}_j$ and that satisfies all the clauses in $\mathcal{C}_j$. Such a partial assignment specifies $\left(1 + k + \binom{k}{2}\right)$ unknowns: all the X_S such that $j \in S$. The MULTEQ function is then used to check the consistency between the partial assignments to the variables from different A_j's.

To formalize this intuition, we first define more notations. Let $\widetilde{m} = 3k\lceil m/k \rceil$. For every subsets $T \subseteq T' \subseteq \mathcal{Z}$ and every partial assignment $\alpha : T' \to \{0, 1\}$, the restriction of α to T, denoted by $\alpha|_T$ is the function from T to $\{0, 1\}$ where $\alpha|_T(z) = \alpha(z)$ for every $z \in T$. Furthermore, we define the operator $\text{ext}(\alpha)$, which "extends" α to $\widetilde{m}$ bits, i.e., the i-th bit of $\text{ext}(\alpha)$ is $\alpha(z_i)$ if $z_i \in T$ and is zero otherwise. Finally, we use $\text{var}(\mathcal{C}_j)$ to denote the set of all variables that appear in at least one of the clauses from $\mathcal{C}_j$, i.e., $\text{var}(\mathcal{C}_j) = \bigcup_{C \in \mathcal{C}_j} \text{var}(C)$ where $\text{var}(C)$ denotes $\{z_i \in \mathcal{Z} \mid z_i \in C \text{ or } \overline{z}_i \in C\}$.

Now, since our $t(k)$ is exactly $\left|\binom{[k]}{\leqslant 3}\right|$, we can associate each element of $[t]$ with a subset $S \in \binom{[k]}{\leqslant 3}$. Specifically, for each partial assignment $\alpha : \text{var}(\mathcal{C}_j) \to \{0, 1\}$ such that α satisfies all the clauses in $\mathcal{C}_j$, the set A_j contains an element $a_j^\alpha = (a_{j,S}^\alpha)_{S \in \binom{[k]}{\leqslant 3}}$ where, for every $S \in \binom{[k]}{\leqslant 3}$,

$$a_{j,S}^\alpha = \begin{cases} \left(\text{ext}\left(\alpha|_{\nu(S)}\right), \top\right) & \text{if } j \in S, \\ (0^{\widetilde{m}}, \perp) & \text{otherwise.} \end{cases}$$

Fix $S \in \binom{[k]}{\leqslant 3}$. For every $j \in S$, observe that $\nu(S) \subseteq \text{var}(\mathcal{C}_j)$. Moreover, since each $\mathcal{C}_j$ contains at most $\lceil m/k \rceil$ clauses, there are at most $3\lceil m/k \rceil$ variables in $\text{var}(\mathcal{C}_j)$. This means that A_j has size at most $2^{3\lceil m/k \rceil}$. Hence, $(A_1, \dots, A_k)$ is indeed a valid instance of PSP $(k, \mathcal{F}^{\text{MULTEQ}}, N)$ where $N = 2^{3\lceil m/k \rceil}$. For brevity, below we will use f as a shorthand for $\text{MULTEQ}_{m(N,k),k,t(k)}$.

($\Rightarrow$) Suppose that ϕ is satisfiable. Let $\alpha : \mathcal{C} \to \{0,1\}$ be an assignment that satisfies all the clauses. Let $a^*_j = a_j^{\alpha|_{\mathrm{var}(\mathcal{C}_j)}} \in A_j$ for every $j \in [k]$. Observe that, for every $S \in \binom{[k]}{\leqslant 3}$ and every $j \in [k]$, we either have $a^*_{j,S} = (0^{\widetilde{m}}, \perp)$ or $a^*_{j,S} = (\mathrm{ext}(\alpha|_{\nu(S)}), \top)$. This indeed implies that $f(a^*_1, \dots, a^*_k) = 1$.

($\Leftarrow$) Suppose that there exists $(a_1^{\alpha_1}, \dots, a_k^{\alpha_k}) \in A_1 \times \cdots \times A_k$ such that $f(a_1^{\alpha_1}, \dots, a_k^{\alpha_k}) = 1$. We construct an assignment $\alpha : \mathcal{Z} \to \{0,1\}$ as follows. For each $i \in [n]$, pick an arbitrary $j(i) \in [k]$ such that $z_i \in \mathrm{var}(\mathcal{C}_{j(i)})$ and let $\alpha(z_i) = \alpha_{j(i)}(z_i)$. We claim that α satisfies every clause. To see this, consider any clause $C \in \mathcal{C}$. Suppose that C is in the partition $\mathcal{C}_j$. It is easy to check that $f(a_1^{\alpha_1}, \dots, a_k^{\alpha_k}) = 1$ implies that $\alpha|_{\mathrm{var}(C)} = \alpha_j|_{\mathrm{var}(C)}$. Since α_j is a partial assignment that satisfies C, α must also satisfy C. In other words, α satisfies all clauses of ϕ. □

Lemma 6.23. *Assuming* ETH*, for any computable function $T : \mathbb{N} \to \mathbb{N}$, there is no $T(k) \cdot N^{o(k)}$ time algorithm that can solve* PSP$(k, \mathcal{F}^{\textsc{MultEq}}, N)$ *for every $N, k \in \mathbb{N}$.*

Proof. Let $\delta > 0$ be the constant in the running time lower bound in ETH. Suppose for the sake of contradiction that ETH holds but, for some function T, there is a $T(k) \cdot N^{o(k)}$ time algorithm $\mathcal{A}$ that can solve PSP$(k, \mathcal{F}^{\textsc{MultEq}}, N)$ for every $N, k \in \mathbb{N}$. Thus, there exists a sufficiently large k such that the running time of $\mathcal{A}$ for solving PSP$(k, \mathcal{F}^{\textsc{MultEq}}, N)$ is at most $O(N^{\delta k/10})$ for every $N \in \mathbb{N}$.

Given a 3-CNF formula ϕ such that each variable appears in at most three clauses. Let n, m denote the number of variables and the number of clauses of ϕ, respectively. We first run the reduction from Proposition 6.22 on ϕ with this value of k. This produces an instance $(A_1, \dots, A_k)$ of PSP $(k, \mathcal{F}^{\textsc{MultEq}}, N)$ where $N = 2^{3\lceil m/k \rceil}$. Since each variable appears in at most three clauses, we have $m \leqslant 3n$, meaning that $N = O(2^{9n/k})$. By running $\mathcal{A}$ on this instance, we can decide whether ϕ is satisfiable in time $O(N^{\delta k/10}) = O(2^{0.9\delta n})$, contradicting ETH. □

6.4 Communication Protocols and Reduction to MaxCov

In this section, we first introduce a communication model for multiparty communication known in literature as the Simultaneous Message Passing model. Then, we introduce a notion of "efficient" communication protocols, and connect the existence of such protocols to a reduction from PSP to a gap version of MaxCov.

6.4.1 Efficient Protocols in Simultaneous Message Passing Model

The two-player Simultaneous Message Passing (SMP) model was introduced by Yao [Yao79] and has been extensively studied in literature [KN97]. In the multiparty setting, the SMP model is considered popularly with the number-on-forehead model, where each player can see the input of all the other players but not his own [CFL83; Bab+03]. In this chapter, we consider the multiparty SMP model where the inputs are given as in the number-in-hand model (like in [FOZ16; WW15]).

Simultaneous Message Passing Model. Let $f : \{0,1\}^{m\times k} \to \{0,1\}$. In the k-player simultaneous message passing communication model, we have k players each with an input $x_i \in \{0,1\}^m$ and a referee who is given an advice $\mu \in \{0,1\}^*$ (at the same time when the players are given the input). The communication task is for the referee to determine if $f(x_1,\ldots,x_k) = 1$. The players are allowed to only send messages to the referee. In the randomized setting, we allow the players *and* the referee to jointly toss some random coins before sending messages, i.e., we allow public randomness.

Next, we introduce the notion of *efficient* protocols, which are in a nutshell one-round randomized protocols where the players and the referee are in a computationally bounded setting.

Efficient Protocols. Let π be a communication protocol for a problem in the SMP model. We say that π is a (w,r,ℓ,s)-efficient protocol if the following holds:

- The referee receives w bits of advice.
- The protocol is one-round with public randomness, i.e., the following actions happen sequentially:
    1. The players receive their inputs and the referee receives his advice.
    2. The players and the referee jointly toss r random coins.
    3. Each player on seeing the randomness (i.e. results of r coin tosses) deterministically sends an ℓ-bit message to the referee.
    4. Based on the advice, the randomness, and the total $\ell \cdot k$ bits sent from the players, the referee outputs accept or reject.
- The protocol has completeness 1 and soundness s, i.e.,
    - If $f(x_1,\ldots,x_k) = 1$, then there exists an advice on which the referee always accepts.
    - If $f(x_1,\ldots,x_k) = 0$, then, on any advice, the referee accepts with probability at most s.
- The players and the referee are computationally bounded, i.e., all of them perform all their computations in poly(m)-time.

The following proposition follows immediately from the definition of an efficient protocol and will be very useful in later sections for gap amplification.

Proposition 6.24. *Let $z \in \mathbb{N}$ and π be a communication protocol for a problem in the* SMP *model. Suppose π is a (w,r,ℓ,s)-efficient protocol. Then there exists a $(w, z\cdot r, z\cdot\ell, s^z)$-efficient protocol for the same problem.*

Proof. The proof follows by a simple repetition argument. More precisely, we repeat steps 2-4 in the protocol z times, each time using fresh randomness, but note that the z steps of drawing random coins can be clubbed into one step, and the decision by the referee can be reserved till the end of the entire protocol, wherein he accepts if and only if he would accept in each of the individual repetitions. □

6.4.2 Lower Bounds on Gap-MAXCOV

The following theorem is the main conceptual contribution of the chapter: we show below that the existence of efficient protocols can translate (exact) hardness of PSPs to hardness of approximating MAXCOV.

Theorem 6.25. *Let $m : \mathbb{N} \times \mathbb{N} \to \mathbb{N}$ be any function. Let $\mathcal{F} := \{f_{N,k} : \{0,1\}^{m(N,k)\times k} \to \{0,1\}\}_{N,k\in\mathbb{N}}$ be a family of Boolean functions indexed by N, k. Suppose there exists a (w, r, ℓ, s)-efficient protocol*[11] *for $f_{N,k}$ in the k-player* SMP *model for every $N, k \in \mathbb{N}$. Then, there is a reduction from any instance $(A_1, \dots, A_k)$ of* $\mathrm{PSP}(k, \mathcal{F}, N)$ *to 2^w label cover instances $\{\mathcal{L}^\mu\}_{\mu\in\{0,1\}^w}$ such that*

- *The running time of the reduction is $2^{w+r+\ell k}$poly$(m(N, k))$.*
- *Each $\mathcal{L}^\mu = (U^\mu, V^\mu, \Sigma^\mu_U, \Sigma^\mu_V, \{\Pi^\mu_e\})$ has the following parameters:*
  - *$\mathcal{L}^\mu$ has k right super nodes, i.e., $|V^\mu| = k$,*
  - *$\mathcal{L}^\mu$ has 2^r left super nodes, i.e., $|U^\mu| = 2^r$,*
  - *$\mathcal{L}^\mu$'s right alphabet size is at most N right nodes, i.e., $|\Sigma^\mu_V| \leqslant N$,*
  - *$\mathcal{L}^\mu$'s left alphabet size is at most $2^{\ell k}$, i.e., $|\Sigma^\mu_U| \leqslant 2^{\ell k}$.*
- *If $(A_1, \dots, A_k)$ is a YES instance of* $\mathrm{PSP}(k, \mathcal{F}, N)$*, then* $\mathrm{MAXCOV}(\mathcal{L}^\mu) = 2^r$ *for some $\mu \in \{0,1\}^w$.*
- *If $(A_1, \dots, A_k)$ is a NO instance of* $\mathrm{PSP}(k, \mathcal{F}, N)$*, then* $\mathrm{MAXCOV}(\mathcal{L}^\mu) \leqslant s \cdot 2^r$ *for every $\mu \in \{0,1\}^w$.*

Proof. Given a (w, r, ℓ, s)-efficient protocol π of $f_{N,k}$ and an instance $(A_1, \dots, A_k)$ of $\mathrm{PSP}(k, \mathcal{F}, N)$, we will generate 2^w instances of MAXCOV. Specifically, for each $\mu \in \{0,1\}^w$, we construct an instance $\mathcal{L}^\mu = (U^\mu, V^\mu, \Sigma^\mu_U, \Sigma^\mu_V, \{\Pi^\mu_e\})$ of MAXCOV as follows.

- Let $V^\mu = [k]$.
- Let Σ^μ_V be of size N, and, for each $j \in [h]$, we associate each $x_j \in A_j$ with a label in Σ^μ_V.
- Let $U^\mu = \{0,1\}^r$.
- Let Σ^μ_U be of size $2^{\ell k}$. For each left super-node $\gamma \in \{0,1\}^r$, we associate each accepting messages from the k players (i.e. $(m_1, \dots, m_k) \in (\{0,1\}^\ell)^k$ where in the protocol π the referee, on an advice μ and a random string γ, accepts if the messages he received from the k players are $m_1, \dots, m_k$) with a label in Σ^μ_U.
- For each $e = (j, \gamma)$, we add an $(x_j, (m_1, \dots, m_k))$ to Π_e iff m_j is equal to the message that j sends on an input x_j and a random string γ in the protocol π.

[11] w, r, ℓ and s can depend on N and k.

Note here that, since $|A_j|$'s may not be equal, some label in Σ_V^μ might not be associated with any element in A_j for some left super-node j. In this case, we might ignore such a label for the super-node j since it never appears in the constraints. Similar statement holds for the right super-nodes. From this viewpoint, there is a bijection between "valid" right labelings of $\mathcal{L}^\mu$ and elements of $A_1 \times \cdots \times A_k$, where "valid" means that these labels are not used.

Now consider a right labeling σ_{V^μ} of $\mathcal{L}^\mu$ and the corresponding $(x_1, \dots, x_k) \in A_1 \times \cdots \times A_k$. For each random string $\gamma \in \{0,1\}^r$, observe that the referee accepts on an input $(x_1, \dots, x_k)$, an advice μ, and a random string γ if and only if there is a label $(m_1, \dots, m_k)$ of the left super-node γ that is consistent with all of $x_1, \dots, x_k$. Therefore, the acceptance probability of the protocol on advice μ is the same as the fraction of left super-nodes covered by σ_{V^μ}. The completeness and soundness then easily follows:

Completeness. If there exists $(x_1, \dots, x_k) \in A_1 \times \cdots \times A_k$ such that $f_{N,k}(x_1, \dots, x_k) = 1$, then there is an advice $\mu \in \{0,1\}^w$ on which the referee always accepts for this input $(x_1, \dots, x_k)$, meaning that the corresponding labeling covers every left super-node of $\mathcal{L}^\mu$, i.e., $\textsc{MaxCov}(\mathcal{L}^\mu) = 2^r$.

Soundness. If $f_{N,k}(x_1, \dots, x_k) = 0$ for every $(x_1, \dots, x_k) \in A_1 \times \cdots \times A_k$, then, for any advice $\mu \in \{0,1\}^w$, the referee accepts with probability at most s on every input $(x_1, \dots, x_k) \in A_1 \times \cdots \times A_k$. This means that, for any $\mu \in \{0,1\}^w$, no labeling covers more than s fraction of left the super-nodes. In other words, $\textsc{MaxCov}(\mathcal{L}^\mu) \leqslant s \cdot 2^r$ for all $\mu \in \{0,1\}^w$. □

For the rest of this subsection, we will use the following shorthand. Let $\mathcal{L} = (U, V, \Sigma_U, \Sigma_V, \{\Pi_e\})$ be a label cover instance, and we use the shorthand $\mathcal{L}(N, k, r, \ell)$ to say that the label cover instance has the following parameters:

- $\mathcal{L}$ has k right super nodes, i.e., $|V| = k$,
- $\mathcal{L}$ has 2^r left super nodes, i.e., $|U| = 2^r$,
- $\mathcal{L}$ has right alphabet size of at most N, i.e., $|\Sigma_V| \leqslant N$,
- $\mathcal{L}$ has left alphabet size of at most $2^{\ell k}$, i.e., $|\Sigma_U| \leqslant 2^{\ell k}$.

The rest of this section is devoted to combining Theorem 6.25 with the results in Section 6.3 to obtain conditional hardness for the gap-$\textsc{MaxCov}$ problem, assuming that we have efficient protocols with certain parameters. These protocols will be devised in the three subsequent sections.

Understanding the Parameters. Before we state the exact dependency of parameters, let us first discuss some intuition behind it. First of all, if we start with an instance of PSP$(k, \mathcal{F}, N)$, Theorem 6.25 will produce 2^w instances of $\mathcal{L}(N, k, r, \ell)$. Roughly speaking, since we want the lower bounds from PSP to translate to $\textsc{MaxCov}$, we would like the number of instances to be $N^{o(1)}$, meaning that we want $w = o(\log N)$. Recall that in all function families we consider $m = \Theta_k(\log N)$. Hence, this requirement is the same as $w = o_k(m)$. Moreover, we would like the instance size of $\mathcal{L}(N, k, r, \ell)$ to also be $O_k(N)$, meaning that $|U||\Sigma_U| \leqslant 2^{r+\ell k}$ has to be $O_k(N)$. Thus, it suffices to have a protocol where $r + \ell k = o_k(m)$.

If we additionally want the hardness to translate also to k-DOMSET, the parameter dependencies become more subtle. Specifically, applying Theorem 2.27 to the MAXCOV instances results in a blow-up of $|U||V|^{|\Sigma_U|} \leqslant 2^r \cdot k^{2^{\ell k}} = 2^{r+(\log k)\cdot 2^{\ell k}}$. We also want this to be at most $N^{o(1)}$, meaning that we need $r + (\log k) \cdot 2^{\ell k} = o(\log N) = o_k(m)$. In other words, it suffices for us to require that $\ell k < {}^{(\log m)}/_{\beta}$ for some constant $\beta > 1$. The exact parameter dependencies are formalized below.

SETH

Corollary 6.26. *For any $c \in \mathbb{N}$, let $\mathcal{F}_c^{\text{DISJ}}$ be the family of Boolean functions as defined in Definition 6.14. For every $\delta > 0$, suppose there exists a (w, r, ℓ, s)-efficient protocol for* DISJ$_{m,k}$ *in the k-player* SMP *model for every $k \in \mathbb{N}$ and every $m \in \mathbb{N}$, such that $w \leqslant \delta m$ and $r + \ell k = o_k(m)$. Then, assuming* SETH, *for every $\varepsilon > 0$ and integer $k > 1$, no $O(N^{k(1-\varepsilon)})$-time algorithm can distinguish between* MAXCOV$(\mathcal{L}) = 2^r$ *and* MAXCOV$(\mathcal{L}) \leqslant s \cdot 2^r$ *for any label cover instance $\mathcal{L}(N, k, r, \ell)$ for all $N \in \mathbb{N}$. Moreover, if $\ell < {}^{(\log m)}/_{\beta \cdot k}$ for some constant $\beta > 1$, then assuming* SETH, *for every $\varepsilon > 0$ and integer $k > 1$, no $O(N^{k(1-\varepsilon)})$-time algorithm can distinguish between* DOMSET$(G) = k$ *and* DOMSET$(G) \geqslant \left(\frac{1}{s}\right)^{\frac{1}{k}} \cdot k$ *for any graph G with at most $O_k(N)$ vertices, for all $N \in \mathbb{N}$.*

Proof. The proof of the first part of the theorem statement is by contradiction. Suppose there is an $O(N^{k(1-\varepsilon)})$-time algorithm $\mathcal{A}$ for some fixed constant $\varepsilon > 0$ and integer $k > 1$ which can distinguish between MAXCOV$(\mathcal{L}) = 2^r$ and MAXCOV$(\mathcal{L}) \leqslant s \cdot 2^r$ for any label cover instance $\mathcal{L}(N, k, r, \ell)$ for all $N \in \mathbb{N}$. From Proposition 6.15, we have that there exists $c_\varepsilon \in \mathbb{N}$ such that no $O(N^{k(1-\varepsilon/2)})$-time algorithm can solve PSP$(k, \mathcal{F}_{c_\varepsilon}^{\text{DISJ}}, N)$ for all $N \in \mathbb{N}$. Fix $\delta = {}^{\varepsilon}/_{3c_\varepsilon}$. Next, by considering Theorem 6.25 for the case of (w, r, ℓ, s)-efficient protocols, we have that there are 2^w label cover instances $\{\mathcal{L}^\mu\}_{\mu \in \{0,1\}^w}$ which can be constructed in $2^{\delta m(1+o_k(1))}$ time. Note that $2^{\delta m(1+o_k(1))} = N^{k\varepsilon/3(1+o_k(1))}$ by our choice of δ. Thus, we can run $\mathcal{A}$ on each $\mathcal{L}^\mu$ and solve PSP$(k, \mathcal{F}_c^{\text{DISJ}}, N)$ for all $N, k \in \mathbb{N}$ in time less than $N^{k(1-\varepsilon/2)}$. This contradicts Proposition 6.15.

To prove the second part of the theorem statement, we apply the reduction described in Theorem 2.27 and note that $2^r = N^{o(1)}$ and $k^{2^{\ell k}} = 2^{(\log_2 k)\cdot(m(N,k))^{1/\beta}} = N^{o(1)}$. □

The proof of Theorem 6.3 follows by plugging in the parameters of the protocol described in Corollary 6.31 to the above corollary.

ETH

Corollary 6.27. *Let $\mathcal{F}^{\text{MULTEQ}}$ be the family of Boolean functions as defined in Definition 6.19. Suppose there exists a (w, r, ℓ, s)-efficient protocol for* MULTEQ$_{m,k,t}$ *in the k-player* SMP *model for every $k, t, m \in \mathbb{N}$ such that $w + r + \ell k = o_k(m)$. Then, assuming* ETH, *for any computable function $T : \mathbb{N} \to \mathbb{N}$, there is no $T(k) \cdot N^{o(k)}$ time algorithm that can distinguish between* MAXCOV$(\mathcal{L}) = 2^r$ *and* MAXCOV$(\mathcal{L}) \leqslant s \cdot 2^r$ *for any label cover instance $\mathcal{L}(N, k, r, \ell)$ for all $N, k \in \mathbb{N}$. Moreover, if $\ell < {}^{(\log m)}/_{\beta \cdot k}$ for some constant $\beta > 1$, then assuming* ETH, *for any computable function $T : \mathbb{N} \to \mathbb{N}$, there is no $T(k) \cdot N^{o(k)}$ time algorithm that can distinguish between*

$\textsc{DomSet}(G) = k$ *and* $\textsc{DomSet}(G) \geqslant \left(\frac{1}{s}\right)^{\frac{1}{k}} \cdot k$ *for any graph G with at most $O_k(N)$ vertices, for all $N, k \in \mathbb{N}$.*

Proof. The proof of the first part of the theorem statement is by contradiction. Suppose there is an algorithm $\mathcal{A}$ running in time $\widetilde{T}(k) \cdot N^{o(k)}$ for some computable function $\widetilde{T} : \mathbb{N} \to \mathbb{N}$ that can distinguish between $\textsc{MaxCov}(\mathcal{L}) = 2^r$ and $\textsc{MaxCov}(\mathcal{L}) \leqslant s \cdot 2^r$ for any label cover instance $\mathcal{L}(N, k, r, \ell)$ for all $N, k \in \mathbb{N}$. From Lemma 6.23, we have that for any computable function $T : \mathbb{N} \to \mathbb{N}$, there is no $T(k) \cdot N^{o(k)}$ time algorithm that can solve $\mathrm{PSP}(k, \mathcal{F}^{\textsc{MultEq}}, N)$ for every $N, k \in \mathbb{N}$. Next, by considering Theorem 6.25 for the case of (w, r, ℓ, s)-efficient protocols, we have that there are 2^w label cover instances $\{\mathcal{L}^\mu\}_{\mu \in \{0,1\}^w}$ which can be constructed in $2^{o_k(m)}$ time. Note that $2^{o_k(m)} = O_k(N^{o(1)})$ by the choice of $m(N, k)$ in Definition 6.19. Thus, we can run $\mathcal{A}$ on each $\mathcal{L}^\mu$ and solve $\mathrm{PSP}(k, \mathcal{F}^{\textsc{MultEq}}, N)$ for all $N, k \in \mathbb{N}$ in time $\widetilde{T}(k) \cdot N^{o(k)}$. This contradicts Lemma 6.23.

To prove the second part of the theorem statement, we apply the reduction described in Theorem 2.27 and note that $2^r = N^{o(1)}$ and $k^{2^{\ell k}} = 2^{(m(N,k))^{1/\beta} \cdot \log_2 k} = N^{o(1)}$. □

The proof of Theorem 6.2 follows by plugging in the parameters of the protocol described in Corollary 6.33 to the above corollary.

$\mathbf{W[1] \neq FPT}$

Corollary 6.28. *Let $\mathcal{F}^{\textsc{MultEq}}$ be the family of Boolean functions as defined in Definition 6.19. Suppose there exists a (w, r, ℓ, s)-efficient protocol for $\textsc{MultEq}_{m,k,t}$ in the k-player SMP model for every $k, t, m \in \mathbb{N}$ such that $w + r + \ell k < {}^{m}/_{tk}$. Then, assuming $\mathrm{W[1]} \neq \mathrm{FPT}$, for any computable function $T : \mathbb{N} \to \mathbb{N}$, there is no $T(k) \cdot poly(N)$-time algorithm that can distinguish between $\textsc{MaxCov}(\mathcal{L}) = 2^r$ and $\textsc{MaxCov}(\mathcal{L}) \leqslant s \cdot 2^r$ for any label cover instance $\mathcal{L}(N, k, r, \ell)$ for all $N, k \in \mathbb{N}$. Moreover, if $r < {}^{m}/_{2tk}$ and $\ell < {}^{(\log m)}/_{\beta \cdot k}$ for some constant $\beta > 1$, then assuming $\mathrm{W[1]} \neq \mathrm{FPT}$, for any computable function $T : \mathbb{N} \to \mathbb{N}$, there is no $T(k) \cdot poly(N)$-time algorithm that can distinguish between $\textsc{DomSet}(G) = k$ and $\textsc{DomSet}(G) \geqslant \left(\frac{1}{s}\right)^{\frac{1}{k}} \cdot k$ for any graph G with at most $O_k(N)$ vertices, for all $N, k \in \mathbb{N}$.*

Proof. The proof of the first part of the theorem statement is by contradiction. Suppose there is an algorithm $\mathcal{A}$ running in time $\widetilde{T}(k) \cdot \mathrm{poly}(N)$ for some computable function $\widetilde{T} : \mathbb{N} \to \mathbb{N}$ that can distinguish between $\textsc{MaxCov}(\mathcal{L}) = 2^r$ and $\textsc{MaxCov}(\mathcal{L}) \leqslant s \cdot 2^r$ for any label cover instance $\mathcal{L}(N, k, r, \ell)$ for all $N, k \in \mathbb{N}$. From Lemma 6.21, we have that for any computable function $T : \mathbb{N} \to \mathbb{N}$, there is no $T(k) \cdot \mathrm{poly}(N)$ time algorithm that can solve $\mathrm{PSP}(k, \mathcal{F}^{\textsc{MultEq}}, N)$ for every $N, k \in \mathbb{N}$. Next, by considering Theorem 6.25 for the case of (w, r, ℓ, s)-efficient protocols, we have that there are 2^w label cover instances $\{\mathcal{L}^\mu\}_{\mu \in \{0,1\}^w}$ which can be constructed in $2^{m(N,k)/k \cdot t(k)} \cdot \mathrm{poly}(m(N, k))$ time. Note that $2^{m(N,k)/k \cdot t(k)} = O(N)$ and $\mathrm{poly}(m(N, k)) = N^{o(1)}$ by the choice of $m(N, k)$ in Definition 6.19. Thus, we can run $\mathcal{A}$ on each $\mathcal{L}^\mu$ and solve $\mathrm{PSP}(k, \mathcal{F}^{\textsc{MultEq}}, N)$ for all $N, k \in \mathbb{N}$ in time less than $\widetilde{T}(k) \cdot \mathrm{poly}(N)$. This contradicts Lemma 6.21.

To prove the second part of the theorem statement, we apply the reduction described in Theorem 2.27 and note that $2^r = O(\sqrt{N})$ and $k^{2^{\ell k}} = 2^{(m(N,k))^{1/\beta} \cdot \log_2 k} = N^{o(1)}$. □

The proof of Theorem 6.1 follows by plugging in the parameters of the protocol described in Corollary 6.33 to the above corollary.

k-SUM Hypothesis

Corollary 6.29. *Let $\mathcal{F}^{\textsc{SumZero}}$ be the family of Boolean functions as defined in Definition 6.11. Suppose there exists a (w, r, ℓ, s)-efficient protocol for $\textsc{SumZero}_{m,k}$ in the k-player* SMP *model for every $m, k \in \mathbb{N}$, such that $w + r + \ell k = o_k(m)$. Then assuming the k-*SUM *Hypothesis, for every integer $k \geqslant 3$ and every $\varepsilon > 0$, no $O(N^{\lceil k/2 \rceil - \varepsilon})$-time algorithm can distinguish between $\textsc{MaxCov}(\mathcal{L}) = 2^r$ and $\textsc{MaxCov}(\mathcal{L}) \leqslant s \cdot 2^r$ for any label cover instance $\mathcal{L}(N, k, r, \ell)$ for all $N \in \mathbb{N}$. Moreover, if $\ell < {}^{(\log m)}/_{\beta \cdot k}$ for some constant $\beta > 1$, then assuming the k-*SUM *Hypothesis, for every $\varepsilon > 0$ no $O(N^{\lceil k/2 \rceil - \varepsilon})$-time algorithm can distinguish between $\textsc{DomSet}(G) = k$ and $\textsc{DomSet}(G) \geqslant \left(\frac{1}{s}\right)^{\frac{1}{k}} \cdot k$ for any graph G with at most $O_k(N)$ vertices for all $N \in \mathbb{N}$.*

Proof. The proof of the first part of the theorem statement is by contradiction. Suppose there is an algorithm $\mathcal{A}$ running in time $O(N^{\lceil k/2 \rceil - \varepsilon})$ for some fixed constant $\varepsilon > 0$ and some integer $k \geqslant 3$ that can distinguish between $\textsc{MaxCov}(\mathcal{L}) = 2^r$ and $\textsc{MaxCov}(\mathcal{L}) \leqslant s \cdot 2^r$ for any label cover instance $\mathcal{L}(N, k, r, \ell)$ for all $N \in \mathbb{N}$. From Proposition 6.12, we have that no $O(N^{\lceil k/2 \rceil - \varepsilon/2})$-time algorithm can solve $\text{PSP}(k, \mathcal{F}^{\textsc{SumZero}}, N)$ for all $N \in \mathbb{N}$. Next, by considering Theorem 6.25 for the case of (w, r, ℓ, s)-efficient protocols, we have that there are 2^w label cover instances $\{\mathcal{L}^\mu\}_{\mu \in \{0,1\}^w}$ which can be constructed in $2^{o_k(m)}$ time. Note that $2^{o_k(m)} = O_k(N^{o(1)})$ by the choice of $m(N, k)$ in Definition 6.11. Thus, we can run $\mathcal{A}$ on each $\mathcal{L}^\mu$ and solve $\text{PSP}(k, \mathcal{F}^{\textsc{SumZero}}, N)$ for all $N \in \mathbb{N}$ in time $O(N^{\lceil k/2 \rceil - \varepsilon})$. This contradicts Proposition 6.12.

To prove the second part of the theorem statement, we apply the reduction described in Theorem 2.27 and note that $2^r = N^{o(1)}$ and $k^{2^{\ell k}} = 2^{(m(N,k))^{1/\beta} \cdot \log_2 k} = N^{o(1)}$. □

The proof of Theorem 6.4 follows by plugging in the parameters of the protocol described in Corollary 6.39 to the above corollary.

6.5 An Efficient Protocol for Set Disjointness

Set Disjointness has been extensively studied primarily in the two-player setting (i.e., $k = 2$). In that setting, we know that the randomized communication complexity is $\Omega(m)$ [KS92; Raz92; Bar+04], where m is the input size of each player. Surprisingly, [AW09] showed that the MA-complexity of two-player set disjointness is $\widetilde{O}(\sqrt{m})$. Their protocol was indeed an $(\widetilde{O}(\sqrt{m}), O(\log m), \widetilde{O}(\sqrt{m}), {}^{1}/$ efficient protocol for the case when $k = 2$. Recently, [ARW17a] improved (in terms of the message size) the protocol to be an $({}^{m}/_{\log m}, O(\log m), O((\log m)^3), {}^{1}/_{2})$-efficient protocol for the case when $k = 2$. Both these results can be extended naturally for all $k > 1$, to give an

$({}^{mk}/_{\log m}, O_k(\log m), O_k((\log m)^3), {}^{1}/_{2})$-efficient protocol. However, this does not suffice for proving Theorem 6.3 since we need a (w, r, ℓ, s)-efficient protocol with $w = o(m)$ *and* $\ell = o(\log m)$. Fortunately, Rubinstein [Rub18] recently showed that the exact framework of the MA-protocols as in [AW09; ARW17a] but with the use of algebraic geometric codes instead of Reed Muller or Reed Solomon codes gives the desired parameters in the two-player case. Below we naturally extend Rubinstein's protocol to the k-player setting. This extension was suggested to us by Rubinstein [Rub17a].

Theorem 6.30. *There is a polynomial function $\hat{\ell} : \mathbb{N} \times \mathbb{N} \to [1, \infty)$ such that for every $k \in \mathbb{N}$ and every $\alpha \in \mathbb{N}$, there is a protocol for k-player* $\textsc{Disj}_{m,k}$ *in the* SMP *model which is an $({}^{m}/_{\alpha}, \log_2 m, \hat{\ell}(k, \alpha), {}^{1}/_{2})$-efficient protocol, where each player is given m bits as input, and the referee is given at most ${}^{m}/_{\alpha}$ bits of advice.*

Proof. Fix $k, m, \alpha \in \mathbb{N}$. Let q be the smallest prime greater than $\hat{q}(k)$ such that $q > (2\alpha\hat{r}(k))^2$, where the functions $\hat{r}$ and $\hat{q}$ are as defined in Theorem 6.8. Let $\mathcal{G} = \mathbb{F}_{q^2}$. Let $T = 2\alpha\hat{r}(k)\log_2 q$.

We associate $[m]$ with $[T] \times [m/T]$ and write the input $\mathbf{x}_j \in \{0, 1\}^m$ as vectors $\mathbf{x}_j^1, \ldots, \mathbf{x}_j^T$ where $\mathbf{x}_j^t \in \{0, 1\}^{m/T}$. For every $j \in [k]$, Player j computes $A_{m/T}(\mathbf{x}_j^t)$ for every $t \in [T]$. We denote the block length of $A_{m/T}$ by d. From the systematicity guaranteed by Theorem 6.8 we have that $A_{m/T}(\mathbf{x}_j^t) \mid_{[m/T]} = \mathbf{x}_j^t$. Also, notice that for all $t \in [T]$, we have $\underset{j\in[k]}{\wedge} \mathbf{x}_j^t = 0^{m/T}$ if and only if $\prod_{j\in[k]} A_{m/T}(\mathbf{x}_j^t) = 0^{m/T}$.

With the above observation in mind, we define the marginal sum $\Gamma \in \mathcal{G}^d$ as follows:

$$\forall i \in [d],\ \Gamma_i = \sum_{t\in[T]} \prod_{j\in[k]} A_{m/T}(\mathbf{x}_j^t)_i.$$

Again, notice that for all $t \in [T]$, we have $\underset{j\in[k]}{\wedge} \mathbf{x}_j^t = 0^{m/T}$ if and only if $\Gamma_i = 0$ for all $i \in [m/T]$. This follows from the following:

- For all $i \in [m/T]$, we have $A_{m/T}(\mathbf{x}_j^t)_i \in \{0, 1\}$ and thus $\prod\limits_{j\in[k]} A_{m/T}(\mathbf{x}_j^t)_i \in \{0, 1\}$.
- The characteristic of $\mathcal{G}$ being greater than $(2\alpha\hat{r}(k)) \cdot \sqrt{q} \geqslant (2\alpha\hat{r}(k)) \cdot \log_2 q = T$.

More importantly, we remark that Γ is a codeword[12] in $B_{m/T}$. This follows because $B_{m/T}$ is a degree k closure code of $A_{m/T}$. To see this, in Definition 6.7, set $t = k$, $r = k \cdot T$, and $P[x_{1,1}, \ldots, x_{k,T}] = \sum_{i\in[T]} \prod_{j\in[k]} x_{i,j}$.

The protocol

1. Merlin sends the referee Φ which is allegedly equal to the marginal sums codeword Γ defined above.

[12] We would like to remark that we use the multiplicity of Algebraic Geometric codes to find a non-trivial advice. On a related note, Meir [Mei13] had previously shown that error correcting codes with the multiplicity property suffice to show the IP theorem (i.e., the IP=PSPACE result).

2. All players jointly draw $r \in [d]$ uniformly at random.

3. For every $j \in \{1, \dots, k\}$, Player j sends to the referee $A_{m/T}(\mathbf{x}_j^t)_r, \ \forall t \in [T]$.

4. The referee accepts if and only if both of the following hold:

$$\forall i \in [m/T], \quad \Phi_i = 0 \tag{6.1}$$

$$\Phi_r = \sum_{t\in[T]} \prod_{j\in[k]} A_{m/T}(\mathbf{x}_j^t)_r. \tag{6.2}$$

Analysis

Advice Length. To send the advice, Merlin only needs to send a codeword in $B_{m/T}$ to the verifier. This means that the advice length (in bits) is no more than $\log_2 q^2$ times the block length, which is $d \leqslant (m/T)\,\hat{r}(k) = m/2\alpha\log_2 q$ where the equality comes from our choice of T and the inequality from Theorem 6.8.

Message Length. Each player sends T elements of $\mathcal{G}$. Hence, the message length is $T\log_2 q^2 \leqslant \alpha\hat{r}(k)(2\log_2 q)^2$. Recall that q can be upper bounded by a polynomial in k and α. Hence, the message length is upper bounded entirely as a polynomial in k and α as desired.

Randomness. The number of coin tosses is $\log_2(d) \leqslant \log_2\left(m\hat{r}(k)/T\right) = \log_2\left(m/2\alpha\log_2 q\right) < \log_2 m$.

Completeness. If the k sets are disjoint, Merlin can send the true Γ, and the verifier always accepts.

Soundness. If the k sets are not disjoint and Φ is actually Γ, then (6.1) is false and the verifier always rejects. On the other hand, if $\Phi \neq \Gamma$, then, since both are codewords of $B_{m/T}$, from Theorem 6.8 their relative distance must be at least $1/2$. As a result, with probability at least $1/2$, $\Phi_r \neq \Gamma_r$. Since the right hand side of (6.2) is simply Γ_r, the verifier will reject for such r. Hence, the rejection probability is at least $1/2$. □

The following corollary follows immediately by applying Proposition 6.24 with $z = (\log_2 m)/2k\cdot\ell(k,\alpha)$ to the above theorem.

Corollary 6.31. *There is a polynomial function $\hat{s} : \mathbb{N}\times\mathbb{N} \to [1,\infty)$ such that for every $k \in \mathbb{N}$ and every $\alpha \in \mathbb{N}$ there is a protocol for k-player* Disj$_{m,k}$ *in the* SMP *model which is an $\left(m/\alpha, O\left((\log_2 m)^2\right), (\log_2 m)/2k, (1/m)^{1/\hat{s}(k,\alpha)}\right)$-efficient protocol, where each player is given m bits as input, and the referee is given at most m/α bits of advice.*

6.6 An Efficient Protocol for Multi-Equality

Equality has been extensively studied, primarily in the two-player setting (i.e., $k = 2$). In that setting, when public randomness is allowed, we know that the randomized communication complexity is $O(1)$ [Yao79; KN97], and the protocols can be naturally extended to the k-player SMP model that yields a randomized communication complexity of $O(k)$. There are many protocols which achieve this complexity bound but for the purposes of proving Theorems 6.1 and 6.2, we will use the protocol where the players encode their input using a fixed good binary code and then send a jointly agreed random location of the encoded input to the referee who checks if all the messages he received are equal. Below we extend that protocol for MULTI-EQUALITY.

Theorem 6.32. *For some absolute constant $\delta > 0$, for every $t, k \in \mathbb{N}$ and every $m \in \mathbb{N}$ such that m is divisible by t, there is a $(0, \log m + O(1), 2t, 1-\delta)$-efficient protocol for* $\textsc{MultEq}_{m,k,t}$ *in the k-player* SMP *model.*

Proof. Let $\mathcal{C} = \{C_m : \{0,1\}^m \to \{0,1\}^{d(m)}\}_{m\in\mathbb{N}}$ be the family of good codes with rate at least ρ and relative distance at least δ as guaranteed by Fact 6.5. Fix $m, k, t \in \mathbb{N}$ as in the theorem statement. Let $m' := {}^{m}\!/_{t}$.

The protocol

1. All players jointly draw $i^\star \in [d(m')]$ uniformly at random.
2. If player j's input is $x_j = (x_{j,1}, y_{j,1}, \ldots, x_{j,t}, y_{j,t})$, then he sends $(C(x_{j,1})_{i^*}, y_{j,1}, \ldots, C(x_{j,t})_{i^*}, y_{j,t})$ to the referee.
3. The referee accepts if and only if the following holds:

$$\textsc{MultEq}_{2t,k,t}((C(x_{1,1})_{i^*}, y_{1,1}, \ldots, C(x_{1,t})_{i^*}, y_{1,t}), \ldots, (C(x_{k,1})_{i^*}, y_{k,1}, \ldots, C(x_{k,t})_{i^*}, y_{k,t})) = 1.$$

Analysis

Parameters of the Protocol. In the first step of the protocol all the players jointly draw $i^\star \in [d(m')]$ uniformly, which requires $\lceil \log d(m') \rceil \leqslant \log m' + \log(1/\rho) \leqslant \log m + O(1)$ random bits. Then, for every $j \in [k]$, player j sends the referee $2t$ bits. Finally, since the code $C_{m'}$ is efficient, it is easy to see that the players and referee run in poly(m)-time.

Completeness. If $\textsc{MultEq}_{m,k,t}(x_1, \ldots, x_k) = 1$, then, for every $q \in [t]$ and $i, j \in [k]$, we have $(y_{i,q} = \perp) \vee (y_{j,q} = \perp) \vee (x_{i,q} = x_{j,q}) = 1$. This implies that, for every $q \in [t], i, j \in [k]$ and $i^* \in [d(m)]$, we have $(y_{i,q} = \perp) \vee (y_{j,q} = \perp) \vee (C(x_{i,q})_{i^*} = C(x_{j,q})_{i^*})$. In other words, $\textsc{MultEq}_{2t,k}((C(x_{1,1})_{i^*}, y_{1,1}, \ldots, C(x_{1,t})_{i^*}, y_{1,t}), \ldots, (C(x_{k,1})_{i^*}, y_{k,1}, \ldots, C(x_{k,t})_{i^*}, y_{k,t})) = 1$ for every $i^* \in [d(m)]$, meaning that the referee always accepts.

Soundness. Suppose that $\textsc{MultEq}_{m,k,t}(x_1, \ldots, x_k) = 0$. Then, there exists some $q \in [t]$ and $i, j \in [k]$ such that $y_{i,q} = \top$, $y_{j,q} = \top$ and $x_{i,q} \neq x_{j,q}$. Since the code C has relative distance at least δ, $C(x_{i,q})$ and $C(x_{j,q})$ must differ on at least δ fraction of the coordinates. When the randomly selected i^* is such a coordinate, we have that $\textsc{MultEq}_{2t,k}((C(x_{1,1})_{i^*}, y_{1,1}, \ldots, C(x_{1,t})_{i^*}, y_{1,t}), \ldots, (C(x_{k,1})_{i^*}, y_{k,1}, .$
0, i.e., the referee rejects. Hence, the referee rejects with probability at least δ. □

The following corollary follows immediately by applying Proposition 6.24 to the above theorem with $z = {}^{\log_2 m}\!/_{4kt}$.

Corollary 6.33. *For every $t, k \in \mathbb{N}$ and every $m \in \mathbb{N}$ such that m is divisible by t, there is a $(0, O((\log m)^2), {}^{(\log_2 m)}/_{2k}, ({}^1/_m)^{1/O(kt)})$-efficient protocol for* $\textsc{MultEq}_{m,k,t}$ *in the k-player* SMP *model.*

6.7 An Efficient Protocol for Sum-Zero

The Sum-Zero problem has been studied in the SMP model and efficient protocols with the following parameters have been obtained.

Theorem 6.34 ([Nis94]). *For every $k \in \mathbb{N}$ and $m \in \mathbb{N}$ there is a $(0, O(\log(m+\log k)), O(\log(m+\log k)), {}^1/_2)$-efficient protocol for* $\textsc{SumZero}_{m,k}$ *in the k-player* SMP *model.*

The above protocol is based on a simple (yet powerful) idea of picking a small random prime p and checking if the numbers sum to zero modulo p. Viola put forth a protocol with better parameters than the above protocol (i.e., smaller message length) using specialized hash functions and obtained the following:

Theorem 6.35 ([Vio15]). *For every $k \in \mathbb{N}$ and $m \in \mathbb{N}$ there is a $(0, O(m), O(\log k), {}^1/_2)$-efficient protocol for* $\textsc{SumZero}_{m,k}$ *in the k-player* SMP *model.*

In order to prove Theorem 6.4, we need a protocol with $o(m)$ randomness *and* $o(\log m)$ message length, and both the protocols described above do not meet these conditions. We show below that the above two results can be composed to get a protocol with $O(\log k)$ communication complexity and $O_k(\log m)$ randomness. In fact, we will use a slightly different protocol from [Vio15]: namely, the protocol for the Sum-Zero($\mathbb{Z}_p$) problem as stated below.

Definition 6.36 (Sum-Zero($\mathbb{Z}_p$) Problem). *Let $k, m, p \in \mathbb{N}$. $\mathbb{Z}_p$-$\textsc{SumZero}_{m,k} : (\{0,1\}^m)^k \to \{0,1\}$ is defined by*

$$\mathbb{Z}_p\text{-}\textsc{SumZero}_{m,k}(x_1, \ldots, x_k) = \begin{cases} 1 \text{ if } \sum_{i\in[k]} x_i = 0 \mod p, \\ 0 \text{ otherwise}, \end{cases}$$

where we think of each x_i as a number in $[-2^{m-1}, 2^{m-1} - 1]$.

Theorem 6.37 ([Vio15]). *For every $p, k \in \mathbb{N}$ and $m \in \mathbb{N}$, there is a $(0, O(\log p), O(\log k), {}^1/_2)$-efficient protocol[13] for $\mathbb{Z}_p$-$\textsc{SumZero}_{m,k}$ in the k-player* SMP *model.*

Below is our theorem which essentially combines Theorem 6.34 and Theorem 6.37.

Theorem 6.38. *For every $k \in \mathbb{N}$ and $m \in \mathbb{N}$ there is a $(0, O(\log(m + \log k)), O(\log k), {}^3/_4)$-efficient protocol for* $\textsc{SumZero}_{m,k}$ *in the k-player* SMP *model.*

Proof. Let π^* be the protocol of Viola from Theorem 6.37.

[13] As written in [Vio15], the protocol has the following steps. First, the players send some messages to the referee. Then the players and the referee jointly draw some random coins, and finally the players send some more messages to the referee. However, we note that the first and second steps of the protocol can be swapped in [Vio15] to obtain an efficient protocol as the drawing of randomness do not depend on the messages sent by the players to the referee.

The protocol

Let $t = 2(m - 1 + \log k)$. Let $p_1, \ldots, p_t$ be the first t primes.

1. All players jointly draw $i^\star \in [t]$ uniformly at random.
2. The players and the referee run $\pi^\star$ where each player now has input $y_i = x_i \mod p_{i^\star}$, and the referee accepts if and only if $\sum_{i\in[k]} y_i = 0 \mod p_{i^\star}$.

Notice that the above protocol is still an efficient protocol as the first step of the above protocol can be combined with the draw of random coins in the first step of $\pi^\star$ to form a single step in which the players and the referee jointly draw random coins from the public randomness.

Analysis

Randomness. In the first step of the protocol all the players jointly draw $i^\star \in [t]$ uniformly, which requires $\lceil \log_2 t \rceil$ random bits. Then, the players draw $O(\log p_{i^\star})$ additional random coins for $\pi^\star$. The bound on the randomness follows by noting that $p_{i^\star} \leqslant p_t = O(t \log t)$.

Message Length. For every $j \in [k]$, Player j sends the referee $O(\log k)$ bits as per $\pi^\star$.

Completeness. If the k numbers sum to zero, then for any $p \in \mathbb{N}$, they sum to zero mod p and thus the referee always accepts.

Soundness. If the k numbers do not sum to zero, then let $\mathsf{s}(\pi^\star)$ be the soundness of $\pi^\star$ and let S be the subset of the first t primes defined as follows:

$$S = \left\{ p_i \,\middle|\, i \in [t], \sum_{j\in[k]} x_j = 0 \bmod p_i \right\}$$

It is clear that the referee rejects with probability at least $(1 - {}^{|S|}/_{t}) \cdot (1 - \mathsf{s}(\pi^\star))$. Therefore, it suffices to show that $|S| < {}^{t}/_{2}$ as $\mathsf{s}(\pi^\star) \leqslant {}^{1}/_{2}$. Let $x := \sum_{j\in[k]} x_j$. We have that $x \in [-k \cdot 2^{m-1}, k \cdot 2^{m-1}]$ and that, for every $p \in S$, p divides x. Since $x \neq 0$, we know that $|x| \geqslant \prod_{p\in S} p \geqslant \prod_{i=1}^{|S|} p_i \geqslant 2^{|S|}$. Since $|x| \leqslant k \cdot 2^{m-1}$, we have that $|S| < m - 1 + \log k = t$, and the proof follows. □

The following corollary follows immediately by applying Proposition 6.24 with $z = {}^{\log_2 m}/_{2k\cdot c \log k}$ to the above theorem, where c is some constant such that the message length of the protocol in Theorem 6.38 is at most $c \log k$.

Corollary 6.39. *For every $k, m \in \mathbb{N}$ there is a $\left(0, O((\log(m + \log k))^2), {}^{(\log_2 m)}/_{2k}, ({}^{1}/_{m})^{1/O(k \log k)}\right)$-efficient protocol for* $\textsc{SumZero}_{m,k}$ *in the k-player* SMP *model.*

6.8 Connection to Fine-Grained Complexity

In this section, we will demonstrate the conditional hardness of problems in P by basing them on the conditional hardness of PSP. First, we define the problem in P of interest to this section.

Definition 6.40 (k-linear form inner product)**.** *Let $k, m \in \mathbb{N}$. Given $x_1, \dots, x_k \in \mathbb{R}^m$, we define the inner product of these k vectors as follows:*

$$\langle x_1, \dots, x_k \rangle = \sum_{i \in [m]} \prod_{j \in [k]} x_i(j),$$

where $x_i(j)$ denotes the j^{th} coordinate of the vector x_i.

Definition 6.41 (k-chromatic Maximum Inner Product)**.** *Let $k \in \mathbb{N}$. Given k collections $A_1, A_2, \dots, A_k$ each of N vectors in $\{0,1\}^D$, where $D = N^{o(1)}$, and an integer s, the k-chromatic Maximum Inner Product (*MIP*) problem is to determine if there exists $a_i \in A_i$ for all $i \in [k]$ such that $\langle a_1, \dots, a_k \rangle \geqslant s$.*

We continue to use the shorthand $\mathcal{L}(N, k, r, \ell)$ introduced in Section 6.4.2. We define Unique MaxCov to be the MaxCov problem with the following additional structure: for every labeling σ_V and any left super-node $u \in U$, there is at most one label in Σ_U which satisfies all the constraints $\{\Pi_{(u,v)}\}_{v \in V}$. We remark that the reduction in Theorem 6.25 already produces instances of Unique MaxCov. This follows from the proof of Theorem 6.25 by noting that on every random string, each player sends a message to the referee in a deterministic way. Finally, we have the following connection between unique MaxCov and MIP.

Theorem 6.42. *Let $N, k, r, \ell \in \mathbb{N}$. There is a reduction from any* Unique MaxCov *instance $\mathcal{L}(N, k, r, \ell)$ to k-chromatic* MIP *instance $(A_1, \dots, A_k, s)$ such that*

- *For all $i \in [k]$, $|A_i| \leqslant N$, $s = 2^r$, and $D \leqslant 2^{r+\ell k}$.*
- *The running time of the reduction is $O(Nk \cdot 2^{r+\ell k})$.*
- *For any integer s^*, there exists $(a_1, \dots, a_k) \in A_1 \times \cdots \times A_k$ such that $\langle a_1, \dots, a_k \rangle \geqslant s^*$ if and only if* MaxCov$(\mathcal{L}) \geqslant s^*$.

Proof. For every $i \in [k]$, we associate each A_i with the i-th right super-node $v_i \in V$. Each pair (v_i, β) where $\beta \in \Sigma_V$ corresponds to a vector in A_i; the corresponding vector $a^{(v_i,\beta)} \in A_i$ is constructed as follows. There are $|U| \cdot |\Sigma_U|$ coordinates, each corresponding to $(u, \alpha) \in U \times \Sigma_U$. Let the $(u, \alpha)^{\text{th}}$ coordinate of $a^{(v_i,\beta)}$ be 1 if $(\alpha, \beta) \in \Pi_{(u,v_i)}$; otherwise, it is set to 0. It is easy to see that $|A_i| \leqslant N$ and $D \leqslant 2^{r+\ell k}$, and the running time of the reduction is $O(Nk \cdot 2^{r+\ell k})$. It is also easy to see that if MaxCov$(\mathcal{L}) \geqslant s^*$ then the vectors $a^{(v_i,\sigma_V(v_i))}$ corresponding to the right labeling σ_V resulting in the maximum cover, have inner product at least s^*.

Now, suppose that there exist $a^{(v_i,\beta_i)} \in A_i$ for all $i \in [k]$ such that $\langle a^{(v_1,\beta_1)}, \dots, a^{(v_k,\beta_k)} \rangle \geqslant s^*$. Let σ_V be the right labeling where $\sigma_V(v_i) = \beta_i$. For each coordinate (u, α) at which all the vectors $a^{(v_1,\beta_1)}, \dots, a^{(v_k,\beta_k)}$ are one, this means that the labeling σ_V covers u; moreover, since from the uniqueness property, no left super-node u is double counted. As a result, we have that:

$$\textsc{MaxCov}(\mathcal{L}) \geqslant |\{i \in [2^r] \mid U_i \text{ is covered by } S\}| = \langle a_1, \dots, a_k \rangle \geqslant s^*. \qquad \square$$

The results for hardness of approximation for problems in P is obtained by simply fixing the value of k in the above theorem to some universal constant (such as $k = 2$). For example, by fixing $k = 2$ and applying the above reduction to Corollary 6.26, we recover the main result of [ARW17a] on bichromatic MIP. This is not surprising as under SETH, our framework is just a generalization of the distributed PCP framework of [ARW17a].

Furthermore, we demonstrate the flexibility of our framework by fixing $k = 3$ and applying the above reduction to Corollary 6.29, to establish a hardness of approximation result of trichromatic MIP under the 3-SUM Hypothesis as stated below. We note here that the running time lower bound is only $N^{2-o(1)}$, which is likely not tight since the lower bound from SETH is $N^{3-o(1)}$.

Theorem 6.43. *Assuming the 3-SUM Hypothesis, for every $\varepsilon > 0$, no $O(N^{2-\varepsilon})$ time algorithm can, given three collections A, B, and C each of N vectors in $\{0,1\}^D$, where $D = N^{o(1)}$, and an integer s, distinguish between the following two cases:*

Completeness. *There exists $a \in A, b \in B, c \in C$ such that $\langle a, b, c\rangle \geqslant s$.*

Soundness. *For every $a \in A, b \in B, c \in C$, $\langle a, b, c\rangle \leqslant s/2^{(\log N)^{1-o(1)}}$.*

Proof. We apply Proposition 6.24 with $z = {}^{m}/_{(\log_2 m)^2}$ and $k = 3$ to Theorem 6.38, to obtain a $\left(0, O({}^{m}/_{\log m}), O({}^{m}/_{(\log m)^2}), ({}^{1}/_{2})^{m^{1-o(1)}}\right)$-efficient protocol for $\textsc{SumZero}_{m,3}$ in the 3-player SMP model. By plugging in the parameters of the above protocol to Corollary 6.29, we obtain that assuming the 3-SUM hypothesis, for every $\varepsilon > 0$, no $O(N^{2-\varepsilon})$-time algorithm can distinguish between $\textsc{MaxCov}(\mathcal{L}) = 2^r$ and $\textsc{MaxCov}(\mathcal{L}) \leqslant ({}^{1}/_{2})^{(\log N)^{1-o(1)}} \cdot 2^r$ for any label cover instance $\mathcal{L}(N, 3, r, \ell)$ for all $N \in \mathbb{N}$. The proof of the theorem concludes by applying Theorem 6.42 to the above hardness of MaxCov (Note that Theorem 6.25 provides a reduction from $\text{PSP}(k, \mathcal{F}, N)$ to Unique MaxCov). □

6.9 Discussion and Open Questions

We showed the parameterized inapproximability results for k-DomSet under W[1] $\neq$ FPT, ETH, SETH and k-SUM Hypothesis, which almost resolve the complexity status of approximating parameterized k-DomSet. Although we showed the W[1]-hardness of the problem, the exact version of k-DomSet is W[2]-complete. Thus, a remaining question is whether approximating k-DomSet is W[2]-hard:

Open Question 1. *Can we base total inapproximability of k-*DomSet *on* W[2] $\neq$ FPT*?*

We note that even 1.01-approximation of k-DomSet is not known to be W[2]-hard.

Another direction is to look beyond parameterized complexity questions. As mentioned earlier, Abboud et al. [ARW17a; Rub18] used the hardness of approximating of PCP-Vectors as a starting point of their inapproximability results of problems in P. Since MaxCov is equivalent to PCP-Vectors when the number of right super-nodes is two, it may be possible that MaxCov for larger number of right super-nodes can also be used to prove hardness of problems in P as well. At the moment, however, we do not have any natural candidate in this direction (see Section 6.8 for further discussions).

Chapter 7

Inapproximability from Gap-ETH I: k-Clique and k-Induced Subgraph with Hereditary Property

In the CLIQUE problem, we are given a graph G and an integer k, and the goal is to determine whether G contains a k-clique as a subgraph. Along with DOMSET, CLIQUE is one of problems in Karp's list of NP-complete problems [Kar72]. Hence, the focus has been shifted to its optimization version, called MAXIMUM CLIQUE, where the goal is to find a maximum-size clique in G. The obvious algorithm which outputs a single vertex achieves an approximation ratio of n, where n is the number of vertices of G. There are several algorithm that slightly beats this trivial algorithm, with the best approximation ratio known being $\frac{n \cdot \text{polyloglog}(n)}{\log^3 n}$ [Fei04].

On the other hand, MAXIMUM CLIQUE is arguably the first natural combinatorial optimization problem studied in the context of hardness of approximation; in a seminal work of Feige, Goldwasser, Lovász, Safra and Szegedy (henceforth FGLSS) [Fei+91], a connection was made between interactive proofs and hardness of approximating CLIQUE. This connection paves the way for later works on CLIQUE and other developments in the field of hardness of approximations; indeed, the FGLSS reduction will serve as part of our proof as well. The FGLSS reduction, together with the PCP theorem [AS98; Aro+98] and gap amplification via randomized graph products [BS92], immediately implies n^{ε} ratio inapproximability of CLIQUE for some constant $\varepsilon > 0$ under the assumption that $\mathsf{NP} \subseteq \mathsf{BPP}$. Following Feige et al.'s work, there had been a long line of research on approximability of CLIQUE [Bel+93; FK00; BGS98; BS94], which culminated in Håstad's work [Hås96]. In [Hås96], it was shown that CLIQUE cannot be approximated to within a factor of $n^{1-\varepsilon}$ in polynomial time unless $\mathsf{NP} \subseteq \mathsf{ZPP}$; this was later derandomized by Zuckerman who showed a similar hardness under the assumption $\mathsf{NP} \nsubseteq \mathsf{P}$ [Zuc07]. Since then, better inapproximability ratios are known [EH00; Kho01; KP06], with the best ratio being $n/2^{(\log n)^{3/4+\varepsilon}}$ for every $\varepsilon > 0$ (assuming $\mathsf{NP} \nsubseteq \mathsf{BPTIME}(2^{(\log n)^{O(1)}})$) due to Khot and Ponnuswami [KP06].

The parameterized variant of the problem, denoted by k-CLIQUE, is known to be complete for the class W[1], rendering the problem intractable even in the parameterized version. Chen et al.shows a nearly tight running time lower bound, which rules out any $T(k) \cdot n^{o(k)}$-time algorithm for k-CLIQUE for any function T [Che+04; Che+06]. This matches the trivial $n^{O(k)}$ algorithm to within a constant factor in the exponent.

Given that each of the two techniques alone does not seem to make the problem tractable, it has been asked whether it is possible to combine approximation and parameterization to achieve some non-trivial algorithm for the problem. We note here that, unlike k-DOMSET, the trivial algorithm for k-CLIQUE already gives k-approximation. Hence, we consider the problem *totally FPT inapproximable* if there is no $o(k)$-approximation algorithm that runs in FPT time. (See Section 2.6.2.)

Research Question 3. *Is k-CLIQUE totally FPT inapproximable?*

The only known hardness prior to our work is that of Bonnet et al. [Bon+15] who showed that k-CLIQUE is hard to approximate to within any constant factor under Gap-ETH. There had also been an attempt to prove hardness of approximation of k-CLIQUE under a different assumption in parameterized complexity [KS16], although this assumption turned out to be false [Kay14].

Another problem consider in this chapter is the problem of finding maximum induced subgraph with hereditary property. Recall that a property Π is said to be *hereditary* if, for all $G \in \Pi$, all induced subgraphs of G also belong to Π. For instance, Π could be "planarity" or "3-colorability". In the MAXIMUM INDUCED SUBGRAPH WITH PROPERTY Π problem, we are given a graph G and we would like to find a largest set of vertices $S \subseteq V(G)$ such that the induced subgraph $G[S]$ belongs to Π. Note that this problem contains MAXIMUM CLIQUE as a special case, since we can simply set Π to be the set of all cliques.

The complexity of finding and approximating maximum subgraph with hereditary properties have also been studied since the 1980s [LY80; LY93; FK05]; specifically, Feige and Kogan showed that, for every non-trivial property Π (i.e., Π such that infinite many subgraphs satisfy Π and infinitely many subgraphs do not satisfy Π), the problem is hard to approximate to within $n^{1-\varepsilon}$ factor for every $\varepsilon > 0$ unless $\mathsf{NP} \subseteq \mathsf{ZPP}$ [FK05]. We also note that non-trivial approximation algorithms for the problem are known; for instance, when the property fails for some clique or some independent set, a polynomial time $O\left(\frac{n(\log\log n)^2}{(\log n)^2}\right)$-approximation algorithm is known [Hal00].

The parameterized variant of the problem, denoted by k-INDUCED SUBGRAPH WITH HEREDITARY PROPERTY, was studied by Khot and Raman [KR00] who proved the following dichotomy theorem. If Π contains all independent sets but not all cliques or if Π contains all cliques but not all independent sets, then the problem is W[1]-hard. Otherwise, the problem is in FPT.

Once again, similar to k-CLIQUE, we can also ask whether k-INDUCED SUBGRAPH WITH HEREDITARY PROPERTY is totally FPT inapproximability for these "hard" properties Π; or inversely, whether there are better than $o(k)$-FPTapproximation algorithms for the problem.

Our Results

The main result of this chapter is that k-CLIQUE is totally FPT inapproximable. Furthermore, we show an even stronger result that it is inherently enumerative. Recall from Section 2.6.3 that inherently enumerative states that the problem the trivial enumeration algorithm is essentially the best possible (up to a constant factor in the exponent), even in the approximation (i.e. gap) setting.

Our result in k-CLIQUE and all subsequent FPT inapproximability results in this part will be based on Gap-ETH. The (obvious) benefit of starting with Gap-ETH is that, unlike in the previous chapter, we now begin with gap in hardness of approximation. Hence, our tasks now amount to only retaining or amplifying this gap.

Similar to the previous chapter, the hardness is shown via a reduction from MAXCOV, except that this time our MAXCOV instance satisfies the projection property, which allows us to reduce to CLIQUE. The Gap-ETH-hardness of MAXCOV is proved in Section 7.1. Then, in Section 7.2, we show the inherently enumerativeness of MAXIMUM CLIQUE using the reduction from [Fei+91]. Finally, in Section 7.3, we argue why this also implies total FPT inapproximability of the problem of finding maximum induced subgraph with hereditary property (for similar “hard” properties Π as in [KR00]). Note here that, unlike for MAXIMUM CLIQUE, we only get weakly inherently enumerativeness for the latter problem, i.e., the running time lower bound achieved is not yet tight.

7.1 Hardness of Approximation from MAXCOV with Projection Property

One straightforward algorithm for MAXCOV is to enumerate all the possible right labeling σ_V, which takes $O^\star(|\Sigma_V|^{|V|})$ time, for which our hardness in the previous section matches. The other natural straightforward algorithm to determine whether $\text{MAXCOV}(\mathcal{L}) < r$ is to enumerate all possible subsets $S \subseteq U$ of size r and the possible S-labeling $\sigma_S : S \to \Sigma_U$; this runs in $O^\star((|U| \cdot |\Sigma_U|)^r)$ time. We will show that this algorithm is also essentially the best possible, as stated below. This will serve as the starting point of all hardness results in this section.

Theorem 7.1 (MAXCOV with Projection Property)**.** *Assuming Gap-ETH, there exist constants $\delta, \rho > 0$ such that, for any positive integers $k \geqslant r \geqslant \rho$, no algorithm can take a label cover instance $\mathcal{L}$ with $|U| = k$ and distinguish between the following cases in $O_{k,r}(|\mathcal{L}|^{\delta r})$ time:*

- $\text{MAXCOV}(\mathcal{L}) = k$ *and*
- $\text{MAXCOV}(\mathcal{L}) < r$.

This holds even when $|\Sigma_V| = O(1)$ and Π has the projection property.

We also note here that the label cover instances above has the projection property, unlike the ones from the previous section. As we will see soon, this projection property is crucial in the reduction from MAXCOV to Maximum Clique.

Towards proving Theorem 7.1, first observe that, by applying the clause-variable reduction (Definition 3.20), ETH can be restated as the following hardness of MAXCOV for small alphabet:

Observation 7.2. *Assuming Gap-ETH (Hypothesis 3), there exist constants $\varepsilon, \delta > 0$ such that no algorithm can take a label cover instance Γ and can distinguish between the following cases in $O(2^{\delta|U|})$ time:*

- $\textsc{MaxCov}(\Gamma) = |U|$, *and*
- $\textsc{MaxCov}(\Gamma) < (1 - \varepsilon)|U|$.

This holds even when $|\Sigma_U|, |\Sigma_V| = O(1)$, $|U| = \Theta(|V|)$ and Π has the projection property.

The proof of Theorem 7.1 proceeds by *compressing the left vertex set U* of a label cover instance from Observation 7.2. More specifically, each new left vertex will be a subset of left vertices in the original instance. One could think of these subsets as random subsets where each vertex is included with probability $\Theta(1/k)$; however, the only property of random subsets we will need is that they form a disperser, as defined in Definition 2.20.

The idea of using dispersers to amplify gap in hardness of approximation bears a strong resemblance to the classical randomized graph product technique [BS92]. Indeed, similar approaches have been used before, both implicitly [BGS98] and explicitly [Zuc96b; Zuc96a; Zuc07]. In fact, even the reduction we use below has been studied before by Zuckerman [Zuc96b; Zuc96a]!

What differentiates our proof from previous works is the setting of parameters. Since the reduction size (specifically, the left alphabet size $|\Sigma_U|$) blows up exponentially in the subset size and previous results aim to prove NP-hardness of approximating $\textsc{Clique}$, the subset sizes are chosen to be small (i.e. $O(\log |U|)$). On the other hand, we will choose it to be $\Theta_\varepsilon(|U|/r)$ since we would like to only prove a running time lower bound of the form $|\mathcal{L}|^{\Omega(r)}$. Interestingly, dispersers for our regime of parameters are easier to construct deterministically by slightly modifying the sets from Section 2.9. The exact dependency of parameters can be found in the claim below. Throughout the proof, k, r should be thought of as constants where $k \gg r$; these are the same k, r as the ones in Theorem 7.1.

Claim 7.3 (Deterministic Construction of Dispersers)**.** *For any $k, q \in \mathbb{N}$ and any integer $m \geqslant q^{k+1}$, let $\mathcal{U}$ be any m-element set. Then, there is a collection $\mathcal{I} = \{I_1, \dots, I_k\}$ of k subsets of $\mathcal{U}$ with the following properties with $\alpha := 1/q$.*

- *(Size) Each of $I_1, \dots, I_k$ has size at most $2\alpha m$.*
- *(Disperser) For any $\eta > 0$, $\mathcal{I}$ is a $(\lceil \ln(1/\eta)/\alpha \rceil, \eta)$-disperser.*

Moreover, such a collection $\mathcal{T}$ can be deterministically constructed in time $O(m \cdot q^k)$.

Proof. Let $z = \lfloor m/q^k \rfloor$. To define the sets, we first partition $\mathcal{U}$ into two parts $\mathcal{U}^0, \mathcal{U}^1$, where $\mathcal{U}^0$ is of size $q^k \cdot z$ and $\mathcal{U}^1$ is of size $m - |\mathcal{U}^0| < q^k$. We associate the elements of $\mathcal{U}^0$ with $[q]^k \times [z]$. Let $T_1, \dots, T_k$ be the sets as in Definition 2.16. We define the set $I_1, \dots, I_k \subseteq \mathcal{U}$ by $I_i = (T_i \times [z]) \cup \mathcal{U}^1$. The disperser property of $\mathcal{I} = \{I_1, \dots, I_k\}$ follows immediately from Proposition 2.22 (with $\ell = 1$). Finally, each set is of size at most $q^{k-1} \cdot z + q^k \leqslant 2\alpha m$, where the inequality comes from our assumption that $m \geqslant q^{k+1}$. □

With the above claim ready, we move on to prove Theorem 7.1.

Proof of Theorem 7.1. First, we take a label cover instance $\widetilde{\mathcal{L}} = (\widetilde{G} = (\widetilde{U}, \widetilde{V}, \widetilde{E}), \Sigma_{\widetilde{U}}, \Sigma_{\widetilde{V}}, \widetilde{\Pi})$ from Observation 7.2, where $|\Sigma_{\widetilde{U}}|, |\Sigma_{\widetilde{V}}| = O(1)$ and $|\widetilde{U}| = \Theta(|\widetilde{V}|)$. Moreover, let us rename the vertices in $\widetilde{U}$ and $\widetilde{V}$ so that $\widetilde{U} = [m]$ and $\widetilde{V} = [n]$. Note that it might be useful for the readers to think of $\widetilde{\mathcal{L}}$ as a 3-SAT instance where $\widetilde{U}$ is the set of clauses and $\widetilde{V}$ is the set of variables.

We recall the parameter ε from Observation 7.2 and the parameters k, r from the statement of Theorem 7.1. We also introduce a new parameter $q = \lfloor k/\ln(1/\varepsilon) \rfloor$, and let $\alpha = 1/q$.

The new label cover instance $\mathcal{L} = (G = (U, V, E), \Sigma_U, \Sigma_V, \Pi)$ is defined as follows.

- The right vertices and right alphabet set remain unchanged, i.e., $V = \widetilde{V}$ and $\Sigma_V = \Sigma_{\widetilde{V}}$.
- There will be k vertices in U, each corresponding to a set I_i as constructed[1] by Claim 7.3 with q as specified above and universe $\mathcal{U} = [m]$.
- The left alphabet set Σ_U is $\Sigma_{\widetilde{U}}^{\lceil 2\alpha m \rceil}$. For each $I \in U$, we view each label $\alpha \in \Sigma_U$ as a tuple $(\alpha_u)_{u \in I} \in (\Sigma_{\widetilde{U}})^I$; this is a partial assignment to all vertices $u \in I$ in the original instance $\widetilde{\Gamma}$.
- We create an edge between $I \in U$ and $v \in V$ in E if and only if there exists $u \in I$ such that $uv \in \widetilde{E}$. More formally, $E = \{(I, v) \mid I \cap N_{\widetilde{G}}(v) \neq \emptyset\}$.
- Finally, we define the constraint $\Pi_{(I,v)}$ for each $(I, v) \in E$. As stated above, we view each $\alpha \in \Sigma_U$ as a partial assignment $(\alpha_u)_{u \in I}$ for $I \subseteq \widetilde{U}$. The constraint $\Pi_{(I,v)}$ then contains all (α, β) such that (α_u, β) satisfies the constraint $\widetilde{\Pi}_{uv}$ for every $u \in I$ that has an edge to v in $\widetilde{\Gamma}$. More precisely, $\Pi_{(I,v)} = \{(\alpha, \beta) = ((\alpha_u)_{u \in I}, \beta) \mid \forall u \in I \cap N_{\widetilde{G}}(v), (\alpha_u, \beta) \in \widetilde{\Pi}_{(u,v)}\}$.

Readers who prefer the 3-SAT/CSP viewpoint of label cover may think of each I_i as a collection of clauses in the 3-SAT instance that are joined by an operator **AND**, i.e., the assignment must satisfy all the clauses in I_i simultaneously in order to satisfy I_i.

We remark that, if $\widetilde{\Pi}$ has the projection property, Π also has projection property.

Completeness. Suppose there is a labeling $(\sigma_{\widetilde{U}}, \sigma_{\widetilde{V}})$ of $\widetilde{\mathcal{L}}$ that covers all $|\widetilde{U}|$ left-vertices. We take $\sigma_V = \sigma_{\widetilde{V}}$ and construct σ_U by setting $\sigma_U(I) = (\sigma_{\widetilde{U}}(u))_{u \in I}$ for each $I \in U$. Since $(\sigma_{\widetilde{U}}, \sigma_{\widetilde{V}})$ covers all the vertices of $\widetilde{U}$, (σ_U, σ_V) also covers all the vertices of U. Therefore, $\textsc{MaxCov}(\Gamma) = |U|$.

Soundness. To analyze the soundness of the reduction, observe that Claim 7.3 implies that $\{I_1, \dots, I_k\}$ is an (r, ε)-disperser. Conditioned on this event happening, we will prove the soundness property, i.e., that if $\textsc{MaxCov}(\widetilde{\mathcal{L}}) < (1 - \varepsilon)|\widetilde{U}|$, then $\textsc{MaxCov}(\mathcal{L}) < r$.

We will prove this by contrapositive. Assume that there is a labeling (σ_U, σ_V) that covers at least r left vertices $I_{i_1}, \cdots, I_{i_r} \in U$. We construct a labeling $(\sigma_{\widetilde{U}}, \sigma_{\widetilde{V}})$ as follows. First, $\sigma_{\widetilde{V}}$ is simply set to σ_V. Moreover, for each $u \in I_{i_1} \cup \cdots \cup I_{i_r}$, let $\sigma_{\widetilde{U}}(u) = (\sigma_U(I_{i_j}))_u$ where $j \in [r]$ is an index such that $u \in I_{i_j}$; if there are multiple such j's, just pick an arbitrary one. Finally, for $u \in U \setminus (I_{i_1} \cup \cdots \cup I_{i_r})$, we set $\sigma_{\widetilde{U}}(u)$ arbitrarily.

[1] The assumption $m \geqslant q^{k+1}$ can be assumed w.l.o.g. since both q, k are constrants in our setting.

We claim that, every $u \in I_{i_1} \cup \cdots \cup I_{i_r}$ is covered by $(\sigma_{\widetilde{U}}, \sigma_{\widetilde{V}})$ in the original instance $\widetilde{\mathcal{L}}$. To see that this is the case, recall that $\sigma_{\widetilde{U}}(u) = (\sigma_U(I_{i_j}))_u$ for some $j \in [r]$ such that $u \in I_{i_j}$. For every $v \in V$, if $(u, v) \in E$, then, from how the constraint $\Pi_{(I_{i_j}, v)}$ is defined, we have $(\sigma_{\widetilde{U}}(u), \sigma_{\widetilde{V}}(v)) = (\sigma_U(I_{i_j})_u, \sigma_V(v)) \in \widetilde{\Pi}_{uv}$. In other words, u is indeed covered by $(\sigma_{\widetilde{U}}, \sigma_{\widetilde{V}})$.

Hence, $(\sigma_{\widetilde{U}}, \sigma_{\widetilde{V}})$ covers at least $|I_{i_1} \cup \cdots \cup I_{i_r}| \geqslant (1 - \varepsilon)m$, where the inequality comes from the definition of dispersers. Thus, MaxCov$(\widetilde{\Gamma}) \geqslant (1 - \varepsilon)|\widetilde{U}|$, completing the soundness proof.

Running Time Lower Bound. Our construction gives a MaxCov instance $\mathcal{L}$ with $|U| = k$ and $|\Sigma_U| = |\Sigma_{\widetilde{U}}|^{\lceil 2\alpha m \rceil} = 2^{\Theta(m \ln(1/\varepsilon)/r)}$, whereas $|V|$ and $|\Sigma_V|$ remain n and $O(1)$ respectively. Assume that Gap-ETH holds and let δ_0 be the constant in the running time lower bound in Observation 7.2. Let δ be any constant such that $0 < \delta < \frac{\delta_0}{c \ln(1/\varepsilon)}$ where c is the constant such that $|\Sigma_U| \leqslant 2^{cm \ln(1/\varepsilon)/r}$.

Suppose for the sake of contradiction that, for some $k \geqslant r \geqslant 1/\delta$, there is an algorithm that distinguishes whether MaxCov$(\mathcal{L}) = k$ or MaxCov$(\mathcal{L}) < r$ in $O_{k,r}(|\mathcal{L}|^{\delta r})$ time. Observe that, in our reduction, $|U|, |V|, |\Sigma_V| = |\Sigma_U|^{o(1)}$. Hence, the running time of the algorithm on input Γ is at most $O_{k,r}(|\Sigma_U|^{\delta r(1+o(1))}) \leqslant O_{k,r}(|\Sigma_U|^{\delta_0 \varepsilon r/c}) \leqslant O(2^{\delta_0 m})$ where the first inequality comes from our choice of δ and the second comes from $|\Sigma_U| \leqslant 2^{cm \ln(1/\varepsilon)/r}$. Thanks to the completeness and soundness of the reduction, this algorithm can also distinguish whether MaxCov$(\widetilde{\mathcal{L}}) = |\widetilde{U}|$ or MaxCov$(\widetilde{\Gamma}) < (1 - \varepsilon)|\widetilde{U}|$ in time $O(2^{\delta_0 m})$. From Observation 7.2, this is a contradiction. □

7.2 Maximum Clique

We will next prove our hardness for parameterized Maximum Clique. Observe that we can check if there is a clique of size r by checking if any subset of r vertices forms a clique, and there are $\binom{|V(G)|}{r} = O(|V(G)|^r)$ possible such subsets. We show that this is essentially the best we can do even when we are given a promise that a clique of size $q \gg r$ exists:

Theorem 7.4. *Assuming Gap-ETH, there exist constants $\delta, r_0 > 0$ such that, for any positive integers $q \geqslant r \geqslant r_0$, no algorithm can take a graph $G = (V, E)$ and distinguish between the following cases in $O_{q,r}(|V|^{\delta r})$ time:*

- Clique$(G) \geqslant q$ *and*
- Clique$(G) < r$.

The above theorem simply follows from plugging the FGLSS reduction below to Theorem 7.1.

Theorem 7.5 ([Fei+91])**.** *Given a label cover instance $\mathcal{L} = (G = (U, V, E), \Sigma_U, \Sigma_V, \Pi)$ with projection property, there is a reduction that produces a graph $H_{\mathcal{L}} = (V_{\mathcal{L}}, E_{\mathcal{L}})$ such that $|V_{\mathcal{L}}| = |U||\Sigma_U|$ and* Clique$(H_{\mathcal{L}}) =$ MaxCov$(\mathcal{L})$*. The reduction takes $O(|V_{\mathcal{L}}|^2 \cdot |V|)$ time.*

For clarity, we would like to note that, while the original graph defined in [Fei+91] is for multi-prover interactive proof, analogous graphs can be constructed for CSPs and label cover instances as well. In particular, in our case, the graph $H_{\mathcal{L}} = (V_{\mathcal{L}}, E_{\mathcal{L}})$ can be defined as follows:

- The vertex set $V_{\mathcal{L}}$ is simply $U \times \Sigma_U$.
- There is an edge between two vertices $(u, \alpha), (u', \alpha') \in V_{\mathcal{L}}$ if and only if, $\Pi_{(u,v)}(\alpha) = \Pi_{(u',v)}(\alpha')$ (i.e., recall that we have a projection constraint, so we can represent the constraint $\Pi_{(u,v)}$ as a function $\Pi_{(u,v)} : \Sigma_U \to \Sigma_V$.)

Proof of Theorem 7.4. Assume that Gap-ETH holds and let δ, ρ be the constants from Theorem 7.1. Let $r_0 = \max\{\rho, 2/\delta\}$. Suppose for the sake of contradiction that, for some $q \geqslant r \geqslant r_0$, there is an algorithm $\mathbb{A}$ that distinguishes between $\text{CLIQUE}(G) \geqslant q$ and $\text{CLIQUE}(G) < r$ in $O_{q,r}(|V(G)|^{\delta r})$ time.

Given a label cover instance $\mathcal{L}$ with projection property, we can use $\mathbb{A}$ to distinguish whether $\text{MAXCOV}(\mathcal{L}) \geqslant q$ or $\text{MAXCOV}(\mathcal{L}) < r$ as follows. First, we run the FGLSS reduction to produce a graph $H_{\mathcal{L}}$ and we then use $\mathbb{A}$ to decide whether $\text{CLIQUE}(H_{\mathcal{L}}) \geqslant q$ or $\text{CLIQUE}(H_{\mathcal{L}}) < r$. From $\text{CLIQUE}(H_{\mathcal{L}}) = \text{MAXCOV}(\mathcal{L})$, this indeed correctly distinguishes between $\text{MAXCOV}(\mathcal{L}) \geqslant q$ and $\text{MAXCOV}(\mathcal{L}) < r$; moreover, the running time of the algorithm is $O_{q,r}(|V_{\mathcal{L}}|^{\delta r}) + O(|V_{\mathcal{L}}|^2 \cdot |\mathcal{L}|) \leqslant O_{q,r}(|\mathcal{L}|^{\delta r})$ where the term $O(|V_{\mathcal{L}}|^2 \cdot |\mathcal{L}|)$ comes from the running time used to produce $H_{\mathcal{L}}$. From Theorem 7.1, this is a contradiction, which concludes our proof. □

As a corollary of Theorem 7.4, we immediately arrive at FPT inapproximability of k-CLIQUE:

Corollary 7.6 (Clique is inherently enumerative)**.** *Assuming Gap-ETH,* MAXIMUM CLIQUE *is inherently enumerative and thus totally FPT inapproximable.*

7.3 Maximum Induced Subgraph with Hereditary Properties

In this section, we prove the hardness of maximum induced subgraphs with hereditary property. Let Π be a graph property. We say that a subset $S \subseteq V(G)$ has property Π if $G[S] \in \Pi$. Denote by $A_\Pi(G)$ the maximum cardinality of a set S that has property Π.

Recall that Khot and Raman [KR00] proved a dichotomy theorem for the problem: if Π contains all independent sets but not all cliques or if Π contains all cliques but not all independent sets, then the problem is W[1]-hard. For all other Π's, the problem is in FPT. We extend Khot and Raman's dichotomy theorem to hold even for FPT approximation as stated more precisely below.

Theorem 7.7. *Let Π be any hereditary property.*

- *If Π contains all independent sets but not all cliques or vice versa, then computing $A_\Pi(G)$ is weakly inherently enumerative (and therefore totally FPT inapproximable).*
- *Otherwise, $A_\Pi(G)$ can be computed exactly in FPT.*

Surprisingly, the fact that there is a gap in the optimum of our starting point helps make our reduction simpler than that of Khot and Raman. For convenience, let us focus only on properties Π's which contain all independent sets but not all cliques. The other case can be proved analogously. The main technical result is summarized in the following lemma.

Theorem 7.8. *Let Π be any graph property that contains all independent sets but not all cliques. Then there is a function $g_\Pi = \omega(1)$ such that the following holds:*

- *If $\alpha(G) \geqslant q$, then $A_\Pi(G) \geqslant q$.*
- *If $A_\Pi(G) \geqslant r$, then $\alpha(G) \geqslant g_\Pi(r)$.*

Proof. Since Π contains all independent set, when $\alpha(G) \geqslant q$, we always have $A_\Pi(G) \geqslant q$.

Now, to prove the converse, let $g_\Pi(r)$ denote $\max_{H \in \Pi, |V(H)|=r} \alpha(H)$. If $A_\Pi(G) = r$, then there exists a subset $S \subseteq V(G)$ of size r that has property Π; from the definition of g_Π, $\alpha(H) \geqslant g_\Pi(r)$, which implies that $\alpha(G) \geqslant g_\Pi(r)$ as well. Hence, we are only left to show that $g_\Pi = \omega(1)$.

To show that this is the case, recall the Ramsey theorem.

Theorem 7.9 (Ramsey's Theorem). *For any $s, t \geqslant 1$, there is an integer $R(s,t)$ s.t. every graph on $R(s,t)$ vertices contains either a s-clique or a t-independent set. Moreover, $R(s,t) \leqslant \binom{s+t-2}{s-1}$.*

From our assumption of Π, there exists a fixed integer s_Π such that Π does not contain an s_Π-clique. Hence, from Ramsey's Theorem, $g_\Pi(r) \geqslant \max\{t \mid R(s_\Pi, t) \leqslant r\}$. In particular, this implies that $g_\Pi(r) \geqslant \Omega_{s_\Pi}(r^{1/(s_\Pi - 1)})$. Hence, $\lim_{r\infty} g_\Pi(r) = \infty$ (i.e. $g_\Pi = \omega(1)$) as desired. □

In other words, the identical transformation $G \mapsto G$ is a $(q, g_\Pi(r))$-FPT gap reduction from CLIQUE to Maximum Induced Subgraph with property Π. Hence, by applying Proposition 2.12, we immediately arrive at the following corollary.

Corollary 7.10. *Assuming Gap-ETH, for any property Π that contains all independent sets but not all cliques (or vice versa),* MAXIMUM INDUCED SUBGRAPH WITH PROPERTY Π *is $\Omega(g_\Pi)$-weakly inherently enumerative where g_Π is the function from Theorem 7.8.*

We remark here that, for some properties, g_Π can be much larger than the bound given by the Ramsey's Theorem; for instance, if Π is planarity, then the Ramsey's Theorem only gives $g_\Pi(r) = \Omega(r^{1/5})$ but it is easy to see that, for planar graphs, there always exist an independent set of linear size and $g_\Pi(r)$ is hence as large as $\Omega(r)$.

7.4 Discussion and Open Questions

In this chapter, we prove total FPT inapproximability of MAXIMUM CLIQUE and the problem of finding maximum subgraph with hereditary properties (for the "hard" properties). Since these results (and all subsequent results in this part) are based on Gap-ETH, the obvious question is whether we can relax the assumption to ETH or even W[1] $\neq$ FPT. We refrain from discussing this issue here, but rather provide a more complete view in Chapter 12. (See Directions 3 and 4.)

Another interesting question, which is slightly beyond parameterized complexity, is how far we can push q in Theorem 7.4 in terms of $n = |V|$. Of course, this is not the usual settings in parameterized complexity since we typically view q as either a constant or another parameter independet of n. However, it is not hard to check that our reduction in fact gives a lower bound even when $q = n^{\gamma}$ for some (small) constant $\gamma > 0$ that depends on the parameter in Gap-ETH.

Furthermore, recall that, in the NP-hardness regime, strong NP-hardness of approximation with factor $n^{1-\varepsilon}$ is known for any $\varepsilon > 0$ [Hås96; Zuc07]. These results implies that, there is no polynomial time algorithm that, given a graph G, can distinguish between $\textsc{Clique}(G) \geqslant n^{1-\varepsilon}$ and $\textsc{Clique}(G) \leqslant n^{\varepsilon}$. As a result, we could ask, in the "parameterized" setting, whether we can push q all the way to $n^{1-\varepsilon}$. In other words, the question here can be phrased as follows:

Open Question 2. *Let $\varepsilon > 0$ be any constant. Is there an FPT (in k) time algorithm that can distinguish between* $\textsc{Clique}(G) \geqslant n^{1-\varepsilon}$ *and* $\textsc{Clique}(G) < k$*?*

If the answer to this question is negative, then the proof might involve combining the technique in this chapter with the aforementioned NP-hardness results. The latter involves constructing PCP with small free bits (see [BGS98; Hås96] for definition); however, such constructions require the starting hardness of label cover to have a large gap. It is unclear how to get such a large gap from Gap-ETH without using parallel repetition [Raz98]; however, doing so results in a label cover instance of size $N \geqslant \Omega(n^2)$ and hence the running time lower bound is of the form $2^{\Omega(\sqrt{N})}$, which breaks down the reductions in this chapter.

Chapter 8

Inapproximability from Gap-ETH II: k-Biclique, k-Induced Matching on Bipartite Graphs and Densest k-Subgraph

We continue our study of parameterized approximability. In this chapter, we consider the following three problems: MAXIMUM BALANCED BICLIQUE, MAXIMUM INDUCED MATCHING on bipartite graphs and DENSEST k-SUBGRAPH.

Maximum Balanced Biclique. In the MAXIMUM BALANCED BICLIQUE problem, we are given a bipartite graph G and would like to find the largest k such that the k-biclique $K_{k,k}$ is a subgraph of G. NP-hardness for the exact version of the problem was stated as (without proof) in [GJ79, page 196]; several proofs of this exist such as one provided in [Joh87]. While this problem bears a strong resemblance to the MAXIMUM CLIQUE Problem, inapproximability of the latter cannot be directly translated to that of the former; in fact, despite numerous attempts, not even constant factor NP-hardness of approximation of the Maximum Balanced Biclique problem is known. Fortunately, under stronger assumptions, hardness of approximation for the problem is known: n^{ε}-factor hardness of approximation is known under Feige's random 3SAT hypothesis [Fei02] or $\mathsf{NP} \not\subseteq \bigcap_{\varepsilon>0}\mathsf{BPTIME}(2^{n^{\varepsilon}})$ [Kho06], and $n^{1-\varepsilon}$-factor hardness of approximation is known under strengthening of the Unique Games Conjecture [Bha+16a; Man17b]. To the best of our knowledge, no non-trivial approximation algorithm for the problem is known.

The parameterized version of the problem, denoted by k-BICLIQUE, had been a well-known open problem in the field of parameterized complexity [DF13]. It was not until a few years ago that the exact version of the problem is shown to be W[1]-hard in the breakthrough work of Lin [Lin15]. Lin's proof also implies that, assuming the randomized ETH, the problem does not admit $T(k) \cdot n^{o(\sqrt{k})}$-time (exact) algorithm.

Maximum Induced Matching on Bipartite Graphs. In the MAXIMUM INDUCED MATCHING problem, we are given a graph G and the goal is to find a maximum number of vertices that induce a matching in G. The problem was proved to be NP-hard independently by Stockmeyer and Vazirani [SV82] and Cameron [Cam89]. The approximability of the problem was first studied by Duckworth et al. [DMZ05] who showed that the problem is APX-hard, even on bipartite graphs of degree three. Elbassioni et al. [Elb+09] then showed that the problem is hard to approximate to within $n^{1/3-\varepsilon}$ factor for every $\varepsilon > 0$, unless NP$\subseteq$ ZPP. Chalermsook et al. [CLN13] later improved the ratio to $n^{1-\varepsilon}$ for every $\varepsilon > 0$.

The parameterized version of the problem, denoted by k-INDUCED MATCHING, has also been studied. In particular, the problem was shown to be W[1] to solve exactly in [MT09], and that this remains true even when restricted to bipartite input graphs [MS09]. In fact, the reduction in [MT09] is from k-CLIQUE, and the produced graph always have maximum induced matching of size exactly two times the size of the maximum clique in the original graph. Hence, our result in the previous chapter immediately implies that the problem is totally FPT inapproximable on general graphs. As a result, we will focus on the FPT approximability of k-INDUCED MATCHING on bipartite graphs, for which the reduction in [MT09] does not trivially yield such a hardness.

Densest k-Subgraph. Chapter 4 provides a rather comprehensive literature review on the (non-parameterized) DENSEST k-SUBGRAPH (DkS), and hence we do not repeat it here. The parameterized version of DkS (where k is the parameter) is clearly W[1]-hard to solve exactly, since it generalizes k-CLIQUE. On the other hand, to the best of our knowledge, no parameterized hardness of approximation was known before.

We also note here that there is a rather straightforward k-approximation algorithm for the problem: pick a vertex with highest degree and select $k-1$ of its neighbor. (If it has less than $k-1$ neighbors, then just put all its neighbors in the set.) While the algorithm is very simple, there is no known FPT algorithm that gives $o(k)$-approximation for DkS.

Our Results

In this chapter, we show, assuming Gap-ETH, that both k-BICLIQUE and k-INDUCED MATCHING on bipartite graphs are totally FPT inapproximable, by showing that their corresponding optimization problems are weakly inherently enumerative. Note here that, unlike MAX CLIQUE in the previous chapter, the running time lower bounds for these problems that we achieve are not yet tight. In particular, we prove $\Omega(\sqrt{r})$-weakly inherently enumerativeness for the problems, while it could still be possible that the problems are in fact inherently enumerative (or equivalently $\Omega(r)$-weakly inherently enumerative). Nonetheless, our results already implies, for instance, that approximating k-BICLIQUE to within any constant factor requires $n^{\Omega(k)}$ time, which matches the best known running time lower bound even for the *exact* version of the problem from [Lin15].

We also observe that the total FPT inapproximability almost immediately implies hardness of approximation for DkS to within a factor of $k^{o(1)}$ that holds even against FPT algorithms. Note that, unlike all of our previous FPT hardness of approximation results, the inapproximability ratio here is not yet tight, as the best known algorithm achieves only $O(k)$-approximation.

Unlike the previous two chapters, we will not reduce from any label cover problem; the starting point for the results here will instead be the reduction from Chapter 4. By interpreting this construction in a different perspective, we can modify it in such a way that we arrive at a stronger form of inherently enumerative hardness for CLIQUE. This is done in Section 8.1. In Section 8.2, we argue why this gives the weakly inherently enumerativeness for MAXIMUM BALANCED BICLIQUE. Then, we show how to reduce to MAXIMUM INDUCED MATCHING on bipartite graphs in Section 8.2.1. Finally, in Section 8.3, we show prove FPT hardness of approximation of DkS.

8.1 Rephrasing the Reduction from Chapter 4 as a Parameterized Inapproximability of Clique-vs-Biclique

The main theorem of this section is the following theorem, which is a stronger form of Theorem 7.4 in that the soundness not only rules out cliques, but also rules out bicliques as well.

Theorem 8.1. *Assuming randomized Gap-ETH, there exist constants $\delta, \rho > 0$ such that, for any positive integers $q \geqslant r \geqslant \rho$, no algorithm can take a graph G and distinguish between the following cases in $O_{q,r}(|V(G)|^{\delta\sqrt{r}})$ time:*

- CLIQUE$(G) \geqslant q$.
- BICLIQUE$(G) < r$.

The weakly inherently enumerativeness (and therefore totally FPT inapproximability) of MAXIMUM BALANCED BICLIQUE and MAXIMUM INDUCED MATCHING on bipartite graphs follows easily from Theorem 8.1. We will show these results in the subsequent sections; for now, let us turn our attention to the proof of the theorem.

The theorem is again shown via a reduction from Gap-3SAT. The properties and parameters of the reduction is stated below in Theorem 8.2. It is obvious to see Theorem 8.2 implies Theorem 8.1.

Theorem 8.2. *For any $d, \varepsilon > 0$, there is a constant $\gamma = \gamma(d, \varepsilon) > 0$ such that there exists a randomized reduction that takes in a parameter r and a 3-*SAT *instance ϕ with n variables and m clauses where each variable appears in at most d constraints and produces a graph $G_{\phi,r} = (V_{\phi,r}, E_{\phi,r})$ such that, for any sufficiently large r (depending only on d, ε but not n), the following properties hold with high probability:*

- *(Size) $N := |\widetilde{V}_{\phi,r}| \leqslant 2^{O_{d,\varepsilon}(n/\sqrt{r})}$.*
- *(Completeness) if ϕ is satisfiable, then* CLIQUE$(\widetilde{G}_{\phi,r}) \geqslant N^{\gamma/\sqrt{r}}$.
- *(Soundness) if* $\mathrm{val}(\phi) \leqslant 1 - \varepsilon$, *then* BICLIQUE$(\widetilde{G}_{\phi,r}) < r$.

As mentioned earlier, our result builds upon an intermediate lemma used to prove the hardness of approximating DENSEST k-SUBGRAPH in Chapter 4. Due to this, it will be easier to describe the new reduction in terms of the original reduction from Chapter 4; in this regard, our reduction can be viewed as vertex subsampling (with appropriate probability p) of the graph produced by the reduction from Chapter 4. Recall that, the main soundness result (Theorem 4.4) in Chapter 4 is there are few bicliques in the graph constructed there. As a result, if we select p appropriately, we should be able to make sure that these few bicliques do not remain in the subsampled graph.

Proof of Theorem 8.2. Let $\lambda < 1$ be the constant from Theorem 4.4. We select $\ell = \frac{4n}{\sqrt{\lambda r}}$ and $p = 2^{\frac{\lambda \ell^2}{2n}} / \binom{n}{\ell}$. Let $G_{\phi,\ell} = (V_{\phi,\ell}, E_{\phi,\ell})$ be the graph constructed in Section 4.1. Our graph $\widetilde{G}_{\phi,r} = (\widetilde{V}_{\phi,r}, \widetilde{E}_{\phi,r})$ is the induced subgraph of $G_{\phi,\ell}$, where each vertex is kept in $\widetilde{V}_{\phi,r}$ with probability p independent of each other.

Size. Since each vertex in $V_{\phi,\ell}$ is included that $\widetilde{V}_{\phi,r}$ independently w.p. p, we have $\mathbb{E}[|\widetilde{V}_{\phi,r}|] = p|V_{\phi,\ell}| = 2^{\ell + \frac{\lambda \ell^2}{2n}} \leqslant 2^{2\ell}$. Hence, from Chernoff bound, $|\widetilde{V}_{\phi,r}| \leqslant 2^{10\ell} = 2^{\Omega_{d,\varepsilon}(n/\sqrt{r})}$ w.h.p.

Completeness. Suppose that ϕ is satisfiable. Then, there exists a clique C of size $\binom{n}{\ell}$ in $G_{\phi,\ell}$. From how $\widetilde{G}_{\phi,r}$ is defined, $C \cap \widetilde{V}_{\phi,r}$ induces a clique in $\widetilde{G}_{\phi,r}$. Moreover, $\mathbb{E}[|C \cap \widetilde{V}_{\phi,r}|] = p|C| = 2^{\frac{\lambda \ell^2}{2n}}$. Again, from Chernoff bound, $\textsc{Clique}(\widetilde{G}_{\phi,r}) \geqslant 2^{\frac{\lambda \ell^2}{4n}}$ w.h.p. Combined with the above bound on N, $\textsc{Clique}(\widetilde{G}_{\phi,r}) \geqslant N^{\gamma/\sqrt{r}}$ w.h.p. for $\gamma := \sqrt{\lambda}/40 = O_{d,\varepsilon}(1)$.

Soundness. Suppose that $\mathrm{val}(\phi) \leqslant 1 - \varepsilon$. Consider any subsets $S, T \subseteq V_{\phi,\ell}$ that is a copy of $K_{r,r}$ in $G_{\phi,\ell}$. From how $\widetilde{G}_{\phi,r}$ is defined, $\textsc{Biclique}(\widetilde{G}_{\phi,r}) \geqslant r$ if and only if, for at least one such pair (S, T), $S \cup T \subseteq \widetilde{V}_{\phi,r}$. The probability of this event occurring is bounded above by

$$\sum_{\substack{S,T \subseteq V_{\phi,\ell} \\ S,T \text{ is a copy of } K_{r,r} \text{ in } G_{\phi,\ell}}} \Pr[S, T \subseteq \widetilde{V}_{\phi,r}] \leqslant 2^{4n} \left(2^{-\lambda \ell^2/n} \binom{n}{\ell}\right)^{2r} \cdot p^{2r} = 2^{4n} \left(2^{-\frac{\lambda \ell^2}{2n}}\right)^{2r} = o(1),$$

where the first inequality is from Theorem 4.4 and each vertex is included independently w.p. p.

As a result, the subsampled graph $\widetilde{G}_{\phi,r}$ is $K_{r,r}$-free with high probability as desired. □

8.2 Maximum Balanced Biclique

We now give a simple reduction from the "Clique vs Biclique" problem (from Chapter 8.1) to Maximum Balanced Biclique, which yields FPT inapproximability of the latter.

Lemma 8.3. *For any graph $G = (V, E)$, let $B_e[G] = (V_{B_e[G]}, E_{B_e[G]})$ be the bipartite graph whose vertex set is $V_{B_e[G]} := V \times [2]$ and two vertices $(u, i), (v, j)$ are connected by an edge if and only if $(u, v) \in E$ or $u = v$, and $i \neq j$. Then the following properties hold for any graph G.*

- $\textsc{Biclique}(B_e[G]) \geqslant \textsc{Clique}(G)$.
- $\textsc{Biclique}(B_e[G]) \leqslant 2\textsc{Biclique}(G) + 1$.

Proof. It is easy to see that $\textsc{Biclique}(B_e[G]) \geqslant \textsc{Clique}(G)$ since, for any $C \subseteq V$ that induces a clique in G, $C \times [2] \subseteq V_{B_e[G]}$ induces a $|C|$-biclique in $B_e[G]$.

To see that $\textsc{Biclique}(B_e[G]) \leqslant 2\textsc{Biclique}(G) + 1$, consider any $S \subseteq V_{B_e[G]}$ that induces a k-biclique in $B_e[G]$. Note that S can be partitioned into $S_1 = S \cap (V \times \{1\})$ and $S_2 = S \cap (V \times \{2\})$.

Now consider the projections of S_1 and S_2 into $V(G)$, i.e., $T_1 = \{v : (v, 1) \in S\}$ and $T_2 = \{v : (v, 2) \in S\}$. Note that $|T_1| = |T_2| = k$. Since $S_1 \cup S_2$ induces a biclique in $B_e[G]$, we have, for every $u \in T_1$ and $v \in T_2$, either $u = v$ or $(u, v) \in E$. Observe that if there were no former case (i.e., $T_1 \cap T_2 = \emptyset$), then we would have a k-biclique in G. Even if $T_1 \cap T_2 \neq \emptyset$, we can still get back a $\lfloor k/2 \rfloor$-biclique of G by uncrossing the sets T_1 and T_2 in a natural way by assigning half of the intersection to T_1 and the other half to T_2. To be formal, we partition $T_1 \cap T_2$ into roughly equal sets U_1 and U_2 (i.e., $||U_1| - |U_2|| \leqslant 1$), and we then define new sets T_1' and T_2' by

$$T_1' = (T_1 \setminus T_2) \cup U_1 \text{ and } T_2' = (T_2 \setminus T_1) \cup U_2.$$

It is not hard to see that G has an edge between every pair of vertices between T_1', T_2' and that $|T_1'|, |T_2'| \geqslant \lfloor k/2 \rfloor$. Thus, $\textsc{Biclique}(G) \geqslant \lfloor k/2 \rfloor \geqslant (k-1)/2$. Therefore, $\textsc{Biclique}(B_e[G]) \leqslant 2\textsc{Biclique}(G) + 1$ as desired. □

Thanks to the above lemma, we can conclude that the reduction $G \mapsto B_e[G]$ is a $(2q, (r+1)/2)$-FPT gap reduction from the "Clique vs Biclique" problem to Maximum Balanced Biclique, although the former is not a well-defined optimization problem. Nevertheless, it is easy to check that a proof along the line of Proposition 2.12 still works and it gives the following result:

Corollary 8.4. *Assuming randomized Gap-ETH,* Maximum Balanced Biclique *are $\Omega(\sqrt{r})$-weakly inherently enumerative and thus totally FPT inapproximable.*

It is worth noting here that the Maximum Edge Biclique problem, a well-studied variant of the Maximum Balanced Biclique problem where the goal is instead to find a (not necessarily balanced) complete bipartite subgraph of a given bipartite graph that contains as many edges as possible, is in FPT; this is because the optimum is at least the maximum degree, but, when the degree is bounded above by r, all bicliques can be enumerated in $2^{O(r)}\text{poly}(n)$ time.

8.2.1 Maximum Induced Matching on Bipartite Graphs

Next, we prove the FPT inapproximability for the Maximum Induced Matching problem on bipartite graphs. Again, the proof will be a simple reduction from Theorem 8.1. The argument below is similar to that used in Lemma IV.4 of [CLN13]. We include it here for completeness. (Here, we use $\text{IM}(G)$ to denote the number of edges in the maximum induced matching in G.)

Lemma 8.5. *For any graph $G = (V, E)$, let $B_e[\overline{G}] = (V_{B_e[\overline{G}]}, E_{B_e[\overline{G}]})$ be the bipartite graph whose vertex is $V_{B_e[\overline{G}]} := V \times [2]$ and two vertices $(u, i), (v, j)$ are connected by an edge if and only if $(u, v) \notin E$ or $u = v$, and $i \neq j$. Then, the following properties hold for any graph G.*

- $\text{IM}(B_e[\overline{G}]) \geqslant \textsc{Clique}(G)$.
- $\text{IM}(B_e[\overline{G}]) \leqslant 2\textsc{Biclique}(G) + 1$.

Proof. Consider any $S \subseteq V$ that induces a clique in G. It is obvious that $S \times [2] \subseteq V_{B_e[\overline{G}]}$ induces a matching in $B_e[\overline{G}]$.

Next, consider any induced matching matching $\{(u_1, v_1), \dots, (u_m, v_m)\}$ of size m. Assume w.l.o.g. that $u_1, \dots, u_m \in V \times \{1\}$ and $v_1, \dots, v_m \in V \times \{2\}$. Define $\pi_1 : V \times [2] \to V$ to be a projection operator that projects on to the first coordinate.

Let $S_1 = \pi_1(\{u_1, \dots, u_{\lfloor m/2 \rfloor}\})$ and $S_2 = \pi_1(\{v_{\lceil m/2 \rceil + 1}, \dots, v_m\})$. From the definition of $B_e[\overline{G}]$ and from the fact that there is no edge between $(S_1 \times \{1\})$ and $(S_2 \times \{2\})$, it is easy to check that $S_1 \cap S_2 = \emptyset$ and, for every $u \in S_1$ and $v \in S_2$, $(u, v) \in E$. In other words, (S_1, S_2) is an occurrence of $\lfloor m/2 \rfloor$ in G. Hence, we can conclude that $\text{IM}(B_e[\overline{G}]) \leqslant 2\textsc{Biclique}(G) + 1$. □

Similar to Biclique, it is easy to see that the above reduction implies the following FPT inapproximability for Maximum Induced Matching on Bipartite Graphs.

Corollary 8.6. *Assuming randomized Gap-ETH,* Maximum Induced Matching *on bipartite graphs are $\Omega(\sqrt{r})$-weakly inherently enumerative and thus totally FPT inapproximable.*

8.3 Densest k-Subgraph

Finally, we will show FPT inapproximability result for Densest k-Subgraph. Alas, we are not able to show $o(k)$-ratio FPT inapproximability, which would have been optimal since the trivial algorithm gives an $O(k)$-approximation for the problem. Nonetheless, we will show an $k^{o(1)}$-factor FPT inapproximability for the problem. We note here that below we will state the result as if k is the parameter; this is the same as using the optimum as the parameter, since (in the non-trivial case) the optimum is always between $\lfloor k/2 \rfloor$ and $\binom{k}{2}$ (inclusive).

To prove the hardness, recall the Kővári-Sós-Turán (KST) Theorem (Theorem 2.13), which basically states that if a graph does not contain small bicliques, then it is sparse. By applying KST Theorem to our hardness for Biclique (Theorem 8.1), we immediately arrive at the following.

Theorem 8.7. *Assuming Gap-ETH, there exist a constant $\delta > 0$ and an integer $\rho > 0$ such that, for any integer $q \geqslant r \geqslant \rho$, no algorithm can take a graph $G = (V, E)$ and distinguish between the following cases in $O_{q,r}(|V|^{\delta\sqrt{r}})$ time:*

- *G contains a q-clique.*
- *Every q-subgraph of G has density at most $O(q^{-1/r})$.*

From the above theorem, it is easy to show the $k^{o(1)}$-factor FPT inapproximability of Densest k-Subgraph (with perfect completeness) as formalized below.

Lemma 8.8. *Assuming randomized Gap-ETH, for every function $f = o(1)$ and every function t, there is no $t(k) \cdot n^{O(1)}$-time algorithm such that, given an integer k and any n-vertex graph G that contains a k-clique, always output $S \subseteq V$ of size k such that $G[S]$ has density at least $k^{-f(k)}$.*

Proof. Suppose for the sake of contradiction that there is a $t(k) \cdot |V|^D$-time algorithm $\mathbb{A}$ that, given an integer k and any graph $G = (V, E)$ that contains a k-clique, always outputs $S \subseteq V$ of size k with density at least $k^{-f(k)}$ for some function $f = o(1)$, some function t and some constant $D > 0$.

Let $r = \max\{\lceil \rho \rceil, \lceil (D/\delta)^2 \rceil\}$ where ρ is the constant from Chapter 8.7. Note that $O(q^{-1/r}) = q^{O(1)/\log q - 1/r}$. Now, since $\lim_{q \to \infty} f(q) + O(1)/\log q = 0$, there exists a sufficiently large q such that the term $O(q^{-1/r})$ is less than $q^{-f(q)}$. In other words, $\mathbb{A}$ can distinguish between the two cases in Chapter 8.7 in time $t(q) \cdot n^D = O_{q,r}(|V|^{\delta\sqrt{r}})$, which would break Gap-ETH. □

8.4 Discussion and Open Questions

In this chapter, we show total FPT inapproximability of k-BICLIQUE and k-INDUCED MATCHING on bipartite graphs. We further show $k^{o(1)}$ factor FPT hardness of approximation for DkS.

There are still many open questions remained. First, as discussed earlier before, the running time lower bound we get for k-BICLIQUE and k-INDUCED MATCHING are not yet tight, and they might actually be inherently enumerative. We remark here that, even for the exact version of k-BICLIQUE, an algorithm with running time $n^{o(k)}$ has not yet been rule out (even under say Gap-ETH). This presents us with the following question, which we have to answer first, before moving to tight running time lower bounds for approximation algorithms:

Open Question 3. *Is there a $T(k) \cdot n^{o(k)}$-time algorithm for (exact) k-BICLIQUE?*

In Chapter 11, we discuss slightly how techniques introduced in that chapter might help with the above question. However, there are still many steps needed to be done. For instance, we still do not know how to prove tight running time lower bound even for the ONE-SIDED k-BICLIQUE problem, which is discussed in more details in Section 11.8.

It should also be noted that, while we did not explicitly state the running time lower bounds for DkS, it was quite poor. For instance, even if we are only looking for some constant factor inapproximability result, the running time we can rule out is at most $T(k) \cdot n^{o(\log k)}$. The $\log k$ comes because, in order for KST Theorem to produce a constant gap, we must consider $r = O(\log k)$ size bicliques in the soundness. However, we can determine whether our graph contains such a biclique (and even list all of them) in $n^{O(r)} = n^{O(\log k)}$ time. Notice that this barrier holds, even when we get the tight lower bound for k-BICLIQUE. As a result, we are left with the following question; the answer of which likely requires an approach that does not involves using the KST Theorem to argue about the density.

Open Question 4. *Is there an $O(1)$-approximation $T(k) \cdot n^{o(k)}$-time algorithm for* DkS*?*

Apart from the running time lower bound, the inapproximability factor for DkS itself is still not tight. As mentioned earlier, whereas our hardness of approximation factor is $k^{o(1)}$, there is no known FPT time algorithm that achieves $o(k)$ approximation ratio. Hence, it is natural to ask:

Open Question 5. *Is there a $o(k)$-approximation FPT time algorithm for* $\mathrm{D}k\mathrm{S}$*?*

Next chapter can be considered an attempt to make a progress to the above question. In particular, we consider the PARAMETERIZED 2-CSP problem. While it will be convenient to state in a slightly different form in the next chapter, it can actually be stated as a *colorful* version of $\mathrm{D}k\mathrm{S}$, where, in addition to the graph G, each vertex is colored by one out of k colors, and we are now allowed to pick only one vertex of each color. There, we show that this problem is hard to approximate to within $k^{1-o(1)}$ factor. Nonetheless, it is currently unclear how the techniques developed there could help in improving hardness of approximation for $\mathrm{D}k\mathrm{S}$.

Finally, the reductions in this section is randomized, where the randomization comes from the vertex subsampling step. It is unclear to us how to derandomize the reductions, and we leave that as an open question as well.

Chapter 9

Inapproximability from Gap-ETH III: Parameterized 2-CSPs, Directed Steiner Network, k-Unique Set Cover

In this chapter, we study the 2-ary constraint satisfaction problems (2-CSPs). Recall from Chapter 3 that the 2-CSP problem can be stated as follows: given a constraint graph $G = (V, E)$, an alphabet set Σ and, for each edge $\{u, v\} \in E$, a constraint $C_{uv} \subseteq \Sigma \times \Sigma$, the goal is to find an assignment $\sigma : V \to \Sigma$ that satisfies as many constraints as possible, where a constraint C_{uv} is said to be satisfied by σ if $(\sigma(u), \sigma(v)) \in C_{uv}$. Throughout the chapter, we use k to denote the number of variables $|V|$, n to denote the the alphabet size $|\Sigma|$, and N to denote the instance size nk.

Constraint satisfaction problems and their inapproximability have been studied extensively since the proof of the PCP theorem in the early 90's [AS98; Aro+98]. Most of the effort has been directed towards understanding the approximability of CSPs with constant arity and constant alphabet size, leading to a reasonable if yet incomplete understanding of the landscape [Hås01; Kho02; Kho+07; Rag08; AM09; Cha16]. When the alphabet size grows, the sliding scale conjecture of [Bel+93] predicts that the hardness of approximation ratio will grow as well, and be at least polynomial in the alphabet size n. This has been confirmed for values of n up to $2^{(\log N)^{1-\delta}}$, see [RS97; AS03; Din+11]. Proving the same for n that is polynomial in N is the so-called polynomial sliding scale conjecture and is still quite open. Before we proceed, let us note that the aforementioned results of [RS97; AS03; Din+11] work only for arity strictly larger than two and, hence, do not imply inapproximability for 2-CSPs. We will discuss the special case of 2-CSPs in details below.

The polynomial sliding scale conjecture has been approached from different angles. In [DHK15] the authors try to find the smallest arity and alphabet size such that the hardness factor is polynomial in n, and in [Din16] the conjecture is shown to follow (in some weaker sense) from Gap-ETH, which we discuss in more details later. In this chapter, we focus on yet another angle, which is to separate n and k and ask whether it is hard to approximate constant arity CSPs to within a factor that is polynomial in k (but possibly not polynomial in n). Observe here that obtaining NP-hardness of poly(k) factor is likely to be as hard as obtaining one with poly(N); this is because

CSPs can be solved exactly in time $n^{O(k)}$, which means that, unless NP is contained in subexponential time (i.e. $\mathsf{NP} \not\subseteq \bigcap_{\varepsilon>0} \mathsf{DTIME}(2^{n^{\varepsilon}})$), NP-hard instances of CSPs must have $k = \text{poly}(N)$.

This motivates us to look for hardness of approximation from assumptions stronger than $\mathsf{P} \neq \mathsf{NP}$. Specifically, our results will be based on ETH and Gap-ETH. Firstly, we show that, unless ETH fails, no polynomial time algorithm can approximate 2-CSPs to within an almost linear ratio in k, as stated below. This is almost optimal since there is a straightforward $(k/2)$-approximation for any 2-CSP, by simply satisfying all constraints that touch the variable with highest degree.

Theorem 9.1 (Main Theorem). *Assuming ETH, for any constant $\rho > 0$, no algorithm can, given a 2-CSP instance Γ with alphabet size n and k variables such that the constraint graph is the complete graph on the k variables, distinguish between the following two cases in polynomial time:*

- *(Completeness)* $\text{val}(\Gamma) = 1$*, and,*
- *(Soundness)* $\text{val}(\Gamma) < 2^{(\log k)^{1/2+\rho}}/k$*.*

To paint a full picture of how our result stands in comparison to previous results, let us state what is know about the approximability of 2-CSPs; due to the vast literature regarding 2-CSPs, we will focus only the regime of large alphabets which is most relevant to our setting. In terms of NP-hardness, the best known inapproximability ratio is $(\log N)^c$ for every constant $c > 0$; this follows from Moshkovitz-Raz PCP [MR10] and the Parallel Repetition Theorem for the low soundness regime [DS14]. Assuming a slightly weaker assumption that NP is not contained in quasipolynomial time (i.e. $\mathsf{NP} \not\subseteq \bigcup_{c>0} \mathsf{DTIME}(n^{(\log n)^c})$), 2-CSP is hard to approximate to within a factor of $2^{(\log N)^{1-\delta}}$ for every constant $\delta > 0$; this can be proved by applying Raz's original Parallel Repetition Theorem [Raz98] to the PCP Theorem. In [Din16], the author observed that running time for parallel repetition can be reduced by looking at unordered sets instead of ordered tuples. This observation implies that[1], assuming ETH, no polynomial time $N^{1/(\log\log\log N)^c}$-approximation algorithm exists for 2-CSPs for some constant $c > 0$. Moreover, under Gap-ETH, it was shown that, for every sufficiently small $\varepsilon > 0$, an N^{ε}-approximation algorithm must run in time $N^{\Omega(\exp(1/\varepsilon))}$. Note that, while this latest result comes close to the polynomial sliding scale conjecture, it does not quite resolve the conjecture yet. In particular, even the weak form of the conjecture postulates that there exists $\delta > 0$ for which no polynomial time algorithm can approximate 2-CSPs to within N^{δ} factor of the optimum. This statement does not follow from the result of [Din16]. Nevertheless, the Gap-ETH-hardness of [Din16] does imply that, for any $f = o(1)$, no polynomial time algorithm can approximate 2-CSPs to within a factor of $N^{f(N)}$.

In all hardness results mentioned above, the constructions give 2-CSP instances in which the alphabet size n is smaller than the number of variables k. In other words, even if we aim for an inapproximability ratio in terms of k instead of N, we still get the same ratios as stated above. Thus, our result is the first hardness of approximation for 2-CSPs with $\text{poly}(k)$ factor. Note again that our result rules out any polynomial time algorithm and not just $N^{O(\exp(1/\varepsilon))}$-time algorithm

[1] In [Din16], only the Gap-ETH-hardness result is stated. However, the ETH-hardness result follows rather easily.

ruled out by [Din16]. Moreover, our ratio is almost linear in k whereas the result of [Din16] only holds for ε that is sufficiently small depending on the parameters of the Gap-ETH Hypothesis.

An interesting feature of our reduction is that it produces 2-CSP instances with the alphabet size n that is much larger than k. Of course, this should remind us of the setting of 2-CSPs parameterized by the number of variables k! We show that, even in this parameterized setting, the trivial algorithm is still essentially optimal (up to lower order terms), assuming Gap-ETH:

Theorem 9.2. *Assuming Gap-ETH, for any constant $\rho > 0$ and any function g, no algorithm can, given a 2-CSP instance Γ with alphabet size n and k variables such that the constraint graph is the complete graph on the k variables, distinguish between the following two cases in $g(k) \cdot (nk)^{O(1)}$ time:*

- *(Completeness)* $\text{val}(\Gamma) = 1$*, and,*
- *(Soundness)* $\text{val}(\Gamma) < 2^{(\log k)^{1/2+\rho}}/k$*.*

To the best of our knowledge, the only previous inapproximability result for parameterized 2-CSPs is from [CFM18]. There the authors showed that, assuming Gap-ETH, no $k^{o(1)}$-approximation $g(k) \cdot (nk)^{O(1)}$-time algorithm exists; this is shown via a simple reduction from parameterized inapproximability of DENSEST-k SUBGRAPH from the previous chapter. Our result is a direct improvement over this result.

We end our discussion on 2-CSPs by noting that several approximation algorithms have also been devised for 2-CSPs with large alphabets [Pel07; CHK11; KKT16; MM17; Chl+17b]. In particular, while our results suggest that the trivial algorithm achieves an essentially optimal ratio in terms of k, non-trivial approximation is possible when we measure the ratio in terms of N instead of k: specifically, a polynomial time $O(N^{1/3})$-approximation algorithm is known [CHK11].

Direct Steiner Network. As a corollary of our hardness of approximation results for 2-CSPs, we obtain an inapproximability result for DIRECTED STEINER NETWORK (DSN) with polynomial ratio in terms of the number of demand pairs. In DSN (sometimes referred to as DIRECTED STEINER FOREST [FKN12; Chl+17a]), we are given an edge-weighed directed graph G and a set $\mathcal{D}$ of k demand pairs $(s_1, t_1), \dots, (s_k, t_k) \in V \times V$ and the goal is to find a subgraph H of G with minimum weight such that there is a path in H from s_i to t_i for every $i \in [k]$. DSN was first studied in the approximation algorithms context by Charikar et al. [Cha+99] who gave a polynomial time $\widetilde{O}(k^{2/3})$-approximation algorithm for the problem. This ratio was later improved to $O(k^{1/2+\varepsilon})$ for every $\varepsilon > 0$ by Chekuri et al. [Che+11]. Later, a different approximation algorithm with similar approximation ratio was proposed by Feldman et al. [FKN12].

Algorithms with approximation ratios in terms of the number of vertices n have also been devised [FKN12; Ber+13; Chl+17a; AB17]. In this case, the best known algorithm is that of Berman et al. [Ber+13], which yields an $O(n^{2/3+\varepsilon})$-approximation for every constant $\varepsilon > 0$ in polynomial time. Moreover, when the graph is unweighted (i.e. each edge costs the same), Abboud and Bodwin recently gave an improved $O(n^{0.5778})$-approximation algorithm for the problem [AB17].

On the hardness side, there exists a known reduction from 2-CSP to DSN that preserves approximation ratio to within polynomial factor[2] [DK99]. Hence, known hardness of approximation of 2-CSPs translate immediately to that of DSN: it is NP-hard to approximate to within any polylogarithmic ratio [MR10; DS14], it is hard to approximate to within $2^{\log^{1-\varepsilon} n}$ factor for every $\varepsilon > 0$ unless $\mathsf{NP} \subseteq \mathsf{QP}$ [Raz98], and it is Gap-ETH-hard to approximate to within $n^{o(1)}$ factor [Din16]. Note that, since k is always bounded above by n^2, all these hardness results also hold when n is replaced by k in the ratios. Recently, this reduction was also used by Chitnis et al. [CFM18] to rule out $k^{o(1)}$-FPT-approximation algorithm for DSN parameterized by k assuming Gap-ETH. Alas, none of these hardness results achieve ratios that are polynomial in either n or k and it remains open whether DSN is hard to approximate to within a factor that is polynomial in n or in k.

By plugging our hardness result for 2-CSPs into the reduction, we immediately get ETH-hardness and Gap-ETH-hardness of approximating DSN to within a factor of $k^{1/4-o(1)}$ as stated below.

Corollary 9.3. *Assuming ETH, for any constant $\rho' > 0$, there is no polynomial time $\frac{k^{1/4}}{2^{(\log k)^{1/2+\rho'}}}$-approximation algorithm for DSN.*

Corollary 9.4. *Assuming Gap-ETH, for any constant $\rho' > 0$ and any function g, there is no $g(k) \cdot (nk)^{O(1)}$-time $\frac{k^{1/4}}{2^{(\log k)^{1/2+\rho'}}}$-approximation algorithm for DSN.*

In other words, if one wants a polynomial time approximation algorithm with ratio depending only on k and not on n, then the algorithms of Chekuri et al. [Che+11] and Feldman et al. [FKN12] are roughly within a square of the optimal algorithm. To the best of our knowledge, these are the first inapproximability results of DSN whose ratios are polynomial in terms of k. Again, Corollary 9.4 is a direct improvement over the FPT inapproximability result from [CFM18] which, under the same assumption, rules out only $k^{o(1)}$-factor FPT-approximation algorithm.

Unique Set Cover. Another consequence of our hardness of 2-CSP is an inapproximability result for the (parameterized) k-UNIQUE SET COVER. This problem can be most easily thought of as a promise problem where we are given a set system $(\mathcal{U}, \mathcal{S})$ with a promise that there exist $S_1, \dots, S_k \in \mathcal{S}$ that *uniquely* covers[3] $\mathcal{U}$, and the goal is to find minimum number of subsets from $\mathcal{S}$ that covers $\mathcal{U}$ (not necessarily uniquely). Interestingly, the NP-hardness of approximation for SET COVER of [Fei98] (and subsequent works [Mos12; DS14]) immediately implies the NP-hardness of approximating UNIQUE SET COVER; that is, the solutions in the completeness case of their instances uniquely cover the universe. However, our construction from Chapter 6 does not achieve this.

On a technical level, this stems from the fact that the starting label cover instance in Chapter 6 does not have a desired "right-to-left projection property" (see Section 9.7 for more details). Nonetheless, here we observe that we can rephrase our 2-CSP hardness of approximation in terms

[2] That is, for any non-decreasing function ρ , if DSN admits $\rho(nk)$-approximation in polynomial time, then 2-CSP also admits $\rho(nk)^c$-approximation polynomial time for some absolute constant c.

[3] That is, each element $u \in \mathcal{U}$ appears in exactly one of the subsets.

of a hardness of MAXCOV where the instance has this desired projection property. In doing so, we manage to prove an FPT inapproximability result for k-UNIQUE SET COVER with factor $k^{1/2-o(1)}$:

Theorem 9.5. *Assuming Gap-ETH, for any function g, no algorithm can, given a* SET COVER *instance $(\mathcal{U}, \mathcal{S}, k)$, distinguish between the following two cases in $g(k) \cdot (nk)^{O(1)}$ time:*

- *(Completeness) There exists $S_1, \dots, S_k \in \mathcal{S}$ that covers each element of $\mathcal{U}$ exactly once, and,*
- *(Soundness) No $k^{3/2-o(1)}$ sets from $\mathcal{S}$ covers $\mathcal{U}$.*

We remark here that, similar to k-DOMSET, there is in fact no known $f(k)$-FPT-approximation algorithm for k-UNIQUE SET COVER, which means that there is still a possibility that this problem is totally FPT inapproximable. Unfortunately, this seems out of reach of the current techniques; please refer to Section 9.8 for additional discussion regarding this.

Agreement tests

Our main result is proved through an agreement testing argument. In agreement testing there is a universe $\mathcal{U}$, a collection of subsets $S_1, \dots, S_k \subseteq \mathcal{U}$, and for each subset S_i we are given a local function $f_{S_i} : S_i \to \{0, 1\}$. A pair of subsets are said to *agree* if their local functions agree on every element in the intersection. The goal is, given a non-negligible fraction of agreeing pairs, to deduce the existence of a global function $g : \mathcal{U} \to \{0, 1\}$ that (approximately) coincides with many of the local functions. For a more complete description see [DK17].

Agreement tests capture a natural local to global statement and are present in essentially all PCPs, for example they appear explicitly in the line vs. line and plane vs. plane low degree tests [RS96; AS03; RS97]. Note that these tests impose algebraic structures on the functions f's (e.g. to be low degree). On the other hand, our agreement theorem is "combinatorial", in that $f_{S_1}, \dots, f_{S_k}$ are allowed to be any function. This more closely resembles the so-called *direct product testing*, which has exactly the same setting as us, except that typically the sets $S_1, \dots, S_k$ are taken as all ℓ-size subsets of $[n]$ [Imp+10; IJK09; DG08; DN17; IKW12]. However, the number of sets $k = \binom{n}{\ell}$ is large compare to our setting where k is very small (i.e. independent of n).

Previous works have studied the regime of "small k" as well, which corresponds to what is called *derandomized* direct product tests in literature [Imp+10; IKW12; GK18]. Nevertheless, the existing results require the fraction of agreeing pairs to be relatively large compared to ours.

More specifically, our agreement theorem works when the universe is $[n]$ and the subsets $S_1, \dots, S_k$ have $\Omega(n)$ elements each and are "in general position", namely they behave like subsets chosen independently at random. A convenient feature about this setting is, for instance, that every pair of subsets intersect.

Since we are aiming for a large gap, the agreement test must work (i.e., yield a global function) with a very small fraction of agreeing pairs, which in our case is close to $1/k$. In this small agreement regime the idea, as pioneered in the work of Raz-Safra [RS97], is to zero in on a sub-collection of subsets that is (almost) perfectly consistent. From this sub-collection it is easy to recover a global function and show that it coincides almost perfectly with the local functions in the sub-collection. A major difference between our combinatorial setting and the algebraic setting of Raz-Safra is the lack of "distance" in our case: we can not assume that two distinct local functions differ on many points (in contrast, this is a key feature of low degree polynomials). We overcome this by considering different "strengths" of agreement, depending on the fraction of points on which the two subsets agree. This notion too is present in several previous works on combinatorial agreement tests [IKW12; DN17].

Organization of the Chapter. In the next section, we describe our reduction and give an overview of the proof. Then, in Section 9.2, we define additional notions and state some preliminaries. We proceed to provide the full proof of our main agreement theorem in Section 9.3. Using this agreement theorem, we deduce the soundness of our reduction in Section 9.4. We then plug in the parameters and prove the inapproximability results for 2-CSPs in Section 9.5. In Section 9.6, we show how the hardness of approximation result for 2-CSPs imply inapproximability for DSN as well. Section 9.7 contains the hardness of approximation proof of k-UNIQUE SET COVER. Finally, we conclude our work with some discussions and open questions in Section 6.9.

9.1 Proof Overview

Like other (Gap-)ETH-hardness of approximation results, our proof is based on a subexponential time reduction from the gap version of 3-SAT to our problem of interest, 2-CSPs. Before we describe our reduction, let us define more notations for 2-CSPs and 3-SAT, to facilitate our explanation.

2-CSPs. For notational convenience, we will modify the definition of 2-CSPs slightly so that each variable is allowed to have different alphabets; this definition is clearly equivalent to the more common definition used above. Specifically, an instance Γ of 2-CSP now consists of (1) a constraint graph $G = (V, E)$, (2) for each vertex (or variable) $v \in V$, an alphabet set Σ_v, and, (3) for each edge $\{u, v\} \in E$, a constraint $C_{uv} \subseteq \Sigma_u \times \Sigma_v$. Additionally, to avoid confusion with 3-SAT, we refrain from using the word *assignment* for 2-CSPs and instead use *labeling*, i.e., a labeling of Γ is a tuple $\sigma = (\sigma_v)_{v \in V}$ such that $\sigma_v \in \Sigma_v$ for all $v \in V$. An edge $\{u, v\} \in E$ is said to be *satisfied* by a labeling σ if $(\sigma_u, \sigma_v) \in \Sigma_u \times \Sigma_v$. Similar to before, the value of a labeling σ, denoted by $\text{val}(\sigma)$, is defined as the fraction of edges that it satisfies, i.e., $|\{\{u, v\} \in E \mid (\sigma_u, \sigma_v) \in C_{uv}\}|/|E|$. The goal of 2-CSPs is to find σ with maximum value; we denote the such optimal value by $\text{val}(\Gamma)$, i.e., $\text{val}(\Gamma) = \max_\sigma \text{val}(\sigma)$.

3-SAT. An instance Φ of 3-SAT consists of a variable set X and a clause set $\mathcal{C}$ where each clause is a disjunction of at most three literals. For any assignment $\psi : \mathrm{X} \to \{0, 1\}$, $\text{val}(\psi)$ denotes the fraction of clauses satisfied by ψ. The goal is to find an assignment ψ that satisfies as many clauses as possible; let $\text{val}(\Phi) = \max_\psi \text{val}(\psi)$ denote the fraction of clauses satisfied by such assignment. For each $C \in \mathcal{C}$, we use $\text{var}(C)$ to denote the set of variables whose literals appear in C. We extend this notation naturally to sets of clauses, i.e., for every $T \subseteq \mathcal{C}$, $\text{var}(T) = \bigcup_{C \in T} \text{var}(C)$.

Our Construction

Before we state our reduction, let us again reiterate the objective of our reduction. Roughly speaking, given a 3-SAT stance $\Phi = (\mathrm{X}, \mathcal{C})$, we would like to produce a 2-CSP instance Γ_Φ such that

- (Completeness) If $\mathrm{val}(\Phi) = 1$, then $\mathrm{val}(\Gamma_\Phi) = 1$,
- (Soundness) If $\mathrm{val}(\Phi) < 1 - \varepsilon$, then $\mathrm{val}(\Gamma_\Phi) < k^{o(1)}/k$ where k is number of variables of Γ_Φ,
- (Reduction Time) The time it takes to produce Γ_Φ should be $2^{o(m)}$ where $m = |\mathcal{C}|$,

where $\varepsilon > 0$ is some absolute constant.

Observe that, when plugging a reduction with these properties to Gap-ETH, we directly arrive at the claimed $k^{1-o(1)}$ inapproximability for 2-CSPs. However, for ETH, since we start with a decision version of 3-SAT without any gap, we have to first invoke the PCP theorem to produce an instance of the gap version of 3-SAT before we can apply our reduction. Since the shortest known PCP has a polylogarithmic blow-up in the size (see Theorem 2.2), the running time lower bound for gap 3-SAT will not be exponential anymore, rather it will be of the form $2^{\Omega(m/\mathrm{polylog} m)}$ instead. Hence, our reduction will need to produce Γ_Φ in $2^{o(m/\mathrm{polylog} m)}$ time. As we shall see later in Section 9.5, this will also be possible with appropriate settings of parameters.

With the desired properties in place, we now move on to state our reduction. In addition to a 3-CNF formula Φ, the reduction also takes in a collection $\mathcal{T}$ of subsets of clauses of Φ. For now, the readers should think of the subsets in $\mathcal{T}$ as random subsets of $\mathcal{C}$ where each element is included in each subset independently at random with probability α, which will be specified later. As we will see below, we only need two simple properties that the subsets in $\mathcal{T}$ are "well-behaved" enough and we will later give a deterministic construction of such well-behaved subsets. With this in mind, our reduction can be formally described as follows.

Definition 9.6 (The Reduction). *Given a 3-CNF formula $\Phi = (\mathrm{X}, \mathcal{C})$ and a collection $\mathcal{T}$ of subsets of $\mathcal{C}$, we define a 2-CSP instance $\Gamma_{\Phi,\mathcal{T}} = (G = (V, E), \Sigma, \{C_{uv}\}_{\{u,v\} \in E})$ as follows:*

- *The graph G is the complete graph where the vertex set is $\mathcal{T}$, i.e., $V = \mathcal{T}$ and $E = \binom{\mathcal{T}}{2}$.*
- *For each $T \in \mathcal{T}$, the alphabet set $\Sigma_{\mathcal{T}}$ is the set of all partial assignments to $\mathrm{var}(T)$ that satisfies every clause in T, i.e., $\Sigma_T = \{\psi_T : \mathrm{var}(T) \to \{0, 1\} \mid \forall C \in T, \psi_T \text{ satisfies } C\}$.*
- *For every $T_1 \neq T_2 \in \mathcal{T}$, (ψ_{T_1}, ψ_{T_2}) is included in $C_{T_1T_2}$ if and only if they are consistent, i.e., $C_{T_1T_2} = \{(\psi_{T_1}, \psi_{T_2}) \in \Sigma_{T_1} \times \Sigma_{T_2} \mid \forall x \in \mathrm{var}(T_1) \cap \mathrm{var}(T_2), \psi_{T_1}(x) = \psi_{T_2}(x)\}$.*

Let us now examine the properties of the reduction. The number of vertices in $\Gamma_{\Phi,\mathcal{T}}$ is $k = |\mathcal{T}|$. For the purpose of the proof overview, α should be thought of as $1/\mathrm{polylog} m$ whereas k should be thought of as much larger than $1/\alpha$ (e.g. $k = \exp(1/\alpha)$). For such value of k, all random sets in $\mathcal{T}$ will have size $O(\alpha m)$ w.h.p., meaning that the reduction time is $2^{m/\mathrm{polylog} m}$ as desired.

Moreover, when Φ is satisfiable, it is not hard to see that $\mathrm{val}(\Gamma_{\Phi,\mathcal{T}}) = 1$; more specifically, if $\psi : \mathrm{X} \to \{0,1\}$ is the assignment that satisfies every clause of Φ, then we can label each vertex $T \in \mathcal{T}$ of $\Gamma_{\Phi,\mathcal{T}}$ by $\psi|_{\mathrm{var}(T)}$, the restriction of ψ on $\mathrm{var}(T)$. Since ψ satisfies all the clauses, $\psi|_{\mathrm{var}(T)}$ satisfies all clauses in T, meaning that this is a valid labeling. Moreover, since these are restrictions of the same global assignment ψ, they are all consistent and every edge is satisfied.

Hence, we are only left to show that, if $\mathrm{val}(\Phi) < 1 - \varepsilon$, then $\mathrm{val}(\Gamma_{\Phi,\mathcal{T}}) < k^{o(1)}/k$; this is indeed our main technical contribution. We will show this by contrapositive: assuming that $\mathrm{val}(\Gamma_{\Phi,\mathcal{T}}) \geqslant k^{o(1)}/k$, we will "decode" back an assignment to Φ that satisfies $1 - \varepsilon$ fraction of clauses.

9.1.1 Soundness Analysis as an Agreement Theorem

Our task at hand can be viewed as agreement testing. Informally, in agreement testing, the input is a collection $\{f_S\}_{S \in \mathcal{S}}$ of local functions $f_S : S \to \{0,1\}$ where $\mathcal{S}$ is a collection of subsets of some universe $\mathcal{U}$ such that, for many pairs S_1 and S_2, f_{S_1} and f_{S_2} agree, i.e., $f_{S_1}(x) = f_{S_2}(x)$ for all $x \in S_1 \cap S_2$. An agreement theorem says that there must be a global function $g : \mathcal{U} \to \{0,1\}$ that coincides (exactly or approximately) with many of the local functions, and thus explains the pairwise "local" agreements. In our case, a labeling $\sigma = \{\sigma_T\}_{T \in \mathcal{T}}$ with high value is exactly a collection of functions $\sigma_T : \mathrm{var}(T) \to \{0,1\}$ such that, for many pairs of T_1 and T_2, σ_{T_1} and σ_{T_2} agrees. The heart of our soundness proof is an agreement theorem that recovers a global function $\psi : \mathrm{X} \to \{0,1\}$ that approximately coincides with many of the local functions σ_T's and thus satisfies $1 - \varepsilon$ fraction of clauses of Φ. To discuss the agreement theorem in more details, let us define several additional notations, starting with those for (approximate) agreements of a pair of functions:

Definition 9.7. *For any universe $\mathcal{U}$, let $f_{S_1} : S_1 \to \{0,1\}$ and $f_{S_2} : S_2 \to \{0,1\}$ be any two functions whose domains S_1, S_2 are subsets of $\mathcal{U}$. We use the following notations for (dis)agreements of these two functions:*

- *Let* $\mathrm{disagr}(f_{S_1}, f_{S_2})$ *denote the number of $x \in S_1 \cap S_2$ that f_{S_1} and f_{S_2} disagree on, i.e.,* $\mathrm{disagr}(f_{S_1}, f_{S_2}) = |\{x \in S_1 \cap S_2 \mid f_{S_1}(x) \neq f_{S_2}(x)\}|$.
- *For any $\zeta \geqslant 0$, we say that f_{S_1} and f_{S_2} are* ζ-consistent *if* $\mathrm{disagr}(f_{S_1}, f_{S_2}) \leqslant \zeta|\mathcal{U}|$, *and we say that the two functions are* ζ-inconsistent *otherwise. For $\zeta = 0$, we sometimes drop 0 and refer to these simply as* consistent *and* inconsistent *(instead of 0-consistent and 0-inconsistent).*
- *We use $f_{S_1} \overset{\zeta}{\approx} f_{S_2}$ and $f_{S_1} \overset{\zeta}{\not\approx} f_{S_2}$ as shorthands for ζ-consistency and ζ-inconsistency respectively. Again, for $\zeta = 0$, we may drop 0 from the notations and simply use $f_{S_1} \approx f_{S_2}$ and $f_{S_1} \not\approx f_{S_2}$.*

Next, we define the notion of agreement probability for any collection of functions:

Definition 9.8. *For any $\zeta \geqslant 0$ and any collection $\mathcal{F} = \{f_S\}_{S \in \mathcal{S}}$ of functions, the ζ-agreement probability, denoted by agr$_\zeta(\mathcal{F})$ is the probability that f_S is ζ-consistent with $f_{S'}$ where S and S' are chosen independently uniformly at random from $\mathcal{S}$, i.e., agr$_\zeta(\mathcal{F}) = \Pr_{S,S' \in \mathcal{S}}[f_S \overset{\zeta}{\approx} f_{S'}]$. When $\zeta = 0$, we will drop 0 from the notation and simply use agr$(\mathcal{F})$.*

Our main agreement theorem, which works when each $S \in \mathcal{S}$ is a large "random" subset, says that, if agr$(\mathcal{F})$ is noticeably large, then there exists a global function that is approximately consistent with many of the local functions in $\mathcal{F}$. This is stated more precisely (but still informally) below.

Theorem 9.9 (Informal; See Theorem 9.17)**.** *Let $\mathcal{S}$ be a collection of k independent random αn-element subsets of $[n]$. The following holds with high probability: for any $\beta > 0$ and any collection of functions $\mathcal{F} = \{f_S\}_{S \in \mathcal{S}}$ such that $\delta :=$ agr$(\mathcal{F}) \geqslant k^{o_{\beta,\alpha}(1)}/k$, there exist a function $g : [n] \to \{0, 1\}$ and a subcollection $\mathcal{S}'$ of size $\delta k^{1-o_{\beta,\alpha}(1)}$ such that $g \overset{\beta}{\approx} f_{S'}$ for all $S' \in \mathcal{S}'$.*

To see that Theorem 9.9 implies our soundness, let us view a labeling $\sigma = \{\sigma_T\}_{T \in \mathcal{T}}$ as a collection $\mathcal{F} = \{f_S\}_{S \in \mathcal{S}}$ where $\mathcal{S} = \{\text{var}(T) \mid T \in \mathcal{T}\}$ and $f_{\text{var}(T)}$ is simply σ_T. Now, when val(σ) is large, agr$(\mathcal{F})$ is large as well. Moreover, while the sets $S \in \mathcal{S}$ are not random subsets of variables but rather variable sets of random subsets of clauses, it turns out that these sets are "well-behaved" enough for us to apply Theorem 9.9. This yields a global function $\psi : \mathrm{X} \to \{0, 1\}$ that are β-consistent with many σ_T's. Note that, if instead of β-consistency we had exact consistency, then we would have been done because ψ must satisfy all clauses that appear in any T such that ψ is consistent with σ_T; since there are many such T's and these are random sets, ψ indeed satisfies almost all clauses. A simple counting argument shows that this remains true even with approximate consistency, provided that most clauses appear in at least a certain fraction of such T's (an assumption which holds for random subsets). Hence, the soundness of our reduction follows from Theorem 9.9, and we devote the rest of this section to outline an overview of its proof.

Optimality of the parameters of Theorem 9.9. Before we proceed to the overview, we would like to note that the size of the subcollection $\mathcal{S}'$ in Theorem 9.9 is nearly optimal. This is because, we can partition $\mathcal{S}$ into $1/\delta$ subcollections $\mathcal{S}_1, \dots, \mathcal{S}_{1/\delta}$ each of size δk and, for each $i \in [1/\delta]$, randomly select a global function $g_i : [n] \to \{0, 1\}$ and let each f_S be the restriction of g_i to S for each $S \in \mathcal{S}_i$. In this way, we have agr$(\mathcal{F}) \geqslant \delta k$ and any global function can be (approximately) consistent with at most δk local functions. This means that $\mathcal{S}'$ can be of size at most δk in this case and, up to a $k^{o_{\beta,\alpha}(1)}$ multiplicative factor, Theorem 9.9 yields almost a largest possible $\mathcal{S}'$.

9.1.2 A Simplified Proof: $\delta \geqslant k^{o(1)}/k^{1/2}$ Regime

We now sketch the proof of Theorem 9.9. Before we describe how we can find g when $\delta \geqslant k^{o_{\beta,\alpha}(1)}/k$, let us sketch the proof assuming a stronger assumption that $\delta \geqslant \Theta_{\alpha,\beta}(1)/k^{1/2}$. Note that this simplified proof already implies a $k^{1/2-o(1)}$ factor ETH-hardness of approximating 2-CSPs. In the next subsection, we will then proceed to refine the arguments to handle smaller values of δ.

Let us consider the *consistency graph* of $\mathcal{F}$. This is the graph $G^{\mathcal{F}}$ whose vertex set is $\mathcal{S}$ and there is an edge between S_1 and S_2 if and only if f_{S_1} and f_{S_2} are consistent. Note that the number of edges in $G^{\mathcal{F}}$ is equal to $\frac{k^2\delta - k}{2}$, where the subtraction of k comes from the fact that $\delta = \text{agr}(\mathcal{F})$ includes the agreement of each set and itself (whereas $G^{\mathcal{F}}$ does not).

Previous works on agreement testers exploit particular structures of the consistency graph to decode a global function. One such property that is relevant to our proof is the notion of *almost transitivity* defined by Raz and Safra in the analysis of their test [RS97]. More specifically, a graph $G = (V, E)$ is said to be q-transitive for some $q > 0$ if, for every non-edge $\{u, v\}$ (i.e. $\{u, v\} \in \binom{V}{2} \setminus E$), u and v can share at most q common neighbors[4]. Raz and Safra showed that their consistency graph is $(k^{1-\Omega(1)})$-transitive where k denotes the number of vertices of the graph. They then proved a generic theorem regarding $(k^{1-\Omega(1)})$-transitive graphs that, for any such graph, its vertex set can be partitioned so that the subgraph induced by each partition is a clique and that the number of edges between different partitions is small. Since a sufficiently large clique corresponds to a global function in their setting, they can then immediately deduce that their result.

Observe that, in our setting, a large clique also corresponds to a global function that is consistent with many local functions. In particular, suppose that there exists $\mathcal{S}' \subseteq \mathcal{S}$ of size sufficiently large such that $\mathcal{S}$ induces a clique in $G^{\mathcal{F}}$. Since $f_{S'}$'s are perfectly consistent with each other for all $S' \in \mathcal{S}'$, there is a global function $g : [n] \to \{0, 1\}$ that is consistent with all such $f_{S'}$'s. Hence, if we could show that our consistency graph $G^{\mathcal{F}}$ is $(k^{1-\Omega(1)})$-transitive, then we could use the same argument as Raz and Safra's to deduce our desired result. Alas, our graph $G^{\mathcal{F}}$ does not necessarily satisfy this transitivity property; for instance, consider any two sets $S_1, S_2 \in \mathcal{S}$ and let f_{S_1}, f_{S_2} be such that they disagree on only one variable, i.e., there is a unique $x \in S_1 \cap S_2$ such that $f_{S_1}(x) \neq f_{S_2}(x)$. It is possible that, for every $S \in \mathcal{S}$ that does not contain x, f_S agrees with both f_{S_1} and f_{S_2}; in other words, every such S can be a common neighbor of S_1 and S_2. Since each variable x appears roughly in only $\Theta(\alpha)$ fraction of the sets, there can be as many as $(1 - \Theta(\alpha))k = (1 - o(1))k$ common neighbors of S_1 and S_2 even when there is no edge between S_1 and S_2!

Fortunately for us, a weaker statement holds: if f_{S_1} and f_{S_2} disagree on more than ζn variables (instead of just one variable as above), then S_1 and S_2 have at most $O(\ln(1/\zeta)/\alpha)$ common neighbors in $G^{\mathcal{F}}$. Here ζ should be thought of as β^2 times a small constant which will be specified later. To see why this statement holds, observe that, since every $S \in \mathcal{S}$ is a random subset that includes each clause $x \in [n]$ with probability α, Chernoff bound implies that, for every subcollection $\widetilde{\mathcal{S}} \subseteq \mathcal{S}$ of size $\Omega(\ln(1/\zeta)/\alpha)$, $\bigcup_{S \in \widetilde{\mathcal{S}}} S$ contains all but $O(\zeta)$ fraction of variables. Let $\widetilde{\mathcal{S}}_{S_1,S_2} \subseteq \mathcal{S}$ denote the set of common neighbors of S_1 and S_2. It is easy to see that S_1 and S_2 can only disagree on variables that do not appear in $\bigcup_{S \in \widetilde{\mathcal{S}}_{S_1,S_2}} S$. If $\widetilde{\mathcal{S}}_{S_1,S_2}$ is of size $\Omega(\ln(1/\zeta)/\alpha)$, then $\bigcup_{S \in \widetilde{\mathcal{S}}_{S_1,S_2}} S$ contains all but $O(\zeta)$ fraction of variables, which means that S_1 and S_2 disagrees only on $O(\zeta)$ fraction of variables. By selecting the constant appropriately inside $O(\cdot)$, we arrive at the claim statement.

[4] In [RS97], the transitivity parameter q is used to denote the *fraction* of vertices that are neighbors of both u and v rather than the *number* of such vertices as defined here. However, the latter notion will be more convenient for us.

In other words, while the transitive property does not hold for every edge, it holds for the edges $\{S_1, S_2\}$ where f_{S_1} and f_{S_2} are ζ-inconsistent. This motivates us to define a two-level consistency graph, where the edges with ζ-inconsistent are referred to as the *red* edges whereas the original edges in $G^{\mathcal{F}}$ is now referred to as the *blue* edges. We define this formally below.

Definition 9.10 (Red/blue Graph). *A red-blue graph is an undirected graph $G = (V, E = E_r \cup E_b)$ where its edge set E is partitioned into two sets E_r, the set of red edges, and E_b, the set of blue edges. We use the prefixes "blue-" and "red-" to refer to the quantities of the graph (V, E_b) and (V, E_r) respectively; for instance, u is said to be a blue-neighbor of v if $\{u, v\} \in E_b$.*

Definition 9.11 (Two-Level Consistency Graph). *Given a collection of functions $\mathcal{F} = \{f_S\}_{S \in \mathcal{S}}$ and a real number $0 \leqslant \zeta \leqslant 1$, the two-level consistency graph $G^{\mathcal{F},\zeta} = (V^{\mathcal{F},\zeta}, E_r^{\mathcal{F},\zeta} \cup E_b^{\mathcal{F},\zeta})$ is a red-blue graph defined as follows.*

- *The vertex set $V^{\mathcal{F},\zeta}$ is simply $\mathcal{S}$.*
- *The blue edges are the consistent pairs $\{S_1, S_2\}$, i.e., $E_b = \{\{S_1, S_2\} \in \binom{\mathcal{S}}{2} \mid f_{S_1} \approx f_{S_2}\}$.*
- *The red edges are the ζ-inconsistent pairs $\{S_1, S_2\}$, i.e., $E_r = \{\{S_1, S_2\} \in \binom{\mathcal{S}}{2} \mid f_{S_1} \overset{\zeta}{\not\approx} f_{S_2}\}$.*

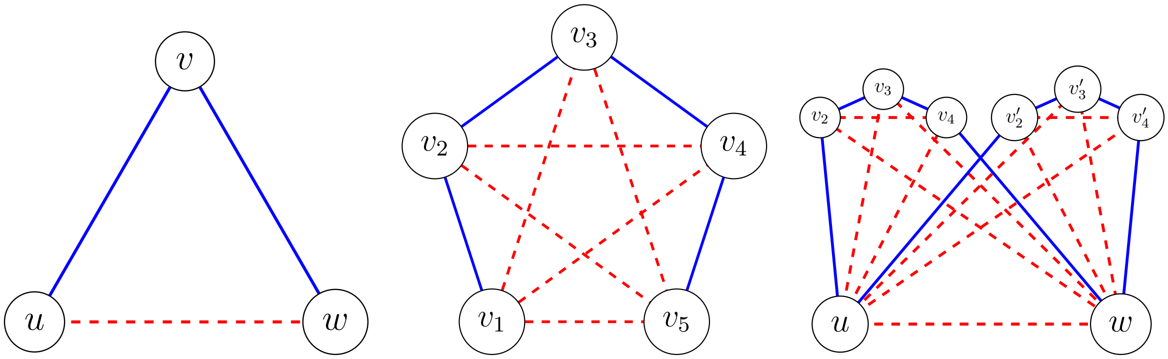
Note that S_1, S_2 constitute neither a blue nor a red edge when $0 < \operatorname{disagr}(f_{S_1}, f_{S_2}) \leqslant \zeta n$.

Now, the transitivity property we argue above can be stated as follows: for every red-edge $\{S_1, S_2\}$ of $G^{\mathcal{F},\zeta}$, there are at most $O(\ln(1/\zeta)/\alpha)$ different S's such that both $\{S, S_1\}$ and $\{S, S_2\}$ are blue edges. For brevity, let us call any red-blue graph $G = (V, E_r \cup E_b)$ *q-red/blue-transitive* if, for every red edge $\{u, v\} \in E_r$, u and v have at most q common blue-neighbors. We will now argue that in any q-red/blue-transitive of average blue-degree d, there exists a subset $U \subseteq V$ of size $\Omega(d)$ such that only $O(qk/d^2)$ fraction of pairs of vertices in U are red edges.

Before we prove this, let us state why this is useful for decoding the desired global function g. Observe that such a subset U of vertices in the two-level consistency graph translates to a subcollection $\mathcal{S}' \subseteq \mathcal{S}$ such that, for all but $O(qk/d^2)$ fraction of pairs of sets $S_1, S_2 \subseteq \mathcal{S}'$, $\{S_1, S_2\}$ does not form a red edge. Recall from definition of red edges that, for such S_1, S_2, f_{S_1} and f_{S_2} disagrees on at most ζn variables. In other words, $\mathcal{S}'$ is similar to a clique in the (not two-level) consistency graph, except that (1) $O(qk/d^2)$ fraction of pairs $\{S_1, S_2\}$ are allowed to disagree on as many variables as they like, and (2) even for the rest of pairs, the guarantee now is that they agree on all but at most ζn variables, instead of total agreement as in the previous case of clique. Fortunately, this still suffices to find g that is $O(\sqrt{qk/d^2 + \zeta})$-consistent with $\Omega(d)$ functions. One way construct such a global function is to simply assign each $g(x)$ according to the majority of $f_S(x)$ for all $S \in \mathcal{S}'$ such that $x \in S$. (This is formalized in Section 9.3.3.) Note that in our case $q = O(\ln(1/\zeta)/\alpha)$ and $d = \Omega(\delta k)$. Hence, if we pick $\zeta \ll \beta^2$ and $\delta \gg (q^{1/2}/\beta)/k^{1/2} = O_{\beta,\alpha}(1)/k^{1/2}$, we indeed get a global function g that is β-consistent with $\Omega(\delta k)$ local functions.

We now move on to sketch how one can find such an "almost non-red subgraph". For simplicity, let us assume that every vertex has the same blue-degree (i.e. (V, E_b) is d-regular). Let us count the number of *red-blue-blue triangle* (or *rbb triangle*), which is a 3-tuple (u, v, w) of vertices in

V such that $\{u, v\}, \{v, w\}$ are blue edges whereas $\{u, w\}$ is a red edge. An illustration of a rbb triangle can be found in Figure 9.1a. The red/blue transitivity can be used to bound the number of rbb triangles as follows. For each $(u^*, w^*) \in V^2$, since the graph is q-red/blue-transitive there are at most q rbb triangle with $u = u^*$ and $w = w^*$. Hence, in total, there can be at most qk^2 rbb triangles. As a result, there exists $v^* \in V$ such that the number of rbb triangles (u, v, w) such that $v = v^*$ is at most qk. Let us now consider the set $U = N_b(v^*)$ that consists of all blue-neighbors of v^*. There can be at most qk red edges with both endpoints in $N_b(v^*)$ because each such edge corresponds to a rbb triangle with $v = v^*$. From our assumption that every vertex has blue degree d, we indeed have that $|U| = d$ and that the fraction of pairs of vertices in U that are linked by red edges is $O(qk/d^2)$ as desired. This completes our overview for the case $\delta \geqslant \Theta_{\beta,\alpha}(1)/k^{1/2}$.



(a) a red-blue-blue triangle (b) a red-filled 4-walk (c) disjoint red-filled 4-walks

Figure 9.1: Illustrations of red-filled walks. The red edges are represented by red dashed lines whereas the blue edges are represented by blue solid lines. Figure 9.1a and Figure 9.1b demonstrate a red-filled 2 walk (aka rbb triangle) and a red-filled 4-walk respectively. Figure 9.1c shows two disjoint red-filled 4-walks.

9.1.3 Towards $\delta = k^{o(1)}/k$ Regime

To handle smaller δ, we need to first understand why the approach above fails to work for $\delta \leqslant 1/k^{1/2}$. To do so, note that the above proof sketch can be summarized into three main steps:

(1) Show that the two-level consistency graph $G^{\mathcal{F}}$ is q-red/blue-transitive for some $q = k^{o(1)}$.

(2) Use red/blue transitivity to find a large subgraph of $G^{\mathcal{F}}$ with few induced red edges.

(3) Decode a global function from such an “almost non-red subgraph”.

The reason that we need $\delta \gg 1/k^{1/2}$, or equivalently $d \gg k^{1/2}$, lies in Step 2. Although not stated as such earlier, our argument in this step can be described as follows. We consider all length-2 blue-walks, i.e., all $(u, v, w) \in V^3$ such that $\{u, v\}$ and $\{v, w\}$ are both blue edges, and, using the red/blue transitivity of the graph, we argue that, for almost of all these walks, $\{u, w\}$ is not a red edge (i.e. (u, v, w) is not a rbb triangle), which then allows us to find an almost non-red subgraph. For this argument to work, we need the number of length-2 blue-walks to far exceed the number of rbb triangles. The former is kd^2 whereas the latter is bounded above by k^2q in q-red/blue-transitive graphs. This means that we need $kd^2 \gg k^2q$, which implies that $d \gg k^{1/2}$.

To overcome this limitation, we instead consider all length-ℓ blue-walks for $\ell > 2$ and we will define a "rbb-triangle-like" structure on these walks. Our goal is again to show that this structure appears rarely in random length-ℓ blue-walks and we will then use this to find a subgraph that allows us to decode a good assignment for Φ. Observe that the number of length-ℓ blue walks is kd^ℓ. We also hope that the number of "rbb-triangle-like" structures is still small; in particular, we will still get a similar bound $k^{2+o(1)}$ for such generalized structure, similar to our previous bound for the red-blue-blue triangles. When this is the case, we need $kd^\ell \geqslant k^{2+o(1)}$, meaning that when $\ell = \omega(1)$ it suffices to select $d = k^{o(1)}$, which yields $k^{1-o(1)}$ factor inapproximability as desired. To facilitate our discussion, let us define notations for ℓ-walks here.

Definition 9.12 (ℓ-Walks)**.** *For any red/blue graph $G = (V, E_r \cup E_b)$ and any integer $\ell \geqslant 2$, an* ℓ-blue-walk *in G is an $(\ell + 1)$-tuple of vertices $(v_1, v_2, \dots, v_{\ell+1}) \in V^{\ell+1}$ such that every pair of consecutive vertices are joined by a blue edge, i.e., $\{v_i, v_{i+1}\} \in E_b$ for every $i \in [\ell]$. For brevity, we sometimes refer to ℓ-blue walks simply as* ℓ-walks. *We use $\mathcal{W}_\ell^G$ to denote the set of all ℓ-walks in G.*

Note here that a vertex can appears multiple times in a single ℓ-walk.

One detail we have yet to specify in the proof is the structure that generalizes the rbb triangle for ℓ-walks where $\ell > 2$. Like before, this structure will enforce the two end points of the walk to be joined by a red edge, i.e., $\{v_1, v_{\ell+1}\} \in E_r$. Additionally, we require every pair of non-consecutive vertices to be joined by a red edge. We call such a walk a *red-filled ℓ-walk* (see Figure 9.1b):

Definition 9.13 (Red-Filled ℓ-Walks)**.** *For any red/blue graph $G = (V, E_r \cup E_b)$, a* red-filled ℓ-walk *is an ℓ-walk $(v_1, v_2, \dots, v_{\ell+1})$ such that every pair of non-consecutive vertices is joined by a red edge, i.e., $\{v_i, v_j\} \in E_r$ for every $i, j \in [\ell + 1]$ such that $j > i + 1$. Let $\widehat{\mathcal{W}}_\ell^G$ denote the set of all red-filled ℓ-walks in G. Moreover, for every $u, v \in V$, let $\widehat{\mathcal{W}}_\ell^G(u, v)$ denote the set of all red-filled ℓ-walks from u to v, i.e., $\mathcal{W}_\ell^G(u, v) = \{(v_1, \dots, v_{\ell+1}) \in \widehat{\mathcal{W}}_\ell^G \mid v_1 = u \land v_{\ell+1} = v\}$.*

As mentioned earlier, we will need a generalized transitivity property that works not only for rbb triangles but also for our new structure, i.e. the red-filled ℓ-walks. This can be defined analogously to q-red/blue transitivity as follows.

Definition 9.14 ((q, ℓ)-Red/Blue Transitivity)**.** *For any positive integers $q, \ell \in \mathbb{N}$, a red/blue graph $G = (V, E_r \cup E_b)$ is said to be* (q, ℓ)-red/blue-transitive *if, for every pair of vertices $u, v \in V$ that are joined by a red edge, there exists at most q red-filled ℓ-walks starting at u and ending at v, i.e., $|\widehat{\mathcal{W}}_\ell^G(u, v)| \leqslant q$.*

Using a similar argument to before, we can show that, when $\mathcal{S}$ consists of random subsets where each element is included in a subset with probability α, the two-level agreement graph is (q, ℓ)-red/blue transitive for some parameter q that is a function of only α and ℓ. When $1/\alpha$ and ℓ are small enough in terms of k, q can made to be $k^{o(1)}$. (The full proof can be found in Section 9.3.1.)

Once this is proved, it is not hard (using a similar argument as before) to show that, when $d \gg (kq)^{1/\ell}$, most ℓ-walks are not red-filled, i.e., $|\mathcal{W}^G_\ell| \gg |\widehat{\mathcal{W}}^G_\ell|$. Even with this, it is still unclear how we can get back a "clique-like" subgraph; in the case of $\ell = 2$ above, this implies that a blue-neighborhood induces few red edges, but the argument does not seem to generalize to larger ℓ. Fortunately, it is still quite easy to find a large subgraph that a non-trivial fraction of pairs of vertices do *not* form red edges; specifically, we will find two subsets $U_1, U_2 \subseteq V$ each of size d such that for at least $1/\ell^2$ fraction of $(u_1, u_2) \in U_1 \times U_2$, $\{u_1, u_2\}$ is not a red edge. To find such sets, observe that, if $|\mathcal{W}^G_\ell| \geqslant 2|\widehat{\mathcal{W}}^G_\ell|$, then for a random $(v_1, \dots, v_{\ell+1}) \in \mathcal{W}^G_\ell$ the probability that there exists non-consecutive vertex v_i, v_j in the walk that are joined by a red edge is at least $1/2$. Since there are less than $\ell^2/2$ such i, j, union bound implies that there must be non-consecutive i^*, j^* such that the probability that v_{i^*}, v_{j^*} are not joined by a red edge is at least $1/\ell^2$. Let us assume without loss of generality that $i^* < j^*$; since they are not consecutive, we have $i^* + 1 < j^*$.

Let us consider v_{i^*+1}, v_{j^*-1}. By a simple averaging argument, there must be u^* and w^* such that, conditioning on $v_{i^*+1} = u^*$ and $v_{j^*+1} = w^*$, the probability that $\{v_{i^*}, v_{j^*}\} \notin E_r$ is at least $1/\ell^2$. However, this conditional probability is exactly equal to fraction of $(u_1, u_2) \in N_b(u^*) \times N_b(w^*)$ that u_1 and u_2 are not joined by a red edge. Recall again that $N_b(v)$ is used to denote the set of all blue-neighbors of v. Thus, $U_1 = N_b(u^*)$ and $U_2 = N_b(w^*)$ are the sets with desired property.

We are still not done yet since we have to use these sets to decode back the global function g. This is still not obvious: the guarantee we have for our sets U_1, U_2 is rather weak since we only know that at least $1/\ell^2$ of the pairs of vertices from the two sets do not form red edges. This is in contrast to the $\ell = 2$ case where we have a subgraph such that almost all induced edges are *not* red.

To see how to overcome this barrier, recall that a pair S_1, S_2 that does not form a red edge corresponds to $f_{S_1} \overset{\zeta}{\approx} f_{S_2}$. As a thought experiment, let us think of the following scenario: if instead of just ζ-consistency, these pairs satisfy (exact) consistency, then we can consider the collection $\widetilde{\mathcal{F}} = \{f_S\}_{S \in \widetilde{U}}$ where $\widetilde{U} = U_1 \cup U_2$. This is a collection of $\Theta(d)$ local functions such that $\text{agr}(\widetilde{\mathcal{F}}) \geqslant \Omega(1/\ell^2)$. Thus, when $d \gg \ell^4$, we are in the regime where $\text{agr}(\widetilde{\mathcal{F}}) \gg 1/d^{1/2}$, meaning that we can apply our earlier argument (for the $\delta \geqslant k^{o(1)}/k^{1/2}$ regime) to recover g!

The approach in the previous paragraph of course does not work directly because we only know that $\Omega(1/\ell^2)$ fraction of the pairs $\{S_1, S_2\} \subseteq \widetilde{U}$ are ζ-consistent, not exactly consistent. However, we can still try to mimic the proof in the regime $\delta \geqslant k^{o(1)}/k^{1/2}$ and define a red/blue graph in such a way that such ζ-consistent pairs are now blue edges. Naturally, the red edges will now be the ζ'-inconsistent pairs for some $\zeta' > \zeta$. In other words, we consider the *generalized two-level consistency graph* defined as follows.

Definition 9.15 (Generalized Two-Level Consistency Graph). *Given a collection of functions $\mathcal{F} = \{f_S\}_{S \in \mathcal{S}}$ and two real numbers $0 \leqslant \zeta \leqslant \zeta' \leqslant 1$, the generalized two-level consistency graph $G^{\mathcal{F},\zeta,\zeta'} = (V^{\mathcal{F},\zeta,\zeta'}, E_r^{\mathcal{F},\zeta,\zeta'} \cup E_b^{\mathcal{F},\zeta,\zeta'})$ is a red/blue graph defined as follows.*

- *The vertex set $V^{\mathcal{F},\zeta,\zeta'}$ is simply $\mathcal{S}$.*

- *The blue edges are the ζ-consistent pairs $\{S_1, S_2\}$, i.e., $E_b^{\mathcal{F},\zeta,\zeta'} = \{\{S_1, S_2\} \in \binom{S}{2} \mid f_{S_1} \overset{\zeta}{\approx} f_{S_2}\}$.*

- *The red edges are the ζ'-inconsistent pairs $\{S_1, S_2\}$, i.e., $E_r^{\mathcal{F},\zeta,\zeta'} = \{\{S_1, S_2\} \in \binom{S}{2} \mid f_{S_1} \overset{\zeta'}{\not\approx} f_{S_2}\}$.*

As its name suggests, the generalized two-level consistency graph is a generalization of the two-level consistency graph from Definition 9.11; namely $G^{\mathcal{F},0,\zeta}$ in the more general definition coincides with $G^{\mathcal{F},\zeta}$ in the original definition.

Now, it is not hard to show that when $\zeta' \gg \zeta/\alpha$, the graph $G^{\mathcal{F},\zeta,\zeta'}$ is again q-red/blue transitive for some q that depends only on α and ζ. This means that we can apply our argument from the $\delta \geqslant 1/k^{1/2-o(1)}$ regime on the graph $G^{\widetilde{\mathcal{F}},\zeta,\zeta'}$, which yields a subset $U \subseteq \widetilde{U}$ such that almost all pairs $\{S_1, S_2\} \subseteq U$ are ζ'-consistent. By selecting the parameters appropriately, such an almost non-red subgraph once again gives us the desired global function. This wraps up our proof overview.

9.2 Additional Preliminaries

9.2.1 Parameters of Well-Behaved Subsets

We next recall two properties of collections of subsets, which will be needed in our soundness analysis. First, recall that, in our proof overview for the weaker $k^{1/2-o(1)}$ factor hardness, we need the following to show the red/blue transitivity of the consistency graph: for any r subsets from the collection, their union must contain almost all clauses. Here r is a positive integer that effects the red/blue transitivity parameter. This coincides with the definition of *dispersers* (Definition 2.20). For walks with larger length, we need a stronger property that any union of r intersections of ℓ subsets are large. Recall that this is exactly the notion of *intersection dispersers* in Definition 2.21.

Another property we need is that any sufficiently large subcollection $\widetilde{\mathcal{S}}$ of $\mathcal{S}$ is "sufficiently uniform" in the sense of Definition 2.18. More specifically, recall that the *uniformity* condition requires that almost all clauses appear in not too small number of subsets in $\widetilde{\mathcal{S}}$. This is used when we decode a good assignment from an almost non-red subgraph.

Using standard concentration bounds, it is not hard to show that, when m is sufficiently large, a collection of random subsets where each element is included in each subset independently with probability α is an $(1/O(\alpha^\ell), \ell, O(1))$-disperser and every subcollection of size $\Omega(1/\alpha)$ is $(\alpha, O(1))$ uniform. This can be easily derandomized using the construction from Section 2.9, as stated formally below.

Lemma 9.16 (Deterministic Construction of Well-Behaved Subsets)**.** *For any $k, \ell, q \in \mathbb{N}$ and any integer $m \geqslant q^{k+1}$, let $\mathcal{U}$ be any m-element set. Then, there is a collection $\mathcal{T}$ of k subsets of $\mathcal{U}$ with the following properties with $\alpha := 1/q$.*

- *(Size) Every subset in $\mathcal{T}$ has size at most $2\alpha m$.*

- *(Intersection Disperser) For any $\eta > 0$, $\mathcal{T}$ is a $(\lceil \ln(1/\eta)/\alpha^\ell \rceil, \ell, \eta)$-intersection disperser.*
- *(Uniformity) For any $\mu > 0$, any subcollection $\widetilde{\mathcal{T}} \subseteq \mathcal{T}$ of size $\lceil 8\ln(1/\mu)/\alpha \rceil$ is $(\alpha/2, \mu)$-uniform.*

Moreover, such a collection $\mathcal{T}$ can be deterministically constructed in time $O(m \cdot q^k)$.

Proof. Let $z = \lfloor m/q^k \rfloor$. To define the sets, we first partition $\mathcal{U}$ into two parts $\mathcal{U}^0, \mathcal{U}^1$, where $\mathcal{U}^0$ is of size $q^k \cdot z$ and $\mathcal{U}^1$ is of size $m - |\mathcal{U}^0| < q^k$. We associate the elements of $\mathcal{U}^0$ with $[q]^k \times [z]$. Let $T_1, \dots, T_k$ be the sets as in Definition 2.16 with $t = 1$. We define the set $T'_1, \dots, T'_k \subseteq \mathcal{U}$ by $T'_i = (T_i \times [z]) \cup \mathcal{U}^1$. The intersection disperser property and uniformity of $\mathcal{T} = \{T'_1, \dots, T'_k\}$ follows immediately from Propositions 2.22 and 2.19 respectively. Finally, each set is of size at most $q^{k-1} \cdot z + q^k \leqslant 2\alpha m$, where the inequality comes from our assumption that $m \geqslant q^{k+1}$. □

Let us turn our focus back to our main technical contribution: the agreement testing theorem.

9.3 The Main Agreement Theorem

The main goal of this section is to prove the following agreement theorem, which is the formal version of Theorem 9.9 and is also the main technical contribution of this chapter.

Theorem 9.17. *For any $0 < \eta, \zeta, \gamma, \mu < 1$ and $r, \ell, k, h, n, d \in \mathbb{N}$ such that $\ell \geqslant 2$, let $\mathcal{S}$ be any collection of k subsets of $[n]$ such that $\mathcal{S}$ is (r, ℓ, ζ)-intersection disperser and every subcollection $\widetilde{\mathcal{S}} \subseteq \mathcal{S}$ of size h is (γ, μ)-uniform, and let $\mathcal{F} = \{f_S\}_{S \in \mathcal{S}}$ be any collection of functions. If $\delta := agr(\mathcal{F}) \geqslant \frac{10+64(r\ell)^2 k^{1/\ell}}{k}$, then there exists a subcollection $\mathcal{S}' \subseteq \mathcal{S}$ of size at least $\frac{\delta k}{256\ell^2}$ and a function $g : [n] \to \{0,1\}$ such that $g \overset{\beta}{\approx} f_S$ for all $S \in \mathcal{S}'$ where*

$$\beta = 2\sqrt{\frac{65536 h \ell^6}{\delta k} + \mu + 2\zeta/\gamma}.$$

While the parameters of the theorem can be confusing, when each subset in $\mathcal{S}$ is a random αn-size subset of $[n]$, the parameters we are interested in are as follows: μ and η both go to 0 as n goes to infinity, h and γ depend only on α, and, r is $O(1/\alpha^\ell)$. Since we want the requirement on soundness as weak as possible, we want to minimize $(r\ell)^2 k^{1/\ell} = 2^{O_\alpha(\ell + (\log k)/\ell)}$. Hence, our best choice is to let $\ell = \sqrt{\log k}$, which indeed yields the $k/2^{(\log k)^{1/2+\rho}}$ ratio inapproximability for 2-CSPs.

To prove this theorem, we follow the general outline as stated in the proof overview section. In particular, the proof contains five main steps, as elaborated below.

(1) First, we will show that when $\mathcal{S}$ is an intersection disperser with appropriate parameters, then the two-level consistency graph $G^{\mathcal{F},\zeta}$ satisfies (q, ℓ)-red/blue transitivity for certain q, ℓ.

(2) Second, we argue that, for any red/blue transitive graphs that contains sufficiently many blue edges, we can find a large subset $\widetilde{U}$ of vertices such that a reasonably large fraction of pairs $\{S_1, S_2\} \subseteq \widetilde{U}$ are non-red. This is done by counting red-filled ℓ-walks for an appropriate ℓ.

(3) We then focus on $\widetilde{\mathcal{F}} = \{f_S\}_{\widetilde{U}}$ and show, using a uniformity condition of $\mathcal{S}$, that the generalized two-level consistency graph $G^{\widetilde{\mathcal{F}},\zeta,\zeta'}$ is red/blue transitive with certain parameters.

(4) Next, counting rbb triangles reveals a large "almost non-red subgraph" in the graph $G^{\widetilde{\mathcal{F}},\zeta,\zeta'}$.

(5) Finally, we decode a global function from this almost non-red subgraph.

This section is organized as follows. In Subsection 9.3.1, we show transitivity properties of the two-level and generalized two-level consistency graphs, i.e., Steps 1 and 3. Subsection 9.3.2 contains a structural lemma regarding an existence of a large subgraph with certain non-red density in red/blue transitive graphs; this lemma is at the heart of Steps 2 and 4. Next, in Subsection 9.3.3, we prove Step 5. Finally, in Subsection 9.3.4, we put these parts together and prove Theorem 9.17.

9.3.1 Red/Blue-Transitivity of (Generalized) Two-Level Consistency Graph

Red/Blue-Transitivity from Intersection Disperser

The first step in our proof is to show that the two-level consistency graph $G^{\mathcal{F},\zeta}$ is red/blue-transitive, assuming that $\mathcal{S}$ is an intersection disperser. Specifically, our main lemma is the following:

Lemma 9.18. *If $\mathcal{S}$ is an (r, ℓ, ζ)-intersection disperser, then, for any $\mathcal{F} = \{f_S\}_{S\in\mathcal{S}}$, $G^{\mathcal{F},\zeta}$ is $((r\ell)^{2(\ell-1)}, \ell)$-red/blue-transitive.*

We note here that both in Lemma 9.18 and Claim 9.19 below, the transitivity property holds not only for ℓ-walks as specified in the statements, but also for $(\ell+1)$-walks. However, since the latter does not yield any improvement to our main results, we work with only ℓ-walks, which makes the calculations cleaner.

In other words, we would like to show that, for every $S_1, S_2 \in \mathcal{S}$ that are joined by a red edge in $G^{\mathcal{F},\zeta}$, there are at most $(r\ell)^{2(\ell-1)}$ red-filled ℓ-walks from S_1 to S_2. The intersection disperser does not immediately imply such a bound, due to the requirement in the definition that the subcollections are disjoint. Rather, it only directly implies a bound on number of *disjoint* ℓ-walks from S_1 to S_2, where two ℓ walks from S_1 to S_2, $(T_1 = S_1, \dots, T_{\ell+1} = S_2), (T'_1 = S_1, \dots, T'_{\ell+1} = S_2) \in \mathcal{W}_\ell^{G^{\mathcal{F},\zeta}}(S_1, S_2)$, are said to be *disjoint* if they do not share any vertex except the starting and ending vertices, i.e., $\{T_2, \dots, T_\ell\} \cap \{T'_2, \dots, T'_\ell\} = \emptyset$. Note that multiple walks sharing starting and ending vertices are said to be disjoint if they are mutually disjoint. The following claim follows almost immediately from definition of intersection dispersers:

Claim 9.19. *If $\mathcal{S}$ is an (r, ℓ, ζ)-intersection disperser, then, for any $\mathcal{F} = \{f_S\}_{S\in\mathcal{S}}$, any integer $2 \leqslant p \leqslant \ell$ and any $\{S_1, S_2\} \in E_r^{\mathcal{F},\zeta}$, there are less than r disjoint p-walks from S_1 to S_2 in $G^{\mathcal{F},\zeta}$.*

Proof. Suppose for the sake of contradiction that $\mathcal{S}$ is an (r, ℓ, ζ)-intersection disperser but there exist $\mathcal{F} = \{f_S\}_{S \in \mathcal{S}}, 2 \leqslant p \leqslant \ell$ and $\{S_1, S_2\} \in E_r^{\mathcal{F},\zeta}$ such that there are at least r disjoint p-walks from S_1 to S_2. Let these walks be $(T_{1,1} = S_1, T_{1,2}, \dots, T_{1,p}, T_{1,p+1} = S_2), \dots, (T_{r,1} = S_1, T_{r,2}, \dots, T_{r,p}, T_{r,p+1} = S_2) \in \mathcal{W}_p^{G^{\mathcal{F},\zeta}}(S_1, S_2)$.

For each $i \in [r]$, consider any $x \in \bigcap_{j=1}^{p+1} T_{i,j}$. Apriori this intersection may be empty but since $\mathcal{S}$ is an intersection disperser this usually does not occur. Since $\{T_{i,j}, T_{i,j+1}\} \in E_b^{\mathcal{F},\zeta}$ for every $j \in [p]$, we have

$$f_{S_1}(x) = f_{T_{i,1}}(x) = f_{T_{i,2}}(x) = \cdots = f_{T_{i,p}}(x) = f_{T_{i,p+1}}(x) = f_{S_2}(x).$$

Hence, for every $x \in \bigcup_{i=1}^r \left(\bigcap_{j=1}^{p+1} T_{q,j}\right)$, $f_{S_1}(x) = f_{S_2}(x)$. Let T^* denote $\bigcup_{i=1}^r \left(\bigcap_{j=2}^{p} T_{q,j}\right)$. Since $T_{i,1} = S_1$ and $T_{i,p+1} = S_2$ for all $i \in [r]$, we have

$$\bigcup_{i=1}^{r} \left(\bigcap_{j=1}^{p+1} T_{q,j}\right) = (S_1 \cap S_2) \cap T^*.$$

In other words, f_{S_1} and f_{S_2} can only disagree on variables outside of T^*. However, since $\mathcal{S}$ is an (r, ℓ, ζ)-intersection disperser, we have $|T^*| \geqslant (1 - \zeta)n$. Hence, $\mathrm{disagr}(S_1, S_2) \leqslant \zeta n$, which contradicts with $\{S_1, S_2\} \in E_r^{\mathcal{F},\zeta}$. □

Since all 2-walks from S_1 to S_2 are disjoint, the above claim immediately gives a bound on the number of red-filled 2-walks from S_1 to S_2. To bound the number of red-filled walks of larger lengths, we will use induction on the length of the walks. Suppose that we have bounded the number of red-filled i-walks sharing starting and ending vertices for $i \leqslant z - 1$. The key idea in the proof is that we can use this inductive hypothesis to show that, for any $S_1, S_2, S \in \mathcal{S}$, few z-walks from S_1 to S_2 contain a given S. Here we say that a z-walk $(T_1 = S_1, \dots, T_z = S_2)$ from S_1 to S_2 *contains* S if $S \in \{T_2, \dots, T_z\}$. This implies that for a given z-walk from S_1 to S_2 there are only few walks that are not disjoint from it. This allows us to show that, if there are too many z-walks, then there must also be many disjoint z-walks as well, which would violate Claim 9.19. A formal proof of Lemma 9.18 based on this intuition is given below.

Proof of Lemma 9.18. For every integer i such that $2 \leqslant i \leqslant \ell$, let $P(i)$ denote the following statement: for every $S_1, S_2 \in \mathcal{S}$, $|\widehat{\mathcal{W}}_i^{G^{\mathcal{F},\zeta}}(S_1, S_2)| \leqslant (ri)^{2(i-1)}$. For convenience, let $B_i = (ri)^{2(i-1)}$ for every $2 \leqslant i \leqslant \ell$.

Base Case. Since every different 2-walks from S_1 to S_2 are disjoint, Claim 9.19 immediately implies that the number of 2-walks from S_1 to S_2 is at most $r \leqslant B_2$.

Inductive Step.

Suppose that, for some integer z such that $3 \leqslant z \leqslant \ell$, $P(3), \dots, P(z-1)$ are true. We will show that $P(z)$ is true. To do so, let us first prove that, for any fixed starting and ending vertices, any vertex cannot appears in too many red-filled z-walks, as stated in the following claim.

Claim 9.20. *For all $S_1, S_2, S \in \mathcal{S}$, the number of red-filled z-walks from S_1 to S_2 containing S in $G^{\mathcal{F},\zeta}$ is at most $B_z/(zr)$.*

Proof. First, observe that the number of red-filled z-walks from S_1 to S_2 containing S is at most the sum over all positions $2 \leqslant j \leqslant z$ of the number of z-walks from S_1 to S_2 such that the j-th vertex in the walk is S. More formally, the number of red-filled z-walks from S_1 to S_2 containing S is

$$|\{(T_1, \dots, T_{z+1}) \in \widehat{\mathcal{W}}_z^{G^{\mathcal{F},\zeta}}(S_1, S_2) \mid \exists 2 \leqslant j \leqslant z, T_j = S_j\}| \leqslant \sum_{j=2}^{z} |\{(T_1, \dots, T_{z+1}) \in \widehat{\mathcal{W}}_z^{G^{\mathcal{F},\zeta}}(S_1, S_2) \mid T_j = S\}|.$$

Now, for each $2 \leqslant j \leqslant z$, to bound the number of red-filled z-walks from S_1 to S_2 whose j-th vertex is S, let us consider the following three cases based on the value of j:

1. $3 \leqslant j \leqslant z - 1$. Observe that, for any such walk $(T_1 = S_1, T_2, \dots, T_j = S, \dots, T_z, T_{z+1} = S_2)$, the subwalk $(T_1 = S_1, \dots, T_j = S)$ and $(T_j = S, \dots, T_{z+1} = S_2)$ must be red-filled walks as well. Since the numbers of red-filled $(j-1)$-walks from S_1 to S and red-filled $(z+1-j)$-walks from S to S_2 are bounded by B_{j-1} and B_{z+1-j} respectively (from the inductive hypothesis), there are at most B_{j-1} choices of $(T_1 = S_1, \dots, T_j = S)$ and B_{z+1-j} choices of $(T_j = S, \dots, T_{z-1}, T_z = S_2)$. Hence, there are at most $B_{j-1}B_{z+1-j}$ red-filled z-walks from S_1 to S_2 whose j-th vertex is S.

2. $j = 2$. In this case, the subwalk $(T_j = S, \dots, T_{z+1} = S_2)$ must be a red-filled $(z-1)$-walk from S to S_2. Hence, the number of red-filled z-walks from S_1 to S_2 where $T_j = S$ is bounded above by B_{z-1}.

3. $j = z$. Similar to the previous case, we also have the bound of B_{z-1}.

For convenience, let $B_1 = 1$. The above argument gives us the following bound for every $2 \leqslant j \leqslant z$:

$$|\{(T_1, \dots, T_{z+1}) \in \widehat{\mathcal{W}}_z^{G^{\mathcal{F},\zeta}}(S_1, S_2) \mid T_j = S\}| \leqslant B_{j-1}B_{z+1-j}.$$

Summing this over j, we have the following upper bound on the number of red-filled z-walks from S_1 to S_2 containing S:

$$\sum_{j=2}^{z} B_{j-1}B_{z+1-j} = \sum_{j=2}^{z} (r(j-1))^{2(j-2)}(r(z+1-j))^{2(z-j)} \leqslant \sum_{j=2}^{z} (rz)^{2(z-2)} \leqslant B_z/(zr),$$

which concludes the proof of the claim. ⌟

Having proved the above claim, it is now easy to show that $P(z)$ is true. Suppose for the sake of contradiction that there exists $S_1, S_2 \in \mathcal{S}$ such that $|\widehat{\mathcal{W}}_z^{G^{\mathcal{F},\zeta}}(S_1, S_2)| > B_z$. Consider the following procedure of selecting disjoint walks from $\widehat{\mathcal{W}}_z^{G^{\mathcal{F},\zeta}}(S_1, S_2)$. First, initialize $U = \widehat{\mathcal{W}}_z^{G^{\mathcal{F},\zeta}}(S_1, S_2)$ and repeat the following process as long as $U \neq \emptyset$: select any $(T_1, \dots, T_{z+1}) \in U$ and remove every $(T'_1, \dots, T'_{z+1})$ that is not disjoint with $(T_1, \dots, T_{z+1})$ from U. Observe that, each time a walk $(T_1, \dots, T_{z+1})$ is selected, the number of walks removed from U is at most B_z/r; this is because each removed walk must contain at least one of $T_2, \dots, T_z$, but, from the above claim, each of these vertices are contained in at most $B_z/(zr)$ walks. Since we start with more than B_z walks, at least r walks are picked. These walks are disjoint z-walks starting from S_1 and S_2, which, due to Claim 9.19, is a contradiction. Thus, $P(z)$ is true as desired.

Hence, $P(\ell)$ is true, which, by definition, implies that $G_{\mathcal{F}}$ is $((r\ell)^{2(\ell-1)}, \ell)$-red/blue-transitive. □

Red/Blue-Transitivity from Uniformity

In Step 3 of our proof, we need to show red/blue-transitivity of the generalized two-level consistency graph $G^{\mathcal{F},\zeta,\zeta'}$. This is encapsulated in the following lemma.

Lemma 9.21. *If every subcollection $\widetilde{\mathcal{S}} \subseteq \mathcal{S}$ of size r is (γ, μ)-uniform, then, for any $\zeta \geqslant 0$, $\zeta' \geqslant \mu + 2\zeta/\gamma$ and $\mathcal{F} = \{f_S\}_{S \in \mathcal{S}}$, the generalized two-level consistency graph $G^{\mathcal{F},\zeta,\zeta'}$ is r-red/blue transitive.*

The proof of the lemma is quite simple. The key observation is that, if S_1 and S_2 are joined by a red edge and T is a common blue-neighbor in the graph $G^{\mathcal{F},\zeta,\zeta'}$, then it means that T only hits a small number (i.e. $2\zeta n$) of the variables on which f_{S_1} and f_{S_2} disagree. In other words, such variables appear less frequently in common blue-neighbors of S_1 and S_2. If the common-blue neighbor set is of size r, this contradicts the fact that the set is (γ, μ)-uniform. This intuition is formalized below.

Proof of Lemma 9.21. Suppose for the sake of contradiction that $G^{\mathcal{F},\zeta,\zeta'}$ is not r-red/blue transitive. That is, there exist $S_1, S_2 \in \mathcal{S}$ that are joined by a red edge such that there are r red-filled 2-walks (i.e. rbb triangle) from S_1 to S_2. Suppose that these walks are $(S_1, T_1, S_2), (S_1, T_2, S_2), \cdots, (S_1, T_r, S_2)$.

For every $i \in [r]$, since (S_1, T_i, S_2) is a 2-walk, $\{S_1, T_i\}$ and $\{S_2, T_i\}$ are blue edges. This implies that

$$\mathrm{disagr}(f_{S_1}, f_{T_i}), \mathrm{disagr}(f_{S_2}, f_{T_i}) \leqslant \zeta n. \tag{9.1}$$

On the other hand, we can lower bound $\mathbb{E}_{i \in [r]}[\mathrm{disagr}(f_{S_1}, f_{T_i}) + \mathrm{disagr}(f_{S_2}, f_{T_i})]$ as follows. First, let let $\mathrm{X}^{\mathrm{disagr}}$ denote the set of all $x \in S_1 \cap S_2$ such that $f_{S_1}(x) \neq f_{S_2}(x)$; since $\{S_1, S_2\}$ is a red edge, we have $|\mathrm{X}^{\mathrm{disagr}}| > \zeta' n$. We can rearrange $\mathbb{E}_{i \in [r]}[\mathrm{disagr}(f_{S_1}, f_{T_i}) + \mathrm{disagr}(f_{S_2}, f_{T_i})]$ as

$$\begin{aligned}
&\mathbb{E}_{i \in [r]}[\mathrm{disagr}(f_{S_1}, f_{T_i}) + \mathrm{disagr}(f_{S_2}, f_{T_i})] \\
&= \sum_{x \in [n]} \left(\Pr_{i \in [r]}[x \in (S_1 \cap T_i) \wedge f_{S_1}(x) \neq f_{T_i}(x)] + \Pr_{i \in [r]}[x \in (S_2 \cap T_i) \wedge f_{S_2}(x) \neq f_{T_i}(x)] \right) \\
&\geqslant \sum_{x \in \mathrm{X}^{\mathrm{disagr}}} \left(\Pr_{i \in [r]}[x \in T_i \wedge f_{S_1}(x) \neq f_{T_i}(x)] + \Pr_{i \in [r]}[x \in T_i \wedge f_{S_2}(x) \neq f_{T_i}(x)] \right) \\
&\geqslant \sum_{x \in \mathrm{X}^{\mathrm{disagr}}} \left(\Pr_{i \in [r]}[x \in T_i \wedge (f_{S_1}(x) \neq f_{T_i}(x) \vee f_{S_2}(x) \neq f_{T_i}(x))] \right) \\
&= \sum_{x \in \mathrm{X}^{\mathrm{disagr}}} \Pr_{i \in [r]}[x \in T_i] \qquad (9.2)
\end{aligned}$$

We remark here that the second inequality comes from union bound, whereas the last equality follows from the fact that $(f_{S_1}(x) \neq f_{T_i}(x)) \vee (f_{S_2}(x) \neq f_{T_i}(x))$ is always true when $f_{S_1}(x) \neq f_{S_2}(x)$.

Recall that $\{T_1, \dots, T_r\} \subseteq \mathcal{S}$ is a subcollection of size r and is thus (γ, μ)-uniform. Let $\mathrm{X}_{\geqslant\gamma}$ be the set of all $x \in [n]$ that appears in at least γ fraction of T_i's. The (γ, μ)-uniformity of $\{T_1, \dots, T_r\}$ implies that $|\mathrm{X}_{\geqslant\gamma}| \geqslant (1-\mu)n$. From this and from $|\mathrm{X}^{\text{disagr}}| > \zeta' n$, we can lower bound the right hand side of (9.2) further as follows:

$$\sum_{x \in \mathrm{X}^{\text{disagr}}} \Pr_{T_i \in \mathcal{T}}[x \in T_i] \geqslant \sum_{x \in \mathrm{X}^{\text{disagr}} \cap \mathrm{X}_{\geqslant\gamma}} \Pr_{T_i \in \mathcal{T}}[x \in T_i] \geqslant \gamma |\mathrm{X}^{\text{disagr}} \cap \mathrm{X}_{\geqslant\gamma}| > \gamma(\zeta' - \mu)n \geqslant 2\zeta n \quad (9.3)$$

where the last inequality comes from our assumption that $\zeta' \geqslant \mu + 2\zeta/\gamma$.

Finally, combining (9.1), (9.2) and (9.3) yields the desired contradiction. □

9.3.2 Finding Almost Non-Red Subgraph in Red/Blue-Transitive Graph

Recall that in two steps of our proofs, we need to utilize the red/blue transitivity of the (generalized) two-level consistency graph to find a large subgraph with certain number of non-red pairs:

- Specifically, in Step 2, we would like to show that, for appropriate values of q and ℓ, any (q, ℓ)-red/blue transitive graph with sufficiently many blue edges must contain a sufficiently large subgraph whose significant (i.e. $1/\ell^2$) fraction of pairs of vertices are non-red.
- Additionally, in Step 4, we need to show that any $o(d^2/k)$-red/blue transitive graph with sufficiently many blue edges must contain a sufficiently large subgraph such that almost all pairs of its vertices are non-red.

It turns out that a single lemma stated below suffices for both steps. In particular, the lemma below returns a subgraph such that roughly $1/\binom{\ell_0}{2}$ fraction of pairs of its vertices are non-red. Plugging in $\ell_0 = \ell$ recovers our former objective whereas setting $\ell_0 = 2$ satisfies the latter.

Lemma 9.22. *For every $k_0, q_0, \ell_0, d_0 \in \mathbb{N}$ such that $\ell_0 \geqslant 2$ and every k_0-vertex (q_0, ℓ_0)-red/blue-transitive graph $G = (V, E_r \cup E_b)$ such that $|E_b| \geqslant 2k_0 d_0$, there exist subsets of vertices $U_1, U_2 \subseteq V$ each of size at least d_0 such that $|\{(u,v) \in U_1 \times U_2 \mid \{u,v\} \notin E_r\}| \geqslant |U_1||U_2|(1 - \frac{q_0 k_0}{d_0^{\ell_0}})/\binom{\ell_0}{2}$. Moreover, when $\ell_0 = 2$, the previous statement remains true even with an additional requirement that $U_1 = U_2$.*

The proof of Lemma 9.22 below is exactly as sketched earlier in Subsection 9.1.

Proof of Lemma 9.22. We start by preprocessing the graph so that every vertex has blue-degree at least d_0. In particular, as long as there exists a vertex v whose blue-degree is at most d_0, we remove v from G. Let $G' = (V', E'_r \cup E'_b)$ be the graph at the end of this process. Note that we remove less than $k_0 d_0$ blue edges in total. Since at the beginning $|E_b| \geqslant 2k_0 d_0$, we have $|E'_b| \geqslant k_0 d_0$. Observe also that G' remains (q_0, ℓ_0)-red/blue-transitive.

Since V' is (q_0, ℓ_0)-red/blue-transitive, we can bound the number of red-filled ℓ_0-walk as follows.

$$|\widehat{\mathcal{W}}^{G'}_{\ell_0}| = \sum_{\substack{u,v \in V' \\ \{u,v\} \in E'_r}} |\widehat{\mathcal{W}}^{G'}_{\ell_0}(u,v)| \leqslant \sum_{\substack{u,v \in V' \\ \{u,v\} \in E'_r}} q_0 \leqslant q_0 k_0^2.$$

Moreover, notice that $|\mathcal{W}^{G'}_{\ell_0}| \geqslant (k_0 d_0) \cdot d_0^{\ell_0 - 1} = k_0 d_0^{\ell_0}$; this is simply because there are at least $k_0 d_0$ choices for (v_1, v_2) (i.e. all blue edges) and, for any $(v_1, \dots, v_{i-1})$, there are at least d_0 choices for v_i.

Hence, we have $|\widehat{\mathcal{W}}^{G'}_{\ell_0}|/|\mathcal{W}^{G'}_{\ell_0}| \leqslant q_0 k_0 / d_0^{\ell_0}$. This implies that $1 - q_0 k_0 / d_0^{\ell_0} \leqslant \Pr_{(v_1, \dots, v_{\ell_0+1}) \in \mathcal{W}^{G'}_{\ell_0}}[(v_1, \dots, v_{\ell_0+1}) \notin$ $\widehat{\mathcal{W}}^{G'}_{\ell_0}]$. This probability can be further rearranged as follows.

$$\Pr_{(v_1, \dots, v_{\ell_0+1}) \in \mathcal{W}^{G'}_{\ell_0}}[(v_1, \dots, v_{\ell_0+1}) \notin \widehat{\mathcal{W}}^{G'}_{\ell_0}] = \Pr_{(v_1, \dots, v_{\ell_0+1}) \in \mathcal{W}^{G'}_{\ell_0}}[\exists i, j \in [\ell_0 + 1] \text{ such that } j > i + 1, \{v_i, v_j\} \notin E'_r]$$

$$(\text{Union Bound}) \leqslant \sum_{\substack{i,j \in [\ell_0+1] \\ j > i+1}} \Pr_{(v_1, \dots, v_{\ell_0+1}) \in \mathcal{W}^{G'}_{\ell_0}}[\{v_i, v_j\} \notin E'_r].$$

Now, note that the number of pairs of $i, j \in [\ell_0 + 1]$ such that $j > i + 1$ is $\binom{\ell_0+1}{2} - \ell_0 = \binom{\ell_0}{2}$. This implies that there exists one such i, j such that $\Pr_{(v_1, \dots, v_{\ell_0+1}) \in \mathcal{W}^{G'}_{\ell_0}}[\{v_i, v_j\} \notin E'_r] \geqslant (1 - \frac{q_0 k_0}{d_0^{\ell_0}}) / \binom{\ell_0}{2}$. The probability $\Pr_{(v_1, \dots, v_{\ell_0+1}) \in \mathcal{W}^{G'}_{\ell_0}}[\{v_i, v_j\} \notin E'_r]$ can now be bounded as follows.

$$\begin{aligned}
&\Pr_{(v_1, \dots, v_{\ell_0+1}) \in \mathcal{W}^{G'}_{\ell_0}}[\{v_i, v_j\} \notin E'_r] \\
&= \sum_{u,v} \Pr_{(v_1, \dots, v_{\ell_0+1}) \in \mathcal{W}^{G'}_{\ell_0}}[\{v_i, v_j\} \notin E'_r \mid v_{i+1} = u \wedge v_{j-1} = v] \Pr_{(v_1, \dots, v_{\ell_0+1}) \in \mathcal{W}^{G'}_{\ell_0}}[v_{i+1} = u \wedge v_{j-1} = v] \\
&\leqslant \left(\max_{u,v} \Pr_{(v_1, \dots, v_{\ell_0+1}) \in \mathcal{W}^{G'}_{\ell_0}}[\{v_i, v_j\} \notin E'_r \mid v_{i+1} = u \wedge v_{j-1} = v] \right) \left(\sum_{u,v} \Pr_{(v_1, \dots, v_{\ell_0+1}) \in \mathcal{W}^{G'}_{\ell_0}}[v_{i+1} = u \wedge v_{j-1} = v] \right) \\
&= \max_{u,v} \Pr_{(v_1, \dots, v_{\ell_0+1}) \in \mathcal{W}^{G'}_{\ell_0}}[\{v_i, v_j\} \notin E'_r \mid v_{i+1} = u \wedge v_{j-1} = v]
\end{aligned}$$

where the summation and maximization is taken over all $u, v \in V'$ such that $\Pr_{(v_1, \dots, v_{\ell_0+1}) \in \mathcal{W}^{G'}_{\ell_0}}[v_{i+1} = u \wedge v_{j-1} = v]$ is non-zero. Hence, we can conclude that there exists $u^*, v^* \in V'$ such that

$$\Pr_{(v_1, \dots, v_{\ell_0+1}) \in \mathcal{W}^{G'}_{\ell_0}}[\{v_i, v_j\} \notin E'_r \mid v_{i+1} = u^* \wedge v_{j-1} = v^*] \geqslant (1 - \frac{q_0 k_0}{d_0^{\ell_0}}) / \binom{\ell_0}{2}.$$

The expression on the left is exactly $|\{(u, v) \in N_b(u^*) \times N_b(v^*) \mid \{u, v\} \notin E'_r\}| / (|N_b(u^*)| \cdot |N_b(v^*)|)$. From this and from every vertex in G' has blue-degree at least d_0, $U_1 = N_b(u^*), U_2 = N_b(v^*)$ are the desired sets. Finally, observe that, when $\ell = 2$, we must have $i = 1$ and $j = 3$, resulting in $v_{i+1} = v_{j-1}$; this implies that $u^* = v^*$ and we have $U_1 = U_2$. □

9.3.3 Majority Decoding of an Almost Non-Red Subgraph

In the last step of our proof, we will decode a global function g from a sufficiently large almost non-red subgraph in the two-level consistency graph $G^{\mathcal{F},\zeta,\zeta'}$. Recall that an almost non-red subgraph in $G^{\mathcal{F},\zeta,\zeta'}$ simply corresponds to a subcollection $\mathcal{S}'$ such that, for almost all pairs $(S_1, S_2) \in \mathcal{S}' \times \mathcal{S}'$, f_{S_1} is ζ'-consistent with f_{S_2}. The main result of this subsection is that, given such $\mathcal{S}'$, we can find a global function g that approximately agrees with most of the local functions in the subcollection. This is stated more precisely below.

Lemma 9.23. *Let $\mathcal{F} = \{f_S\}_{S \in \mathcal{S}'}$ be a collection of functions such that agr$_{\zeta'}(\mathcal{F}) \geqslant 1 - \kappa$. Then, the function $g : [n] \to \{0, 1\}$ defined by $g(x) = \operatorname{Majority}_{\substack{S\in\mathcal{S}'\\ x\in S}}(f_S(x))$ satisfies*

$$\mathbb{E}_{S\in\mathcal{S}'}\left[\operatorname{disagr}(g, f_S)\right] \leqslant n\sqrt{\kappa + \zeta'}.$$

Proof. Recall that $\operatorname{agr}_{\zeta'}(\mathcal{F}) \geqslant 1-\kappa$ is equivalent to $\Pr_{S_1,S_2\in\mathcal{S}'}\left[f_{S_1}(x) \overset{\zeta'}{\approx} f_{S_2}(x)\right] \geqslant 1-\kappa$. Hence,

$$\mathbb{E}_{S_1,S_2\in\mathcal{S}'}[\operatorname{disagr}(f_{S_1}, f_{S_2})] \leqslant \Pr_{S_1,S_2\in\mathcal{S}'}[f_{S_1} \overset{\zeta'}{\not\approx} f_{S_2}]\cdot n + \Pr_{S_1,S_2\in\mathcal{S}'}[f_{S_1} \overset{\zeta'}{\approx} f_{S_2}]\cdot(\zeta' n) \leqslant (\kappa+\zeta')n. \tag{9.4}$$

We can then lower bound the expression on the left hand side as follows.

$$\begin{aligned}
\mathbb{E}_{S_1,S_2\in\mathcal{S}'}[\operatorname{disagr}(f_{S_1}, f_{S_2})] &= \sum_{x\in[n]} \Pr_{S_1,S_2\in\mathcal{S}'}[x\in S_1 \wedge x \in S_2 \wedge f_{S_1}(x) \neq f_{S_2}(x)] \\
&\geqslant \sum_{x\in[n]} \Pr_{S_1,S_2\in\mathcal{S}'}[x\in S_1 \wedge x \in S_2 \wedge f_{S_1}(x) \neq g(x) \wedge f_{S_2}(x) = g(x)] \\
&= \sum_{x\in[n]} \Pr_{S_1\in\mathcal{S}'}[x\in S_1 \wedge f_{S_1}(x) \neq g(x)] \Pr_{S_2\in\mathcal{S}'}[x\in S_2 \wedge f_{S_2}(x) = g(x)] \\
\text{(Since } g(x) = \operatorname{Majority}_{\substack{S\in\mathcal{S}'\\ x\in S}}(f_S(x))\text{)} &\geqslant \sum_{x\in[n]} \Pr_{S_1\in\mathcal{S}'}[x\in S_1 \wedge f_{S_1}(x) \neq g(x)] \Pr_{S_2\in\mathcal{S}'}[x\in S_2 \wedge f_{S_2}(x) \neq g(x)] \\
&= \sum_{x\in[n]} \left(\Pr_{S\in\mathcal{S}'}[x\in S \wedge f_S(x) \neq g(x)]\right)^2 \\
\text{(Power Mean Inequality)} &\geqslant \frac{1}{n}\left(\sum_{x\in[n]} \Pr_{S\in\mathcal{S}'}[x\in S \wedge f_S(x) \neq g(x)]\right)^2 \\
&= \frac{1}{n}\left(\mathbb{E}_{S\in\mathcal{S}'}\left[\operatorname{disagr}(g, f_S)\right]\right)^2.
\end{aligned} \tag{9.5}$$

Combining (9.4) and (9.5) gives the desired bound. □

9.3.4 Putting Things Together: Proof of Theorem 9.17

Finally, we put all five steps together as outlined at the beginning of this section. This is formalized below. Note that Theorem 9.17 follows from the theorem below simply by Markov inequality.

Theorem 9.24. *For any $0 < \eta, \zeta, \gamma, \mu < 1$ and $r, \ell, k, h, n, d \in \mathbb{N}$ such that $\ell \geqslant 2$, let $\mathcal{S}$ be any collection of k subsets of $[n]$ such that $\mathcal{S}$ is (r, ℓ, ζ)-intersection disperser and every subcollection $\widetilde{\mathcal{S}} \subseteq \mathcal{S}$ of size h is (γ, μ)-uniform, and let $\mathcal{F} = \{f_S\}_{S \in \mathcal{S}}$ be any collection of functions. If $\delta :=$ agr$(\mathcal{F}) \geqslant \frac{10+64(r\ell)^2 k^{1/\ell}}{k}$, then there exists a subcollection $\mathcal{S}' \subseteq \mathcal{S}$ of size at least $\frac{\delta k}{128\ell^2}$ and a function $g : [n] \to \{0, 1\}$ such that*

$$\mathbb{E}_{S \in \mathcal{S}'}\left[\text{disagr}(g, f_S)\right] \leqslant n\sqrt{\frac{65536 h \ell^6}{\delta k}} + \mu + 2\zeta/\gamma.$$

Proof. Observe that agr$(\mathcal{F})$ directly corresponds to the number of blue edges $|E_b^{\mathcal{F},\zeta}|$ in the two-level consistency graph $G^{\mathcal{F},\zeta}$. In particular, agr$(\mathcal{F}) = \delta$ means that the number of blue edges is $(\delta k^2 - k)/2$. Since $\mathcal{S}$ is a (r, ℓ, ζ)-intersection disperser, Lemma 9.18 implies that $G^{\mathcal{F},\zeta}$ is $((r\ell)^{2(\ell-1)}, \ell)$-red/blue-transitive. Let $d = \lfloor \frac{|E_b^{\mathcal{F},\zeta}|}{2k} \rfloor = \lfloor \frac{\delta k - 1}{4} \rfloor$; since $\delta \geqslant \frac{10+64(r\ell)^2 k^{1/\ell}}{k}$, we have $d \geqslant (r\ell)^2 k^{1/\ell}$.

Applying Lemma 9.22 with $G = G^{\mathcal{F},\zeta}, k_0 = k, \ell_0 = \ell, q_0 = (r\ell)^{2(\ell-1)}$ and $d_0 = d$, we can conclude that there exist subsets $U_1, U_2 \subseteq V^{\mathcal{F},\zeta}$ each of size at least d such that

$$\frac{|\{(u, v) \in U_1 \times U_2 \mid \{u, v\} \notin E_r^{\mathcal{F},\zeta}\}|}{|U_1||U_2|} \geqslant \frac{1 - (r\ell)^{2(\ell-1)} k/d^\ell}{\binom{\ell}{2}} \geqslant \frac{1}{\ell^2}$$

where the last inequality follows from our aforementioned lower bound on d and from $\ell \geqslant 2$.

Next, observe that, if we let U_1' and U_2' be random subsets of U_1, U_2 of size d, then we have

$$\mathbb{E}_{U_1', U_2'}\left[\frac{|\{(u', v') \in U_1' \times U_2' \mid \{u', v'\} \notin E_r^{\mathcal{F},\zeta}\}}{d^2}\right] = \frac{|\{(u, v) \in U_1 \times U_2 \mid \{u, v\} \notin E_r^{\mathcal{F},\zeta}\}|}{|U_1||U_2|}.$$

As a result, there exists $\widetilde{U}_1, \widetilde{U}_2$ each of size exactly d such that

$$\frac{|\{(\tilde{u}, \tilde{v}) \in \widetilde{U}_1 \times \widetilde{U}_2 \mid \{\tilde{u}, \tilde{v}\} \notin E_r^{\mathcal{F},\zeta}\}|}{d^2} \geqslant \frac{1}{\ell^2}. \tag{9.6}$$

Now, let $\widetilde{U} = \widetilde{U}_1 \cup \widetilde{U}_2$. (9.6) implies that the number of $\{\tilde{u}, \tilde{v}\} \subseteq \widetilde{U}$ such that $\{\tilde{u}, \tilde{v}\} \notin E_r^{\mathcal{F},\zeta}$ is at least $d^2/(2\ell^2) - d$ where the factor of 2 comes from the fact that each pair $\{\tilde{u}, \tilde{v}\}$ is double counted in the left hand side of (9.6) and the subtraction of d comes from the fact that the left hand side of (9.6) also count the case where $\tilde{u} = \tilde{v}$.

Now, let $\widetilde{\mathcal{F}} = \{f_S\}_{S \in \widetilde{U}}$, $\zeta' = \mu + 2\zeta/\gamma$ and consider the two-level consistency graph $G^{\widetilde{\mathcal{F}},\zeta,\zeta'}$. Observe that $\{\tilde{u}, \tilde{v}\}$ is a blue edge in this new graph $G^{\widetilde{\mathcal{F}},\zeta,\zeta'}$ if and only if it is not a red edge in the original graph $G^{\mathcal{F},\zeta}$. Hence, the bound derived in the previous paragraph implies that

$$|E_b^{\widetilde{\mathcal{F}},\zeta,\zeta'}| \geqslant \frac{d^2}{2\ell^2} - d.$$

Let $d' = |E_b^{\widetilde{\mathcal{F}},\zeta,\zeta'}|/(2|\widetilde{U}|) \geqslant |E_b^{\widetilde{\mathcal{F}},\zeta,\zeta'}|/(4d) = d/(8\ell^2) - 1/4$. Recall that $d = \lfloor(\delta k - 1)/4\rfloor$; from $\delta \geqslant (10 + 64(r\ell)^2)/k$, we have $d \geqslant 8\ell^2$ and $d \geqslant \delta k/8$. Hence, we have $d' \geqslant d/(16\ell^2) \geqslant \delta k/(128\ell^2)$.

Furthermore, by Lemma 9.21 and from our assumption that every subcollection of $\mathcal{S}$ of size h is (γ, μ)-uniform, the graph $G^{\mathcal{F},\zeta,\zeta'}$ is h-red/blue transitive. Applying Lemma 9.22 with $G = G^{\mathcal{F},\zeta,\zeta'}, k_0 = |\widetilde{U}| \leqslant 2d, \ell_0 = 2, q_0 = h$ and $d_0 = d'$, there must be a set $U' \subseteq \widetilde{U}$ of size at least d' such that

$$\frac{|\{(u', v') \in U' \times U' \mid \{u', v'\} \notin E_r^{\mathcal{F},\zeta,\zeta'}\}|}{|U'|^2} \geqslant 1 - \frac{2hd}{(d')^2} \geqslant 1 - \frac{512h\ell^4}{d} \geqslant 1 - \frac{65536h\ell^6}{\delta k} \tag{9.7}$$

where the last two inequalities follow from $d' \geqslant d/(16\ell^2)$ and $d' \geqslant \delta k/(128\ell^2)$ respectively.

Let $\mathcal{F}' = \{f_S\}_{S \in U'}$. Observe that the expression on the left hand side of (9.7) is simply $\operatorname{agr}_{\zeta'}(\mathcal{F}')$. Hence, by Lemma 9.23, there exists a function $g : [n] \to \{0, 1\}$ such that

$$\mathbb{E}_{S \in U'}[\operatorname{disagr}(g, f_S)] \leqslant n\sqrt{\frac{65536h\ell^6}{\delta k}} + \zeta' = n\sqrt{\frac{65536h\ell^6}{\delta k}} + \mu + 2\zeta/\gamma.$$

In other words, U' is the desired subcollection, which completes our proof. □

9.4 Soundness Analysis of the Reduction

We will next use our agreement theorem to analyze the soundness of our reduction. The soundness of our reduction can be stated more precisely as follows:

Theorem 9.25. *For any $\Delta \in \mathbb{N}$, let Φ be any 3-CNF formula with variable set X and clause set $\mathcal{C}$ such that each variable appears in at most Δ clauses. Moreover, for any $0 < \eta, \zeta, \gamma, \mu < 1$ and $r, \ell, k, h, d \in \mathbb{N}$ such that $\ell \geqslant 2$, let $\mathcal{T}$ be any collection of k subsets of $\mathcal{C}$ such that $\mathcal{T}$ is (r, ℓ, ζ)-intersection disperser and every subcollection $\widetilde{\mathcal{T}} \subseteq \mathcal{T}$ of size h is (γ, μ)-uniform. If $\operatorname{val}(\Phi) < 1 - \mu - (3\Delta/\gamma)\sqrt{4\Delta\mu + 6\Delta\zeta/\gamma}$, then*

$$\operatorname{val}(\Gamma_{\Phi,\mathcal{T}}) < \frac{10 + 64(r\ell)^2 k^{1/\ell} + 65536h\ell^2/\mu}{k}.$$

Again, we will prove the contrapositive that if $\operatorname{val}(\Gamma_{\Phi,\mathcal{T}})$ is large, then $\operatorname{val}(\Phi)$ is also large. Recall that $\operatorname{val}(\Gamma_{\Phi,\mathcal{T}})$ being large implies that there exists a labeling $\sigma = \{\sigma_T\}_{T \in \mathcal{T}}$ with high agreement probability. We would like to apply our agreement testing theorem. Note however that Theorem 9.24 only applies when the subsets of *variables* are "well-behaved" (i.e. satisfies uniformity and is an intersection disperser). However, in our construction, the subset of variables are not random, rather they are variable set of random subsets of clauses. Hence, we will first need to translate the "well-behavedness" from subsets of clauses to their corresponding variable sets; this is shown in Section 9.4.1. Once this is in place, we can apply Theorem 9.24, which gives us a global assignment that approximately agrees with many σ_T's. We show in Section 9.4.2 that such assignment satisfies most of the constraint, which implies that $\operatorname{val}(\Phi)$ must be large as desired. The full proof of Theorem 9.25 can then be found in Section 9.4.3.

9.4.1 Well-Behave Subsets of Clauses vs Well-Behave Subsets of Variables

For convenient, let us define an additional notation:

Definition 9.26. *Let Φ be any 3-CNF formula and $\mathcal{T}$ be any subset of clauses of Φ. We use $\mathcal{S}_{\Phi,\mathcal{T}}$ to denote the collection $\{\text{var}(T)\}_{T\in\mathcal{T}}$ of subsets of variables.*

Note that the subsets in $\mathcal{S}_{\Phi,\mathcal{T}}$ are indeed the variable sets of our labeling $\sigma = \{\sigma_T\}_{T\in\mathcal{T}}$. Moreover, it is rather straightforward to see that both uniformity and intersection disperser conditions translate from $\mathcal{T}$ to $\mathcal{S}_{\Phi,\mathcal{T}}$ with little loss in parameters, provided that each variable appears in bounded number of clauses. These observations are formalized and proved below.

Lemma 9.27. *Suppose that every variable in Φ appears in at least one and at most Δ clauses. If $\mathcal{T}$ is (γ, μ)-uniform, then $\mathcal{S}_{\Phi,\mathcal{T}}$ is $(\gamma, 3\Delta\mu)$-uniform.*

Proof. First, observe that, since each variable appears in at most Δ clauses, we have $n \geqslant m/\Delta$. Now, let $\mathcal{C}_{\geqslant\gamma} = \{C \in \mathcal{C} \mid \Pr_{T\in\mathcal{T}}[C \in T] \geqslant \gamma\}$ and $\mathrm{X}_{\geqslant\gamma} = \{x \in \mathrm{X} \mid \Pr_{S\in\mathcal{S}_{\Phi,\mathcal{T}}}[x \in S] \geqslant \gamma\}$. Recall that (γ, μ)-uniformity of $\mathcal{T}$ implies that $|C_{\geqslant\gamma}| \geqslant (1-\mu)m$. Observe that any $x \in \text{var}(\mathcal{C}_{\geqslant\gamma})$ must also be contained in $\mathrm{X}_{\geqslant\gamma}$. Since every variable appears in at least one clauses, we have that every variable $x \notin \mathrm{X}_{\geqslant\gamma}$ must be in $\text{var}(\mathcal{C} \setminus \mathcal{C}_{\geqslant\gamma})$. As a result, $|\mathrm{X} \setminus \mathrm{X}_{\geqslant\gamma}| \leqslant 3\mu m$. From this and from $n \geqslant m/\Delta$, we arrive at the desired conclusion. □

Lemma 9.28. *Suppose that every variable in Φ appears in at least one and at most Δ clauses. If $\mathcal{T}$ is an (r, ℓ, η)-intersection disperser, then $\mathcal{S}_{\Phi,\mathcal{T}}$ is $(r, \ell, 3\Delta\eta)$-intersection disperser.*

Proof. Consider any r disjoint subcollections $\mathcal{S}^1 = \{S_{1,1}, \dots, S_{1,p_1}\}, \cdots, \mathcal{S}^r = \{S_{r,1}, \dots, S_{r,p_r}\} \subseteq \mathcal{S}_{\Phi,\mathcal{T}}$ each of size at most ℓ. From our definition of $\mathcal{S}_{\Phi,\mathcal{T}}$, there is an r disjoint subcollections $\mathcal{T}^1 = \{T_{1,1}, \dots, T_{1,p_1}\}, \dots, \mathcal{T}^r = \{T_{r,1}, \dots, T_{r,p_r}\} \subseteq \mathcal{T}$ such that $S_{i,j} = \text{var}(T_{i,j})$ for all $i \in [r]$ and $j \in [p_i]$. Observe that

$$\bigcup_{i=1}^{r} \left(\bigcap_{S\in\mathcal{S}^i} S \right) \supseteq \text{var}\left(\bigcup_{i=1}^{r} \left(\bigcap_{T\in\mathcal{T}^i} T \right) \right).$$

Moreover, since $\mathcal{T}$ is an (r, ℓ, η)-intersection disperser, we have $|\bigcup_{i=1}^{r} \left(\bigcap_{T\in\mathcal{T}^i} T\right)| \geqslant (1-\eta)m$. As a result, since each variable appears in at least one clause, we indeed have $|\bigcup_{i=1}^{r} \left(\bigcap_{S\in\mathcal{S}^i} S\right)| \geqslant n - 3\eta m \geqslant (1 - 3\Delta\eta)n$ as desired. □

9.4.2 Global Function with Many Agreements is a Good Assignment

In this subsection, we show that any global assignment that are approximately consistent with a collection of labels $\{\sigma_T\}_{T\in\mathcal{T}^*}$ must satisfy most of the constraints, assuming that $\mathcal{T}^*$ is sufficiently uniform, which is stated more precisely below.

Lemma 9.29. *Let $\mathcal{T}^*$ be any (γ, μ)-uniform collection of subsets of clauses and σ be any labeling of $\mathcal{T}^*$. If there exists $\psi : \mathrm{X} \to \{0, 1\}$ such that $\mathbb{E}_{T\in\mathcal{T}^*}[\text{disagr}(\psi, \sigma_T)] \leqslant \nu n$, then $\text{val}(\psi) \geqslant 1 - \mu - 3\,\nu\Delta/\gamma$.*

The key to proving that ψ violates few clauses is that, if a clause C is violated, then, for each $T \in \mathcal{T}^*$ that contains T, σ_T and ψ must disagree on at least one of $\mathrm{var}(C)$ because σ_T satisfies C but ψ violates it. Hence, if C appears often in $\mathcal{T}$, then it contributes to many disagreements between σ_T and ψ; the uniformity condition helps us ensure that most C indeed appear often in $\mathcal{T}$. Comparing this lower bound against the assumed upper bound on the expected disagreements gives us the desired result. This intuition is formalized below.

Proof. Let $\mathcal{C}_{\geqslant\gamma}$ denote the set of all clauses that appear in at least γ fraction of $T \in \mathcal{T}^*$, i.e., $\mathcal{C}_{\geqslant\gamma} = \{C \in \mathcal{C} \mid \Pr_{T\in\mathcal{T}^*}[C \in T] \geqslant \gamma\}$. Since $\mathcal{T}^*$ is (γ, μ)-uniform, we have $|\mathcal{C}_{\geqslant\gamma}| \geqslant (1-\mu)m$.

Since each variable x appears in at most Δ clauses, we can obtain the following bound:

$$\begin{aligned}
\mathbb{E}_{T\in\mathcal{T}^*}[\mathrm{disagr}(\psi, \sigma_T)] &= \sum_{x\in \mathrm{X}} \Pr_{T\in\mathcal{T}^*}[x \in \mathrm{var}(T) \wedge \sigma_T(x) \neq \psi(x)] \\
&\geqslant \frac{1}{\Delta} \sum_{C\in\mathcal{C}_{\geqslant\gamma}} \sum_{x\in\mathrm{var}(C)} \Pr_{T\in\mathcal{T}^*}[x \in \mathrm{var}(T) \wedge \sigma_T(x) \neq \psi(x)] \\
&\geqslant \frac{1}{\Delta} \sum_{C\in\mathcal{C}_{\geqslant\gamma}} \sum_{x\in\mathrm{var}(C)} \Pr_{T\in\mathcal{T}^*}[C \in T \wedge \sigma_T(x) \neq \psi(x)] \\
\text{(Union Bound)} &\geqslant \frac{1}{\Delta} \sum_{C\in\mathcal{C}_{\geqslant\gamma}} \Pr_{T\in\mathcal{T}^*}\left[C \in T \wedge \left(\bigvee_{x\in\mathrm{var}(C)} \sigma_T(x) \neq \psi(x)\right)\right] \\
&\geqslant \frac{\gamma}{\Delta} \sum_{C\in\mathcal{C}_{\geqslant\gamma}} \Pr_{T\in\mathcal{T}^*}\left[\bigvee_{x\in\mathrm{var}(C)} \sigma_T(x) \neq \psi(x) \,\middle|\, C \in T\right] \qquad (9.8)
\end{aligned}$$

Note here that we use the fact that each variable appears in at most Δ clauses in the first inequality and that the last inequality follows from the fact that each $C \in \mathcal{C}_{\geqslant\gamma}$ appears in at least γ fraction of $T \in \mathcal{T}^*$. The rest of the inequalities are trivial.

Let $\mathcal{C}_{\mathrm{UNSAT}}$ denote the set of clauses violated by ψ. Observe that, for any $C \in \mathcal{C}_{\mathrm{UNSAT}}$ and any $T \in \mathcal{T}^*$ such that $C \in T$, σ_T must disagree with ψ on at least one of $x \in \mathrm{var}(C)$; this is simply because C is satisfied by σ_T but violated by ψ. In other words, for every $C \in C_{\mathrm{UNSAT}}$, we have

$$\Pr_{T\in\mathcal{T}^*}\left[\bigvee_{x\in\mathrm{var}(C)} \sigma_T(x) \neq \psi(x) \,\middle|\, C \in T\right] = 1. \qquad (9.9)$$

(9.8), (9.9) and the assumption that $\mathbb{E}_{T\in\mathcal{T}^*}[\mathrm{disagr}(\psi, \sigma_T)] \leqslant \nu n$ imply that

$$\nu\Delta n/\gamma \geqslant |C_{\geqslant\gamma} \cap \mathcal{C}_{\mathrm{UNSAT}}|.$$

Since $C_{\geqslant\gamma} \geqslant m(1-\mu)$ and $n \leqslant 3m$ (from every variable appears in at least one clause), we can conclude that $|\mathcal{C}_{\mathrm{UNSAT}}| \leqslant \mu m + 3\nu\Delta m/\gamma$. As a result, $\mathrm{val}(\psi) \geqslant 1 - \mu - 3\nu\Delta/\gamma$ as desired. □

9.4.3 Putting Things Together: Proof of Theorem 9.25

Proof of Theorem 9.25. We may assume w.l.o.g. that each variable appears in at least one clause.

We will prove the theorem by contrapositive. Suppose that $\mathrm{val}(\Gamma_{\Phi,\mathcal{T}}) \geqslant \frac{10+64(r\ell)^2k^{1/\ell}+65536h\ell^2/\mu}{k}$. This means that there exists a labeling $\sigma = \{\sigma_T\}_{T\in\mathcal{T}}$ such that $\mathrm{val}(\sigma) \geqslant \frac{10+64(r\ell)^2k^{1/\ell}+65536h\ell^2/\mu}{k}$; this also means that, if we view σ as a collection of functions $\mathcal{F} = \{f_S\}_{S\in\mathcal{S}_{\Phi,\mathcal{T}}}$ where $f_{\mathrm{var}(T)} = \sigma_T$, then $\mathrm{agr}(\mathcal{F}) \geqslant \frac{10+64(r\ell)^2k^{1/\ell}+65536h\ell^2/\mu}{k}$. Let $\delta = \mathrm{agr}(\mathcal{F})$.

Furthermore, Lemmas 9.28 and 9.27 imply that $\mathcal{S}_{\Phi,\mathcal{T}}$ is an $(r, \ell, 3\Delta\zeta)$-intersection disperser and every subcollection of $\mathcal{S}_{\Phi,\mathcal{T}}$ of size h is $(\gamma, 3\Delta\mu)$-uniform respectively. This enables us to apply Theorem 9.24 on $\mathcal{F}$, which yields a subcollection $\mathcal{S}' \subseteq \mathcal{S}_{\Phi,\mathcal{T}}$ of size at least $\frac{\delta k}{128\ell^2} \geqslant h$ and $g : \mathrm{X} \to \{0, 1\}$ such that

$$\mathbb{E}_{S\in\mathcal{S}'}\left[\mathrm{disagr}(g, f_S)\right] \leqslant n\sqrt{\frac{65536h\ell^6}{\delta k} + 3\Delta\mu + 6\Delta\zeta/\gamma} \leqslant n\sqrt{4\Delta\mu + 6\Delta\zeta/\gamma}$$

where the second inequality follows from $\delta k \geqslant 65536h\ell^2/\mu$.

Let $\mathcal{S}^*$ be the subcollection of $\mathcal{S}'$ of size h that minimizes $\mathbb{E}_{S\in\mathcal{S}^*}\left[\mathrm{disagr}(g, f_S)\right]$. Observe that $\mathbb{E}_{S\in\mathcal{S}^*}\left[\mathrm{disagr}(g, f_S)\right] \leqslant \mathbb{E}_{S\in\mathcal{S}'}\left[\mathrm{disagr}(g, f_S)\right] \leqslant n\sqrt{4\Delta\mu + 6\Delta\zeta/\gamma}$. This is equivalent to saying that there exists a subcollection $\mathcal{T}^* \subseteq \mathcal{T}$ of size h such that $\mathbb{E}_{T\in\mathcal{T}^*}\left[\mathrm{disagr}(g, \sigma_T)\right] \leqslant n\sqrt{4\Delta\mu + 6\Delta\zeta/\gamma}$.

Since every subcollection of $\mathcal{T}$ of size h is (γ, μ)-uniform, we can apply Lemma 9.29 to infer that $\mathrm{val}(\Phi) \geqslant 1 - \mu - (3\Delta/\gamma)\sqrt{4\Delta\mu + 6\Delta\zeta/\gamma}$ as desired. □

9.5 Proof of Inapproximability Results of 2-CSPs

The inapproximability results for 2-CSPs can be shown simply by plugging in appropriate parameters to Theorem 9.25. More specifically, for ETH-hardness, since there is a polylog$m$ loss in the PCP Theorem (Theorem 2.2), we need to select our $\alpha = 1/\mathrm{polylog} m$ so that the size (and running time) of the reduction is $2^{o(m)}$. Now, observe that the parameter r in Theorem 9.25 for the intersection disperser property grows with $(1/\alpha)^\ell$ (see Lemma 9.16). Since the soundness guarantee in Theorem 9.25 is of the form $k^{O(1/\ell)}(r\ell)^{O(1)}/k = k^{O(1/\ell)}(1/\alpha)^{O(\ell)}/k$, it is minimized when ℓ is roughly $\sqrt{\log k}$, which yields the soundness of $2^{(\log k)^{1/2+o(1)}}/k$. Other parameters are chosen accordingly.

Proof of Theorem 9.1. Let c, ε, Δ be constants from Theorem 2.2.

For any 3-CNF formula $\widetilde{\Phi}$ with m clauses, let us first apply the nearly-linear size PCP from Theorem 2.2 to produce a 3-CNF formula Φ with $m' = O(m \log^c m)$ clauses. Let us also define the following parameters:

- $q = \lfloor \log m \rfloor^{c+1}$ and $\alpha = \frac{1}{q} = \frac{1}{\lfloor \log m \rfloor^{c+1}}$,
- $\gamma = \frac{\alpha}{2} = \frac{1}{2\lfloor \log m \rfloor^{c+1}}$,

- $\mu = \frac{\varepsilon^2\gamma^2}{288\Delta^3} = \Theta_{\varepsilon,\Delta}\left(\frac{1}{(\log m)^{2c+2}}\right)$,
- $\zeta = \frac{\varepsilon^2\gamma^3}{432\Delta^3} = \Theta_{\varepsilon,\Delta}\left(\frac{1}{(\log m)^{3c+3}}\right)$,
- $\ell = (\log m)^{1/4}$,
- $r = \lceil\frac{\ln(2/\zeta)}{\alpha^\ell}\rceil = 2^{\Theta_{\varepsilon,\Delta,c}((\log m)^{1/4}\log\log m)}$,
- $h = \lceil 8\ln(2/\mu)/\alpha\rceil = \Theta_{\varepsilon,\Delta,c}((\log m)^{c+1})$,
- $k = 2^{\ell^2} = 2^{\sqrt{\log m}}$.

We then use Lemma 9.16 with the above parameters $q, \alpha, \mu, \zeta, k, \ell$ to construct a collection $\mathcal{S}$ of subsets of clauses of Φ such that the following conditions hold.

- Every subset in $\mathcal{S}$ has size at most $2\alpha m' = o(m)$.
- $\mathcal{S}$ is a (r, ℓ, ζ)-disperser.
- Any subcollection $\widetilde{\mathcal{S}} \subseteq \mathcal{S}$ of size h is $(\alpha/2, \mu)$-uniform.

Note that, for our choices of parameter, q^{k+1} is $2^{2^{O(\sqrt{\log m})}}$; this means that, for sufficiently large m, we indeed have $m' \geqslant m \geqslant q^{k+1}$ and the running time needed to produce $\mathcal{S}$ is $O(m \cdot q^k) = O(m^2)$. Note that we assume without loss of generality here that $m' \geqslant m$; if this is not the case, we can simply copy the formula Φ $\lceil m/m'\rceil$ times using new variables each time, which does not change the value of the formula.

We now consider the 2-CSP instance $\Gamma_{\Phi,\mathcal{S}}$. Observe that the running time used to create $\Gamma_{\Phi,\mathcal{S}}$ (and hence also the size of $\Gamma_{\Phi,\mathcal{S}}$) is no more than $\text{poly}(k)\cdot 2^{o(\alpha m')} = 2^{o(m)}$. Moreover, if $\text{val}(\widetilde{\Phi}) = 1$, then $\text{val}(\Phi) = 1$ and it is easy to see that $\text{val}(\Gamma_{\Phi,\mathcal{S}}) = 1$ as well.

On the other hand, if $\text{val}(\widetilde{\Phi}) < 1$, then $\text{val}(\Phi) < 1 - \varepsilon$. Due to our choice of parameters, we can apply Theorem 9.25, which implies that

$$\text{val}(\Gamma_{\Phi,\mathcal{S}}) < \frac{O((r\ell)^2k^{1/\ell} + h\ell^2/\mu)}{k} = 2^{O_{\varepsilon,\Delta,c}((\log m)^{1/4}\log\log m)}/k = 2^{O_{\varepsilon,\Delta,c}(\sqrt{\log k}\log\log k)}/k.$$

For sufficiently large m (depending only on $c, \varepsilon, \Delta, \rho$), this term is at most $2^{(\log k)^{1/2+\rho}}/k$. Hence, if there exists a polynomial time that can distinguish the two cases in the theorem statement, we can run this algorithm on $\Gamma_{\Phi,\mathcal{S}}$ to decide whether $\widetilde{\Phi}$ is satisfiable in $2^{o(m)}$ time, contradicting ETH. □

For Gap-ETH-hardness, we do not incur a loss of polylog$m$ from the PCP Theorem anymore. Thus, it suffices to chose α to be any function that converges to zero as k goes to ∞ (e.g. $\alpha = 1/\log\log k$), and k can now be independent of m. The rest of the analysis remains unchanged.

Proof of Theorem 9.2. Let $\delta, \varepsilon, \Delta$ be the constants from Theorem 9.2. For any positive integer k, define the parameters as follows:

- $q = \lfloor \log\log k \rfloor$ and $\alpha = \frac{1}{q} = \frac{1}{\lfloor \log\log k \rfloor}$,
- $\gamma = \frac{\alpha}{2} = \frac{1}{2\lfloor \log\log k \rfloor}$,
- $\mu = \frac{\varepsilon^2\gamma^2}{288\Delta^3} = \Theta_{\varepsilon,\Delta}\left(\frac{1}{(\log\log k)^2}\right)$,
- $\zeta = \frac{\varepsilon^2\gamma^3}{432\Delta^3} = \Theta_{\varepsilon,\Delta}\left(\frac{1}{(\log\log k)^3}\right)$,
- $\ell = \sqrt{\log k}$,
- $r = \lceil \frac{\ln(2/\zeta)}{\alpha^\ell} \rceil = 2^{\Theta_{\varepsilon,\Delta,c}(\sqrt{\log k}\log\log\log k)}$,
- $h = \lceil 8\ln(2/\mu)/\alpha \rceil = \Theta_{\varepsilon,\Delta,c}(\log\log k)$.

Consider any 3-CNF formula Φ with m clauses such that each variable appears in at most Δ clauses. We then use Lemma 9.16 with the above parameters $q, \alpha, \mu, \eta, k, \ell$ to construct a collection $\mathcal{S}$ of subsets of clauses of Φ such that the following conditions hold.

- Every subset in $\mathcal{S}$ has size at most $2\alpha m$.
- $\mathcal{S}$ is a (r, ℓ, η)-disperser.
- Any subcollection $\widetilde{\mathcal{S}} \subseteq \mathcal{S}$ of size h is $(\alpha/2, \mu)$-uniform.

Note that, for our choices of parameter, the parameter q^{k+1} is a function of k; this means that, for sufficiently large m (which depends on k), we indeed have $m \geqslant q^{k+1}$ and the running time needed to produce $\mathcal{S}$ is $O_k(m)$.

We now consider the 2-CSP instance $\Gamma_{\Phi,\mathcal{S}}$. Observe that the running time used to create $\Gamma_{\Phi,\mathcal{S}}$ (and hence also the size of $\Gamma_{\Phi,\mathcal{S}}$) is no more than $\text{poly}(k) \cdot 2^{O(\alpha m)} = 2^{O(m/\log\log k)}$. Moreover, if $\text{val}(\Phi) = 1$, it is easy to see that $\text{val}(\Gamma_{\Phi,\mathcal{S}}) = 1$ as well.

Suppose that $\text{val}(\Phi) < 1 - \varepsilon$. Due to our choice of parameters, we can apply Theorem 9.25, which implies that

$$\text{val}(\Gamma_{\Phi,\mathcal{S}}) < \frac{O(k^{1/\ell}(r\ell)^2) + h\ell^2/\mu}{k} = 2^{O_{\varepsilon,\Delta}(\log\log k/\sqrt{\log k})}/k.$$

For sufficiently large k (depending on $\varepsilon, \Delta, \rho$), this term is at most $2^{(\log k)^{1/2+\rho}}/k$.

If there exists a $g(k) \cdot (nk)^D$-time algorithm that can distinguish the two cases in the theorem statement for some constant D, then pick sufficiently large k such that the time needed to produce $\Gamma_{\Phi,\mathcal{S}}$ is $O(2^{\delta m})$ and its size is at most $2^{\delta m/D}$, and that $\text{val}(\Gamma_{\Phi,\mathcal{S}}) < 2^{1/(\log k)^{1/2+\rho}}/k$ whenever $\text{val}(\Phi) < 1 - \varepsilon$. When we run this algorithm on $\Gamma_{\Phi,\mathcal{S}}$ for such k, the algorithm can distinguish between $\text{val}(\Phi) = 1$ and $\text{val}(\Phi) < 1 - \varepsilon$ in $O(2^{\delta m})$ time, which contradicts Gap-ETH. □

9.6 Inapproximability of Directed Steiner Network

We now move on to prove hardness of approximation of DSN by simply plugging in the our main theorems to known reductions from 2-CSPs to DSN. The properties of the reduction are stated in the lemma below. Note that, while the reduction is attributed to Dodis and Khanna [DK99], the lemma below is extracted from [CFM18] since, in [DK99], the full description and its properties are left out due to space constraint.

Lemma 9.30 ([CFM18, Lemma 27]). *There exists a polynomial time reduction that, given a 2-CSP instance*[5] Γ *with the constraint graph being a complete graph on k variables, produces an edge-weighted directed graph $G = (V, E)$ and a set of demand pairs $\mathcal{D} = \{(s_1, t_1), \dots, (s_{k'}, t_{k'})\}$ such that*

- *(Completeness) If* $\mathrm{val}(\Gamma) = 1$*, then there exists a subgraph H of cost 1 that satisfies all demands.*
- *(Soundness) If* $\mathrm{val}(\Gamma) < \gamma$*, every subgraph satisfying all demand pairs has cost more than* $\sqrt{2/\gamma}$.
- *(Parameter Dependency)* $k' = k^2 - k$.

Note that the exponent $1/4$ in the hardness of approximating DSN comes from two places: we lose a square factor in the parameter (i.e. $k' = \Theta(k^2)$) and another square factor in the gap.

Proof of Corollary 9.3. Suppose for the sake of contradiction that, for some constant $\rho' > 0$, there exists a polynomial time $2^{(\log k')^{1/2+\rho'}}/(k')^{1/4}$-approximation algorithm where k' is the number of demand pairs; let us call this algorithm $\mathbb{A}$. Moreover, let ρ be any constant smaller than ρ'.

Given a 2-CSP instance Γ with complete constraint graph on k variables, we invoke Lemma 9.30 to produce a DSN instance (G, D) where $|D| = k' = k^2 - k$. From the completeness of the construction, we have that, if $\mathrm{val}(\Gamma) = 1$, then the optimum of (G, D) is also 1. On the other hand, if $\mathrm{val}(\Gamma) < 2^{(\log k)^{1/2+\rho}}/k$, then the optimum of $(G, \mathcal{D})$ must be more than $\sqrt{2k/2^{(\log k)^{1/2+\rho}}}$, which is at least $(k')^{1/4}/2^{(\log k')^{1/2+\rho'}}$ when k is sufficiently large. Hence, by running algorithm $\mathbb{A}$, we can distinguish these two cases of Γ in polynomial time. From Theorem 9.1, this contradicts ETH. □

Proof of Corollary 9.4. Suppose for the sake of contradiction that, for some constant $\rho' > 0$ and for some function g, there exists a $g(k') \cdot (nk')^{O(1)}$-time $2^{(\log k')^{1/2+\rho'}}/(k')^{1/4}$-approximation algorithm where k' is the number of demand pairs; let us call this algorithm $\mathbb{A}$. Moreover, let ρ be any constant smaller than ρ'.

Given a 2-CSP instance Γ with complete constraint graph on k variables, we invoke Lemma 9.30 to produce a DSN instance (G, D) where $|D| = k' = k^2 - k$. From the completeness of the construction, if $\mathrm{val}(\Gamma) = 1$, then the optimum of (G, D) is also 1. On the other hand, if

[5] Lemma 27 of [CFM18] states this reduction in terms of Maximum Colored Subgraph Isomorphism. However, it is easy to see that the reduction also works with 2-CSPs as well.

$\mathrm{val}(\Gamma) < 2^{(\log k)^{1/2+\rho}}/k$, then the optimum of $(G, \mathcal{D})$ must be more than $\sqrt{2k/2^{(\log k)^{1/2+\rho}}}$, which is at least $(k')^{1/4}/2^{(\log k')^{1/2+\rho'}}$ when k is sufficiently large. Hence, by running algorithm $\mathbb{A}$, we can distinguish these two cases of Γ in time $t(k) \cdot |\Gamma|^{O(1)}$ where $t(k) = g(k^2 - k)$. From Theorem 9.2, this contradicts Gap-ETH. □

9.7 Inapproximability of Unique Set Cover

We now proceed to our final result of this section: the hardness of UNIQUE SET COVER (Theorem 9.5). The proof proceeds in three steps; first, we rephrase our 2-CSP result into MAXCOV hardness in Section 9.7.1. Second, we provide a simple way to reduce the left alphabet in Section 9.7.2 so that it is small enough that we can apply Feige's reduction from Section 2.11, which we do so in Section 9.7.3. The key point here is that the label cover instance we construct from our 2-CSP has a projection property but from *right to left*, which is the reverse of the usual projection direction in other sections; this right-to-left projection property indeed provides the uniqueness guarantee in the completeness.

9.7.1 Rephrasing 2-CSP as MAXCOV

The first step in the reduction is to reframe our hardness in terms of MAXCOV with projection property. To do so, recall that our 2-CSP instance $\Gamma_{\Phi,\mathcal{T}}$ (as in Definition 9.6) is an instance of the form: Σ_T is a subset of $\{0,1\}^{\mathrm{var}(T)}$ and the constraint between two vertices T_1, T_2 just checks that $\psi_{T_1}|_{T_1 \cap T_2} = \psi_{T_2}|_{T_1 \cap T_2}$ where ψ_{T_1}, ψ_{T_2} are the labels assigned to T_1, T_2 respectively. This naturally corresponds to a label cover instance $\mathcal{L} = (U, V, \Sigma_U, \Sigma_V, E, \{\Pi_e\}_{e \in E})$ as follows:

- Each vertex in $\Gamma_{\Phi,\mathcal{T}}$ is a right vertex in the $\mathcal{L}$, i.e., $V = \mathcal{T}$.
- For each $T \in V$, the right alphabet for T is Σ_T.
- Each constraint in $\Gamma_{\Phi,\mathcal{T}}$ becomes a left vertex in $\mathcal{L}$, i.e., $U = \binom{\mathcal{T}}{2}$.
- For each $\{T_1, T_2\} \in U$, the left alphabet for $\{T_1, T_2\}$ is $\{0,1\}^{T_1 \cap T_2}$.
- There is an edge from every $\{T_1, T_2\}$ to T_1 and T_2, and the constraint belonging to such edge checks whether the right label projected on $T_1 \cap T_2$ is the same as the left label; that is, $\Pi_{(\{T_1,T_2\},T_i)} = \{(\psi, \psi_{T_i}) \mid \psi_{T_i}|_{T_1 \cap T_2} = \psi\}$ for $i \in [2]$.

Note that this label cover has a projection property, in the sense that, for every edge $e \in E$ and every fix $\beta \in \Sigma_V$, there is at most one $\alpha \in \Sigma_U$ such that (α, β) satisfies Π_e. This is unlike other label cover instances in this thesis where the projections goes from left to right. As a result, we will refer to this new property as *right-to-left projection property*. Indeed, the right-to-left projection property is crucial here, as it will give us the uniqueness in the completeness of SET COVER. Similar to the left-to-right projection situation, it will be convenient to think of each constraint Π_e as a function $\pi_e : \Sigma_V \to \Sigma_U$; we will use this convention for the rest of this section.

Next, observe that any labeling σ_V of V simply corresponds to an assignment in the original 2-CSP instance $\Gamma_{\Phi,\mathcal{T}}$ and the left node $\{T_1, T_2\} \in U$ is covered iff the corresponding constraint is satisfied in $\Gamma_{\Phi,\mathcal{T}}$. As a result, the following corollary is immediate from our main lemma from our main theorem (Theorem 9.2).

Corollary 9.31. *Assuming Gap-ETH, for any constant $\rho > 0$ and any function g, no algorithm can, given a label cover instance $\mathcal{L} = (U, V, \Sigma_U, \Sigma_V, \{\pi_e\}_{e \in E})$ with right-to-left propjection property of size n and with k right supernodes, distinguish between the following two cases in $g(k) \cdot (nk)^{O(1)}$ time:*

- *(Completeness) $\mathcal{L}$ is satisfiable (i.e. $\textsc{MaxCov}(\mathcal{L}) = |U|$), and,*
- *(Soundness) $\textsc{MaxCov}(\mathcal{L}) < (2^{(\log k)^{1/2+\rho}}/k) \cdot |U|$.*

9.7.2 Left Alphabet Reduction

Now, we would ideally like to plug our label cover instance from Corollary 9.31 to the reduction from Section 2.11 and arrive at the desired hardness for Set Cover. At the moment, however, we cannot quite to this yet, since our left alphabet $|\Sigma_U|$ can be as large as n and, since the blow-up in the reduction is exponential in $|\Sigma_U|$, this could result in an exponential time reduction. Nonetheless, this is not a hard issue to overcome, since it is simple to reduce the alphabet size of label cover instances with right-to-left projection property, as stated below.

Lemma 9.32. *For any parameter $\delta > 0$, there is a polynomial time algorithm that, given a label cover instance $\mathcal{L} = (U, V, E, \Sigma_U, \Sigma_V, \{\pi_e\}_{e \in E})$ of size n, produces another label cover instance $\mathcal{L}' = (U', V, E, \Sigma_{U'}, \Sigma_V, \{\pi'_e\}_{e \in E'})$ with the same right supernodes and alphabets such that*

- *(Completeness) If $\mathcal{L}$ is satisfiable (i.e. $\textsc{MaxCov}(\mathcal{L}) = |U|$), then $\mathcal{L}'$ is also satisfiable (i.e. $\textsc{MaxCov}(\mathcal{L}') = |U'|$).*
- *(Soundness) $\frac{\textsc{MaxCov}(\mathcal{L}')}{|U'|} \leqslant \frac{\textsc{MaxCov}(\mathcal{L})}{|U|} + \delta$.*
- *(Left Alphabet Size) $|\Sigma_{U'}| = O(1/\delta)$.*

The proof proceeds by replacing each left alphabet with an error correcting code with distance $1 - \delta$; the point here is that, if a labeling σ_V does not cover $u \in U$, then at least two of u neighbors v_1, v_2 "disagree", i.e., $\pi_{(v,u)}(\sigma_V(v_1)) \neq \pi_{(v,u)}(\sigma_V(v_2))$. Since we are replacing Σ_U with an error correcting code with distance $1 - \delta$, they will still disagree on all but δ fraction of the coordinates. This indeed ensures the soundness of the reduction. (In other words, we "compose" the communication protocol for equality with the original constraint.) Below we use the Hadamard codes only because the relationship between their alphabet sizes and distances are simple. In general, one could use any code such that the relative distance is $1 - \Omega(1/q)$ where q is the alphabet size.

Proof of Lemma 9.32. We may assume w.l.o.g. that $\delta \geqslant 1/n$, as otherwise the alphabet size already satisfies $|\Sigma_U| = O(1/\delta)$ and there is no need to modify the instance $\mathcal{L}$ at all.

Let q be the smallest prime such that $q \geqslant 1/\delta$ and $t = \lceil \log_q |\Sigma_U| \rceil$. Consider the Hadamard code $C : \mathbb{F}_q^t \to \mathbb{F}_q^{q^t}$ with alphabet size q, message length t, block length q^t and relative distance $1 - 1/q$. We may associate each label in Σ_U with an element of $\mathbb{F}_q^{q^t}$. With this in mind, we can define our new label cover instance $\mathcal{L}'$ as follows:

- Let $U' = U \times [q^t]$ and $\Sigma_{U'} = \mathbb{F}_q$.
- We add edges in E' between each $(u, j) \in U \times [q^t]$ to all neighbors $v \in V$ of u.
- We define the constraint $\pi_{(v,(u,j))}$ by
$$\pi_{(v,(u,j))}(\beta) := C(\pi_{(v,u)}(\beta))_j.$$
(In other words, we take the j coordinate of the codeword for $\pi_{(v,u)}(\beta)$.)

It is obvious that the completeness and alphabet size properties are satisfied. We now argue the soundness property. Let us consider any right labeling $\sigma_V : V \to \Sigma_V$. From definition of MaxCov, at most $\frac{\text{MaxCov}(\mathcal{L})}{|U|}$ fraction of vertices in U are covered by σ_V in the original instance $\mathcal{L}$. Let us now consider any vertex u not covered by σ_V in $\mathcal{L}$; this implies that there exists two neighbors v_1, v_2 of u such that $\pi_{(v,u)}(\sigma_V(v_1)) \neq \pi_{(v,u)}(\sigma_V(v_2))$.

Now, for each $j \in [q^t]$, if σ_V covers (u, j) in the new instance $\mathcal{L}'$, it must be that $C(\pi_{(v,u)}(\sigma_V(v_1)))_j$ and $C(\pi_{(v,u)}(\sigma_V(v_2)))_j$ are equal. However, since C has relative distance $1 - 1/q$, this equality can only hold for $1/q \leqslant \delta$ fraction of indices j. In other words, at most δ fraction of vertices of the form $(u, \star)$ can be covered by σ_V, for all u that is not covered in the original instance $\mathcal{L}$. As a result, we indeed have that the fraction of vertices in U' that can be covered by σ_V in $\mathcal{L}'$ is at most $\frac{\text{MaxCov}(\mathcal{L})}{|U|} + \delta$ as desired. This indeed implies that $\frac{\text{MaxCov}(\mathcal{L}')}{|U'|} \leqslant \frac{\text{MaxCov}(\mathcal{L})}{|U|} + \delta$. □

By applying the above transformation to Corollary 9.31 with $\delta = 1/k$, we have at the following:

Corollary 9.33. *Assuming Gap-ETH, for any constant $\rho > 0$ and any function g, no algorithm can, given a label cover instance $\mathcal{L} = (U, V, \Sigma_U, \Sigma_V, \{\pi_e\}_{e \in E})$ with right-to-left projection property of size n and with k right supernodes, distinguish between the following two cases in $g(k) \cdot (nk)^{O(1)}$ time:*

- *(Completeness) $\mathcal{L}$ is satisfiable (i.e.* MaxCov$(\mathcal{L}) = |U|$*), and,*
- *(Soundness)* MaxCov$(\mathcal{L}) < (2^{(\log k)^{1/2+\rho}}/k) \cdot |U|$.

Moreover, this holds even when the left alphabet size $\mathcal{L}$ is $O(k)$.

Finally, observe that in our reduction the instance $\mathcal{L}$ is bi-regular with left degree two, as a result, by applying Lemma 2.23, we get the following hardness for MinLab.

Corollary 9.34. *Assuming Gap-ETH, for any constant $\rho > 0$ and any function g, no algorithm can, given a label cover instance $\mathcal{L}$ with right-to-left propjection property of size n and with k right supernodes, distinguish between the following two cases in $g(k) \cdot (nk)^{O(1)}$ time:*

- *(Completeness) $\mathcal{L}$ is satisfiable (i.e. $\textsc{MinLab}(\mathcal{L}) = k$), and,*
- *(Soundness) $\textsc{MinLab}(\mathcal{L}) > (\sqrt{k}/2^{(\log k)^{1/2+\rho}}) \cdot k$.*

Moreover, this holds even when the left alphabet size $\mathcal{L}$ is $O(k)$.

9.7.3 Putting Things Together

We can prove our hardness of approximation for Unique Set Cover by simply plugging in the label cover from Corollary 9.34 to the reduction from Section 2.11.

Proof of Theorem 9.5. Let $\mathcal{L} = (U, V, \Sigma_U, \Sigma_V, \{\pi_e\}_{e \in E})$ be any label cover instance of size n with right-to-left projection property such that $|V| = k$ and the left alphabet Σ_U is of size at most $O(k)$. We use the reduction from Section 2.11 to produce a set system $\mathcal{U}, \mathcal{S}$. Notice that the size of $|\mathcal{U}|$ is at most $O(n \cdot k^{|\Sigma_U|}) = O(n \cdot k^k)$. Hence, the reduction is an FPT reduction as desired.

(Completeness) Suppose that there exists a labeling (σ_U, σ_V). Recall that, as argued in the proof of Lemma 2.25, we can select the subsets $S_{v,\sigma_V(v)}$ for all $v \in V$ to cover $\mathcal{U}$. Observe further that, when $\mathcal{L}$ has the right-to-left projection property, these k subsets in fact covers each element exactly once: x of $\mathcal{U}^u$ where $\mathrm{x} \in N_G(u)^{\Sigma_U}$ is covered by the set $S_{x_{\sigma_U(u)},\sigma_V(x_{\sigma_U(u)})}$ only.

(Soundness) If $\textsc{MinLab}(\mathcal{L}) > (\sqrt{k}/2^{(\log k)^{1/2+\rho}}) \cdot k$, then, from Lemma 2.25, we have that $\textsc{SetCov}(\mathcal{U}, \mathcal{S}) \geqslant \textsc{MinLab}(\mathcal{L}) > (\sqrt{k}/2^{(\log k)^{1/2+\rho}}) \cdot k = k^{3/2-o(1)}$.

As a result, if there exists an FPT algorithm that can distinguish the two cases in Theorem 9.5, it can distinguish the two cases in Corollary 9.34 in FPT time, which would violate Gap-ETH. □

9.8 Discussion and Open Questions

In this chapter, we show that 2-CSP is ETH-hard to approximate to within a factor of $k^{1-o(1)}$ where k denotes the number of variables. This ratio is nearly optimal since a trivial algorithm yields an $O(k)$-approximation for the problem. Under Gap-ETH, we strengthen our result by improving the lower order term in the inapproximability factor and ruling out not only polynomial time algorithm but FPT algorithms parameterized by k.

Of course the polynomial sliding scale conjecture still remains open after our work and, as touched upon in the introduction, resolving the conjecture will help advance our understanding of approximability of many problems. Even without fully resolving the conjecture, it may still be good to further study the interaction between the number of variables k and the alphabet size n. For instance, while we show the inapproximability result with ratio almost linear in k, the dependency between n and k is quite bad; in particular, in our ETH-hardness reduction, n is $2^{2^{(\log k)^d}}$ for some constant $d > 0$. Would it be possible to improve this dependency (say, to $n = k^{\text{polylog} k}$)?

Note that, in the parameterized setting, k must be independent of n and hence the question above does not apply to this regime.

As touched upon briefly earlier, for k-UNIQUE SET COVER, there is no known $g(k)$-FPT-approximation algorithm for any g. However, our hardness only rules out a factor of $k^{1/2-o(1)}$. Hence, it remains open whether the problem is totally FPT inapproximable:

Open Question 6. *Is k-UNIQUE SET COVER totally FPT inapproximable?*

On this front, let us also note a natural barrier if we are to use the approach of reducing from MINLAB with right-to-left projection property as in this chapter (and previous works). We claim that this will not give a hardness of approximation with factor more than 2^k. The reason is that we may merge two left vertices in a label cover instance with the same set of neighbors. The right-to-left projection property implies that such merging will never blow up the left alphabet size to more than the original right alphabet size $|\Sigma_V|$. Summarizing, we can always assume that there are at most 2^k left super-nodes. For each left super-node, we can always pick at most k labels on the right to cover this node. As a result, we may select at most $k \cdot 2^k$ labels on the right, which is the limit even in the soundness case. Hence, the gap we can hope to get is at most 2^k. In other words, to answer Question 6 in the positive, one has to deviate from this general approach.

Another interesting research direction is to try to prove similar hardness results for other problems. As mentioned in the previous chapter, DkS is one obvious candidate in this direction. However, as discussed in Chapter 4, it is typically quite challenging to transfer hardness from CSPs to DkS; for instance, in the NP-hardness regime, CSP is quite well understood, whereas not even a constant factor NP-hardness for DkS is known.

Chapter 10

Inapproximability from Gap-ETH IV: Even Set and Shortest Vector Problems

The study of error-correcting codes gives rise to many interesting computational problems. One of the most fundamental among these is the problem of computing the distance of a linear code. In this problem, which is commonly referred to as the *Minimum Distance Problem (*MDP*)*, we are given as input a generator matrix $\mathbf{A} \in \mathbb{F}_2^{n \times m}$ of a binary[1] linear code and an integer k. The goal is to determine whether the code has distance at most k. Recall that the distance of a linear code is $\min_{\mathbf{0} \neq \mathbf{x} \in \mathbb{F}_2^m} \|\mathbf{A}\mathbf{x}\|_0$ where $\| \cdot \|_0$ denote the the Hamming norm.

The study of this problem dates back to at least 1978 when Berlekamp et al. [BMT78] conjectured that it is NP-hard. This conjecture remained open for almost two decades until it was positively resolved by Vardy [Var97a; Var97b]. Later, Dumer et al. [DMS03] strengthened this by showed that, even *approximately* computing the minimum distance of the code is hard. Specifically, they showed that, unless $\mathsf{NP} = \mathsf{RP}$, no polynomial time algorithm can distinguish between a code with distance at most k and one whose distance is greater than $\gamma \cdot k$ for any constant $\gamma \geqslant 1$. Furthermore, under stronger assumptions, the ratio can be improved to superconstants and even almost polynomial. Dumer et al.'s result has been subsequently derandomized by Cheng and Wan [CW12a] and further simplified by Austrin and Khot [AK14] and Micciancio [Mic14].

While the aforementioned intractability results rule out not only efficient algorithms but also efficient approximation algorithms for MDP, it does not yet rule out FPT algorithms with the natural parameter k. Note that k-MDP can be solved in $(mn)^{O(k)}$ time, as we can enumerate through all vectors $\mathbf{y}$ with Hamming norm at most k and try to solve $\mathbf{A}\mathbf{x} = \mathbf{y}$. In Parameterized Complexity language, this means that k-MDP belongs to the class XP.

The parameterized complexity of k-MDP was first questioned by Downey et al. [Dow+99], who showed that parameterized variants of several other coding-theoretic problems, including the Nearest Codeword Problem and the Nearest Vector Problem[2] which we will discuss in more details in Section 10, are W$[1]$-hard. Thereby, assuming the widely believed W$[1] \neq$ FPT hypoth-

[1] Note that MDP can be defined over larger fields as well; we discuss more about this in Section 10.5.

[2] The Nearest Vector Problem is also referred to in the literature as the Closest Vector Problem.

esis, these problems are rendered intractable from the parameterized perspective. Unfortunately, Downey et al. fell short of proving such hardness for k-MDP and left it as an open problem:

Research Question 4. *Is k-MDP fixed parameter tractable?*

Although almost two decades have passed, the above question remains unresolved to this day, despite receiving significant attention from the community. In particular, the problem was listed as an open question in the seminal 1999 book of Downey and Fellows [DF99] and has been reiterated numerous times over the years [Dem+07; Fel+12; GKS12; FM12; DF13; Cyg+14; Cyg+15; Bha+16c; Cyg+17; Maj17]. This problem is one of the few questions that remained open from the original list of Downey and Fellows [DF99]. In fact, in their second book [DF13], Downey and Fellows even include this problem as one of the six[3] "most infamous" open questions in the area of Parameterized Complexity.

Another question posted in Downey et al.'s work [Dow+99] that remains open is the parameterized *Shortest Vector Problem (k-SVP)* in lattices. The input of k-SVP (in the ℓ_p norm) is an integer $k \in \mathbb{N}$ and a matrix $\mathbf{A} \in \mathbb{Z}^{n \times m}$ representing the basis of a lattice, and we want to determine whether the shortest (non-zero) vector in the lattice has length at most k, i.e., $\min_{\mathbf{0} \neq \mathbf{x} \in \mathbb{Z}^m} \|\mathbf{A}\mathbf{x}\|_p \leqslant k$. Again, k is the parameter of the problem. It should be noted here that, similar to [Dow+99], we require the basis of the lattice to be integer-value, which is sometimes not enforced in literature (e.g. [Emd81; Ajt98]). This is because, if $\mathbf{A}$ is allowed to be any matrix in $\mathbb{R}^{n \times m}$, then parameterization is meaningless because we can simply scale $\mathbf{A}$ down by a large multiplicative factor.

The (non-parameterized) Shortest Vector Problem (SVP) has been intensively studied, motivated partly due to the fact that both algorithms and hardness results for the problem have numerous applications. Specifically, the celebrated LLL algorithm for SVP [LLL82] can be used to factor rational polynomials, and to solve integer programming (parameterized by the number of unknowns) [Len83] and many other computational number-theoretic problems (see e.g. [NV10]). Furthermore, the hardness of (approximating) SVP has been used as the basis of several cryptographic constructions [Ajt98; AD97; Reg03; Reg05]. Since these topics are out of scope of our thesis, we refer the interested readers to the following surveys: [Reg06; MR09; NV10; Reg10].

On the computational hardness side of the problem, van Emde-Boas [Emd81] was the first to show that SVP is NP-hard for the ℓ_∞ norm, but left open the question of whether SVP on the ℓ_p norm for $1 \leqslant p < \infty$ is NP-hard. It was not until a decade and a half later that Ajtai [Ajt96] showed, under a randomized reduction, that SVP for the ℓ_2 norm is also NP-hard; in fact, Ajtai's hardness result holds not only for exact algorithms but also for $(1+o(1))$-approximation algorithms as well. The $o(1)$ term in the inapproximability ratio was then improved in a subsequent work of Cai and Nerurkar [CN99]. Finally, Micciancio [Mic00] managed to achieve a factor that is bounded away from one. Specifically, Micciancio [Mic00] showed (again under randomized reductions) that SVP on the ℓ_p norm is NP-hard to approximate to within a factor of $\sqrt[p]{2}$ for every $1 \leqslant p < \infty$. Khot [Kho05] later improved the ratio to any constant, and even to $2^{\log^{1/2-\varepsilon}(nm)}$ under a stronger assumption. Haviv and Regev [HR07] subsequently simplified the gap amplification step

[3] So far, two of the six problems have been resolved: that of parameterized complexity of k-Biclique [Lin15] and that of parameterized approximability of k-Dominating Set (Section 6).

of Khot and, in the process, improved the ratio to almost polynomial. We note that both Khot's and Haviv-Regev reductions are also randomized and it is still open to find a deterministic NP-hardness reduction for SVP in the ℓ_p norms for $1 \leqslant p < \infty$ (see [Mic12]); we emphasize here that such a reduction is not known even for the exact (not approximate) version of the problem. For the ℓ_∞ norm, the following stronger result due to Dinur is known [Din02]: SVP in the ℓ_∞ norm is NP-hard to approximate to within $n^{\Omega(1/\log\log n)}$ factor (under a *deterministic* reduction).

Very recently, fine-grained studies of SVP have been initiated [BGS17; AS18]. The authors of [BGS17; AS18] showed that SVP for any ℓ_p norm cannot be solved (or even approximated to some constant strictly greater than one) in subexponential time assuming the existence of a certain family of lattices[4] and the (randomized) Gap-ETH.

As with MDP, Downey et al. [Dow+99] were the first to question the parameterized tractability of k-SVP (for the ℓ_2 norm). Once again, Downey and Fellows included k-SVP as one of the open problems in both of their books [DF99; DF13]. As with Open Question 4, this question remains unresolved to this day:

Research Question 5. *Is k-SVP fixed parameter tractable?*

We remark here that, similar to k-MDP, k-SVP also belongs to XP, as we can enumerate over all vectors with norm at most k and check whether it belongs to the given lattice. There are only $(mn)^{O(k^p)}$ such vectors, and the lattice membership of a given vector can be decided in polynomial time. Hence, this is an $(nm)^{O(k^p)}$-time algorithm for k-SVP.

Our Results

The main result of this chapter is a resolution to the previously mentioned Open Questions 4 and 5: more specifically, we prove that k-MDP and k-SVP (on ℓ_p norm for any $p > 1$) are intractable assuming randomized Gap-ETH. In fact, our result is stronger than stated here as we rule out not only exact FPT algorithms but also FPT *approximation* algorithms as well.

With this in mind, we can state our results starting with the parameterized intractability of k-MDP, more concretely (but still informally), as follows:

Theorem 10.1. *Assuming randomized Gap-ETH, there is no γ-FPT-approximation algorithm for k-MDP for any constant $\gamma \geqslant 1$.*

Notice that our above result rules out FPT approximation algorithms with *any* constant approximation ratio for k-MDP. In contrast, we can only prove FPT inapproximability with *some* constant ratio for k-SVP in ℓ_p norm for $p > 1$. These are stated more precisely below.

Theorem 10.2. *For any $p > 1$, there exists $\gamma_p > 1$ such that the following holds. Assuming randomized Gap-ETH, there is no γ_p-FPT-approximation algorithm for k-SVP in ℓ_p norm.*

[4] This additional assumption is only needed for $1 \leqslant p \leqslant 2$. For $p > 2$, their hardness is conditional only on the deterministic Gap-ETH.

We remark that our result does not yield hardness for SVP in the ℓ_1 norm and this remains an interesting open question. Section 10.5 contains discussion on this problem. We also note that, for Theorem 10.2 and onwards, we are only concerned with $p \neq \infty$; this is because, for $p = \infty$, the problem is NP-hard to approximate even when $k = 1$ [Emd81]!

Nearest Codeword Problem and Nearest Vector Problem

Similar to the NP-hardness of approximation proofs of MDP and SVP, our proofs proceed by first showing FPT hardness of approximation of the non-homogeneous variants of k-MDP and k-SVP called the *k-Nearest Codeword Problem* (k-NCP) and the *k-Nearest Vector Problem* (k-NVP) respectively. For both k-NCP and k-NVP, we are given a target vector $\mathbf{y}$ (in $\mathbb{F}_2^n$ and $\mathbb{Z}^n$, respectively) in addition to $(\mathbf{A}, k)$, and the goal is to find whether there is any $\mathbf{x}$ (in $\mathbb{F}_2^m$ and $\mathbb{Z}^m$, respectively) such that the (Hamming and ℓ_p, respectively) norm of $\mathbf{Ax} - \mathbf{y}$ is at most k.

As an intermediate step of our proof, we show that the k-NCP and k-NVP problems are hard to approximate[5] (see Theorem 10.4 and Theorem 10.5 respectively). This should be compared to Downey et al. [Dow+99], in which the authors show that both problems are W$[1]$-hard to solve exactly. Therefore our inapproximability result significantly improves on their work to rule out even $k^{1/2-o(1)}$ factor FPT-approximation algorithm, albeit we need the stronger Gap-ETH assumption (in comparison to W$[1] \neq$ FPT from [Dow+99]).

We end this section by remarking that the computational complexity of both (non-parameterized) NCP and NVP are also thoroughly studied (see e.g. [Mic01; Din+03; Ste93; Aro+97; Gol+99] in addition to the references for MDP and SVP), and indeed the inapproximability results of these two problems form the basis of hardness of approximation for MDP and SVP. We would like to emphasize that while W[1]-hardness results were known for k-NCP and k-NVP, it does not seem possible to transfer them to W[1]-hardness results for k-MDP and k-SVP; we really need parameterized *inapproximability* results for k-NCP and k-NVP to be able to transfer them to (slightly weaker) inapproximability results for k-MDP and k-SVP.

Subsequent Work. After the publication of the conference version of the work on which this chapter is based [Bha+18], Bonnet et al. [Bon+18] showed W[1]-hardness of approximation results for k-NCP and k-NVP. When combined with our reductions (Lemmas 10.6 and 10.10) to k-MDP and k-SVP respectively, one arrives at W[1]-hardness of approximation (via randomized reductions) for both problems, thereby resolving the complexity of both problems up to whether the reductions can be derandomized.

[5] While our k-MDP result only applies for $\mathbb{F}_2$, it is not hard to see that our intermediate reduction for k-NCP actually applies for any finite field $\mathbb{F}_q$ too.

10.1 Additional Preliminaries

Before we prove our results, we need a few results from error-correcting codes. First, we say that an error-correcting code $C : \Sigma^m \to \Sigma^h$ is *systematic*[6] if $C(\mathbf{x})|_{[h]} = C(\mathbf{x})$ for all $\mathbf{x} \in \Sigma^m$.

10.1.1 BCH Codes

Throughout this chapter, the BCH codes are crucial in constructing the gadgets. Their parameters are stated more precisely below.

Theorem 10.3 (BCH Code [Hoc59; BR60]). *For any choice of $h, d \in \mathbb{N}$ such that $h + 1$ is a power of two and that $d \leqslant h$, there exists a linear code over $\mathbb{F}_2$ with block length h, message length $h - \left\lceil \frac{d-1}{2} \right\rceil \cdot \log(h+1)$ and distance d. Moreover, the generator matrix of this code can be computed in poly(h) time.*

10.1.2 Tensor Product of Codes

Finally, we define the tensor product of codes which will be used to amplify the gap in hardness of approximation of k-MDP. Consider two linear codes $\mathcal{C}_1 \subseteq \mathbb{F}_2^m$ (generated by $\mathbf{G}_1 \in \mathbb{F}_2^{m \times m'}$) and $\mathcal{C}_2 \subseteq \mathbb{F}_2^n$ (generated by $\mathbf{G}_2 \in \mathbb{F}_2^{n \times n'}$). Then the tensor product of the two codes $\mathcal{C}_1 \otimes \mathcal{C}_2 \subseteq \mathbb{F}_2^{m \times n}$ is defined as

$$\mathcal{C}_1 \otimes \mathcal{C}_2 = \{\mathbf{G}_1 \mathbf{X} \mathbf{G}_2^\top | \mathbf{X} \in \mathbb{F}_2^{m' \times n'}\}.$$

We will only need two properties of tensor product codes. First, the generator matrix of the tensor products of two linear codes $\mathcal{C}_1, \mathcal{C}_2$ with generator matrices $\mathbf{G}_1, \mathbf{G}_2$ can be computed in polynomial time in the size of $\mathbf{G}_1, \mathbf{G}_2$. Second, the distance of $\mathcal{C}_1 \otimes \mathcal{C}_2$ is exactly the product of the distances of the two codes, i.e.,

$$\Delta(\mathcal{C}_1 \otimes \mathcal{C}_2) = \Delta(\mathcal{C}_1)\Delta(\mathcal{C}_2).$$

10.2 Inapproximability of MLD and NVP

We now proceed to prove our results, starting with k-NCP and k-NVP. For both, we show, assuming Gap-ETH, that no $k^{1/2-o(1)}$-factor FPT approximation exists for both problems. We do this by a (simple) reduction from UNIQUE SET COVER from the previous section. This reduction is due to Arora et al. [Aro+97] who use the reduction to prove NP-hardness of approximation for the (non-parameterized) NCP. The hardness of approximation for k-NCP is stated and proved below; notice that in the YES (completeness) case, we actually have a stronger property than usual that the "solution" vector $\mathbf{x}$ is also sparse. While this is not needed for inapproximability of k-NCP, it will be used for the subsequent proof of inapproximability of k-MDP.

[6] Note that this definition is different than s-systematic used in Chapter 6.

Theorem 10.4. *Assuming Gap-ETH, no FPT time algorithm can, given a matrix $\mathbf{A} \in \mathbb{F}_2^{n \times m}$, a vector $\mathbf{y} \in \mathbb{F}_2^n$ and a positive integer $k \in \mathbb{N}$, distinguish between the following two cases:*

- *(Completeness) There exists $\mathbf{x} \in \mathcal{B}_m(\mathbf{0}, k)$ such that $\|\mathbf{Ax} - \mathbf{y}\|_0 = k$.*
- *(Soundness) For any $\mathbf{x} \in \mathbb{F}_2^m$, $\|\mathbf{Ax} - \mathbf{y}\|_0 > k^{3/2-o(1)}$.*

Proof. We reduce from UNIQUE SET COVER; let $(\mathcal{U}, \mathcal{S}, k)$ be any instance of UNIQUE SET COVER. Label elements in the universe by $u_1, \dots, u_m$, and the subsets by $S_1, \dots, S_M$. First, we construct a matrix $\mathbf{B} \in \mathbb{F}_2^{N \times M}$ where B_{ij} is the indicator whether u_i belongs to S_j.

We now define $\mathbf{A} \in \mathbb{F}_2^{n \times m}$ where $n = \lceil k^{3/2} \rceil N + M$ and $m = M$ by

$$\mathbf{A} = \begin{bmatrix} k \otimes \mathbf{B} \\ \mathrm{Id}_M \end{bmatrix},$$

and let $\mathbf{y} = \mathbf{1}_{\lceil k^{3/2} \rceil N} \otimes \mathbf{0}_m$ be the vector whose last m coordinates are zeros and the rest are ones. This completes the description of our reduction. It is clear that this is an FPT reduction.

(Completeness) Suppose that there exists a subset $S_{i_1}, \dots, S_{i_k}$ that uniquely covers $\mathcal{U}$. Let $\mathbf{x} \in \mathbb{F}_2^m$ be such that $x_{i_1}, \dots, x_{i_k}$ are ones, and the remaining coordinates are zeros. Clearly, $\mathbf{x} \in \mathcal{B}_m(\mathbf{0}, k)$. It is also simple to see that $\mathbf{Bx} = \mathbf{1}_N$, which means that $\|\mathbf{Ax} - \mathbf{y}\|_0 = \|\mathbf{x}\|_0 = k$.

(Soundness) We claim that, for any $\mathbf{x} \in \mathbb{F}_2^m$, we have $\|\mathbf{Ax} - \mathbf{y}\|_0 \geqslant \min\{k^{3/2}, \textsc{SetCov}(\mathcal{U}, \mathcal{S})\}$. To see that this is the case, consider any $\mathbf{x} \in \mathbb{F}_2^m$. If $\mathbf{Bx} \neq \mathbf{1}_m$, then we immediately have $\|\mathbf{Ax} - \mathbf{y}\|_0 \geqslant k^{3/2} \|\mathbf{Bx} - \mathbf{1}_m\|_0 \geqslant k^{3/2}$ as desired. On the other hand, if $\mathbf{Bx} = \mathbf{1}_m$, then let $i_1, \dots, i_\ell$ be the coordinates of $\mathbf{x}$ that are ones. Observe that $S_{i_1}, \dots, S_{i_\ell}$ must cover the universe $\mathcal{U}$, as otherwise the coordinate corresponding to the uncovered element of $\mathbf{Bx}$ would be zero. As a result, we have $\ell \geqslant \textsc{SetCov}(\mathcal{U}, \mathcal{S})$; in other words, we have $\|\mathbf{x}\|_0 \geqslant \textsc{SetCov}(\mathcal{U}, \mathcal{S})$. Hence, in this case, we also have $\|\mathbf{Ax} - \mathbf{y}\|_0 \geqslant \|\mathbf{x}\|_0 \geqslant \textsc{SetCov}(\mathcal{U}, \mathcal{S})$.

Hence, if there is an FPT time algorithm that can distinguish the two cases in Theorem 10.4, then it can also distinguish the two cases in Theorem 9.5, which would break Gap-ETH. □

Note here that if we repeat the proof above but with operations in $\mathbb{Z}$ instead of $\mathbb{F}_2$ and with $\| \cdot \|_p^p$ in place of $\| \cdot \|_0$, then we arrive at the hardness for NVP, as formalized below. Due to this similarity, we shall not duplicate the whole proof again.

Theorem 10.5. *Let $p \geqslant 1$ be any constant. Assuming Gap-ETH, no FPT time algorithm can, given a matrix $\mathbf{A} \in \mathbb{Z}^{n \times m}$, a vector $\mathbf{y} \in \mathbb{Z}^n$ and a positive integer $k \in \mathbb{N}$, distinguish between the following two cases:*

- *(Completeness) There exists $\mathbf{x} \in \mathbb{Z}$ such that $\|\mathbf{Ax} - \mathbf{y}\|_p^p = k$.*
- *(Soundness) For any $\mathbf{x} \in \mathbb{Z}^m$, $\|\mathbf{Ax} - \mathbf{y}\|_p^p > k^{3/2-o(1)}$.*

10.3 Inapproximability of k-MDP

In this section, we prove our main theorem regarding k-MDP (Theorem 10.1). As stated in the introduction, this is shown via a reduction from inapproximability of NCP; the main properties of the reduction is stated below. (Here we omit the dimensions to avoid unnecessary cumbersomeness.)

Lemma 10.6. *For any constants $\gamma' > 4$ and $\gamma \geqslant 1$ such that $\gamma < \frac{2\gamma'}{4+\gamma'}$, there is a polynomial-time reduction that takes in an* NCP *instance* $(\mathbf{B}, \mathbf{y}, t)$ *and produces an* MDP *instance* $(\mathbf{A}, k)$ *such that*

- *(Completeness) If there exists $\mathbf{x} \in \mathcal{B}(\mathbf{0}, t)$ such that $\|\mathbf{Bx} - \mathbf{y}\|_0 \leqslant t$, then, with probability $t^{-O(t)}$, there exists $\mathbf{z} \neq \mathbf{0}$ such that $\|\mathbf{Az}\|_0 \leqslant k$.*
- *(Soundness) If $\|\mathbf{Bx} - \mathbf{y}\| > \gamma' \cdot t$ for all $\mathbf{x}$, then $\|\mathbf{Az}\|_0 > \gamma \cdot k$ for all $\mathbf{z}$.*
- *(Bounded Parameter) $k = O(t)$.*

Before we prove Lemma 10.6, let us first argue why it implies our main theorem. Notice that, if we have an algorithm that can solve the gap problem for MDP with gap γ, then the above reduction implies that we can solve the gap problem for NCP with gap γ' (with high probability) as well, since we can run the reduction $t^{O(t)}$ times and run the algorithm for MDP for each of the produced instance. If the algorithm says YES in any of the instance, we output YES. Otherwise, output NO. Recall that, from Theorem 10.4, the gap version of NCP is hard for any $\gamma' > 1$ (and in fact even for $\gamma' = t^{1/2-o(1)}$). As a result, we have the following:

Lemma 10.7. *Let $1 \leqslant \gamma < 2$ be a constant. Assuming randomized Gap-ETH, there is no γ-FPT-approximation algorithm for k-*MDP.

Gap Amplification. Finally, the gap γ can be boosted to any constant using the now standard technique of tensoring the code (c.f. [DMS03],[AK14]). Recall from Section 10.1.2 that if we take an error-correcting code $\mathcal{C}$ and tensor with itself, then we arrive at the code $\mathcal{C} \otimes \mathcal{C}$ with distance equal to $\Delta(\mathcal{C})^2$. Hence, if there is an γ^2-FPT-approximation algorithm for k-MDP, then we can run this on $\mathcal{C} \otimes \mathcal{C}$ and get a γ-approximation for $\Delta(\mathcal{C})$. In other words, by repeatedly tensoring the instance in Lemma 10.7, we can amplify the gap to be any constant, which implies Theorem 10.1.

Reduction Overview. Now that we have argued why the reduction in Lemma 10.6 implies our main theorem, we turn our attention back to the proof of Lemma 10.6. Our reduction is inspired by the reduction of Dumer, Micciancio and Sudan (henceforth DMS) [DMS03]. There, the authors define the notions of *Locally Dense Codes (*LDC*)* and use it in their reduction. To make the reduction works in the parameterized regime, we define a new notion called *Locally Suffix Dense Codes (*LSDC*)* and show their existence in the next subsection (using BCH codes). Finally, we show how to use them in the reduction in Subsection 10.3.2.

10.3.1 Locally Suffix Dense Codes

Before we formalize the notion of *Locally Suffix Dense Codes* (LSDC), let is give an intuitive explanation of LSDC: informally, LSDC is a linear code $\mathcal{C} \subseteq \mathbb{F}_2^h$ where, given any short prefix $\mathbf{x} \in \mathbb{F}_2^q$ where $q \ll h$ and a random suffix $\mathbf{s} \in \mathbb{F}_2^{h-q}$, we can, with non-negligible probability, find a code that shares the prefix $\mathbf{x}$ and has a suffix that is "close" in Hamming distance to $\mathbf{s}$ (i.e. one should think of r below as roughly $d/2$). More formally, LSDC can be defined as follows.

Definition 10.8. *A Locally Suffix Dense Code (*LSDC*) over $\mathbb{F}_2$ with parameters*[7] (m, q, d, r, δ) *an m-dimensional systematic linear code with minimum distance (at least) d given by its generator matrix $\mathbf{L} \in \mathbb{F}_2^{h \times m}$ such that for any $\mathbf{x} \in \mathbb{F}_2^q$, the following holds:*

$$\Pr_{\mathbf{s} \sim \mathbb{F}_2^{h-q}} \left[\exists \mathbf{z} \in \mathcal{B}_{h-q}(\mathbf{s}, r) : (\mathbf{x} \circ \mathbf{z}) \in \mathbf{L}(\mathbb{F}_2^m) \right] \geqslant \delta. \tag{10.1}$$

We note that our notion of Locally Suffix Dense Codes is closely related and inspired by the notion of Locally Dense Codes (LDC) of Dumer et al. [DMS03]. Essentially speaking, the key differences in the two definitions are that (i) Locally Dense Codes are for the case of $q = 0$, i.e., there is no prefix involved, and (ii) s in LDC is not chosen at random from $\mathbb{F}_2^q$ but rather from $\mathcal{B}_q(\mathbf{0}, r)$. Note that, apart from these, there are other subtle additional requirements in Locally Dense Codes that we do not need in our reduction, such as the requirements that the "center" s is close to not just one but many codewords; however, these are not important and we will not discuss them further.

Unfortunately, the proof of Dumer et al. does not directly give us the desired LSDC; the main issue is that, when there is no prefix, the set of codewords is a linear subspace, and their proof relies heavily on the linear structure of the set (which is also why s is randomly chosen from $\mathcal{B}_q(\mathbf{0}, r)$ instead of $\mathbb{F}_2^q$). However, the set of our interest is $\left\{ \mathbf{z} \in \mathbb{F}^{h-q} \middle| \mathbf{x} \circ \mathbf{z} \in \mathbf{L}(\mathbb{F}_2^m) \right\}$, which is not a linear subspace but rather an affine subspace; Dumer et al.'s argument (specifically Lemma 13 in [DMS03]) does not apply in the affine subspace case.

Below, we provide a different proof than Dumer et al. for the construction of LSDC. Our bound is more related to the *Sphere Packing* (aka Hamming) bound for codes. In particular, we show below that BCH codes, which "near" the Sphere Packing bound gives us LSDC with certain parameters. It should be noted however that the probability guarantee δ that we have is quite poor, i.e. $\delta \geqslant d^{-\Theta(d)}$, but this works for us since d is bounded by a function of the parameter of our problem. On the other hand, this would not work in NP-hardness reductions of [DMS03] (and, on top of this, our codes may not satisfy other additional properties required in LDC).

Lemma 10.9. *For any $q, d \in \mathbb{N}$ such that d is an odd number larger than one, there exist $h, m \in \mathbb{N}$ and $\mathbf{L} \in \mathbb{F}_2^{h \times m}$ which is a* LSDC *with parameters $\left(m, q, d, \frac{d-1}{2}, \frac{1}{d^{d/2}}\right)$. Additionally, the following holds:*

- $h, m \leqslant poly(q, d)$ *and* $m \geqslant q$,

[7] We remark that the parameter h is implicit in specifying LSDC.

- $\mathbf{L}$ *can be computed in poly*(q, d) *time.*

Proof. Let h be the smallest integer such that $h+1$ is a power of two and that $h \geqslant \max\{2q, 10d \log d\}$, and let $m = h - \left(\frac{d-1}{2}\right) \log(h+1)$. Clearly, h and m satisfy the first condition.

Let $\mathbf{L}$ be the generator matrix of the $[h, m, d]_2$ linear code as given by Theorem 10.3. Without loss of generality, we assume that the code is systematic on the first m coordinates. From Theorem 10.3, $\mathbf{L}$ can be computed in poly$(h) =$ poly(q, d) time.

It remains to show that for our choice of $\mathbf{L}$, (10.1) holds for any fixed choice of $\mathbf{x} \in \mathbb{F}_2^q$. Fix a vector $\mathbf{x} \in \mathbb{F}_2^q$ and define the set $\mathcal{C} = \left\{\mathbf{z} \in \mathbb{F}_2^{h-q} \middle| \mathbf{x} \circ \mathbf{z} \in \mathbf{L}(\mathbb{F}_2^m)\right\}$. Since the code generated by $\mathbf{L}$ is systematic on the first $m \geqslant q$ coordinates, we have that $|\mathcal{C}| \geqslant 2^{m-q}$.

Moreover, since the code generated by $\mathbf{L}$ has distance d, every distinct $\mathbf{z}_1, \mathbf{z}_2 \in \mathcal{C}$ are at least d-far from each other (i.e. $\|\mathbf{z}_1 - \mathbf{z}_2\|_0 \geqslant d$). Therefore, for any distinct pair of vectors $\mathbf{z}_1, \mathbf{z}_2 \in \mathcal{C}$, the sets $\mathcal{B}_{h-q}(\mathbf{z}_1, \frac{d-1}{2})$ and $\mathcal{B}_{h-q}(\mathbf{z}_2, \frac{d-1}{2})$ are disjoint. Hence the number of vectors in the union of $\left(\frac{d-1}{2}\right)$-radius Hamming balls around every $\mathbf{z} \in \mathcal{C}$ is at least

$$2^{m-q}\left|\mathcal{B}_{h-q}\left(\mathbf{0}, \frac{d-1}{2}\right)\right| \geqslant 2^{m-q}\binom{h-q}{\frac{d-1}{2}} \geqslant 2^{m-q}\binom{h/2}{\frac{d-1}{2}} \geqslant 2^{m-q}\left(\frac{h}{d-1}\right)^{\frac{d-1}{2}}$$

On the other hand, $|\mathbb{F}_2^{h-q}| = 2^{h-q} = 2^{m-q}(h+1)^{\frac{d-1}{2}}$. Hence, with probability at least $\left(\frac{h}{(d-1)(h+1)}\right)^{\frac{d-1}{2}} \geqslant \frac{1}{d^{d/2}}$, a vector $\mathbf{s}$ sampled uniformly from $\mathbb{F}_2^{h-q}$ lies in $\mathcal{B}_{h-q}\left(\mathbf{z}, \frac{d-1}{2}\right)$ for some vector $\mathbf{z} \in \mathcal{C}$. This is indeed the desired condition in (10.1), which completes our proof. □

10.3.2 The Reduction

In this subsection, we prove the FPT reduction from the inapproximability of k-NCP problem to that of k-MDP (Lemma 10.6). It follows a general outline of the reduction from [DMS03], which is then modified (and simplified) to work in combination with LSDC instead of LDC.

Proof of Lemma 10.6. Let $(\mathbf{B}, \mathbf{y}, t)$ be the input for NCP where $\mathbf{B} \in \mathbb{F}_2^{n \times q}$, $\mathbf{y} \in \mathbb{F}_2^n$, and t is the parameter. We may assume without loss of generality that $t > \frac{\gamma}{2\gamma' - \gamma(4+\gamma')}$. Let d be the smallest odd integer greater than $\gamma' t$. Let $h, m \in \mathbb{N}, \mathbf{L} \in \mathbb{F}_2^{h \times m}$ be as in Lemma 10.9.

We produce an instance $(\mathbf{A}, k)$ for MDP by first sampling a random $\mathbf{s} \sim \mathbb{F}_2^{h-q}$. Then, we set $k = 2t + (d-1)/2$, $\mathbf{s}' = \mathbf{0}_q \circ -\mathbf{s}$ and

$$\mathbf{A} = \begin{bmatrix} \mathbf{B} & \mathbf{0}_{n \times (m-q)} & \mathbf{y} \\ \mathbf{L} & & \mathbf{s}' \end{bmatrix} \in \mathbb{F}_2^{(n+h) \times (m+1)}.$$

Note that the zeros are padded to the right of $\mathbf{B}$ so that the number of rows is the same as that of $\mathbf{L}$.

Since $k = 2t + (d-1)/2 = O_{\gamma'}(t)$ and the reduction clearly runs in polynomial time, we are only left to argue that it appropriately maps completeness and soundness cases.

(Completeness) Suppose that there exists $\mathbf{x} \in \mathcal{B}_q(\mathbf{0}, t)$ such that $\|\mathbf{Bx} - \mathbf{y}\|_0 \leqslant t$. From Lemma 10.9, with probability at least $1/d^{d/2}$, there exists $\mathbf{u} \in \mathcal{B}_{h-q}\left(\mathbf{s}, \frac{d-1}{2}\right)$ such that $\mathbf{x} \circ \mathbf{u} \in \mathbf{L}(\mathbb{F}_2^m)$. From this and from systematicity of $\mathbf{L}$, there exists $\mathbf{z}' \in \mathbb{F}_2^{m-q}$ such that $\mathbf{L}(\mathbf{x} \circ \mathbf{z}') = \mathbf{x} \circ \mathbf{u}$. Conditioned on this, we can pick $\mathbf{z} = \mathbf{x} \circ \mathbf{z}' \circ 1 \in \mathbb{F}_2^{m+1}$, which yields

$$\|\mathbf{Az}\|_0 = \|\mathbf{Bx} - \mathbf{y}\|_0 + \|\mathbf{x}\|_0 + \|\mathbf{u} - \mathbf{s}\|_0 \leqslant 2t + \frac{d-1}{2} = k.$$

(Soundness) Suppose that $\|\mathbf{Bx} - \mathbf{y}\|_0 > \gamma' t$ for all $\mathbf{x} \in \mathbb{F}_2^q$. We will show that $\|\mathbf{Az}\|_0 > \gamma' t$ for all non-zero $\mathbf{z}$; with our choices of k, d and assumption on t, it is simple to check that $\gamma' t > \gamma k$.

To show that $\|\mathbf{Az}\|_0 > \gamma' t$ for all $\mathbf{z} \in \mathbb{F}_2^{m+1} \setminus \{\mathbf{0}\}$, let us consider two cases, based on the last coordinate $\mathbf{z}[m+1]$ of $\mathbf{z}$. Let us write $\mathbf{z}$ as $\mathbf{x} \circ \mathbf{z}' \circ \mathbf{z}[m+1]$, where $\mathbf{x} \in \mathbb{F}_2^q$ and $\mathbf{z}' \in \mathbb{F}_2^{m-q}$.

If $\mathbf{z}[m+1] = 0$, then $\|\mathbf{Az}\|_0 = \|\mathbf{Bx}\|_0 + \|\mathbf{L}(\mathbf{x} \circ \mathbf{z}')\|_0 \geqslant \|\mathbf{L}(\mathbf{x} \circ \mathbf{z}')\|_0 \geqslant d$, where the last inequality comes from the fact that $\mathbf{L}$ is a generator matrix of a code of distance d (and that $\mathbf{z} \neq \mathbf{0}$). Finally, recall that we select $d > \gamma' t$, which yields the desired result for this case.

On the other hand, if $\mathbf{z}_{m+1} = 1$, then $\|\mathbf{Az}\|_0 \geqslant \|\mathbf{Bx} - \mathbf{y}\|_0 \geqslant \gamma' t$.

In conclusion, $\|\mathbf{Az}\|_0 > \gamma' t$ in all cases considered, which completes our proof. □

10.4 Inapproximability of k-SVP: Following Khot's Reduction

We will now prove the parameterized inapproximability of SVP, by reducing from the inapproximability of NVP (Theorem 10.5). This step is almost the same as that of Khot [Kho05], with small changes in parameter selection. Despite this, we repeat the whole argument here (with appropriate adjustments) for completeness.

The main properties of the (randomized) FPT reduction is summarized below. For succinctness, we define a couple of additional notation: let $\mathcal{L}(\mathbf{A})$ denote the lattice generated by the matrix $\mathbf{A} \in \mathbb{Z}^{n \times m}$, i.e., $\mathcal{L}(\mathbf{A}) = \{\mathbf{Ax} \mid \mathbf{x} \in \mathbb{Z}^m\}$, and let $\lambda_p(\mathcal{L})$ denote the length (in the ℓ_p norm) of the shortest vector of the lattice $\mathcal{L}$, i.e., $\lambda_p(\mathcal{L}) = \min_{\mathbf{0} \neq \mathbf{z} \in \mathcal{L}} \|\mathbf{z}\|_p$. Furthermore, for $\mathbf{y} \in \mathbb{Z}^n$, we define $\lambda_p(\mathcal{L}, \mathbf{y})$ as the ℓ_p-distance from $\mathbf{y}$ to the closest vector in $\mathcal{L}$, i.e., $\lambda_p(\mathcal{L}, \mathbf{y}) = \min_{\mathbf{z} \in \mathcal{L}} \|\mathbf{z} - \mathbf{y}\|_p$.

Lemma 10.10. *Fix $p > 1$, and let $\eta \geqslant 1$ be such that $\frac{1}{2} + \frac{1}{2^p} + \frac{(2^p+1)}{\eta} < 1$. There is a randomized polynomial time algorithm that takes in a* NVP *instance* $(\mathbf{B}, \mathbf{y}, t)$ *and produces an* SVP *instance* $(\mathbf{B}_{\text{svp}}, \gamma_p^{-1} l)$ *such that*

- *(Completeness) If $\lambda_p(\mathcal{L}(\mathbf{B}), \mathbf{y})^p \leqslant t$, then with probability* 0.8, $\lambda_p(\mathcal{L}(\mathbf{B}_{\text{svp}}))^p \leqslant \gamma_p^{-1} l$.
- *(Soundness) If $\lambda_p(\mathcal{L}(\mathbf{B}), \mathbf{y})^p > \eta \cdot t$, then with probability* 0.9, $\lambda_p(\mathcal{L}(\mathbf{B}_{\text{svp}}))^p > l$.
- *(Bounded Parameter) $l = \eta \cdot t$.*

Here $\gamma_p := \frac{1}{\frac{1}{2} + (2^p+1)/\eta + 1/2^p}$ is strictly greater than 1 *by our choice of η.*

Combining the above lemma with Theorem 10.5 gives us Theorem 10.2.

We devote the rest of this subsection to describing the reduction (which is similar to that from [Kho05]) and proving Lemma 10.10. In Section 10.4.1, we define the BCH lattice, which is the key gadget used in the reduction. Using the BCH lattice and the NVP instance, we construct the intermediate lattice $\mathbf{B}_{\text{int}}$ in Section 10.4.2. The intermediate lattice serves to blow up the number of "good vectors" for the completeness case, while controlling the number of "bad vectors" for the soundness case. In particular, this step ensures that the number of good vectors in the completeness case (Lemma 10.12) far outnumber the number of bad vectors in the soundness case (Lemma 10.13). Finally, in Section 10.4.3 we compose the intermediate lattice with a random homogeneous constraint (sampled from an appropriate distribution), to give the final SVP instance. The additional random constraint is used to annihilate all bad vectors in the soundness case, while retaining at least one good vector in the completeness case.

For the rest of the section, we fix $(\mathbf{B}, \mathbf{y}, t)$ to be a NVP instance, and set $l := \eta \cdot t$ and $r := \left(\frac{1}{2} + \frac{1}{2^p} + \frac{1}{\eta}\right) l$. For simplicity of calculations, we will assume that both l and r are integers, and that l is even. Furthermore, we say that a vector $\mathbf{u}$ is *good* (for the completeness case) if $\|\mathbf{u}\|_p^p \leqslant \gamma_p^{-1} l$, and we say that $\mathbf{u}$ is *bad* (for the soundness case) if $\|\mathbf{u}\|_p^p \leqslant l$.

10.4.1 The BCH Lattice gadget

We begin by defining the BCH lattices which is the key gadget used in the reduction. Given parameters $l, h \in \mathbb{N}$ where $h+1$ is a power of 2 and $l < h$. Let $g = (l/2) \cdot \log(h+1)$. Theorem 10.3 guarantees that there exists a BCH code with block length h, message length $h - g$ and distance $l + 1$. Let $\mathbf{P}_{\text{BCH}} \in \{0, 1\}^{g \times h}$ be the parity check matrix of such code. The BCH lattice is defined by

$$\mathbf{B}_{\text{BCH}} = \begin{bmatrix} \text{Id}_h & \mathbf{0}_{h \times g} \\ l \cdot \mathbf{P}_{\text{BCH}} & 2l \cdot \text{Id}_g \end{bmatrix} \in \mathbb{Z}^{(h+g) \times (h+g)}.$$

The following lemma, which is simply a restatement[8] of Lemma 4.3 in [Kho05], summarizes the key properties of BCH lattices, as defined above.

Lemma 10.11 ([Kho05]). *Let $\mathbf{B}_{\text{BCH}} \in \mathbb{Z}^{(h+g) \times (h+g)}$ be as above. There exists a randomized polynomial time algorithm that, with probability at least 0.99, returns a vector $\mathbf{s} \in \mathbb{Z}^{h+g}$ such that the following holds: there are at least $\frac{1}{100} 2^{-g} \binom{h}{r}$ distinct vectors $\mathbf{z} \in \mathbb{Z}^{h+g}$ such that $\|\mathbf{B}_{\text{BCH}}\mathbf{z} - \mathbf{s}\|_p^p = r$.*

10.4.2 The Intermediate Lattice

We now define the intermediate lattice. Let $(\mathbf{B}, \mathbf{y}, t)$ be an instance of NVP, where $\mathbf{B} \in \mathbb{Z}^{n \times q}$. The intermediate lattice $\mathbf{B}_{\text{int}}$ is constructed as follows. Let $l = \eta t$. Let h be the smallest power of

[8] In fact, Lemma 10.11 is even weaker than Khot's lemma, since we do not impose a bound on $\|\mathbf{z}\|_p$.

2 such that $h \geqslant \max\{2n, (10^{10}l)^{2\eta}\}$, and let $\mathbf{B}_{\mathrm{BCH}}$ be constructed as above. Then

$$\mathbf{B}_{\mathrm{int}} = \begin{bmatrix} 2\mathbf{B} & \mathbf{0}_{n\times(h+g)} & 2\mathbf{y} \\ \mathbf{0}_{(h+g)\times q} & \mathbf{B}_{\mathrm{BCH}} & \mathbf{s} \end{bmatrix} \in \mathbb{Z}^{(n+h+g)\times(q+h+g+1)}.$$

where $\mathbf{s} \in \mathbb{Z}^{h+g}$ is the vector given by Lemma 10.11.

Bounding Good Vectors in Completeness Case. We now prove a lower bound on the number of good vectors in the completeness case.

Lemma 10.12. *If $\lambda_p(\mathcal{L}(\mathbf{B}), \mathbf{y})^p \leqslant t$, then, with probability at least 0.99, there are at least $h^r\left(200h^{l/2}l^l\right)^{-1}$ good non-zero vectors in $\mathcal{L}(\mathbf{B}_{\mathrm{int}})$.*

Proof. Since $\lambda_p(\mathcal{L}(\mathbf{B}), \mathbf{y})^p \leqslant t$, there exists $\widetilde{\mathbf{x}} \in \mathbb{Z}^q$ such that $\|\mathbf{B}\widetilde{\mathbf{x}} - \mathbf{y}\|_p^p \leqslant t$. From Lemma 10.11, with probability at least 0.99, there exist at least $2^{-g}\binom{h}{r}/100$ distinct vectors $\mathbf{z} \in \mathbb{Z}^{h+g}$ such that $\|\mathbf{B}_{\mathrm{BCH}}\mathbf{z} - \mathbf{s}\|_p^p = r$. For each such $\mathbf{z}$, consider the vector $\mathbf{x} = \widetilde{\mathbf{x}} \circ \mathbf{z} \circ -1$. It follows that $\mathbf{B}_{\mathrm{int}}\mathbf{x} = (2\mathbf{B}\widetilde{\mathbf{x}} - 2\mathbf{y}) \circ (\mathbf{B}_{\mathrm{BCH}}\mathbf{z} - \mathbf{s})$ is a non-zero vector and $\|\mathbf{B}_{\mathrm{int}}\mathbf{x}\|_p^p = 2^p\|\mathbf{B}\widetilde{\mathbf{x}} - \mathbf{y}\|_p^p + \|\mathbf{B}_{\mathrm{BCH}}\mathbf{z} - \mathbf{s}\|_p^p \leqslant 2^p t + r = \gamma_p^{-1} l$. Since the number of such vectors $\mathbf{x}$ is at least the number of distinct coefficient vectors $\mathbf{z}$, it can be lower bounded by

$$\frac{1}{100} \cdot 2^{-g}\binom{h}{r} \geqslant \frac{1}{100} \cdot 2^{-\frac{l}{2}\log(h+1)}\binom{h}{r} \geqslant \frac{1}{100} \cdot \frac{h^r}{r^r(h+1)^{l/2}} \geqslant \frac{1}{200} \cdot \frac{h^r}{l^l h^{l/2}},$$

where the last inequality follows from $r \leqslant l$ and $l < h$. Finally, observe that each $\mathbf{z}$ produces different $\mathbf{B}_{\mathrm{BCH}}\mathbf{z}$ and hence all $\mathbf{B}_{\mathrm{int}}\mathbf{x}$'s are distinct. □

Bounding Bad Vectors in Soundness Case. We next bound the number of bad vectors in the soundness case:

Lemma 10.13. *If $\lambda_p(\mathcal{L}(\mathbf{B}), \mathbf{y})^p > \eta \cdot t$, then the number of bad vectors in $\mathcal{L}(\mathbf{B}_{\mathrm{int}})$ is at most $10^{-5}h^r\left(200h^{l/2}l^l\right)^{-1}$.*

At the heart of the proof is the claim that every bad vector must have even coordinates:

Claim 10.14. *If $\lambda_p(\mathcal{L}(\mathbf{B}), \mathbf{y})^p > \eta \cdot t$, then all coordinates of every bad $\mathbf{u} \in \mathcal{L}(\mathbf{B}_{\mathrm{int}})$ must be even.*

Proof. Let $\mathbf{u}$ be any bad vector in $\mathcal{L}(\mathbf{B}_{\mathrm{int}})$ and let $\mathbf{x} \in \mathbb{Z}^{q+h+g+1}$ be such that $\mathbf{B}_{\mathrm{int}}\mathbf{x} = \mathbf{u}$. We write $\mathbf{x}$ as $\mathbf{x}_1 \circ \mathbf{x}_2 \circ x$ where $\mathbf{x}_1 \in \mathbb{Z}^q$, $\mathbf{x}_2 \in \mathbb{Z}^{m+h}$ and $x \in \mathbb{Z}$. Using this, we can express $\mathbf{u}$ as $\mathbf{B}_{\mathrm{int}}\mathbf{x} = (2\mathbf{B}\mathbf{x}_1 - 2x \cdot \mathbf{y}) \circ (\mathbf{B}_{\mathrm{BCH}}\mathbf{x}_2 - x \cdot \mathbf{s})$. Recall that $\mathbf{u}$ is bad means that $\|\mathbf{u}\|_p^p \leqslant l$, which implies that $\|\mathbf{B}\mathbf{x}_1 - x \cdot \mathbf{y}\|_p^p \leqslant l = \eta \cdot t$. Since $\lambda_p(\mathcal{L}(\mathbf{B}), \mathbf{y})^p > \eta \cdot t$, it must be that $x = 0$.

Note that we now have $\mathbf{u} = (2\mathbf{B}\mathbf{x}_1) \circ (\mathbf{B}_{\mathrm{BCH}}\mathbf{x}_2)$. Let us assume for the sake of contradiction that $\mathbf{u}$ has at least one odd coordinate; it must be that $(\mathbf{B}_{\mathrm{BCH}}\mathbf{x}_2)$ has at least one odd coordinate.

Let us further write $\mathbf{x}_2$ as $\mathbf{x}_2 = \mathbf{w}_1 \circ \mathbf{w}_2$ where $\mathbf{w}_1 \in \mathbb{Z}^m$ and $\mathbf{w}_2 \in \mathbb{Z}^h$. Notice that $\mathbf{B}_{\text{BCH}}\mathbf{x}_2 = \mathbf{w}_1 \circ (l(\mathbf{P}_{\text{BCH}}\mathbf{w}_1 - 2\mathbf{w}_2))$. Since every coordinate of $\mathbf{B}_{\text{BCH}}\mathbf{x}_2$ must be less than l in magnitude, it must be the case that $\mathbf{P}_{\text{BCH}}\mathbf{w}_1 - 2\mathbf{w}_2 = \mathbf{0}$. In other words, $(\mathbf{w}_1 \mod 2)$ is a codeword of the BCH code. However, since the code has distance $l+1$, this means that, if $\mathbf{w}_1$ has at least one odd coordinate, it must have at least $l+1$ odd (non-zero) coordinates, which contradicts $\|\mathbf{u}\|_p^p \leqslant l$. □

Having proved Claim 10.14, we can now prove Lemma 10.13 by a simple counting argument.

Proof of Lemma 10.13. From Claim 10.14, all coordinates of $\mathbf{u}$ must be even. Therefore, $\mathbf{u}$ must have at most $l/2^p$ non-zero coordinates, all of which have magnitude at most $\lfloor l^{1/p} \rfloor \leqslant l-1$. Hence, we can upper bound the total number of such vectors by

$$\Big(2(l-1)+1\Big)^{l/2^p}\binom{n+h+g}{\lfloor \frac{l}{2^p} \rfloor} \leqslant (2l)^l (n+h+g)^{l/2^p} \leqslant (2l)^l (2lh)^{l/2^p} \leqslant (2l)^{2l} h^{l/2^p}$$

where the second-to-last step holds since $g \leqslant \frac{l}{2}\log(h+1) \leqslant lh/2$ and $n \leqslant h/2$. On the other hand,

$$\frac{h^r}{h^{l/2} l^l} = \frac{h^{\left(\frac{1}{2}+\frac{1}{\eta}+\frac{1}{2^p}\right)l}}{h^{l/2} l^l} = h^{l/2^p}(h/l^{\eta})^{l/\eta} \geqslant 10^8 \left((2l)^{2l} h^{l/2^p}\right),$$

which follows from $h \geqslant (10^{10} l)^{2\eta}$. Combining the two bounds completes the proof. □

10.4.3 The Final SVP Instance and Proof of The Main Lemma

Finally, we construct $\mathbf{B}_{\text{svp}}$ from $\mathbf{B}_{\text{int}}$ by adding a random homogeneous constraint similar to [Kho05]. For convenience, let N_g denote the lower bound on the number of distinct coefficient vectors guaranteed by Lemma 10.12 in the completeness case. Similarly, let N_a denote the upper bound on the number of annoying vectors as given in Lemma 10.13. Combining the two lemmas we have $N_g \geqslant 10^5 N_a$, which will be used crucially in the construction and analysis of the final lattice.

Construction of the Final Lattice : Let ρ be any prime number in[9] $\left[10^{-4}N_g, 10^{-2}N_g\right]$. Furthermore, let $\mathbf{r} \overset{\text{unif}}{\sim} [0, \rho-1]^{n+h+g}$ be a uniformly sampled lattice point. We construct $\mathbf{B}_{\text{svp}}$ as

$$\mathbf{B}_{\text{svp}} = \begin{bmatrix} \mathbf{B}_{\text{int}} & 0 \\ l \cdot \mathbf{r}^T \mathbf{B}_{\text{int}} & l \cdot \rho \end{bmatrix} \in \mathbb{Z}^{(n+h+g+1)\times(q+h+g+2)}.$$

This can be thought of as adding a random linear constraint to the intermediate lattice. The choice of parameters ensures that with good probability, in the completeness case, at least one of the good

[9] Note that the density of primes in this range is at least $1/\log N_g = 1/r\log h$. Therefore, a random sample of size $O(r\log h)$ in this range contains a prime with high probability. Since we can test primality for any $\rho \in \left[10^{-4}N_g, 10^{-2}N_g\right]$ in FPT time, this gives us an FPT algorithm to sample such a prime number efficiently .

vectors $\mathbf{x} \in \mathbb{Z}^{q+h+g+1}$ evaluates to 0 modulo ρ on the random constraint, and therefore we can pick $u \in \mathbb{Z}$ such that $\mathbf{B}_{\text{svp}}(\mathbf{x} \circ u) = (\mathbf{B}_{\text{int}}\mathbf{x}) \circ 0$ still has small ℓ_p norm. On the other hand, since $N_a \ll N_g$, with good probability, all of bad vectors in the soundness case evaluate to non-zeros, and hence will contribute a coordinate of magnitude l. This intuition is formalized below.

Proof of Lemma 10.10. Let $\mathbf{B}_{\text{svp}}$ be the corresponding final lattice of $(\mathbf{B}, \mathbf{y}, t)$ as described above. Observe that given the NVP instance $(\mathbf{B}, \mathbf{y}, t)$, we can construct $\mathbf{B}_{\text{svp}}$ in $\text{poly}(n, q, t)$-time.

Moreover, observe that $\mathcal{L}(\mathbf{B}_{\text{svp}})$ is simply $\{\mathbf{u} \circ (l \cdot w) \mid \mathbf{u} \in \mathcal{L}(\mathbf{B}_{\text{int}}), w \equiv \mathbf{r}^T\mathbf{u} \mod \rho\}$.

(Completeness) Suppose that $\lambda_p(\mathcal{L}(\mathbf{B}), \mathbf{y})^p \leqslant t$. We will show that, with probability at least 0.8, $\lambda_p(\mathcal{L}(\mathbf{B}_{\text{svp}}))^p \leqslant \gamma_p^{-1} l$. To do this, we first condition on the event that there exists at least N_g good vectors as guaranteed by Lemma 10.12. Consider any two good vectors $\mathbf{u}_1 \neq \mathbf{u}_2$. Since each entry of $\mathbf{u}_1$ and $\mathbf{u}_2$ is of magnitude at most $(\gamma_p^{-1} l)^{1/p}$, they are pairwise independent modulo $\rho > 2l$. Therefore, instantiating Lemma 5.8 from [Kho05] with the lower bound on the number of good vectors N_g, and our choice of ρ, it follows that with probability at least 0.9, there exists a good vector $\mathbf{u}$ such that $\mathbf{r}^T\mathbf{u} \equiv 0 \mod \rho$, i.e., $\mathbf{u} \circ 0$ belongs to $\mathcal{L}(\mathbf{B}_{\text{svp}})$. Therefore, by union bound, with probability at least 0.8 (over the randomness of Lemma 10.12 and the choice of $\mathbf{r}$), there exists a good $\mathbf{u} \in \mathcal{L}(\mathbf{B}_{\text{int}})$ such that $\mathbf{u} \circ 0$ remains in $\mathcal{L}(\mathbf{B}_{\text{svp}})$.

(Soundness) Suppose that $\lambda_p(\mathcal{L}(\mathbf{B}), \mathbf{y})^p > \eta \cdot t$. Consider any $\mathbf{u} \circ (l \cdot w) \in \mathcal{L}(\mathbf{B}_{\text{svp}})$. If $\|\mathbf{u} \circ (l \cdot w)\|_p^p \leqslant l$, it must be that $\|\mathbf{u}\|_p^p \leqslant l$ and $w = 0$; the latter is equivalent to $\mathbf{r}^T\mathbf{u} \equiv 0 \mod \rho$. However, from Lemma 10.13, there are only N_a bad vectors $\mathbf{u}$ in $\mathcal{L}(\mathbf{B}_{\text{int}})$. For each such non-zero $\mathbf{u}$, the probability that $\mathbf{r}^T\mathbf{u} \equiv 0 \mod \rho$ is exactly $1/\rho$. As a result, by taking union bound over all such $\mathbf{u} \neq \mathbf{0}$, we can conclude that, with probability at least $1 - N_a/\rho \geqslant 0.9$, we have $\lambda_p(\mathcal{L}(\mathbf{B}_{\text{svp}}))^p > l$. This concludes our proof. □

10.5 Discussion and Open Questions

In this chapter, we have shown the parameterized inapproximability of k-Minimum Distance Problem (k-MDP) and k-Shortest Vector Problem (k-SVP) in the ℓ_p norm for every $p > 1$ and their non-homogeneous counterpart k-NCP and k-NVP, assuming (randomized) Gap-ETH.

An immediate open question is whether k-SVP in the ℓ_1 norm is in FPT:

Open Question 7. *Is k-SVP in the ℓ_1 metric fixed-parameter tractable?*

Khot's reduction unfortunately does not work for ℓ_1; indeed, in the work of Haviv and Regev [HR07], they arrive at the hardness of approximating SVP in the ℓ_1 norm by embedding SVP instances in ℓ_2 to instances in ℓ_1 using an earlier result of Regev and Rosen [RR06]. The Regev-Rosen embedding inherently does not work in the FPT regime either, as it produces non-integral lattices. Similar issue applies to an earlier hardness result for SVP on ℓ_1 of [Mic00], whose reduction produces irrational bases.

An additional question regarding k-SVP is whether we can prove hardness of approximation for *every* constant factor. In the conference version of the work that this chapter is based on [Bha+18], this is shown for $p = 2$; however, the question remains open for $p \neq 2$. Please refer to [Bha+18] for a more detailed discussion regarding the barrier to apply gap amplification techniques of [Kho05; HR07], which yields NP-hardness of large factor for approximating SVP.

Furthermore, the Minimum Distance Problem can be defined for linear codes in $\mathbb{F}_p$ for any larger field of size $p > 2$ as well. It turns out that our result does not rule out FPT algorithms for k-MDP over $\mathbb{F}_p$ with $p > 2$. The issue here is that, in our proof of existence of LSDC (Lemma 10.9), we need the co-dimension of the code to be small compared to its distance. In particular, the co-dimension $h - m$ has to be at most $(d/2 + O(1)) \log_p h$ where d is the distance. While the BCH code over binary alphabet satisfies this property, we are not aware of any linear codes that satisfy this for larger fields. It is an intriguing open question to determine whether such codes exist, or whether the reduction can be made to work without existence of such codes.

Since the current reductions for both k-MDP and k-SVP are randomized, it remains open whether we can find deterministic reductions for these problems. As stated in the introduction, even in the non-parameterized setting, NP-hardness of SVP through deterministic reductions is not known. On the other hand, MDP is known to be NP-hard even to approximate under deterministic reductions; in fact, even the Dumer et al.'s reduction [DMS03] that we employ can be derandomized, as long as one has a deterministic construction for Locally Suffix Dense Codes [CW12a; Mic14]. In our settings, if one can deterministically construct Sparse Covering Codes (i.e. derandomize Lemma 10.9), then we would also get a deterministic reduction for k-MDP.

Finally, to the best of our knowledge, there is no $g(k)$-FPT-approximation algorithm for any function g for any of the four problems consider in this chapter. Hence, similar to k-UNIQUE SET COVER, it remains open whether these problems are totally FPT inapproximable:

Open Question 8. *Are k-NCP, k-NVP, k-MDP and k-SVP totally FPT inapproximable?*

Of course, due to the reduction in 10.2, if we can show that k-UNIQUE SET COVER is totally FPT inapproximable, then the total FPT inapproximability of k-NCP and k-NVP follow immediately. On the other hand, k-MDP and k-SVP seems to be much more challenging; all NP-hardness of approximation proofs for the two problems proceed by showing a small factor inapproximability, and then amplify the gap if possible. Such an approach is unlikely to even achieve a polynomial ratio, let alone total inapproximability.

Part III

Problems in P

Chapter 11

Inapproximability in P: Closest Pair and Maximum Inner Product

The Closest Pair of Points problem or *Closest Pair* problem (CP) is a fundamental problem in computational geometry: given n points in a d-dimensional metric space, find a pair of distinct points with the smallest distance between them. The Closest Pair problem for points in the Euclidean plane [SH75; BS76] stands at the origins of the systematic study of the computational complexity of geometric problems [PS85; Man89; KT05; Cor+09]. Since then, this problem has found abundant applications in geographic information systems [Hen06], clustering [Zah71; Alp10], and numerous matching problems (such as stable marriage [Won+07]).

The trivial algorithm for CP examines every pair of points in the point-set and runs in time $O(n^2d)$. Over the decades, there have been a series of developments on CP in low dimensional space for the Euclidean metric [Ben80; HNS88; KM95; SH75; BS76], leading to a deterministic $O(2^{O(d)}n\log n)$-time algorithm [BS76] and a randomized $O(2^{O(d)}n)$-time algorithm [Rab76; KM95]. For low (i.e., constant) dimensions, these algorithms are tight as a matching lower bound of $\Omega(n\log n)$ was shown by Ben-Or [Ben83] and Yao [Yao91] in the *algebraic decision tree* model, thus settling the complexity of CP in low dimensions. On other hand, for very high dimensions (for example, when $d = n$) there are subcubic algorithms [GS17; Ind+04] in the ℓ_1, ℓ_2, and ℓ_∞-metrics[1] using fast matrix multiplication algorithms [Gal14]. However, CP in medium dimensions, i.e., $d = \text{polylog}(n)$, and in various ℓ_p-metrics, have been a focus of study in machine learning and analysis of Big Data [Kle97], and it is surprising that, even with the tools and techniques that have been developed over many decades, when $d = \omega(\log n)$, there is no known subquadratic-time (i.e., $O(2^{o(d)}n^{2-\varepsilon})$-time) algorithm, for CP in any standard distance measure [Ind00; AC09; Ind+04] . The absence of such algorithms was explicitly observed as early as the late nineties by Cohen and Lewis [CL99] but there was not any explanation until recently.

David, Karthik, and Laekhanukit [DKL18] showed that for all $p > 2$, assuming the Orthogonal Vectors Hypothesis (OVH), for every $\varepsilon > 0$, no algorithm running in $n^{2-\varepsilon}$ time can solve CP in the ℓ_p-metric, even when $d = \omega(\log n)$. Their conditional lower bound was based on the conditional

[1] In fact, when $d = n$ there are subcubic algorithms for every ℓ_p-metric, where p is even [Ind+04].

lower bound (again assuming OVH) of Alman and Williams [AW15] for the *Bichromatic Closest Pair* problem[2] (BCP) where we are given two sets of n points in a d-dimensional metric space, and the goal is to find a pair of points, one from each set, with the smallest distance between them. Alman and Williams showed that for all $p \in \mathbb{R}_{\geqslant 1} \cup \{0\}$, assuming OVH, for every $\varepsilon > 0$, no algorithm running in $n^{2-\varepsilon}$ time can solve BCP in the $\omega(\log n)$-dimensional ℓ_p-metric space. Given that [AW15] show their lower bound on BCP for all ℓ_p-metrics, the lower bound on CP of [DKL18] feels unsatisfactory, since the ℓ_2-metric is arguably the most interesting metric to study CP on. On the other hand, the answer to the complexity of CP in the Euclidean metric might be on the positive side, i.e., there might exist an algorithm that performs well in the ℓ_2-metric because there are more tools available, e.g., Johnson-Lindenstrauss' dimension reduction [JL84]. Thus we have the following question:

Open Question 9 ([ARW17a][3][Wil18a; DKL18])**.** *Is there an algorithm running in time $n^{2-\varepsilon}$ for some $\varepsilon > 0$ which can solve* CP *in the Euclidean metric when the points are in $\omega(\log n)$ dimensions?*

Even if the answer to the above question is negative, this does not rule out strong approximation algorithms for CP in the Euclidean metric, which might suffice for all applications. Indeed, we do know of subquadratic approximation algorithms for CP. For example, LSH based techniques can solve $(1+\delta)$-CP (i.e., $(1+\delta)$ factor approximate CP) in $n^{2-\Theta(\delta)}$ time [IM98], but cannot do much better [MNP07; OWZ14]. In a recent breakthrough, Valiant [Val15] obtained an approximation algorithm for $(1+\delta)$-CP with runtime of $n^{2-\Theta\left(\sqrt{\delta}\right)}$. The state of the art is an $n^{2-\widetilde{\Theta}\left(\delta^{1/3}\right)}$-time algorithm by Alman, Chan, and Williams [ACW16]. Can the dependence on δ be improved indefinitely? For the case of $(1+\delta)$-BCP, assuming OVH, Rubinstein [Rub18] answered the question in the negative. Does $(1+\delta)$-CP also admit the same negative answer?

Open Question 10. *Is there an algorithm running in time $n^{2-\varepsilon}$ for some $\varepsilon > 0$ which can solve* $(1+\delta)$-CP *in the Euclidean metric when the points are in $\omega(\log n)$ dimensions for every $\delta > 0$?*

Another important geometric problem is the *Maximum Inner Product* problem (MIP): given n points in the d-dimensional Euclidean space, find a pair of distinct points with the largest inner product. This problem along with its bichromatic variant (*Bichromatic Maximum Inner Product* problem, denoted BMIP) is extensively studied in literature (see [ARW17a] and references therein). Abboud, Rubinstein, and Williams [ARW17a] showed that assuming OVH, for every $\varepsilon > 0$, no $2^{(\log n)^{1-o(1)}}$-approximation algorithm running in $n^{2-\varepsilon}$ time can solve BMIP when $d = n^{o(1)}$. It is a natural question to ask if their inapproximability result can be extended to MIP:

Research Question 6. *Is there an algorithm running in time $n^{2-\varepsilon}$ for some $\varepsilon > 0$ which can solve* γ-MIP *in $n^{o(1)}$ dimensions for even $\gamma = 2^{(\log n)^{1-o(1)}}$?*

[2] We remark that BCP is of independent interest as it's equivalent to finding the *Minimum Spanning Tree* in ℓ_p-metric [Aga+91; KLN99]. Moreover, understanding the fine-grained complexity of BCP has lead to better understanding of the query time needed for *Approximate Nearest Neighbor* search problem (see Razenshteyn's thesis [Raz17] for a survey about the problem) with polynomial preprocessing time [Rub18].

[3] Please see the erratum in [ARW17b].

Our Results

In this chapter we address all three previously mentioned open questions. First, we almost completely resolve Open Question 9. In particular, we show the following.

Theorem 11.1 (Subquadratic Hardness of CP; Informal, See Theorem 11.20)**.** *Let $p \in \mathbb{R}_{\geqslant 1} \cup \{0\}$. Assuming* OVH*, for every $\varepsilon > 0$, no algorithm running in $n^{2-\varepsilon}$ time can solve* CP *in the ℓ_p-metric, even when $d = (\log n)^{\Omega_\varepsilon(1)}$.*

In particular we would like to emphasize that the dimension for which we show the lower bound on CP depends on ε. We would also like to remark that our lower bound holds even when the input point-set of CP is a subset of $\{0,1\}^d$. Finally, we note that the centerpiece of the proof of the above theorem (and also the proofs of the other results that will be subsequently mentioned) is the construction of a dense bipartite graph with low *contact dimension*, i.e., we construct a balanced bipartite graph on n vertices with $n^{2-\varepsilon}$ edges whose vertices can be realized as points in a $(\log n)^{\Omega_\varepsilon(1)}$-dimensional ℓ_p-metric space such that every pair of vertices which have an edge in the graph are at distance exactly 1 and every other pair of vertices are at distance greater than 1. This graph construction is inspired by the construction of locally dense codes introduced by Dumer, Miccancio, and Sudan [DMS03] and uses special density properties of Reed Solomon codes. A detailed proof overview is given in Section 11.1.1.

Next, we improve our result in Theorem 11.1 in some aspects by showing $1 + o(1)$ factor inapproximability of CP even in $O_\varepsilon(\log n)$ dimensions, but can only rule out algorithms running in $n^{1.5-\varepsilon}$ time (as opposed to Theorem 11.1 which rules out exact algorithms for CP running in $n^{2-\varepsilon}$ time). More precisely, we show the following.

Theorem 11.2 (Subquadratic Hardness of gap-CP)**.** *Let $p \in \mathbb{R}_{\geqslant 1} \cup \{0\}$. Assuming* OVH*, for every $\varepsilon > 0$, there exists $\delta(\varepsilon) > 0$ and $c(\varepsilon) > 1$ such that no algorithm running in $n^{1.5-\varepsilon}$ time that can solve $(1 + \delta)$-*CP *in the ℓ_p-metric, even when $d = c \log n$.*

We remark that the $n^{1.5-\varepsilon}$ lower bound on approximate CP is an artifact of our proof strategy and that a different approach or an improvement in the state-of-the-art bound on the number of minimum weight codewords in algebraic geometric codes (which are used in our proof), will lead to the complete resolution of Open Question 10.

It should also be noted that the approximate version of CP and the dimension are closely related. Namely, using standard dimensionality reduction techniques [JL84][4] for $(1 + \delta)$-CP, one can always assume that $d = O_\delta(\log n)$. In other words, hardness of $(1 + \delta)$-CP immediately yields logarithmic dimensionality bound as a byproduct.

Finally, we completely answer Open Question 6 by showing the following inapproximability result for MIP, matching the hardness for BMIP from [ARW17a].

Theorem 11.3 (Subquadratic Hardness of gap-MIP)**.** *Assuming* OVH*, for every $\varepsilon > 0$, no algorithm running in $n^{2-\varepsilon}$ time can solve γ-*MIP *for any $\gamma \leqslant 2^{(\log n)^{1-o(1)}}$, even when $d = n^{o(1)}$.*

[4] In fact, since our results apply to $\{0,1\}$-vectors, simply subsampling coordinates would also work.

Recently, there have been a lot of results connecting BCP or $(1+o(1))$-BCP to other problems (see [Rub18; Che18a; Che18b; CW19]). Now such connections can be extended to CP as well. For example, the following conditional lower bound follows from [Rub18] for gap-CP in the edit distance metric and for completeness a proof is given in Appendix 11.7.

Theorem 11.4 (Subquadratic Hardness of gap-CP in edit distance metric)**.** *Assuming* OVH*, for every $\varepsilon > 0$, there exists $\delta(\varepsilon) > 0$ and $c(\varepsilon) > 1$ such that no algorithm running in $n^{1.5-\varepsilon}$ time can solve* $(1+\delta)$-CP *in the edit distance metric, even when $d = c \log n \log \log n$.*

11.1 Proof Overview

In this section, we provide an overview of our proofs. For ease of presentation, we will sometimes be informal here; all notions and proofs are formalized in subsequent sections. Our overview is organized as follows. First, in Subsection 11.1.1, we outline our proof of running time lower bounds for exact CP (Theorem 11.1). Then, in Subsection 11.1.2, we abstract part of our reduction using error-correcting codes, and relate them back to the works on locally dense codes [DMS03; CW12b; Mic14] that inspire our constructions. Finally, in Subsection 11.1.3, we briefly discuss how to modify the base construction (i.e. code properties) to give conditional lower bounds for approximate CP and MIP (Theorems 11.2 and 11.3).

11.1.1 Conditional Lower Bound on Exact Closest Pair

In this subsection, we provide a proof overview of a slightly weaker version of Theorem 11.1, i.e., we show that assuming SETH, for every $p \in \mathbb{R}_{\geqslant 1} \cup \{0\}$, no subquadratic time algorithm can solve CP in the ℓ_p-metric when $d = (\log n)^{\omega(1)}$. We prove such a result by reducing BCP in dimension d to CP in dimension $d + (\log n)^{\omega(1)}$, and the subquadratic hardness for CP follows from the subquadratic hardness of BCP established by [AW15]. Note that the results in this chapter remain interesting even if SETH is false, as our reduction shows that BCP and CP are computationally equivalent[5] (up to $n^{o(1)}$ factor in the running time) when $d = (\log n)^{\omega(1)}$. The conditional lower bound on CP is merely a consequence of this computational equivalence. Finally, we note that a similar equivalence also holds between MIP and BMIP.

Understanding an obstacle of [DKL18]. Our proof builds on the ideas of [DKL18] who showed that assuming SETH, for every $p > 2$, no subquadratic time algorithm can solve CP in the ℓ_p-metric when $d = \omega(\log n)$. They did so by connecting the complexity of CP and BCP via the *contact dimension* of the balanced complete bipartite graph (biclique), denoted by $K_{n,n}$. We elaborate on this below.

[5] We can reduce an instance of CP to an instance of BCP by randomly partitioning the input set of CP instance into two, and the optimal closest pair of points will be in different sets with probability $1/2$ (and this reduction can be made deterministic).

To motivate the idea behind [DKL18], let us first consider the trivial reduction from BCP to CP: given an instance A, B of BCP, we simply output $A \cup B$ as an instance of CP. This reduction fails because there is no guarantee on the distances of a pair of points both in A (or both in B). That is, there could be two points $\mathbf{a}, \mathbf{a}' \in A$ such that $\|\mathbf{a} - \mathbf{a}'\|_p$ is much smaller than the optimum of BCP on A, B. If we simply solve CP on $A \cup B$, we might find such $\mathbf{a}, \mathbf{a}'$ as the optimal pair but this does not give the answer to the original BCP problem. In order to circumvent this issue, one needs a gadget that "stretch" pairs of points both in A or both in B further apart while keeping the pairs of points across A and B close (and preserving the optimum of BCP on A, B). It turns out that this notion corresponds exactly to the contact dimension of the biclique, which we define below.

Definition 11.5 (Contact Dimension [Pac80])**.** *For any graph $G = (V, E)$, a mapping $\tau : V \to \mathbb{R}^d$ is said to* realize G (in the ℓ_p-metric) *if for some $\beta > 0$, the following holds for every distinct vertices u, v:*

$$\|\tau(u) - \tau(v)\|_p = \beta \textit{ if } \{u, v\} \in E, \textit{ and,} \tag{11.1}$$

$$\|\tau(u) - \tau(v)\|_p > \beta \textit{ otherwise.} \tag{11.2}$$

The contact dimension *(in the ℓ_p-metric) of G, denoted by $\mathsf{cd}_p(G)$, is the minimum $d \in \mathbb{N}$ such that there exists $\tau : V \to \mathbb{R}^d$ realizing G in the ℓ_p-metric.*

In this chapter, we will be mainly interested in the contact dimension of bipartite graphs. Specifically, [DKL18] only consider the contact dimension of the biclique $K_{n,n}$. Notice that a realization of biclique ensures that vertices on the same side are far from each other while vertices on different sides are close to each other preserving the optimum of BCP; these are exactly the desired properties of a gadget outlined above. Using this, [DKL18] give a reduction from BCP to CP which shows that the two are computationally equivalent whenever $d = \Omega(\mathsf{cd}_p(K_{n,n}))$, as follows.

Let $A, B \subseteq \mathbb{R}^d$ each of cardinality n be an instance of BCP and let $\tau : A \dot\cup B \to \mathbb{R}^{\mathsf{cd}_p(K_{n,n})}$ be a map realizing the biclique $(A \dot\cup B, A \times B)$ in the ℓ_p-metric; we may assume w.l.o.g. that $\beta = 1$. Let δ be the distance between any point in A and any point in B (i.e., δ is an upper bound on the optimum of BCP). Let $\rho > 0$ be such that $\|\tau(\mathbf{a}) - \tau(\mathbf{b})\|_p > 1 + \rho$ for all $\mathbf{a} \in A, \mathbf{b} \in B$ (and this is guaranteed to exist by (11.2)). Moreover, let $k > \delta/\rho$ be any sufficiently large number. Consider the point-sets $\widetilde{A}, \widetilde{B} \subseteq \mathbb{R}^{d+\mathsf{cd}_p(K_{n,n})}$ of cardinality n each defined as

$$\widetilde{A} = \{\mathbf{a} \circ (k \cdot \tau(\mathbf{a})) \mid \mathbf{a} \in A\},\ \widetilde{B} = \{\mathbf{b} \circ (k \cdot \tau(\mathbf{b})) \mid \mathbf{b} \in B\},$$

where $\circ$ denotes the concatenation between two vectors and $k \cdot \mathbf{x}$ denotes the usual scalar-vector multiplication (i.e. scaling $\mathbf{x}$ up by a factor of k). For brevity, we write $\widetilde{\mathbf{a}}$ and $\widetilde{\mathbf{b}}$ to denote $\mathbf{a} \circ (k \cdot \tau(\mathbf{a}))$ and $\mathbf{b} \circ (k \cdot \tau(\mathbf{b}))$ respectively.

We now argue that, if we can find the closest pair of points in $\widetilde{A} \cup \widetilde{B}$, then we also immediately solve BCP for (A, B). More precisely, we claim that $(\mathbf{a}^*, \mathbf{b}^*) \in A \times B$ is a bichromatic closest pair of (A, B) if and only if $(\widetilde{\mathbf{a}^*}, \widetilde{\mathbf{b}^*})$ is a closest pair of $\widetilde{A} \cup \widetilde{B}$.

To see that this is the case, observe that, for cross pairs $(\tilde{\mathbf{a}}, \tilde{\mathbf{b}}) \in \tilde{A} \times \tilde{B}$, (11.1) implies that the distance $\|\tilde{\mathbf{a}} - \tilde{\mathbf{b}}\|_p$ is exactly $(k^p + \|\mathbf{a} - \mathbf{b}\|_p^p)^{1/p}$; hence, among these pairs, $(\widetilde{\mathbf{a}^*}, \widetilde{\mathbf{b}^*})$ is a closest pair iff $(\mathbf{a}^*, \mathbf{b}^*)$ is a bichromatic closest pair in A, B. Notice also that, since the bichromatic closest pair in A, B is of distance at most δ, the closest pair in $\tilde{A} \cup \tilde{B}$ is of distance at most $(k^p + \delta^p)^{1/p} \leqslant k + \delta$.

On the other hand, for pairs both from $\tilde{A}$ or both from $\tilde{B}$, the distance must be at least $k(1 + \rho)$, which is more than $k + \delta$ from our choice of k. As a result, these pairs cannot be a closest pair in $\tilde{A} \cup \tilde{B}$, and this concludes the sketch of the proof.

There are a couple of details that we have glossed over here: one is that the gap ρ cannot be too small (e.g., ρ cannot be as small as $1/2^n$) and the other is that we should be able to construct τ efficiently. Nevertheless, these are typically not an issue.

[DKL18] show that $\mathsf{cd}_p(K_{n,n}) = \Theta(\log n)$ when $p > 2$ and that the realization can be constructed efficiently and with sufficiently large ρ. This implies the subquadratic hardness of CP (by reduction from BCP) in the ℓ_p-metric for all $p > 2$ and $d = \omega(\log n)$. However, it was known that $\mathsf{cd}_2(K_{n,n}) = \Theta(n)$ [FM88]. Thus, they could *not* extend their conditional lower bound to CP in the Euclidean metric[6] even when $d = o(n)$. In fact, this is a serious obstacle as it rules out many natural approaches to reduce BCP to CP in a black-box manner. Elaborating, the lower bound on $\mathsf{cd}_2(K_{n,n})$ rules out local gadget reductions which would replace each point with a composition of that point and a gadget with a small increase in the number of dimensions, as such gadgets can be used to construct a realization of $K_{n,n}$ in the Euclidean metric in a low dimensional space, contradicting the lower bound on $\mathsf{cd}_2(K_{n,n})$.

Overcoming the Obstacle: Beyond Biclique. We overcome the above obstacle by considering dense bipartite graphs, instead of the biclique. More precisely, we show that there exists a balanced bipartite graph $G^* = (A^* \dot{\cup} B^*, E^*)$ on $2n$ vertices such that $|E^*| \geqslant n^{2-o(1)}$ and $\mathsf{cd}_p(G^*)$ is small (i.e. $\mathsf{cd}_p(G^*) \leqslant (\log n)^{\omega(1)}$). We give a construction of such a graph below but before we do so, let us briefly argue why this suffices to show that BCP and CP are computationally equivalent (up to $n^{o(1)}$ multiplicative overhead in the running time) for dimension $d = \Omega(\mathsf{cd}_p(G^*))$.

Let us consider the same reduction which produces $\tilde{A}, \tilde{B}$ as before, but instead of using a realization of the biclique, we use a realization τ of G^*. This reduction is of course incorrect: if $(\mathbf{a}^*, \mathbf{b}^*)$ is not an edge in G^*, then $\|\tau(\mathbf{a}^*) - \tau(\mathbf{b}^*)\|_p$ could be large and, thus the corresponding pair of points $(\widetilde{\mathbf{a}^*}, \widetilde{\mathbf{b}^*}) \in \tilde{A} \times \tilde{B}$, may not be the closest pair. Nevertheless, we are not totally hopeless: if $(\mathbf{a}^*, \mathbf{b}^*)$ is an edge, then we are in good shape and the reduction is correct.

With the above observation in mind, consider picking a random permutation π of $A \cup B$ such that $\pi(A) = A$ and $\pi(B) = B$ and then initiate the above reduction with the map $(\tau \circ \pi)$ instead of τ. Note that $\tau \circ \pi$ is simply a realization of an appropriate permutation G' of G^* (i.e., G' is isomorphic to G^*). Due to this, the probability that we are "lucky" and $(\mathbf{a}^*, \mathbf{b}^*)$ is an edge in G' is $p := |E|/n^2$; when this is the case, solving CP on the resulting instance would give the correct

[6] Note that plugging in the bound on $\mathsf{cd}_2(K_{n,n})$ in the result of [DKL18] yields that assuming SETH, no subquadratic in n running time algorithm can solve CP when $d = \Omega(n)$. This is not a meaningful lower bound as just the input size of CP when $d = \Omega(n)$ is $\Omega(n^2)$.

answer for the original BCP instance. If we repeat this $\log n/p = n^{o(1)}$ times, we would find the optimum of the original BCP instance with high probability.

To recap, even when G^* is not a biclique, we can still use it to give a reduction from BCP to CP, except that the reduction produces multiple (i.e. $\widetilde{O}(n^2/|E^*|)$) instances of CP. We remark here that the reduction can be derandomized: we can deterministically (and efficiently) pick the permutations so that the permuted graphs covers $K_{n,n}$ (see Lemma 11.8). As a minor digression, we would like to draw a parallel here with a recent work of Abboud, Rubinstein, and Williams [ARW17a]. The obstacle raised in [DKL18] is about the impossibility of certain kinds of many-one gadget reductions. We overcame it by designing a reduction from BCP to CP which not only increased the number of dimensions but also the number of points (by creating multiple instances of CP). This technique is also utilized in [ARW17a] where they showed the impossibility of Deterministic Distributed PCPs (Theorem I.2 in [ARW17a]) but then overcame that obstacle by using an advice (which is then enumerated over resulting in multiple instances) to build Non-deterministic Distributed PCPs.

Constructing a dense bipartite graph with low contact dimension. We now proceed to construct the desired graph $G^* = (A^* \cup B^*, E^*)$. Note that any construction of a dense bipartite graph with contact dimension $n^{o(1)}$ is non-trivial. This is because it is known that a random graph has contact dimension $\Omega(n)$ in the Euclidean metric with high probability [RRS89; BL05], and therefore our graph construction must be significantly better than a random graph.

Our realization τ^* of G^* will map into a subset of $\{0,1\}^{(\log n)^{\omega(1)}}$. As a result, we can fix $p = 0$, since a realization of a graph with entries in $\{0,1\}$ in the Hamming-metric also realizes the same graph in every ℓ_p-metric for any $p \neq \infty$.

Fix $g = \omega(1)$. We associate $[n]$ with $\mathbb{F}_q^h$ where $q = \Theta\left((\log n)^g\right)$ is a prime and $h = \Theta\left(\frac{\log n}{g \cdot \log\log n}\right)$. Let $\mathcal{P}$ be the set of all univariate polynomials (in x) over $\mathbb{F}_q$ of degree at most $h-1$. We have that $|\mathcal{P}| = q^h = n$ and associate $\mathcal{P}$ with A^*. Let $\mathcal{Q}$ be the set of all univariate monic polynomials (in x) over $\mathbb{F}_q$ of degree h, i.e.,

$$\mathcal{Q} = \{x^h + p(x) \mid p(x) \in \mathcal{P}\}.$$

We associate the polynomials in $\mathcal{Q}$ with the vertices in B^* (note that $|\mathcal{Q}| = n$). In fact, we view the vertices in A^* and B^* as being uniquely labeled by polynomials in $\mathcal{P}$ and $\mathcal{Q}$ respectively. For notational clarity, we write p_a (resp. p_b) to denote the polynomial in $\mathcal{P}$ (resp. $\mathcal{Q}$) that is associated to $a \in A^*$ (resp. $b \in B^*$).

For every $a \in A^*$ and $b \in B^*$, we include (a, b) as an edge in E^* if and only if the polynomial $p_b - p_a$ (which is of degree h) has h distinct roots. This completes the construction of G^*. We have to now show the following two claims about G^*: (i) $|E^*| = n^{2-O(1/g)} = n^{2-o(1)}$ and (ii) there is $\tau : A^* \dot{\cup} B^* \to \{0,1\}^{(\log n)^{O(g)}} = \{0,1\}^{(\log n)^{\omega(1)}}$ that realizes G^*.

To show (i), let $\mathcal{R}$ be the set of all monic polynomials of degree h with h distinct roots. We have that $|\mathcal{R}| = \binom{q}{h}$. Fix a vertex $a \in A^*$. Its degree in G^* is exactly $|\mathcal{R}| = \binom{q}{h}$. This is because, for every polynomial $r \in \mathcal{R}$, $r + a$ belongs to $\mathcal{Q}$, and therefore $(a, r + a) \in E^*$. This implies the following bound on $|E^*|$:

$$|E^*| = q^h \cdot \binom{q}{h} \geqslant q^h \cdot \frac{q^h}{h^h} > \frac{n^2}{(\log n)^{\Theta((\log n)/(g \cdot \log\log n))}} = n^{2-O(1/g)}.$$

Next, to show (ii), we construct a realization $\tau^* : A^* \dot\cup B^* \to \mathbb{F}_q^q$ of G^*. We note that, it is simple to translate the entries to $\{0, 1\}$ instead of $\mathbb{F}_q$, by replacing $i \in \mathbb{F}_q$ with the i-th standard basis $\mathbf{e}_i \in \{0, 1\}^q$. This would result in a realization $\tau^* : A^* \dot\cup B^* \to \{0, 1\}^{q^2}$ of G^*; notice that the dimension of τ^* is $q^2 = \Theta((\log n)^{2g})$ as claimed.

We define τ^* as follows.

- For every $a \in A^*$, $\tau^*(a)$ is simply the vector of evaluation of p_a on every element in $\mathbb{F}_q$. More precisely, for every $j \in [q]$, the j-th coordinate of $\tau^*(a)$ is $p_a(j-1)$.
- Similarly, for every $b \in B^*$ and $j \in [q]$, the j-th coordinate of $\tau^*(b)$ is $p_b(j-1)$.

We now show that τ^* is indeed a realization of G^*; specifically, we show that τ^* satisfies (11.1) and (11.2) with $\beta = q - h$.

Consider any edge $(a, b) \in E^*$. Notice that $\|\tau^*(a) - \tau^*(b)\|_0$ is the number of $x \in \mathbb{F}_q$ such that $p_b(x) - p_a(x) \neq 0$. By definition of E^*, $p_b - p_a$ is a polynomial with h distinct roots over $\mathbb{F}_q$. Thus, $\|\tau^*(a) - \tau^*(b)\|_0 = q - h = \beta$ as desired.

Next, consider a non-edge $(a, b) \in (A^* \times B^*) \setminus E^*$. Then, we know that $p_b - p_a$ has at most $h - 1$ distinct roots over $\mathbb{F}_q$. Therefore, the polynomial $p_b - p_a$ is non-zero on at least $q - h + 1$ coordinates. This implies that $\|\tau^*(a) - \tau^*(b)\|_0 \geqslant q - h + 1 > \beta$.

Finally, for any distinct $a, a' \in A^*$, we have $\|\tau^*(a) - \tau^*(a')\|_0 \geqslant q - h + 1$ because $p_a - p_{a'}$ is a non-zero polynomial of degree at most $h - 1$ and thus can be zero over $\mathbb{F}_q$ in at most $h - 1$ locations. Similarly, $\|\tau^*(b) - \tau^*(b')\|_0 \geqslant q - h + 1$ for any distinct $b, b' \in B^*$.

This completes the proof sketch for both the claims about G^* and yields Theorem 11.1 for $d = (\log n)^{\omega(1)}$. Finally we remark that in the actual proof of Theorem 11.1, we will set the parameters in the above construction more carefully and achieve the bound $\mathsf{cd}_p(G^*) = (\log n)^{O_\varepsilon(1)}$.

11.1.2 Abstracting the Construction via Error-Correcting Codes

Before we move on to discuss the proofs of Theorems 11.3 and 11.2, let us give an abstraction of the construction in the previous subsection. This will allow us to easily generalize the construction for the aforemention theorems, and also to explain where our motivation behind the construction comes from in the first place.

For notational convenient, we use "code" or "error-correcting code" to refer to a set of codewords $\mathcal{C}$ rather than the mapping C throughout this chapter.

Dense Bipartite Graph with Low Contact Dimension from Codes. In order to construct a balanced bipartite graph G^* on $2n$ vertices with $n^{2-o(1)}$ edges such that $\mathsf{cd}_p(G^*) \leqslant d^*$, it suffices to have a code $\mathcal{C}^*$ with the following properties (for code-related definitions, see Section 2.7):

- $\mathcal{C}^* \subseteq \mathbb{F}_q^\ell$ of cardinality n is a linear code with block length ℓ, and distance Δ over alphabet $\mathbb{F}_q$.

- There exists a *center* $\mathbf{s}^* \in \mathbb{F}_q^\ell$ and $r^* < \Delta$ such that $|\mathcal{C}^*|^{1-o(1)}$ codewords are at Hamming distance exactly r^* from $\mathbf{s}^*$ and no codeword is at distance less than r^* from s^*.

- $q \cdot \ell = d^*$.

We also require that $\mathcal{C}^*$ and s^* can be constructed in poly(n) time; such a requirement holds for all the codes we use, and we shall ignore this requirement for the ease of exposition.

We describe below how to construct G^* from $\mathcal{C}^*$, but first note that the construction of G^* we saw in the previous subsubsection was just showing that Reed Solomon codes [RS60] of block length $q = \Theta((\log n)^g)$ and message length $h = \Theta\left(\frac{\log n}{g\cdot\log\log n}\right)$ over alphabet $\mathbb{F}_q$ with distance $q - h + 1$ has the above properties. The center $\mathbf{s}^*$ in that construction was the evaluation of the polynomial x^h over $\mathbb{F}_q$, and r^* was $q - h$.

In general, to construct G^* from $\mathcal{C}^*$, we first define a subset $S^* \subseteq \mathbb{F}_q^\ell$ of cardinality n as follows:

$$S^* = \{\mathbf{s}^* + \mathbf{c} \mid \mathbf{c} \in \mathcal{C}^*\}.$$

We associate the vertices in A^* with the codewords of $\mathcal{C}^*$ and vertices in B^* with the strings in S^*. For any $(\mathbf{a}, \mathbf{b}) \in A^* \times B^*$, let $(\mathbf{a}, \mathbf{b}) \in E^*$ if and only if $\|\mathbf{b} - \mathbf{a}\|_0 = r^*$. This completes the construction of G^*. We have to now show the following claims about G^*: (i) $|E^*| = n^{2-o(1)}$ and (ii) there is $\tau : A^*\dot\cup B^* \to \{0,1\}^{q\cdot\ell}$ that realizes G^*.

Item (i) follows rather easily from the properties of $\mathcal{C}^*$ and s^*. Let T^* be the subset of $\mathcal{C}^*$ of all codewords which are at distance exactly equal to r^* from $\mathbf{s}^*$. From the definition of $\mathbf{s}^*$, we have $|T^*| = |\mathcal{C}^*|^{1-o(1)}$. Fix $\mathbf{a} \in A^*$. Its degree in G^* is $|T^*| = |\mathcal{C}^*|^{1-o(1)}$. This is because for every codeword $\mathbf{t} \in T^*$ we have that $\mathbf{t} - \mathbf{a}$ is a codeword in $\mathcal{C}^*$ (from the linearity of $\mathcal{C}^*$) and thus $\mathbf{s}^* - \mathbf{t} + \mathbf{a}$ is in S^*, and therefore $(\mathbf{a}, \mathbf{s}^* - \mathbf{t} + \mathbf{a}) \in E^*$.

For item (ii), consider the identity mapping $\tau^* : A^*\dot\cup B^* \to \mathbb{F}_q^\ell$ that maps each string to itself. It is simple to check that τ^* realizes G^* in the Hamming metric (with $\beta = r^*$).

Recall from the previous subsection that given $\tau^* : A^*\dot\cup B^* \to \mathbb{F}_q^\ell$ that realizes G^* in the Hamming metric, it is easy to construct $\tau : A^*\dot\cup B^* \to \{0,1\}^{q\cdot\ell}$ that realizes G^* in the Hamming metric with a q multiplicative factor blow-up in the dimension. This completes the proof of both the claims about G^* and gives a general way to prove Theorem 11.1 given the construction of $\mathcal{C}^*$ and $\mathbf{s}^*$.

Finding Center from Another Code. One thing that might not be clear so far is: where does the center $\mathbf{s}^*$ come from? Here we provide a systematic way to produce such an $\mathbf{s}^*$, by looking at another code that contains $\mathcal{C}^*$. More precisely, let $\mathcal{C}^* \subseteq \widetilde{\mathcal{C}}^* \subseteq \mathbb{F}_q^\ell$ be two linear codes with the same block length and alphabet. Suppose that the distance of $\mathcal{C}^*$ is Δ, the distance of $\widetilde{\mathcal{C}}^*$ is r^* and that $r^* < \Delta$. It is easy to see that, by taking $\mathbf{s}^*$ to be any element of $\widetilde{\mathcal{C}}^* \setminus \mathcal{C}^*$, it holds that every codeword in $\mathcal{C}^*$ is at distance at least r^* from $\mathbf{s}^*$, simply because $\mathbf{s}^*$ and any codeword of $\mathcal{C}^*$ are two distinct codewords of $\widetilde{\mathcal{C}}^*$.

Hence, we are only left to argue that there are many codewords of $\mathcal{C}^*$ that is of distance exactly r^* from $\mathbf{s}^*$. While this is not true in general, we can show by an averaging argument that this is true (for some $\mathbf{s}^* \in \widetilde{\mathcal{C}}^*$) if a large fraction (e.g. $|\mathcal{C}^*|^{-o(1)}$ fraction) of codewords of $\widetilde{\mathcal{C}}^*$ has Hamming weight exactly r^* (see Lemma 11.21).

Indeed, viewing in this light, our previous choice of center for Reed-Solomon code (i.e. evaluation of x^h) is not coincidental: we simply take $\widetilde{\mathcal{C}}^*$ to be another Reed-Solomon code with message length $h + 1$ (whereas the base code $\mathcal{C}^*$ is of message length h).

Comparison to Locally Dense Codes. We end this subsection by remarking that the codes that we seek are very similar to locally dense codes [DMS03; CW12b; Mic14], which is indeed our inspiration. A *locally dense code* is a linear code of block length ℓ and large minimum distance Δ, admitting a ball centered at $\mathbf{s}$ of radius[7] $r < \Delta$ and containing a large (i.e. $\exp(\text{poly}(\ell))$) number of codewords[8]. Such codes are non-trivial to construct and in particular all known constructions of locally dense codes are using codes that beat the Gilbert-Varshamov (GV) bound [Gil52; Var57]; in other words we need to do better than random codes to construct them. This is because (as noted in [DMS03]), for a random code $\mathcal{C} \subseteq \mathbb{F}_q^\ell$ (or any code that does not beat the GV bound), a random point in $\mathbb{F}_q^\ell$ acting as the center contains in expectation less than one codeword in a ball of radius Δ. Of course, this is simply an intuition and not a formal proof that a locally dense code needs to beat the GV bound, since there may be more sophisticated ways to pick a center.

Although the codes we require are similar to locally dense codes, there are differences between the two. Below we list four such differences: the first two makes it *harder* for us to construct our codes whereas the latter two makes it *easier* for us.

- We seek a center $\mathbf{s}^*$ so that no codewords in $\mathcal{C}^*$ lies at distance less than r^*, as opposed to locally dense codes which allows codewords to be close to $\mathbf{s}^*$. This is indeed where our idea of using another code $\widetilde{\mathcal{C}}^* \supseteq \mathcal{C}^*$ comes in, as picking $\mathbf{s}^*$ from $\widetilde{\mathcal{C}}^* \setminus \mathcal{C}^*$ ensures us that no codeword of $\mathcal{C}^*$ is too close to $\mathbf{s}^*$.

- Another difference is that we need the number of codewords at distance r^* from $\mathbf{s}^*$ to be very large, i.e., $|\mathcal{C}^*|^{1-o(1)}$, whereas locally dense codes allow for much smaller number of codewords. Indeed, the deterministic constructions from [CW12b; Mic14] only yield the bound of $2^{O(\sqrt{\log|\mathcal{C}^*|})}$. Hence, these do not directly work for us.

- Locally dense codes requires r to be at most $(1 - \varepsilon)\Delta$ for some constant $\varepsilon > 0$, whereas we are fine with any $r^* < \Delta$. In fact, our Reed-Solomon code based construction above only yields $r^* = \Delta - 1$ which would not suffice for locally dense codes. Nevertheless, as we will see later for inapproximability of CP, we will also need the ratio r^*/Δ to be a constant

[7] Clearly, for the ball to contain more than a single codeword, it must be $r \geqslant \Delta/2$. Here we are interested in balls with radius not much bigger than that, say $r < \gamma \cdot \Delta$ for some constant $1/2 < \gamma < 1$.

[8] Strictly speaking, a locally dense code also requires an auxiliary matrix $\mathbf{T}$ used to index these codewords. However, in previous works, finding $\mathbf{T}$ is typically not hard given the center $\mathbf{s}$. Hence, we ignore $\mathbf{T}$ in our discussion here for the ease of exposition.

bounded away from 1 as well and, since we need a code with these extraordinary properties, they are very hard to find. Indeed, in this case we only manage to prove a weaker lower bound on gap-CP.

- Finally, we remark that locally dense codes are required to be efficiently constructed in $\text{poly}(\log |\mathcal{C}^*|)$ time, which is part of why it is hard to find. Specifically, while [DMS03] shows that an averaging argument works for a random center, derandomizing this is a big issue and a few subsequent works are dedicated solely to this issue [CW12b; Mic14]. On the other hand, brute force search (over all codewords in $\widetilde{\mathcal{C}}^*$) suffices to find a center for us, as we are allowed construction time of $\text{poly}(|\mathcal{C}^*|)$.

11.1.3 Inapproximability of Closest Pair and Maximum Inner Product

In this subsection, we sketch our inapproximability results for MIP and CP. Both these results use the same reduction that we had from BCP to CP, except that we now need stronger properties from the gadget, i.e., the previously used notions of contact dimension does not suffice anymore. Below we sketch the required strengthening of the gadget properties and explain how to achieve them.

Approximate Maximum Inner Product

Observe that the gadget we construct for CP in Subsection 11.1.2 can also be written in terms of inner product as follows: there exists a dense balanced bipartite graph $G^* = (A^* \dot\cup B^*, E^*)$, a mapping $\tau : A^* \dot\cup B^* \to \{0,1\}^{q \cdot \ell}$ such that the following holds.

(i) For all edges $(a, b) \in E^*$, $\langle \tau(a), \tau(b) \rangle = \ell - r^*$.

(ii) For all edges $(a, b) \in (A^* \times B^*) \setminus E^*$, $\langle \tau(a), \tau(b) \rangle < \ell - r^*$.

(iii) For all distinct a, b both from A^* or both from B^*, $\langle \tau(a), \tau(b) \rangle \leqslant \ell - \Delta$.

Notice that we wrote the conditions above in a slightly different way than in previous subsections; previously in the contact dimension notation, (ii) and (iii) would be simply written together as: for all non-edge (a, b), $\langle \tau(a), \tau(b) \rangle < \ell - r^*$. This change is intentional, since, to get gap in our reductions, we only need a gap between the bounds in (i) and (iii) (but not in (ii)). In particular, to get hardness of approximating MIP, we require $\frac{\ell - r^*}{\ell - \Delta}$ to be at least $(1 + \varepsilon)$ for some $\varepsilon > 0$.

From our Reed-Solomon construction above, $\ell - \Delta$ and $\ell - r^*$ are exactly the message length of $\mathcal{C}^*$ minus one and the message length of $\widetilde{\mathcal{C}}^*$ minus one respectively. Previously, we selected these two to be h and $h + 1$. Now to obtain the desired gap, we simply take the larger code $\widetilde{\mathcal{C}}^*$ to be a Reed-Solomon code with larger (i.e. $(1 + \varepsilon)h$) message length[9].

[9] This approach can in fact give not just $(1 + \varepsilon)$ but arbitrarily large constant gap between the two cases. In the actual reduction, we take this gap to be 3 (Theorem 11.31), which makes some computations simpler.

Finally, we note that even with the above gadget, the reduction only gives a small (i.e. $1+o(1)$) factor hardness of approximating MIP (Theorem 11.31). To boost the gap to near polynomial, we simply tensor the vectors with themselves (see Section 11.5).

Approximate Closest Pair

Once again, recall that we have the following gadget from Subsection 11.1.2: there exists a dense balanced bipartite graph $G^* = (A^* \dot\cup B^*, E^*)$, a mapping $\tau : A^* \dot\cup B^* \to \{0,1\}^{q \cdot \ell}$ such that the following holds.

(i) For all edges $(a, b) \in E^*$, $\|\tau(a) - \tau(b)\|_0 = r^*$.

(ii) For all edges $(a, b) \in (A^* \times B^*) \setminus E^*$, $\|\tau(a) - \tau(b)\|_0 > r^*$.

(iii) For all distinct a, b both from A^* or both from B^*, $\|\tau(a) - \tau(b)\|_0 \geqslant \Delta$.

Once again, we need an $(1 + \varepsilon)$ gap between the bounds in (iii) and (i), i.e., $\frac{\Delta}{r^*}$. Unfortunately, we cannot construct such codes using any of the Reed-Solomon code families. We turn to another type of codes that beat the Gilbert-Varshamov bound: Algebraic- Geometric (AG) codes. Similar to the Reed-Solomon code based construction, we take $\mathcal{C}^*$ as an AG code and $\widetilde{\mathcal{C}}^*$ to be a "higher degree" AG code; getting the desired gap simply means that the distance of $\mathcal{C}^*$ must be at least $(1 + \varepsilon)$ times the distance of $\widetilde{\mathcal{C}}^*$.

Recall from Subsection 11.1.2 also that, to bound the density of G^*, we need a lower bound on the number of minimum weight codewords of $\widetilde{\mathcal{C}}^*$. Such bounds for AG codes are non-trivial and we turn to the bounds from [ABV01; Vlă18]. Unfortunately, this only gives G^* with density $|\mathcal{C}^*|^{-1/2-o(1)}$, instead of $|\mathcal{C}^*|^{-o(1)}$ as before. This is indeed the reason that our running time lower bound for approximate CP is only $n^{1.5-\varepsilon}$.

11.2 Additional Preliminaries

11.2.1 Singleton Bound

We will use the following standard bound from coding theory called the *Singleton bound*:

Theorem 11.6 (Singleton bound [Sin64])**.** *For any linear $[N, K, D]_q$ code, $K + D \leqslant N + 1$.*

11.2.2 Miscellaneous Tools

Covering Biclique by Isomorphic Graphs. A useful fact we use to derandomize our reductions is that the biclique can be covered by any dense bipartite graph G with only a few graphs that are isomorphic to G. To state this more formally, let us first define a few notions.

Definition 11.7. *For any graph $G = (V_G, E_G)$ and any permutation $\pi : V_G \to V_G$, we use G_π to denote the graph (V_{G_π}, E_{G_π}) where the vertex set V_{G_π} is equal to V_G and $E_{G_\pi} = \{(\pi(a), \pi(b)) \mid (a, b) \in E_G\}$.*

For brevity, we say that a permutation $\pi : A\dot{\cup}B \to A\dot{\cup}B$ of vertices of a bipartite graph $G = (A\dot{\cup}B, E_G)$ is *side-preserving* if $\pi(A) = A$ and $\pi(B) = B$.

We can now state the result as follows.

Lemma 11.8. *For any bipartite graph $G(A\dot{\cup}B, E_G)$ where $|A| = |B| = n$ and $E_G \neq \emptyset$, there exist side-preserving permutations $\pi_1, \dots, \pi_k : A \cup B \to A \cup B$ where $k \leqslant \frac{2n^2 \ln n}{|E_G|} + 1$ such that*

$$\bigcup_{i\in[k]} E_{G_{\pi_i}} = E_{K_{n,n}}$$

Moreover, such permutations can be found in time $O(n^6 \log n)$.

The proof strategy for Lemma 11.8 is similar to how the greedy approximation algorithms for the set cover problem are analyzed: we show that at each step, we can pick a graph isomorphic to G that covers at least $|E_G|/n^2$ fraction of the remaining edges of the biclique. By doing so, we guarantee that the process ends in $O(\log n) \cdot n^2/|E_G|$ steps. Note however that, there are exponential number of isomorphisms and thus we cannot simply enumerate all isomorphisms to find one that covers the desired fraction of uncovered edges. Nevertheless, it is not hard to see that we can use the method of conditional expectation to find one such isomorphism in polynomial time. This is formalized below.

Lemma 11.9. *For any two bipartite graphs $G = (A\dot{\cup}B, E_G)$ and $H = (A\dot{\cup}B, E_H)$, there exists a side-preserving permutation $\pi : A\dot{\cup}B \to A\dot{\cup}B$ such that*

$$|E_H \cap E_{G_\pi}| \geqslant \frac{|E_G| \cdot |E_H|}{|A| \cdot |B|}.$$

Moreover, such a permutation π can be found (deterministically) in $O((|A| + |B|)^4)$ time.

Proof. Notice that, if we pick $\pi|_A$ and $\pi|_B$ randomly among all permutations of A and B respectively, then, for a fixed $(a, b) \in E_H$, the probability that (a, b) belongs to E_{G_π} is $\frac{|E_G|}{|A|\cdot|B|}$. Thus,

$$\mathbb{E}_\pi\left[|E_H \cap E_{G_\pi}|\right] = \frac{|E_G| \cdot |E_H|}{|A| \cdot |B|}.$$

This proves the existence part of the claim. To deterministically find such a π, we use the method of conditional expectation. Suppose $A\dot{\cup}B = \{1, \dots, n\}$. The algorithm works as follows:

1. Let $V_{\text{assigned}} \leftarrow \emptyset$.
2. For $i = 1, \dots, n$:
   a) If $i \in A$, let $V_{\text{candidate}} = A \setminus V_{\text{assigned}}$. Otherwise, if $i \in B$, let $V_{\text{candidate}} = B \setminus V_{\text{assigned}}$.
   b) For each $k \in V_{\text{candidate}}$, compute the conditional expectation:

$$\mathbb{E}_\pi\left[|E_H \cap E_{G_\pi}| \;\middle|\; \pi(i) = k \wedge \left(\bigwedge_{j=1}^{i-1} \pi(j) = \pi^*(j)\right)\right].$$

   Let k^* be the maximizer for the above conditional expectation. We set $\pi^*(i) = k^*$.

3. Output π^*.

It is simple to see that the conditional expectation never decreases as we fill in the permutation. As a result, we must have $|E_H \cap E_{G_\pi}| \geqslant \frac{|E_G| \cdot |E_H|}{|A| \cdot |B|}$ as desired. Moreover, it is easy to see that the conditional expectation can be computed in time $O(|A| \cdot |B|)$ because, for each edge $(a, b) \in E_H$, we can compute the probability that $(a, b) \in E_{G_\pi}$ in $O(1)$ time. As a result, the overall running time of the algorithm is $O((|A| + |B|)^4)$. □

Finally using Lemma 11.9, we prove Lemma 11.8 using the strategy outlined earlier in this section.

Proof of Lemma 11.8. We describe below an algorithm for finding $\pi_1, \dots, \pi_k$. It works as follows.

1. Let $k \leftarrow 0$.
2. While $E_H := E_{K_{n,n}} \setminus \bigcup_{i \in [k]} E_{G_{\pi_i}}$ is non-empty, do the following:
   a) Let $k \leftarrow k + 1$.
   b) Let $H = (A \dot\cup B, E_H)$.
   c) Use the algorithm from Lemma 11.9 to find π_k such that $|E_H \cap E_{G_{\pi_k}}| \geqslant |E_H| \cdot \frac{|E_G|}{n^2}$.
3. Output $\pi_1, \dots, \pi_k$.

It is obvious that the permutations are all side-preserving permutations and that the union of $E_{G_{\pi_i}}$ over $i \in [k]$ is equal to $E_{K_{n,n}}$. To see that $k \leqslant \frac{2n^2 \ln n}{|E_G|} + 1$, observe that due to the guarantee of Lemma 11.9, $|E_H|$ decreases by a multiplicative factor of (at most) $(1 - |E_G|/n^2) \leqslant e^{-|E_G|/n^2}$ for each permutation picked. Since the set E_H remains non-empty after $k - 1$ permutations are picked, we have $e^{-(k-1) \cdot |E_G|/n^2} \cdot n^2 \geqslant 1$, which implies that $k \leqslant 2n^2 \ln n / |E_G| + 1$ as desired. Finally, the bottleneck in the running time is Step 2c; we execute this step k times and each execution takes $O(n^4)$ time. Thus, the total running time is $O(nk) = O(n^6 \log n)$. □

Translating Finite Fields Vectors to {0, 1}-Vectors. Another simple fact which was already mentioned in the proof overview (Section 11.1) is that, we can embed Hamming metric on alphabet of size q to Hamming metric on Boolean alphabet, with only q multiplicative factor blow-up in the dimension:

Proposition 11.10. *For any $q, N \in \mathbb{N}$, and alphabet Σ such that $|\Sigma| = q$, there exists a mapping $\psi : \Sigma^N \to \{0, 1\}^{q \cdot N}$ such that, for all $\mathbf{v}_1, \mathbf{v}_2 \in \Sigma^N$, we have $\|\psi(\mathbf{v}_1) - \psi(\mathbf{v}_2)\|_0 = 2 \cdot \Delta(\mathbf{v}_1, \mathbf{v}_2)$ and $\langle \psi(\mathbf{v}_1), \psi(\mathbf{v}_2) \rangle = N - \Delta(\mathbf{v}_1, \mathbf{v}_2)$.*

Proof. The mapping ψ simply replaces each coordinate that is equal to $j \in \Sigma$ by the j-th standard basis in the q-dimensional space. More precisely, for $\mathbf{v} = (v_1, \dots, v_N) \in \mathbb{F}_q$, we define

$$\psi(\mathbf{v}) = e_{v_1} \circ e_{v_2} \circ \cdots \circ e_{v_N},$$

where $\circ$ denotes concatenation of vectors and e_j denote the j-th standard basis in $\mathbb{R}^q$, i.e., the vector whose j-th coordinate is one and the remaining coordinates are zeroes.

It is simple to check that this satisfies the two requirements. $\square$

11.2.3 OVH-hardness of Exact Bichromatic Closest Pair

Alman and Williams [AW15] showed the conditional hardness (under OVH) of exact BCP in every ℓ_p-metric even when the point-sets are over $\{0,1\}$ via a Turing reduction from OV. David, Karthik, and Laekhanukit [DKL18] gave an alternate proof of the same result where point-sets were over $\mathbb{R}$ via a many-one reduction from OV. For independent interest, below we give an alternative proof, which is both a many-one reduction and the point-sets are over $\{0,1\}$.

Theorem 11.11. *Assuming* OVH*, for every $\varepsilon > 0$, no algorithm running in time $n^{2-\varepsilon}$ can solve* BCP*, even when the point-sets A, B are subsets of $\{0,1\}^d$ and $d = c_\varepsilon \log n$, for some constant $c_\varepsilon > 1$ (only depending on ε).*

Proof. Let $A, B \subseteq \{0,1\}^d$ where $|A| = |B| = n$ be the input to an OV instance. We build an instance (A', B', α) of BCP where $A', B' \subseteq \{0,1\}^{5d}$, $|A| = |B| = n$, and $\alpha = 2d$, using functions T_A and T_B guaranteed by the following claim.

Claim 11.12. *There are functions $T_A, T_B : \{0,1\} \to \{0,1\}^5$ such that for every $x, y \in \{0,1\}$ we have:*

- *$x \cdot y = 0$ implies $\|T_A(x) - T_B(y)\|_0 = 2$.*
- *$x \cdot y = 1$ implies $\|T_A(x) - T_B(y)\|_0 = 4$.*

For every $i \in [n]$, the i^{th} point of A', say a' is constructed from the i^{th} point of A, say a by simply applying T_A pointwise on each coordinate of a, i.e., $a' = (T_A(a_1), \ldots, T_A(a_d))$. Similarly we apply T_B pointwise on each coordinate of points in B. It is easy to see that there exists $(a'_i, b'_j) \in A' \times B'$ such that $\|a'_i - b'_j\|_0 = 2d$ if and only if $\langle a_i, b_j \rangle = 0$, and otherwise every pair of points in $A' \times B'$ is at Hamming distance at least $2d + 2$. $\square$

Proof of Claim 11.12. We define for all $x, y \in \{0,1\}$, $T_A(x) = (T_A(x)_{0,0}, T_A(x)_{0,1}, T_A(x)_{1,0}, x, 0)$ and $T_B(y) = (T_B(y)_{0,0}, T_B(y)_{0,1}, T_B(y)_{1,0}, 0, y)$, where for all $i, j \in \{0,1\}$ such that $i \cdot j = 0$, we have $T_A(x)_{i,j} = 1$ if and only if $x = i$ and $T_B(y)_{i,j} = 1$ if and only if $y = j$. More succinctly, T_A and T_B are described below as strings and the claim follows by a straight-forward calculation.

$$T_A(0) = 11000 \qquad T_A(1) = 00110 T_B(0) = 10100 \qquad T_B(1) = 01001 \qquad \square$$

11.2.4 Contact Dimension of a Graph

The central gadget in our reduction from BCP to CP is based on the contact dimension of a graph. Below we reproduce its definition from the proof overview (i.e. Definition 11.5) for convenience.

Definition 11.13 (Contact Dimension [Pac80])**.** *For any graph $G = (V, E)$, a mapping $\tau : V \to \mathbb{R}^d$ is said to* realize G (in the ℓ_p-metric) *if for some $\beta > 0$, the following holds:*

(i) For all $(u, v) \in E$, $\|\tau(u) - \tau(v)\|_p = \beta$.

(ii) For all $(u, v) \notin E$, $\|\tau(u) - \tau(v)\|_p > \beta$.

The contact dimension *(in the ℓ_p-metric) of G, denoted by $\mathsf{cd}_p(G)$, is the minimum $d \in \mathbb{N}$ such that there exists $\tau : V \to \mathbb{R}^d$ realizing G in the ℓ_p-metric.*

We may also say that τ *β-realizes G* if we wishes to emphasize the value of β.

Note here that we may view points in $\tau(V)$ as centers of spheres of radius $\beta/2$. No two spheres overlap but they may touch, and G has an edge (u, v) if and only if the spheres centered at $\tau(u)$ and $\tau(v)$ touches.

For a summary of the bounds on $\mathsf{cd}(G)$ for various graphs in the Euclidean metric see [Mae85; FM86; FM88; Mae91] and for a summary of the bounds on $\mathsf{cd}(K_{n,n})$ in various metrics see [DKL18]. For this chapter, the following bounds are relevant.

Theorem 11.14 (Frankl-Maehara [FM88])**.** $(1.286)n - 1 < \mathsf{cd}_2(K_{n,n}) < (1.5)n.$

Theorem 11.15 (David-Karthik-Laekhanukit [DKL18])**.** $\mathsf{cd}_0(K_{n,n}) = n.$

In particular, the above two theorems are the obstacles of the approach of [DKL18] for the ℓ_2 and Hamming metrics respectively. As discussed in the proof overview, we will overcome these barriers by constructing dense bipartite graphs with low contact dimensions in every ℓ_p metrics.

As discussed in Section 11.1.3, we need a generalization of contact dimension in order to show inapproximability for CP. This is formally defined below; it should be noted that the definition only makes sense for bipartite graphs, whereas the original contact dimension is well-defined for any graphs. Moreover, when $\lambda = 1$, the notion of gap contact dimension coincides with the (non-gap) contact dimension in bipartite graphs.

Definition 11.16 (Gap Contact Dimension)**.** *For any bipartite graph $G = (A \dot\cup B, E)$ and $\lambda \geqslant 1$, a mapping $\tau : V \to \mathbb{R}^d$ is said to* λ-gap-realize G (in the ℓ_p-metric) *if for some $\beta > 0$, the following holds:*

(i) For all $(u, v) \in E$, $\|\tau(u) - \tau(v)\|_p = \beta$.

(ii) For all $(u, v) \in (A \times B) \setminus E$, $\|\tau(u) - \tau(v)\|_p > \beta$.

(iii) For all distinct u, v both from A or both from B, $\|\tau(u) - \tau(v)\|_p > \lambda \cdot \beta$.

The λ-gap contact dimension *(in the ℓ_p-metric) of G, denoted by λ-$\mathsf{cd}_p(G)$, is the minimum $d \in \mathbb{N}$ such that there exists $\tau : V \to \mathbb{R}^d$ λ-gap-realizing G in the ℓ_p-metric.*

Again, we may say that τ (β, λ)-gap-realizes G to emphasize the value of β.

Finally, we define an analogous notion for inner product:

Definition 11.17 (Gap Inner Product Dimension)**.** *For any bipartite graph $G = (A\dot{\cup}B, E)$ and $\lambda \geqslant 1$, a mapping $\tau : V \to \mathbb{R}^d$ is said to* λ-gap-IP-realize G *if for some $\beta > 0$, the following holds:*

(i) For all $(u, v) \in E$, $\langle\tau(u), \tau(v)\rangle = \beta$.

(ii) For all $(u, v) \in (A \times B) \setminus E$, $\langle\tau(u), \tau(v)\rangle < \beta$.

(iii) For all distinct u, v both from A or both from B, $\langle\tau(u), \tau(v)\rangle < \beta/\lambda$.

The λ-gap inner product dimension *of G, denoted by λ-$\mathsf{ipd}(G)$, is the minimum $d \in \mathbb{N}$ such that there exists $\tau : V \to \mathbb{R}^d$ λ-gap-*IP*-realizing G.*

We may say that τ (β, λ)-gap-IP-realizes G to emphasize the value of β.

11.3 Lower Bound on (Exact) Closest Pair under OVH

In this section, we prove the subquadratic hardness for CP (assuming OVH) using the efficient construction of a realization of a dense bipartite graph. The construction will be be formally stated below and the details will be given in Section 11.4.2. First, we define the notion of a *log-dense* sequence of integers:

Definition 11.18. *A sequence $(n_i)_{i\in\mathbb{N}}$ of increasing positive integers is said to be* log-dense *if there exists a constant $C \geqslant 1$ such that $\log n_{i+1} \leqslant C \cdot \log n_i$ for all $i \in \mathbb{N}$.*

As outlined in Section 11.1.1 , we use Reed-Solomon codes to construct a family of dense bipartite graphs with low contact dimensions. While the construction does not yield a graph for every number of vertices n, it does yield a graph for a log-dense sequence of numbers of vertices, which turns out to be sufficient for the purpose of the reduction. More formally, we will prove the following in Section 11.4.2.

Theorem 11.19. *For every $0 < \delta < 1$, there exists a log-dense sequence $(n_i)_{i\in\mathbb{N}}$ such that, for every $i \in \mathbb{N}$, there is a bipartite graph $G_i = (A_i\dot{\cup}B_i, E_i)$ where $|A_i| = |B_i| = n_i$ and $|E_i| \geqslant \Omega(n_i^{2-\delta})$, such that $\mathsf{cd}(G_i) = (\log n_i)^{O(1/\delta)}$. Moreover, for all $i \in \mathbb{N}$, a realization $\tau : A_i\dot{\cup}B_i \to \{0,1\}^{(\log n_i)^{O(1/\delta)}}$ of G_i can be constructed in time $n_i^{2+o(1)}$.*

Notice that we did not specify any ℓ_p-metric in the notion of contact dimension above. This is intentional, because our point sets $\tau(A_i\dot{\cup}B_i)$ have coordinate entries in $\{0, 1\}$, for which the distances in the Hamming metric are equivalent (up to power of p) to distances in any ℓ_p-metric ($p \neq \infty$). We also adopt this notational convenience below. Specifically, we will prove the following theorem which states that CP is hard even when the points are from $\{0, 1\}^d$; clearly, this also implies Theorem 11.1 due to the aforementioned equivalence to other ℓ_p-metrics.

Theorem 11.20 (Subquadratic Hardness of $\{0,1\}$-CP)**.** *Assuming* OVH*, for every $\varepsilon > 0$, there exists $s_\varepsilon > 0$ such that no algorithm running in $O(n^{2-\varepsilon})$ time can solve* CP *in the Hamming metric even when $d = (\log n)^{s_\varepsilon}$ and all points have $\{0,1\}$ entries.*

Proof. For any $\varepsilon > 0$, let C_{exp} be the constant such that the dimension guarantee for τ in Theorem 11.19 is at most $(\log n_i)^{C_{\text{exp}}/\varepsilon}$ for $\delta = \varepsilon/2$. We define s_ε as $2 \cdot C_{\text{exp}}/\varepsilon + 2$.

Assume that there exists $\varepsilon > 0$ and an algorithm $\mathcal{A}$ that can solve CP in time $n^{2-\varepsilon}$ in the Hamming metric for any input of n points in $\{0,1\}^{(\log n)^{s_\varepsilon}}$. We will construct an algorithm $\mathcal{A}'$ that solves any instance of BCP in time $n^{2-\varepsilon'}$ for some constant $\varepsilon' > 0$ (to be specified below), on n points in dimension $d := c_{\varepsilon'} \cdot \log n$ with coordinate entries in $\{0,1\}$. Together with Theorem 11.11, this implies that OVH is false, arriving at a contradiction.

Let C_ε denote the log-density constant (i.e. $\sup\limits_i \frac{\log n_{i+1}}{\log n_i}$) of the sequence from Theorem 11.19 for $\delta = \varepsilon/2$, and let ε' be $0.01 \cdot \varepsilon/C_\varepsilon$. The algorithm $\mathcal{A}'$ on input (A, B, α) where $A, B \subseteq \{0,1\}^d$, with $|A| = |B| = n$, and $\alpha \in [d]$, works as follows:

1. Let n' be the largest number in the sequence from Theorem 11.19 with $\delta = \varepsilon/2$ s.t. $n' \leqslant n^{0.1}$.
2. Let $G' = (A'\dot\cup B', E')$ be the graph from Theorem 11.19 with $|A'| = |B'| = n'$, $|E'| \geqslant \Omega((n')^{2-\delta})$, and $\tau : A'\dot\cup B' \to \{0,1\}^{(\log n')^{C_{\text{exp}}/\varepsilon}}$ be a β-realization of G' where $\beta \in \mathbb{N}$.
3. We use the algorithm from Lemma 11.8 to find $\pi_1, \ldots, \pi_k$ where $k = O((n')^\delta \log n')$ such that the union of $E_{G'_{\pi_1}}, \ldots, E_{G'_{\pi_k}}$ is $E_{K_{n',n'}}$.
4. We assume w.l.o.g.[10] that n is divisible by n'. Partition A and B into $A_1, \ldots, A_{n/n'}$ and $B_1, \ldots, B_{n/n'}$ each of size n'. For each $i, j \in [n/n'], t \in [k]$, do the following:
    a) Let τ_t be an appropriate permutation of τ that β-realizes G'_{π_t}. Label the vertices of G'_{π_t} with the points in $A_i \dot\cup B_j$.
    b) Let $\alpha' = \alpha + (d+1) \cdot \beta$, and define A_i^t, B_j^t as
    $$A_i^t = \{\mathbf{a} \circ (\mathbf{1}_{d+1} \otimes \tau_t(\mathbf{a})) \mid \mathbf{a} \in A_i\}, B_j^t = \{\mathbf{b} \circ (\mathbf{1}_{d+1} \otimes \tau_t(\mathbf{b})) \mid \mathbf{b} \in B_j\}$$
    where $\mathbf{1}_{d+1} \otimes \mathbf{v}$ simply denotes $\mathbf{v} \circ \mathbf{v} \circ \cdots \circ \mathbf{v}$, i.e., the concatenation of $d+1$ copies of $\mathbf{v}$.
    c) Run $\mathcal{A}$ on $(A_i^t \dot\cup B_j^t, \alpha')$. If $\mathcal{A}$ outputs YES, then output YES and terminate.
5. If none of the executions of $\mathcal{A}$ returns YES, then output NO.

Observe that the bottleneck in the running time of the algorithm is in the executions of $\mathcal{A}$. The number of executions is $(n/n')^2 \cdot k$ and each execution takes $O((n')^{2-\varepsilon})$ time. Hence, in total the

[10] This is without loss of generality, since if n is not divisible by n', we can use brute force for the remainder points. This requires only $O(n \cdot n' \cdot \log n) = O(n^{1.1} \log n)$ which does not affect the overall asymptotic running time of the algorithm.

running time of the algorithm $\mathcal{A}'$ is $O((n/n')^2 \cdot k \cdot (n')^{2-\varepsilon}) \leqslant O(n^2 \log n \cdot (n')^{-\varepsilon/2})$. Now, from the log-density of the sequence from Theorem 11.19, we have $n' \geqslant n^{0.1/C_\varepsilon} = n^{10\varepsilon'/\varepsilon}$. As a result, the running time of $\mathcal{A}$ is at most $O(n^{2-5\varepsilon'} \log n) \leqslant O(n^{2-\varepsilon'})$ as desired.

To see the correctness of the algorithm, first observe that the dimensions of vectors in A_i^t, B_j^t are at most $d + (d+1) \cdot (\log n')^{C_{\exp}/\varepsilon}$ which is at most $(\log n)^{s_\varepsilon}$ for any sufficiently large n; that is, the calls to $\mathcal{A}$ are valid. Next, observe that, if (A, B, α) is a YES instance of BCP, there must be $i, j \in [n/n']$ and $\mathbf{a}^* \in A_i, \mathbf{b}^* \in B_j$ such that $\|\mathbf{a}^* - \mathbf{b}^*\|_0$ is at most α. Since $G'_{\pi_1}, \dots, G'_{\pi_k}$ covers $K_{n',n'}$, there must be $t \in [k]$ such that $\|\tau_t(\mathbf{a}^*) - \tau_t(\mathbf{b}^*)\|_0 = \beta$. As a result, $\|(\mathbf{a}^* \circ (\mathbf{1}_{d+1} \otimes \tau_t(\mathbf{a}^*))) - (\mathbf{b}^* \circ (\mathbf{1}_{d+1} \otimes \tau_t(\mathbf{b}^*)))\|_0 \leqslant \alpha + (d+1) \cdot \beta = \alpha'$. Thus, $(A_i^t \cup B_j^t, \alpha')$ is a YES instance for CP and $\mathcal{A}'$ outputs YES as desired.

Finally, assume that (A, B, α) is a NO instance of BCP. Consider any $i, j \in [n/n']$ and $t \in [k]$. To argue that $(A_i^t \cup B_j^t, \alpha')$ is a NO instance for CP, we have to show that any two points in $A_i^t \cup B_j^t$ have distance more than α'. To see this, let us consider two cases.

1. Both points are either from A_i^t or from B_j^t. Assume w.l.o.g. that the two points are from A_i^t; let them be $\mathbf{a} \circ (\mathbf{1}_{d+1} \otimes \tau_t(\mathbf{a}))$ and $\mathbf{a}' \circ (\mathbf{1}_{d+1} \otimes \tau_t(\mathbf{a}'))$. Recall that, from the definition of β-realization, $\|\tau_t(\mathbf{a}) - \tau_t(\mathbf{a}')\|_0 > \beta$. Since $\|\tau_t(\mathbf{a}) - \tau_t(\mathbf{a}')\|_0$ is an integer, we must have $\|\tau_t(\mathbf{a}) - \tau_t(\mathbf{a}')\|_0 \geqslant \beta + 1$. As a result, the Hamming distance between the two points is at least $(d+1) \cdot (\beta + 1) > d + (d+1) \cdot \beta = \alpha'$.

2. One of the point is from A_i^t and the other from B_j^t. Let them be $\mathbf{a} \circ (\mathbf{1}_{d+1} \otimes \tau_t(\mathbf{a}))$ and $\mathbf{b} \circ (\mathbf{1}_{d+1} \otimes \tau_t(\mathbf{b}))$. Since (A, B, α) is a NO instance of BCP, $\|\mathbf{a} - \mathbf{b}\|_0 > \alpha$. Furthermore, from definition of β-realization, we must have $\|\tau_t(\mathbf{a}) - \tau_t(\mathbf{b})\|_0 \geqslant \beta$. Combining the two implies that the Hamming distance between $\mathbf{a} \circ (\mathbf{1}_{d+1} \otimes \tau_t(\mathbf{a}))$ and $\mathbf{b} \circ (\mathbf{1}_{d+1} \otimes \tau_t(\mathbf{b}))$ is more than α'.

Hence, $(A_i^t \dot\cup B_j^t, \alpha')$ must be a NO instance for CP for every $t \in [k]$ and $i, j \in [n/n']$. Thus, $\mathcal{A}'$ outputs NO as desired. □

11.4 Gadget Constructions

In this section, we construct all the gadgets that are used in our reductions, including the basic gadget (Theorem 11.19) and more advanced gadgets used for MIP and approximate version of CP.

11.4.1 Finding a Center of a Code via Another Code

At the heart of all our gadgets is the task of finding a code $\mathcal{C}_1$ and a center s such that there are $|\mathcal{C}_1|^{1-o(1)}$ many codewords at Hamming distance exactly equal to r (for some $r > 0$) from s but there is no codeword in $\mathcal{C}_1$ at distance less than r from s. The below lemma is useful in finding such an s.

Lemma 11.21. *Let $\mathcal{C}_1 \subseteq \mathcal{C}_2 \subseteq \mathbb{F}_q^N$ be two linear codes with the same block length N and alphabet $\mathbb{F}_q$ such that $\Delta(\mathcal{C}_2) < \Delta(\mathcal{C}_1)$. Then, there exists a center $\mathrm{s} \in \mathbb{F}_q^N$ such that (1) $\Delta(\mathrm{s}, \mathcal{C}_1) \geqslant \Delta(\mathcal{C}_2)$ and (2) $|\mathcal{B}(\mathrm{s}, \Delta(\mathcal{C}_2)) \cap \mathcal{C}_1|/|\mathcal{C}_1| \geqslant A_{\Delta(\mathcal{C}_2)}(\mathcal{C}_2)/|\mathcal{C}_2|$. Moreover, given $\mathcal{C}_1, \mathcal{C}_2$, such an s can be found in $O(|\mathcal{C}_1| \cdot |\mathcal{C}_2| \cdot qN)$ time.*

Proof. We show that there exists $\mathrm{s} \in \mathcal{C}_2 \setminus \mathcal{C}_1$ such that (2) holds. Note that (1) immediately holds, because $\mathrm{s} - \mathbf{c}$ must be a non-zero codeword of $\mathcal{C}_2$ which implies that $\Delta(\mathrm{s}, \mathbf{c}) \geqslant \Delta(\mathcal{C}_2)$.

To show that there exists $\mathrm{s} \in \mathcal{C}_2 \setminus \mathcal{C}_1$ such that $|\mathcal{B}(\mathrm{s}, \Delta(\mathcal{C}_2)) \cap \mathcal{C}_1| \geqslant |\mathcal{C}_1| \cdot A_{\Delta(\mathcal{C}_2)}/|\mathcal{C}_2|$. We will in fact show a stronger statement: for a random $\mathrm{s} \in \mathcal{C}_2 \setminus \mathcal{C}_1$, we have $\mathbb{E}[|\mathcal{B}(\mathrm{s}, \Delta(\mathcal{C}_2)) \cap \mathcal{C}_1|] \geqslant |\mathcal{C}_1| \cdot A_{\Delta(\mathcal{C}_2)}/|\mathcal{C}_2|$. Consider $\mathbb{E}_{\mathbf{s} \in \mathcal{C}_2 \setminus \mathcal{C}_1}[|\mathcal{B}(\mathrm{s}, \Delta(\mathcal{C}_2)) \cap \mathcal{C}_1|]$. Due to linearity of expectation, we have

$$\begin{aligned}
\mathbb{E}_{\mathbf{s} \in \mathcal{C}_2 \setminus \mathcal{C}_1}[|\mathcal{B}(\mathrm{s}, \Delta(\mathcal{C}_2)) \cap \mathcal{C}_1|] &= \sum_{\mathbf{c} \in \mathcal{C}_1} \Pr_{\mathrm{s} \in \mathcal{C}_2 \setminus \mathcal{C}_1}[\mathbf{c} \in \mathcal{B}(\mathrm{s}, \Delta(\mathcal{C}_2))] \\
&= \sum_{\mathbf{c} \in \mathcal{C}_1} \Pr_{\mathrm{s} \in \mathcal{C}_2 \setminus \mathcal{C}_1}[\Delta(\mathrm{s} - \mathbf{c}) \leqslant \Delta(\mathcal{C}_2)] \\
&= \sum_{\mathbf{c} \in \mathcal{C}_1} \Pr_{\mathrm{s} \in \mathcal{C}_2 \setminus \mathcal{C}_1}[\Delta(\mathrm{s}) \leqslant \Delta(\mathcal{C}_2)] \\
&= |\mathcal{C}_1| \cdot \frac{|(\mathcal{C}_2 \setminus \mathcal{C}_1) \cap \mathcal{B}(\mathbf{0}, \Delta(\mathcal{C}_2))|}{|\mathcal{C}_2 \setminus \mathcal{C}_1|}.
\end{aligned}$$

Now, since $\Delta(\mathcal{C}_1) > \Delta(\mathcal{C}_2)$, we have $\mathcal{C}_1 \cap \mathcal{B}(\mathbf{0}, \Delta(\mathcal{C}_2)) = \{\mathbf{0}\}$. That is, $|(\mathcal{C}_2 \setminus \mathcal{C}_1) \cap \mathcal{B}(\mathbf{0}, \Delta(\mathcal{C}_2))| = |(\mathcal{C}_2 \setminus \{\mathbf{0}\}) \cap \mathcal{B}(\mathbf{0}, \Delta(\mathcal{C}_2))| = A_{\Delta(\mathcal{C}_2)}(\mathcal{C}_2)$. Plugging this back into the above equality, we have

$$\mathbb{E}_{\mathbf{s} \in \mathcal{C}_2 \setminus \mathcal{C}_1}[|\mathcal{B}(\mathrm{s}, \Delta(\mathcal{C}_2)) \cap \mathcal{C}_1|] = |\mathcal{C}_1| \cdot \frac{A_{\Delta(\mathcal{C}_2)}(\mathcal{C}_2)}{|\mathcal{C}_2 \setminus \mathcal{C}_1|} \geqslant |\mathcal{C}_1| \cdot \frac{A_{\Delta(\mathcal{C}_2)}(\mathcal{C}_2)}{|\mathcal{C}_2|}.$$

Thus, there must exist a center $\mathrm{s} \in \mathcal{C}_2 \setminus \mathcal{C}_1$ that satisfies (2) (and also (1)) as desired.

Finally, note that s can be found by a brute force algorithm that tries every $\mathrm{s} \in \mathcal{C}_2$ and check whether (2) is satisfied; this algorithm takes $O(|\mathcal{C}_1| \cdot |\mathcal{C}_2| \cdot qN)$ time. □

11.4.2 Gadgets based on Reed-Solomon Codes

In this subsection, we construct gadgets based on the Reed Solomon codes, which are defined below.

Theorem 11.22 (Reed-Solomon Codes)**.** *For every prime power q, and every $K \leqslant N \leqslant q$, there exists a $[N, K, N-K+1]_q$ linear code, denoted by $\mathsf{RS}_q[N, K]$. The generator matrix of this code can be computed in time poly(N, K, q). Moreover, for every $q \geqslant N \geqslant K_2 > K_1$, we have $\mathsf{RS}_q[N, K_1] \subseteq \mathsf{RS}_q[N, K_2]$.*

In order to find a good center s, we use the following (well-known) bound on the number of minimum weight codewords of Reed Solomon codes (and more generally MDS codes). For a reference of this bound, see e.g. [MS77, Ch. 11, Theorem 6].

Lemma 11.23. *Let $\mathcal{C}$ be any linear $[N, K, D]_q$ code that is MDS. Then, $A_D(\mathcal{C}) = \binom{N}{K-1} \cdot (q-1)$.*

The Basic Gadget: Dense Bipartite Graphs with Low Contact Dimensions

Now we construct a dense bipartite graph with low contact dimension. A proof sketch of this construction was provided in Section 11.1.1 and was formally stated as Theorem 11.19.

Proof of Theorem 11.19. Let q_i be the i-th prime number and let $n_i = (q_i)^{(\lfloor q_i^\delta \rfloor)}$; it is simple to see that the sequence $(n_i)_{i \in \mathbb{N}}$ is log-dense. For $q = q_i$, consider the Reed-Solomon codes $\mathcal{C}_1 = \mathsf{RS}_q[q, K_1]$ and $\mathcal{C}_2 = \mathsf{RS}_q[q, K_2]$ where $K_1 = \lfloor q^\delta \rfloor$ and $K_2 = K_1 + 1$. Applying Lemma 11.21 with $(\mathcal{C}_1, \mathcal{C}_2)$ implies that there exists a center $\mathrm{s} \in \mathcal{C}_2$ such that

$$\begin{aligned}
\frac{|\mathcal{B}(\mathbf{s}, \Delta(\mathcal{C}_2)) \cap \mathcal{C}_1|}{|\mathcal{C}_1|} &\geqslant \frac{A_{\Delta(\mathcal{C}_2)}}{|\mathcal{C}_2|} \\
\text{(By Lemma 11.23)} &= \frac{\binom{q}{K_2-1} \cdot (q-1)}{q^{K_2}} \\
&\geqslant \frac{\left(\frac{q}{K_2-1}\right)^{K_2-1} \cdot (q-1)}{q^{K_2}} \\
&= \frac{q-1}{q} \cdot \left(\frac{1}{K_2-1}\right)^{K_2-1} \\
&= \frac{q-1}{q} \cdot \frac{1}{K_1^{K_1}} \\
&\geqslant \frac{1}{2} \cdot \frac{1}{q^{\delta K_1}} \\
&= \Omega(|\mathcal{C}_1|^{-\delta}),
\end{aligned}$$

where the last equality follows from the fact that $|\mathcal{C}_1| = q^{K_1}$.

We construct the graph $G_i = (A_i, B_i, E_i)$ and a realization τ as follows. Let $A_i = \mathcal{C}_1, B_i = \{\mathbf{s} + \mathbf{c} \mid \mathbf{c} \in \mathcal{C}_1\}$ and $E_i = \{(\mathbf{a}, \mathbf{b}) \in A_i \times B_i \mid \Delta(\mathbf{a}, \mathbf{b}) = \Delta(\mathcal{C}_2)\}$. G_i can be easily realized by applying the mapping $\psi : \mathbb{F}_q^q \to \{0,1\}^{q^2}$ from Proposition 11.10. More precisely, let τ be the restriction of ψ on $A_i \cup B_i$. Below we argue about the density of G_i and that τ is a $2\Delta(\mathcal{C}_2)$-realization of G_i.

- First, notice that $|E_i|$ is exactly $|\mathcal{C}_1| \cdot |\mathcal{B}(\mathbf{s}, \Delta(\mathcal{C}_2)) \cap \mathcal{C}_1| \geqslant \Omega(|\mathcal{C}_1|^{2-\delta}) = \Omega(n_i^{2-\delta})$.
- Second, notice that, for every $\mathbf{v}_1, \mathbf{v}_2$ both from A_i or both from B_i, we have $\mathbf{v}_1 - \mathbf{v}_2 \in \mathcal{C}_1 \setminus \{\mathbf{0}\}$. This implies that $\|\tau(\mathbf{v}_1) - \tau(\mathbf{v}_2)\|_0 = 2\Delta(\mathbf{v}_1, \mathbf{v}_2) \geqslant 2\Delta(\mathcal{C}_1) > 2\Delta(\mathcal{C}_2)$.
- Third, for every $\mathbf{a} \in A_i$ and $\mathbf{b} \in B_i$, we have $\mathbf{a} - \mathbf{b} \in \mathcal{C}_2 \setminus \{\mathbf{0}\}$. Thus, $\Delta(\mathbf{a}, \mathbf{b}) \geqslant \Delta(\mathcal{C}_2)$. Hence, $\|\tau(\mathbf{a}) - \tau(\mathbf{b})\|_0 = 2\Delta(\mathbf{a}, \mathbf{b}) \geqslant 2\Delta(\mathcal{C}_2)$. Moreover, the inequality is an equality if and only if $\Delta(\mathbf{a}, \mathbf{b}) = \Delta(\mathcal{C}_2)$, i.e., $(\mathbf{a}, \mathbf{b}) \in E_i$ as desired.
- Finally, observe that the dimension is $q^2 = (\log n_i)^{O(1/\delta)}$.

As for the running time of constructing G_i and τ, observe that the bottleneck is the running time needed to find the center $\mathbf{s}$; according to Lemma 11.21, $\mathbf{s}$ can be computed in $O(|\mathcal{C}_1| \cdot |\mathcal{C}_2| \cdot q^2) = O(n_i^2 \cdot q^2)$, which is $n_i^{2+o(1)}$ as desired. □

A Gadget for Maximum Inner Product

Now, we build gadgets (stated below) which will be used for proving the inapproximability of MIP.

Theorem 11.24. *For every $0 < \delta < 1$, there exists a log-dense sequence $(n_i)_{i \in \mathbb{N}}$ such that, for every $i \in \mathbb{N}$, there is a bipartite graph $G_i = (A_i \dot\cup B_i, E_i)$ where $|A_i| = |B_i| = n_i$ and $|E_i| \geqslant \Omega(n_i^{2-\delta})$, such that 3-$\mathsf{ipd}(G) = (\log n_i)^{O(1/\delta)}$. Moreover, for all $i \in \mathbb{N}$, a 3-gap-IP-realization $\tau : A_i \dot\cup B_i \to \{0,1\}^{(\log n_i)^{O(1/\delta)}}$ of G_i can be constructed in time $n_i^{4+o(1)}$.*

Proof. The proof here is exactly the same as the proof of Theorem 11.19, except that we will not pick $K_2 = K_1 + 1$, but rather pick $K_2 > 3K_1$ (and n_i accordingly).

More precisely, let q_i be the i-th prime number and let $n_i = (q_i)^{(\lfloor q_i^{0.3\delta}/3 \rfloor)}$; it is simple to see that the sequence $(n_i)_{i \in \mathbb{N}}$ is log-dense. For $q = q_i$, consider the Reed-Solomon codes $\mathcal{C}_1 = \mathsf{RS}_q[q, K_1]$ and $\mathcal{C}_2 = \mathsf{RS}_q[q, K_2]$ where $K_1 = \lfloor q^{0.3\delta}/3 \rfloor$ and $K_2 = 3K_1 + 1$. Similar to the proof of Theorem 11.19, applying Lemma 11.21 with $(\mathcal{C}_1, \mathcal{C}_2)$ implies that there exists $\mathbf{s} \in \mathcal{C}_2 \setminus \mathcal{C}_1$ such that

$$\frac{|\mathcal{B}(\mathbf{s}, \Delta(\mathcal{C}_2)) \cap \mathcal{C}_1|}{|\mathcal{C}_1|} \geqslant \frac{q-1}{q} \cdot \left(\frac{1}{K_2 - 1}\right)^{K_2 - 1} = \frac{q-1}{q} \cdot \frac{1}{(3K_1)^{(3K_1)}} \geqslant \frac{1}{2} \cdot \frac{1}{q^{\delta K_1}} = \Omega(|\mathcal{C}_1|^{-\delta}).$$

We construct the graph $G_i = (A_i, B_i, E_i)$ and a realization τ as follows. Let $A_i = \mathcal{C}_1, B_i = \{\mathbf{s} + \mathbf{c} \mid \mathbf{c} \in \mathcal{C}_1\}$ and $E_i = \{(\mathbf{a}, \mathbf{b}) \in A_i \times B_i \mid \Delta(\mathbf{a}, \mathbf{b}) = \Delta(\mathcal{C}_2)\}$. G_i can be easily 3-gap-IP-realized by applying the mapping $\psi : \mathbb{F}_q^q \to \{0,1\}^{q^2}$ from Proposition 11.10. More precisely, let τ be the restriction of ψ on $A_i \cup B_i$. Below we argue about the density of G_i and that τ is a $(K_2 - 1, 3)$-gap-IP-realization of G_i.

- First, notice that $|E_i|$ is exactly $|\mathcal{C}_1| \cdot |\mathcal{B}(\mathbf{s}, \Delta(\mathcal{C}_2)) \cap \mathcal{C}_1| \geqslant \Omega(|\mathcal{C}_1|^{2-\delta}) = \Omega(n_i^{2-\delta})$.
- Second, for every $\mathbf{v}_1, \mathbf{v}_2$ both from A_i or both from B_i, we have $\mathbf{v}_1 - \mathbf{v}_2 \in \mathcal{C}_1 \setminus \{\mathbf{0}\}$. Thus, $\langle \tau(\mathbf{v}_1), \tau(\mathbf{v}_2) \rangle = q - \Delta(\mathbf{v}_1, \mathbf{v}_2) \leqslant q - \Delta(\mathcal{C}_1) = K_1 - 1 < (K_2 - 1)/3$.
- Third, for every $\mathbf{a} \in A_i$ and $\mathbf{b} \in B_i$, we have $\mathbf{a} - \mathbf{b} \in \mathcal{C}_2 \setminus \{\mathbf{0}\}$. Thus, $\Delta(\mathbf{a}, \mathbf{b}) \geqslant \Delta(\mathcal{C}_2)$. Hence, $\langle \tau(\mathbf{a}), \tau(\mathbf{b}) \rangle = q - \Delta(\mathbf{a}, \mathbf{b}) \leqslant q - \Delta(\mathcal{C}_2) = K_2 - 1$. Moreover, the inequality is an equality if and only if $\Delta(\mathbf{a}, \mathbf{b}) = \Delta(\mathcal{C}_2)$, i.e., $(\mathbf{a}, \mathbf{b}) \in E_i$ as desired.
- Finally, observe that the dimension is $q^2 = (\log n_i)^{O(1/\delta)}$.

Once again, the running time of the construction is $O(|\mathcal{C}_1| \cdot |\mathcal{C}_2| \cdot q^2) \leqslant n_i^{4+o(1)}$. □

11.4.3 Gadgets based on AG Codes

In this subsection, we construct gadgets based on algebraic geometric (AG) codes. The definitions of AG Codes are well beyond the scope of this work and we refer the readers to [Sti08; VNT07] for more thorough introductions.

Once again to find a good center, we need a bound on the number of minimum weight codewords. On this front, we use the following bound[11] from [Vlă18]. Throughout this subsection, we follow the notations from [Vlă18].

Theorem 11.25 (Theorem 4.3 of [Vlă18])**.** *Let q be a prime power, X be a curve of genus g over $\mathbb{F}_q$, let $S \subseteq X(\mathbb{F}_q)$ such that $|S| = N$, and let $a \in \mathbb{N}$ with $1 \leqslant a \leqslant N - 1$. Then, there exists an $\mathbb{F}_q$-positive divisor $D \geqslant 0$, $\deg(D) = a$, such that the corresponding AG Code $\mathcal{C} = \mathcal{C}(X, D, S)$ has minimum distance $N - a$ and*

$$A_{N-a}(\mathcal{C}) \geqslant \frac{\binom{N}{a}}{(\sqrt{q}+1)^{2g}}.$$

We also need the following well-known (central) fact about the parameters of AG codes.

Theorem 11.26. *Let q be a prime power, X be a curve of genus g over $\mathbb{F}_q$, let $S \subseteq X(\mathbb{F}_q)$ such that $|S| = N$, and let $a \in \mathbb{N}$ with $1 \leqslant a \leqslant N-1$. Then, the corresponding AG Code $C = C(X, D, S)$ is a linear code over $\mathbb{F}_q$ with block length N, distance at least $N-a$ and message length $k \geqslant a-g+1$.*

Recall also the tower of functions of Garcia and Stichtenoth [GS96], whose parameters approach the TVZ bound. We note here that, it suffices for us to have the genus approaching $\Omega(N/\sqrt{q})$ and there are also other curves that satisfy this.

Theorem 11.27 ([GS96])**.** *For any $\zeta > 0$ and any square of prime q, there exists a dense sequence[12] $(N_i)_{i\in\mathbb{N}}$ such that there exists a curve X_i with genus at most $\frac{N_i}{\sqrt{q}-1} + \zeta$ where $|X_i(\mathbb{F}_q)| \geqslant N_i$.*

Plugging the bound from [Vlă18] into the above family of curves immediately yields the following:

Lemma 11.28. *For any $\zeta > 0$ and any square of prime q, there exists a dense sequence $(N_i)_{i\in\mathbb{N}}$ such that the following holds. For any $i \in \mathbb{N}$ and any $a_1, a_2 \in \mathbb{N}$ such that $1 \leqslant a_1 < a_2 \leqslant N_i - 1$, there exist linear codes $\mathcal{C}_1 \subseteq \mathcal{C}_2 \subseteq \mathbb{F}_q^{N_i}$ such that the following holds, where $g_i = \frac{N_i}{\sqrt{q}-1} + \zeta$:*

- *$\mathcal{C}_1$ has message length at least $a_1 - g_i + 1$ and distance at least $N_i - a_1$.*

[11] Note that most of the proof of this bound was from [ABV01]; [Vlă18] simply makes the bound more explicit, which is more convenience for us.

[12] A sequence $(N_i)_{i\in\mathbb{N}}$ of increasing positive integers is said to be *dense* if there exists a constant $C \geqslant 1$ such that $N_{i+1} \leqslant C \cdot N_i$ for all $i \in \mathbb{N}$.

- *$\mathcal{C}_2$ has message length at least $a_2 - g_i + 1$ and distance exactly $N_i - a_2$ and*

$$A_{N_i - a_2}(\mathcal{C}_2) \geqslant \frac{\binom{N_i}{a_2}}{(\sqrt{q}+1)^{2g_i}}. \tag{11.3}$$

Moreover, the generator matrices of $\mathcal{C}_1, \mathcal{C}_2$ can be computed in $O\left(\binom{N+a_2-1}{a_2} \cdot |\mathcal{C}_2| \cdot poly(N_i)\right)$ time.

Proof. Let $(N_i)_{i \in \mathbb{N}}$ be a dense sequence as in Theorem 11.27. From Theorem 11.25, there exists an $\mathbb{F}_q$-positive divisor D_2 of degree a_2 such that the corresponding code $\mathcal{C}_2 = C(X_i, D_2, S_i)$ (where $S \subseteq X_i(\mathbb{F}_q)$ of size N_i) satisfies (11.3) and that its distance is $N_i - a_2$; from Theorem 11.26, its message length must also be at least $a_2 - g_i + 1$. Next, let D_1 be any $\mathbb{F}_q$-positive divisor of degree a_1 such that $D_2 - D_1 \geqslant 0$. Let $\mathcal{C}_1 = C(X_i, D_1, S_i)$ be the corresponding AG code; once again, Theorem 11.26 yields the desired bounds on its message length and distance. Finally, observe that $D_2 - D_1 \geqslant 0$ implies that $\mathcal{C}_1 \subseteq \mathcal{C}_2$ as desired.

The main bottleneck to algorithmically construct such codes lies in finding D_2. Nevertheless, the total number of degree-a_2 $\mathbb{F}_q$-positive divisor is only $\binom{N_i+a_2-1}{a_2}$. We can use brute force to enumerate all of them and check whether the corresponding code satisfies (11.3), which further takes $|\mathcal{C}_2|$ time. This results in the claimed running time. □

Finally, we can now construct our gadgets, by an appropriate setting of parameters. In particular, a_1 and a_2 will be selected to be close to each other and to both be slightly larger than $N/\sqrt{q}$. This results in the graphs whose degrees are roughly square root of the number of vertices.

Theorem 11.29. *For every $0 < \delta < 1$, there exist $\mu > 0$ and a log-dense sequence $(n_i)_{i \in \mathbb{N}}$ such that, for every $i \in \mathbb{N}$, there is a bipartite graph $G_i = (A_i \dot{\cup} B_i, E_i)$ where $|A_i| = |B_i| = n_i$ and $|E_i| \geqslant \Omega(n_i^{1.5-\delta})$, such that $(1+\mu)$-$\mathsf{cd}(G) = O(\log n_i)$. Moreover, for all $i \in \mathbb{N}$, a $(\beta, 1+\mu)$-gap-realization $\tau : A_i \dot{\cup} B_i \to \{0,1\}^{O(\log n_i)}$ of G_i can be constructed in time $O(n_i^3)$ for some $\beta = \Theta(\log n_i)$.*

Proof. Once again, the proof here is similar to those of Theorems 11.19 and 11.24, except that we use the (pairs of) AG codes from Lemma 11.28 instead of Reed-Solomon codes.

Let $q \geqslant 49$ be any sufficiently large square of prime and $\zeta > 0$ be any sufficiently small positive real number (both to be precisely specified later).

Let $(N_i)_{i \in \mathbb{N}}$ be the sequence guaranteed by Lemma 11.28. Let $a_1 = N_i \cdot \left(\frac{1}{q^{0.5(1-\delta)}} - \frac{1}{q}\right)$ and $a_2 = \frac{N_i}{q^{0.5(1-\delta)}}$. For convenience, we assume that a_1 and a_2 are integers[13]. Let $\mathcal{C}_1, \mathcal{C}_2$ be the codes given by Lemma 11.28. The sequence $(n_i)_{i \in \mathbb{N}}$ is defined as $n_i = |\mathcal{C}_1|$.

Applying Lemma 11.21 to $(\mathcal{C}_1, \mathcal{C}_2)$ implies that there exists $\mathsf{s} \in \mathcal{C}_2 \setminus \mathcal{C}_1$ such that

$$\frac{|\mathcal{B}(\mathsf{s}, \Delta(\mathcal{C}_2)) \cap \mathcal{C}_1|}{|\mathcal{C}_1|} \geqslant \frac{A_{\Delta(\mathcal{C}_2)}(\mathcal{C}_2)}{|\mathcal{C}_2|}$$

[13] Note that, for sufficiently large N_i, one can take the ceilings (or floors) of the specified values to get integers with negligible affect to the calculations.

$$\begin{aligned}
\text{(From Lemma 11.28)} &\geqslant \frac{\binom{N_i}{a_2}}{(\sqrt{q}+1)^{2g_i} \cdot |C_2|} \\
\text{(Singleton Bound)} &\geqslant \frac{\binom{N_i}{a_2}}{(\sqrt{q}+1)^{2g_i} \cdot q^{a_2+1}} \\
&\geqslant \frac{(N_i/a_2)^{a_2}}{(\sqrt{q}+1)^{2g_i} \cdot q^{a_2+1}} \\
&= \frac{q^{0.5(1-\delta)a_2}}{(\sqrt{q}+1)^{2g_i} \cdot q^{a_2+1}} \\
&= \frac{1}{(\sqrt{q}+1)^{2g_i} \cdot q^{(0.5+0.5\delta)a_2+1}} \\
&= \frac{1}{q^{(0.5+0.5\delta+o(1))a_2}} \\
&= \frac{1}{q^{(0.5+0.5\delta+o(1))(a_1+o(1))}} \\
&= \frac{1}{|\mathcal{C}_1|^{(0.5+0.5\delta+o(1))}} \\
&\geqslant \Omega(|\mathcal{C}_1|^{-0.5-0.5\delta-o(1)}) \qquad (11.4)
\end{aligned}$$

where $o(1)$ terms above denote the terms that go to zero as $q \to \infty$ and $\zeta \to 0$. As a result, by picking q sufficiently large and ζ sufficiently small, the term in (11.4) is at least $\Omega(|\mathcal{C}_1|^{-0.5-\delta})$.

We construct the graph $G_i = (A_i, B_i, E_i)$ and a realization τ as follows. Let $A_i = \mathcal{C}_1, B_i = \{\mathbf{s} + \mathbf{c} \mid \mathbf{c} \in \mathcal{C}_1\}$ and $E_i = \{(\mathbf{a}, \mathbf{b}) \in A_i \times B_i \mid \Delta(\mathbf{a}, \mathbf{b}) = \Delta(\mathcal{C}_2)\}$. G_i can be easily realized by applying the mapping $\psi : \mathbb{F}_q^{N_i} \to \{0,1\}^{N_i \cdot q}$ from Proposition 11.10. More precisely, let τ be the restriction of ψ on $A_i \cup B_i$. Below we argue about the density of G_i and that τ is a $(2\Delta(\mathcal{C}_2), 1+\mu)$-gap-realization of G_i where $\mu = \frac{\Delta(\mathcal{C}_1)-1}{\Delta(\mathcal{C}_2)} - 1$. Note that

$$\mu \geqslant \frac{a_2 - a_1 - 1}{N_i - a_2} = \Omega(1/q).$$

Let us now check that G_i and τ satisfy all the claimed properties:

- First, notice that $|E_i|$ is exactly $|\mathcal{C}_1| \cdot |\mathcal{B}(\mathbf{s}, \Delta(\mathcal{C}_2)) \cap \mathcal{C}_1| \geqslant \Omega(|\mathcal{C}_1|^{1.5-\delta}) = \Omega(n_i^{1.5-\delta})$.

- For any $\mathbf{v}_1 = \psi(\mathbf{c}_1), \mathbf{v}_2 = \psi(\mathbf{c}_2)$ both from X_i or both from Y_i, we have $\mathbf{c}_1 - \mathbf{c}_2 \in \mathcal{C}_1 \setminus \{\mathbf{0}\}$. Hence, $\|\mathbf{v}_1 - \mathbf{v}_2\|_0 = 2 \cdot \Delta(\mathbf{v}_1, \mathbf{v}_2) \geqslant 2 \cdot \Delta(\mathcal{C}_1) > (1+\mu) \cdot (2\Delta(\mathcal{C}_2))$.

- Next, for every $\mathbf{a} \in A_i$ and $\mathbf{b} \in B_i$, we have $\mathbf{a} - \mathbf{b} \in \mathcal{C}_2 \setminus \{\mathbf{0}\}$. Thus, $\Delta(\mathbf{a}, \mathbf{b}) \geqslant \Delta(\mathcal{C}_2)$. Hence, $\|\tau(\mathbf{a}) - \tau(\mathbf{b})\|_0 = 2\Delta(\mathbf{a}, \mathbf{b}) \geqslant 2\Delta(\mathcal{C}_2)$. Moreover, the inequality is an equality if and only if $\Delta(\mathbf{a}, \mathbf{b}) = \Delta(\mathcal{C}_2)$, i.e., $(\mathbf{a}, \mathbf{b}) \in E_i$ as desired.

Given $\mathcal{C}_1, \mathcal{C}_2$, the running time of constructing (X_i, Y_i) is $O(|\mathcal{C}_1| \cdot |\mathcal{C}_2| \cdot q^2) = O(n_i^3)$. Moreover, the running time to construct $\mathcal{C}_1$ and $\mathcal{C}_2$, as given by Lemma 11.28, is

$$\begin{aligned} O\left(\binom{N+a_2-1}{a_2} \cdot |\mathcal{C}_2| \cdot \text{poly}(N_i)\right) &\leqslant O\left((e(N+a_2)/a_2)^{a_2} \cdot |\mathcal{C}_2| \cdot \text{poly}(N_i)\right) \\ &\leqslant O\left((2e\sqrt{q})^{a_2} \cdot |\mathcal{C}_2| \cdot \text{poly}(N_i)\right) \\ &\leqslant O\left(|\mathcal{C}_1| \cdot |\mathcal{C}_2| \cdot \text{poly}(N_i)\right) \\ &\leqslant O(n_i^3), \end{aligned}$$

where the last two inequalities are true for any sufficiently large q. □

11.5 Inapproximability of Maximum Inner Product

In this section, we prove the hardness of approximating MIP. Once again, we show a stronger version (than Theorem 11.3) where every point has Boolean coordinates, as stated below.

Theorem 11.30. *Assuming* OVH, *for every $\varepsilon > 0$, there is no algorithm running in $O(n^{2-\varepsilon})$ time for γ-*MIP *even for points in $\{0,1\}^{n^{o(1)}}$, for any $\gamma \leqslant 2^{(\log n)^{1-o(1)}}$.*

The proof proceeds in two steps: first, we show hardness of approximating MIP in low dimension but with a small $(1 + o(1))$ approximation factor. Second, we use tensor product operation to amplify the gap to be almost polynomial, as stated in Theorem 11.30. More specifically, in the first step, we prove the following:

Theorem 11.31. *Assuming* OVH, *for every $\varepsilon > 0$, there exists $s_\varepsilon > 0$ such that no algorithm running in $O(n^{2-\varepsilon})$ time can solve $\left(1 + \frac{1}{\log\log n}\right)$-*MIP *even for points in $\{0,1\}^{(\log n)^{s_\varepsilon}}$.*

Note that the factor $\frac{1}{\log\log n}$ is not significant, and this can be replaced by any $o(1)$ factor; we use this just to make the calculations more concrete. Before we move on to the proof of Theorem 11.31, let us first show how it implies Theorem 11.30.

Proof of Theorem 11.30 from Theorem 11.31. Let (P, α) be an instance of $\left(1 + \frac{1}{\log\log n}\right)$-MIP where $P \subseteq \{0,1\}^{(\log n)^{s_\varepsilon}}$. For $t = \frac{\log n}{(\log\log n)^2}$, define $P' = \{\mathbf{x}^{\otimes t} \mid \mathbf{x} \in P\}, \alpha' = \alpha^t$ and $\gamma = \left(1 + \frac{1}{\log\log n}\right)^t = 2^{(\log n)^{1-o(1)}}$. The dimension of points in P' is $(\log n)^{s_\varepsilon \cdot t} = n^{o(1)}$. Moreover, it is easy to check, based on the identity $\langle \mathbf{x}^{\otimes t}, \mathbf{y}^{\otimes t} \rangle = \langle \mathbf{x}, \mathbf{y} \rangle^t$, that (P', α') is a YES (resp. NO) instance of γ-MIP iff (P, α) is a YES (resp. NO) instance of $\left(1 + \frac{1}{\log\log n}\right)$-MIP.

In other words, if there is an $O(n^{2-\varepsilon})$ time algorithm for γ-MIP in $n^{o(1)}$ dimension, then there also exists an $O(n^{2-\varepsilon})$ subquadratic time algorithm for $\left(1 + \frac{1}{\log\log n}\right)$-MIP in $(\log n)^{s_\varepsilon}$ dimension. Thus, Theorem 11.30 follows from Theorem 11.31. □

The rest of this section is devoted to proving Theorem 11.31. To do so, we consider the gap-Additive-BMIP problem.

Definition 11.32 (γ-Additive-BMIP problem). *Let $\gamma \geqslant 0$. In the γ-Additive-BMIP problem we are given two sets A, B each of n points in $\{0,1\}^d$ and an integer $\alpha \in [d]$ as input, and the goal is to distinguish between the following two cases.*

- ***Completeness.** There exists $(a, b) \in A \times B$ such that $\langle a, b\rangle \geqslant \alpha$.*
- ***Soundness.** For every $(a, b) \in A \times B$ we have $\langle a, b\rangle < \alpha - \gamma$.*

We need the below hardness result from [Rub18]. Note that the result is stated differently in [Rub18]; for how the result in [Rub18] implies the one below, see Section 3.2 of [Che18a].

Theorem 11.33 ([Rub18]). *Assuming* OVH, *for every $\varepsilon > 0$, there is no algorithm running in $O(n^{2-\varepsilon})$ time for the γ-Additive-BMIP problem, for any $d = \omega(\log n)$ and $\gamma = o(d)$.*

Proof of Theorem 11.31. For any $\varepsilon > 0$, let C_{exp} be the constant such that the dimension of τ in Theorem 11.24 is at most $(\log n_i)^{C_{\text{exp}}/\varepsilon}$ for $\delta = \varepsilon/2$. We define s_ε as $2 \cdot C_{\text{exp}}/\varepsilon + 2$.

Suppose contrapositively that there exists $\varepsilon > 0$ and an algorithm $\mathcal{A}$ that can solve $\left(1 + \frac{1}{\log \log n}\right)$-MIP of dimension $(\log n)^{s_\varepsilon}$ in time $n^{2-\varepsilon}$. We will construct an algorithm $\mathcal{A}'$ that solves $(\log n)$-Additive-BMIP in time $n^{2-\varepsilon'}$ for some constant $\varepsilon' > 0$ (to be specified below) for $d = (\log n \sqrt{\log \log n})$ dimensions. Together with Theorem 11.33, this implies that OVH is false, as desired.

Let C_ε denote the constant of the log-dense sequence from Theorem 11.24 for $\delta = \varepsilon/2$, and let ε' be $0.01 \cdot \varepsilon/C_\varepsilon$. The algorithm $\mathcal{A}'$ on input (A, B, α) where $A, B \subseteq \{0,1\}^d, \alpha \in [d]$ works as follows:

1. Let n' be the largest number in the sequence from Theorem 11.24 with $\delta = \varepsilon/2$ s.t. $n' \leqslant n^{0.1}$.
2. Let $G' = (A' \dot\cup B', E')$ be the graph from Theorem 11.24 with $|A'| = |B'| = n'$, $|E'| \geqslant \Omega((n')^{2-\delta})$, and $\tau : A' \dot\cup B' \to \{0,1\}^{(\log n')^{C_{\text{exp}}/\varepsilon}}$ be a $(\beta, 3)$-gap-IP-relization of G' where $\beta \in \mathbb{N}$.
3. We use the algorithm from Lemma 11.8 to find $\pi_1, \ldots, \pi_k$ where $k = O((n')^\delta \log n')$ such that the union of $E_{G'_{\pi_1}}, \ldots, E_{G'_{\pi_k}}$ is $E_{K_{n',n'}}$
4. We assume w.l.o.g. that n is divisible by n'. Partition A and B into $A_1, \ldots, A_{n/n'}$ and $B_1, \ldots, B_{n/n'}$ each of size n'. For each $i, j \in [n/n'], t \in [k]$, do the following:
    a) Let τ_t be an appropriate permutation of τ that $(\beta, 3)$-gap-IP-realizes G'_{π_t}.
    b) Let $\alpha' = \beta \cdot \alpha + 3d \cdot \beta$, and define A_i^t, B_j^t as
    $$A_i^t = \{(\mathbf{1}_\beta \otimes \mathbf{a}) \circ (\mathbf{1}_{3d} \otimes \tau_t(\mathbf{a})) \mid \mathbf{a} \in A_i\}, B_j^t = \{(\mathbf{1}_\beta \otimes \mathbf{b}) \circ (\mathbf{1}_{3d} \otimes \tau_t(\mathbf{b})) \mid \mathbf{b} \in B_j\}.$$
    c) Run $\mathcal{A}$ on $(A_i^t \dot\cup B_j^t, \alpha')$. If $\mathcal{A}$ outputs YES, then output YES and terminate.
5. If none of the executions of $\mathcal{A}$ returns with YES, then output NO.

Observe that the bottleneck in the running time of the algorithm is in the executions of $\mathcal{A}$. The number of executions is $(n/n')^2 \cdot k$ and each execution takes $O((n')^{2-\varepsilon})$ time. Hence, in total the running time of the algorithm $\mathcal{A}'$ is $O((n/n')^2 \cdot k \cdot (n')^{2-\varepsilon}) \leqslant O(n^2 \log n \cdot (n')^{-\varepsilon/2})$. Now, from the log-density of the sequence from Theorem 11.24, we have $n' \geqslant n^{0.1/C_\varepsilon} = n^{10\varepsilon'/\varepsilon}$. As a result, the running time of $\mathcal{A}$ is at most $O(n^{2-5\varepsilon'} \log n) \leqslant O(n^{2-\varepsilon'})$ as desired.

To see the correctness of the algorithm, first observe that the dimensions of vectors in A_i^t, B_j^t are at most $\beta \cdot d + 3d \cdot (\log n')^{C_{\exp}/\varepsilon}$ which is at most $(\log n)^{s_\varepsilon}$ for any sufficiently large n; that is, the calls to $\mathcal{A}$ are valid. Next, observe that, if (A, B, α) is a YES instance of Additive-BMIP, there must be $i, j \in [n/n']$ and $\mathbf{a}^* \in A_i, \mathbf{b}^* \in B_j$ such that $\langle \mathbf{a}^*, \mathbf{b}^* \rangle$ is at least α. Since $G'_{\pi_1}, \dots, G'_{\pi_k}$ covers $K_{n',n'}$, there must be $t \in [k]$ such that $\langle \tau_t(\mathbf{a}^*), \tau_t(\mathbf{b}^*) \rangle \geqslant \beta$. As a result, $\langle (\mathbf{1}_\beta \otimes \mathbf{a}^*) \circ (\mathbf{1}_{3d} \otimes \tau_t(\mathbf{a}^*), (\mathbf{1}_\beta \otimes \mathbf{b}^*) \circ (\mathbf{1}_{3d} \otimes \tau_t(\mathbf{b}^*)) \rangle \geqslant \beta \cdot \alpha + 3d \cdot \beta = \alpha'$. Thus, $(A_i^t \cup B_j^t, \alpha')$ is a YES instance for MIP and $\mathcal{A}'$ outputs YES as desired.

Finally, let us assume that (A, B, α) is a NO instance of $(\log n)$-Additive-BMIP. Consider any $i, j \in [n/n']$ and $t \in [k]$. To argue that $(A_i^t \cup B_j^t, \alpha')$ is a NO instance for $\left(1 + \frac{1}{\log\log n'}\right)$-MIP, we have to show that any two points in $A_i^t \cup B_j^t$ have inner product less than $\alpha' / \left(1 + \frac{1}{\log\log n'}\right)$. To see this, let us consider two cases.

1. The two points are either both from A_i^t or both from B_j^t. Assume w.l.o.g. that the two points are from A_i^t; let them be $(\mathbf{1}_\beta \otimes \mathbf{a}) \circ (\mathbf{1}_{3d} \otimes \tau_t(\mathbf{a}))$ and $(\mathbf{1}_\beta \otimes \mathbf{a}') \circ (\mathbf{1}_{3d} \otimes \tau_t(\mathbf{a}'))$. Recall that, from Theorem 11.24, we must have $\langle \tau_t(\mathbf{a}), \tau_t(\mathbf{a}') \rangle < \beta/3$. Moreover, since $\mathbf{a}, \mathbf{a}' \in \{0,1\}^d$, we have $\langle \mathbf{a}, \mathbf{a}' \rangle \leqslant d$. Thus, we can conclude that
$$\begin{aligned}\langle (\mathbf{1}_\beta \otimes \mathbf{a}) \circ (\mathbf{1}_{3d} \otimes \tau_t(\mathbf{a})), (\mathbf{1}_\beta \otimes \mathbf{a}') \circ (\mathbf{1}_{3d} \otimes \tau_t(\mathbf{a}')) \rangle &< \beta \cdot d + 3d \cdot (\beta/3) \\ &< (2/3) \cdot \alpha',\end{aligned}$$
which is less than $\alpha' / \left(1 + \frac{1}{\log\log n'}\right)$ for any sufficiently large n.

2. One of the point is from A_i^t and the other from B_j^t. Let them be $(\mathbf{1}_\beta \otimes \mathbf{a}) \circ (\mathbf{1}_{3d} \otimes \tau_t(\mathbf{a}))$ and $(\mathbf{1}_\beta \otimes \mathbf{b}) \circ (\mathbf{1}_{3d} \otimes \tau_t(\mathbf{b}))$. Since (A, B, α) is a NO instance of $(\log n)$-Additive-BMIP, we must have $\langle \mathbf{a}, \mathbf{b} \rangle < \alpha - \log n$. Furthermore, from Theorem 11.24, we must have $\langle \tau_t(\mathbf{a}), \tau_t(\mathbf{b}) \rangle \leqslant \beta$. Combining the two implies that
$$\begin{aligned}\langle (\mathbf{1}_\beta \otimes \mathbf{a}) \circ (\mathbf{1}_{3d} \otimes \tau_t(\mathbf{a})), (\mathbf{1}_\beta \otimes \mathbf{b}) \circ (\mathbf{1}_{3d} \otimes \tau_t(\mathbf{v})) \rangle &< \beta \cdot (\alpha - \log n) + 3d \cdot \beta \\ &= \alpha' - \beta \cdot (\log n) \\ (\text{Since } \alpha' \leqslant 4d\beta) &\leqslant \alpha' \left(1 - \frac{1}{4\sqrt{\log\log n}}\right) \\ &\leqslant \alpha' \left(1 - \frac{1}{\log\log n'}\right) \\ &\leqslant \alpha' / \left(1 + \frac{1}{\log\log n'}\right),\end{aligned}$$
where the second-to-last inequality holds for any sufficiently large n.

Hence, $(A_i^t \dot\cup B_j^t, \alpha')$ must be a NO instance for $\left(1 + \frac{1}{\log\log n'}\right)$-MIP for every $t \in [k]$ and $i, j \in [n/n']$. Thus, $\mathcal{A}'$ outputs NO as desired. □

11.6 Inapproximability of Closest Pair

In this section, we prove the hardness of approximating CP (Theorem 11.2). As usual, we reduce from the bichromatic version of the problem, and the lower bound for the bichromatic version is stated below:

Theorem 11.34 (Rubinstein [Rub18]). *Assuming* OVH*, for every* $\varepsilon > 0$ *there exists* $\kappa > 0$ *such that there is no algorithm running in* $n^{2-\varepsilon}$ *time for* $(1+\kappa)$-BCP *in the Hamming metric. Moreover, this holds even for instances* (A, B, α) *of* $(1+\kappa)$-BCP *when* $d = \Theta_\varepsilon(\log n), \alpha = \Theta_\varepsilon(\log n)$ *and* $A, B \subseteq \{0,1\}^d$.

Again, we prove below the inapproximability of the gap-CP problem for Boolean vectors. Clearly, this immediately implies Theorem 11.2.

Theorem 11.35. *Assuming* OVH*, for every* $\varepsilon > 0$*, there exists* $\theta > 0$ *and* $c > 0$ *such that there is no algorithm running in* $n^{1.5-\varepsilon}$ *time for* $(1+\theta)$-CP *in the Hamming metric for point-set in* $\{0,1\}^{c \cdot \log n}$.

Proof. Assume towards a contradiction that there exists an $\varepsilon > 0$ and an algorithm $\mathcal{A}$ that, for every $\theta > 0$ solves $(1+\theta)$-CP of dimension $c \cdot \log n$ in time $O(n^{1.5-\varepsilon})$, where $c := c(\varepsilon)$ is a constant that will be specified later. Let $\varepsilon' > 0$ be a small constant (depending on ε) that we will specify below and let $\kappa = \kappa(\varepsilon')$ be as in Theorem 11.34. We construct below an algorithm $\mathcal{A}'$ that solves $(1+\kappa)$-BCP in time $O(n^{2-\varepsilon'})$ for any instance (A, B, α) such that $A, B \subseteq \{0,1\}^{O(\log n)}$ and $\alpha = \Theta(\log n)$. Together with Theorem 11.34, this implies that OVH is false, as desired.

Let C_ε denote the constant of the log-dense sequence from Theorem 11.29 for $\delta = \varepsilon/2$, and let ε' be $0.01 \cdot \varepsilon / C_\varepsilon$. Let μ be the constant from Theorem 11.29. Select $\theta > 0$ be a sufficiently small constant such that $\frac{\mu-\theta}{1+\theta} > \frac{\theta}{\kappa-\theta}$.

The algorithm $\mathcal{A}'$ on (A, B, α) where $A, B \subseteq \{0,1\}^{O(\log n)}, \alpha = \Theta(\log n)$ works as follows:

1. Let n' be the largest number in the sequence from Theorem 11.29 with $\delta = \varepsilon/2$ s.t. $n' \leqslant n^{0.1}$.

2. Let $G' = (A' \dot\cup B', E')$ be the graph from Theorem 11.29 with $|A'| = |B'| = n'$, $|E'| \geqslant \Omega((n')^{1.5-\delta})$, and $\tau : A' \dot\cup B' \to \{0,1\}^{O(\log n')}$ be a $(\beta, 1+\mu)$-gap-relization of G' where $\beta \in \mathbb{N}$ and $\beta = \Theta(\log n')$.

3. We use the algorithm from Lemma 11.8 to find $\pi_1, \ldots, \pi_k$ where $k = O((n')^{0.5+\delta} \log n')$ such that the union of $E_{G'_{\pi_1}}, \ldots, E_{G'_{\pi_k}}$ is $E_{K_{n',n'}}$

4. We assume w.l.o.g. that n is divisible by n'. Partition A and B into $A_1, \ldots, A_{n/n'}$ and $B_1, \ldots, B_{n/n'}$ each of size n'. For each $i, j \in [n/n'], t \in [k]$, do the following:

a) Let τ_t be an appropriate permutation of τ that $(\beta, 1+\mu)$-gap-realizes G'_{π_t}.

b) Pick r_1, r_2 such that

$$\frac{\theta}{\kappa-\theta} \cdot \frac{\beta}{\alpha} \leqslant \frac{r_1}{r_2} \leqslant \frac{\mu-\theta}{1+\theta} \cdot \frac{\beta}{\alpha}. \tag{11.5}$$

Notice that the upper and lower bounds are $\Theta(1)$ and they are also $\Theta(1)$ apart. Hence, we can pick these r_1, r_2 so that $r_1, r_2 = \Theta(1)$.

c) Let $\alpha' = r_1 \cdot \alpha + r_2 \cdot \beta$ and define A_i^t, B_j^t as

$A_i^t = \{(\mathbf{1}_{r_1} \otimes \mathbf{a}) \circ (\mathbf{1}_{r_2} \otimes \tau_t(\mathbf{a})) \mid \mathbf{a} \in A_i\}, B_j^t = \{(\mathbf{1}_{r_1} \otimes \mathbf{b}) \circ (\mathbf{1}_{r_2} \otimes \tau_t(\mathbf{b})) \mid \mathbf{b} \in B_j\}.$

d) Run $\mathcal{A}$ on $(A_i^t \cup B_j^t, \alpha')$. If $\mathcal{A}$ outputs YES, then output YES and terminate.

5. If none of the executions of $\mathcal{A}$ returns with YES, then output NO.

Observe that the bottleneck in the running time of the algorithm is in the executions of $\mathcal{A}$. The number of executions is $(n/n')^2 \cdot k$ and each execution takes $O((n')^{1.5-\varepsilon})$ time. Hence, in total the running time of the algorithm $\mathcal{A}'$ is $O((n/n')^2 \cdot k \cdot (n')^{1.5-\varepsilon}) \leqslant O(n^2 \log n \cdot (n')^{-\varepsilon/2})$. Now, from the log-density of the sequence from Theorem 11.29, we have $n' \geqslant n^{0.1/C_\varepsilon} = n^{10\varepsilon'/\varepsilon}$. As a result, the running time of $\mathcal{A}$ is at most $O(n^{2-5\varepsilon'} \log n) \leqslant O(n^{2-\varepsilon})$ as desired.

To see the correctness of the algorithm, first observe that the dimensions of vectors in A_i^t, B_j^t are at most $r_1 \cdot \alpha + r_2 \cdot \beta$ which is $O(\log n')$; that is, the calls to $\mathcal{A}$ are valid. Next, observe that, if (A, B, α) is a YES instance of BCP, there must be $i, j \in [n/n']$ and $\mathbf{a}^* \in A_i, \mathbf{b}^* \in B_j$ such that $\|\mathbf{a}^* - \mathbf{b}^*\|_0$ is at most α. Since $G'_{\pi_1}, \dots, G'_{\pi_k}$ covers $K_{n',n'}$, there must be $t \in [k]$ such that $\|\tau_t(\mathbf{a}^*) - \tau_t(\mathbf{b}^*)\|_0 \leqslant \beta$. As a result, $\|((\mathbf{1}_{r_1} \otimes \mathbf{a}^*) \circ (\mathbf{1}_{r_2} \otimes \tau_t(\mathbf{a}^*)) - ((\mathbf{1}_{r_1} \otimes \mathbf{b}^*) \circ (\mathbf{1}_{r_2} \otimes \tau_t(\mathbf{b}^*)))\|_0 \leqslant r_1 \cdot \alpha + r_2 \cdot \beta = \alpha'$. Thus, $(A_i^t \cup B_j^t, \alpha')$ is a YES instance for CP and $\mathcal{A}'$ outputs YES as desired.

Finally, let us assume that (A, B, α) is a NO instance of $(1+\kappa)$-BCP. Consider any $i, j \in [n/n']$ and $t \in [k]$. To argue that $(A_i^t \cup B_j^t, \alpha')$ is a NO instance for $(1+\theta)$-CP, we have to show that any two points in $A_i^t \cup B_j^t$ have distance more than α'. To see this, let us consider two cases.

1. Both points are either from A_i^t or from B_j^t. Assume w.l.o.g. that they are from A_i^t; let them be $(\mathbf{1}_{r_1} \otimes \mathbf{a}) \circ (\mathbf{1}_{r_2} \otimes \tau_t(\mathbf{a}))$ and $(\mathbf{1}_{r_1} \otimes \mathbf{a}') \circ (\mathbf{1}_{r_2} \otimes \tau_t(\mathbf{a}'))$. Recall that, from the definition of X'_t and Theorem 11.29, we must have $\|\tau_t(\mathbf{a}) - \tau_t(\mathbf{a}')\|_0 > (1+\mu) \cdot \beta$. Thus, the Hamming distance between the two points is more than $r_2 \cdot (1+\mu) \cdot \beta \geqslant (1+\theta) \cdot \alpha'$, where the inequality comes from our choice of r_1, r_2.

2. One of the point is from A_i^t and the other from B_j^t. Let them be $(\mathbf{1}_{r_1} \otimes \mathbf{a}) \circ (\mathbf{1}_{r_2} \otimes \tau_t(\mathbf{a}))$ and $(\mathbf{1}_{r_1} \otimes \mathbf{b}) \circ (\mathbf{1}_{r_2} \otimes \tau_t(\mathbf{b}))$. Since (A, B, α) is a NO instance of $(1+\kappa)$-BCP, $\|\mathbf{a} - \mathbf{b}\|_0 > (1+\kappa) \cdot \alpha$. Moreover, from definition of τ_t, we must have $\|\tau_t(\mathbf{a}) - \tau_t(\mathbf{b})\|_0 \geqslant \beta$. Combining the two implies that the distance between $(\mathbf{1}_{r_1} \otimes \mathbf{a}) \circ (\mathbf{1}_{r_2} \otimes \tau_t(\mathbf{a}))$ and $(\mathbf{1}_{r_1} \otimes \mathbf{b}) \circ (\mathbf{1}_{r_2} \otimes \tau_t(\mathbf{b}))$ is more than $r_1 \cdot (1+\kappa) \cdot \alpha + r_2 \cdot \beta \geqslant (1+\theta) \cdot \alpha'$, where the inequality is once again from our choice of r_1, r_2.

Hence, $(A_i^t \dot\cup B_j^t, \alpha')$ must be a NO instance for $(1+\theta)$-CP for every $t \in [k]$ and $i, j \in [n/n']$. Thus, $\mathcal{A}'$ outputs NO as desired. □

11.7 Inapproximability of Closest Pair in Edit Distance Metric

In this section we prove Theorem 11.4. The proof is almost identical to Rubinstein's [Rub18] proof for the OVH-hardness of gap-BCP in the edit distance metric and uses the following technical tool established in [Rub18].

Lemma 11.36 (Rubinstein [Rub18]). *For large enough $d \in \mathbb{N}$, there is a function $\zeta : \{0,1\}^d \to \{0,1\}^{d'}$, where $d' = O(d \log d)$, such that for all $a, b \in \{0,1\}^d$ the following holds for some constant $\lambda > 0$:*

$$|\mathsf{ed}(\zeta(a), \zeta(b)) - \lambda \cdot \log d \cdot \|a - b\|_0| = o(d').$$

Moreover, for any $a \in \{0,1\}^d$, $\zeta(a)$ can be computed in $2^{o(d)}$ time.

At a high level, ζ picks a random $O(\log d)$-bit string $s_{i,x}$ uniformly and independently for every $(i, x) \in [d] \times \{0,1\}$, and for every vector $u \in \{0,1\}^d$, replaces the i^{th} coordinate u_i by s_{i,u_i}. The claims in the lemma statement follow by the known concentration bounds on the edit distance of random strings [McD89; Lue09]. This construction is further efficiently derandomized by using $\log d$-wise independent strings [Kop13].

Proof of Theorem 11.4. We show that if there exists an algorithm $\mathcal{A}$ running in time $O(n^{1.5-\varepsilon})$ for some $\varepsilon > 0$ that can solve $(1+\delta)$-CP in the edit distance metric for some $\delta > 0$ over point-sets in $\{0,1\}^{d'}$, then $\mathcal{A}$ can be used to solve $(1 + \delta - o(1))$-CP in the Hamming metric in time $O(n^{1.5-\varepsilon})$ over point-sets in $\{0,1\}^d$, where $d' = O(d \log d)$. Together with Theorem 11.35, this implies that OVH is false, as desired.

Let (P, α) be an instance of $(1+\delta)$-CP in the Hamming metric over point-sets in $\{0,1\}^d$. It is clear[14] from the proofs of Theorem 11.34 and Theorem 11.35 that $\alpha = \Omega(d)$. We now define an instance of $(P', \alpha' := (1 + o(1)) \cdot \lambda \log d \cdot \alpha)$ of $(1 + \delta - o(1))$-CP in the edit distance metric as follows. Recall the function ζ from Lemma 11.36 and define the set $P' = \{\zeta(p) \mid p \in P\}$. Notice that for every pair of distinct points $p, q \in P$, we have $|\mathsf{ed}(\zeta(p), \zeta(q)) = \lambda \cdot \log d \cdot \|p - q\|_0| = o(d')$. In other words if we had a pair of distinct points p, q in P such that $\|p - q\|_0 \leqslant \alpha$ then, $\mathsf{ed}(\zeta(p), \zeta(q)) \leqslant \lambda \log d \cdot \alpha + o(d') = (1 + o(1)) \cdot \lambda \log d \cdot \alpha$ and suppose for all pairs of distinct points $p, q \in P$ we had $\|p - q\|_0 > (1 + \delta) \cdot \alpha$ then $\mathsf{ed}(\zeta(p), \zeta(q)) > \lambda \log d \cdot (1 + \delta) \cdot \alpha - o(d') > (1 + \delta - o(1))\lambda \log d \cdot \alpha$, since $\alpha = \Omega(d)$. This completes the analysis of the completeness and soundness cases, and we can conclude that running $\mathcal{A}$ on input (P', α') solves the instance (P, α) of $(1 + \delta)$-CP in the Hamming metric. □

11.8 Discussion and Open Questions

It remains open to completely resolve Open Questions 9 and 10. It is still possible that our framework can be used to resolve these problems: we just need to construct gadgets with better parameters! In particular, to resolve Question 9, we have to improve the dimension bound in Theo-

[14] In fact, one can design a $2^{\alpha} \cdot n \log n$ time algorithm for CP in the Hamming metric, and therefore to assume OVH, we require $\alpha = \Omega(d)$.

rem 11.19 to $O_\delta(\log n_i)$. For Question 10, we just have to improve the bound in Theorem 11.29, i.e., improve the bound on the number of pairs in (3) of Lemma 11.28 to $\Omega(n_i^{2-\delta})$. Following our observation from Lemma 11.21, this motivates us to ask the following purely coding theoretic question, which would imply the desired hardness (if the codes can be constructed in poly($|\mathcal{C}_1|$) time):

Open Question 11. *For every* $0 < \delta < 1$, *are there linear codes* $\mathcal{C}_1 \subseteq \mathcal{C}_2 \subseteq \mathbb{F}_q^N$ *both of block length* N *over alphabet* $\mathbb{F}_q$ *such that the following holds:*

- $\Delta(\mathcal{C}_1) \geqslant (1 + f(\delta)) \cdot \Delta(\mathcal{C}_2)$, *for some* $f : (0, 1) \to (0, 1)$.
- $|A_{\Delta(\mathcal{C}_2)}(\mathcal{C}_2)|/|\mathcal{C}_2| \geqslant |\mathcal{C}_1|^{-\delta}$.

Apart from the aforementioned questions, Rubinstein [Rub18] pointed out an interesting obstacle, aptly dubbed the "triangle inequality barrier", to obtain fine-grained lower bounds against 3-approximation algorithms for BCP (see Open Question 3 in [Rub18]). In the case of CP, this barrier turns out to be against 2-approximation algorithms as noted in [DKL18]. We reiterate this below as an open problem to be resolved:

Open Question 12. *Can we show that assuming* SETH, *for some constant* $\varepsilon > 0$, *no algorithm running in time* $n^{1+\varepsilon}$ *can solve* 2-CP *in* any *metric when the points are in* $\omega(\log n)$ *dimensions?*

Another interesting direction is to extend the hardness of MIP to the k-vector generalization of the problem, called k-MIP. In k-MIP, we are given a set of n points $P \subseteq \mathbb{R}^d$ and we would like to select k distinct points $\mathbf{a}_1, \ldots, \mathbf{a}_k \in P$ that maximizes

$$\langle \mathbf{a}_1, \ldots, \mathbf{a}_k \rangle := \sum_{j \in [d]} (\mathbf{a}_1)_j \cdots (\mathbf{a}_k)_j.$$

Recall that, in Section 6.8, we showed that the k-chromatic variant of k-MIP is hard to approximate but this is not known to be true for k-MIP itself. Our approach seems quite compatible to tackling this problem as well; in particular, if we can construct a certain (natural) generalization of our gadget for MIP, then we would immediately arrive at the inapproximability of k-MIP even for $\{0, 1\}$-entries vectors. The issue in constructing this gadget is that we are now concerned about agreements of more than two vectors, which does not correspond to error-correcting codes anymore and some additional tools are needed to argue for this more general case.

It should be noted that the hardness of approximating k-MIP for $\{0, 1\}$-entry vectors is equivalent to the *one-sided k-biclique* problem [Lin15], in which a bipartite graph is given and the goal is to select k vertices on the right that maximize the number of their common neighbors. The equivalence can be easily seen by viewing the coordinates as the left-hand-side vertices and the vectors as the right-hand-side vertices. The one-sided k-biclique is shown to be W$[1] \neq$ FPT-hard to approximate by Lin [Lin15] who also showed a lower bound of $n^{\Omega(\sqrt{k})}$ for the problem assuming ETH. If the generalization of our gadget for k-MIP works as intended, then this lower bound can be improved to $n^{\Omega(k)}$ under ETH and even $n^{k-o(1)}$ under SETH.

The one-sided k-biclique is closely related to the (two-sided) k-biclique problem, where we are given a bipartite graph and we wish to decide whether it contains $K_{k,k}$ as a subgraph. The k-biclique problem was consider a major open problem in parameterized complexity (see e.g., [DF13]) until it was shown by Lin to be $\mathrm{W}[1] \neq$ FPT-hard [Lin15]. Nevertheless, the running time lower bound known is still not tight: currently, the best lower bound known for this problem is $n^{\Omega(\sqrt{k})}$ both for the exact version (under ETH) [Lin15] and its approximate variant (under GAP-ETH; see Chapter 8). It remains an interesting open question to close the gap between the above lower bounds and the trivial upper bound of $n^{O(k)}$. Progresses on the one-sided k-biclique problem could lead to improved lower bounds for k-biclique problem too, although several additional steps have to be taken care of.

Chapter 12

Discussion and Future Directions

Although this dissertation, together with many other recent works, has advanced our understanding of approximation and hardness between P and NP, the area is still very much in its infancy. To conclude this thesis, we provide several directions that we think are interesting for future research.

Direction 1. *Prove Quasi-Polynomial Hardness for Problems with "Complicated" Algorithms*

As touched upon slightly in Section 1.1.3, the birthday repetition technique has so far been mostly applied upon problems with "simple" algorithms: either it is a bilinear optimization problem of the form $\max_{u,v} u^\top Av$ for which it suffices to enumerate over all sparse v, or it is a problem which has a brute-force algorithm (e.g. VC Dimension). However, there are many other problems which admit quasi-polynomial time algorithms that are far more complicated than these two types of algorithms. These problems include (Approximate) Graph Isomorphism [Bab16; AFK02][1], Directed Steiner Tree [Cha+99], Max-Min Allocation [CCK09] and Maximum Independent Set of Rectangles [AW13; CE16]. It is a natural, yet challenging, direction to apply birthday repetition techniques to these problems, and eventually map down the complexity and approximability of quasi-polynomial time algorithms.

Direction 2. *Toward Completeness Results for Quasi-Polynomial Hardness of Approximation?*

Despite the success of the birthday repetition framework, it has a slight weakness: it does not show that these gap/approximate problems are complete for any complexity class. In contrast, the exact versions of problems such as VC Dimension and dominating set on tournaments are in fact complete for the classes LOGNP and LOGSNP respectively [PY96]. The lack of completeness in the body of works related to birthday repetition is undesirable for a variety of reasons. First, although these gap/approximation problems are proved to be hard under ETH, it is unclear how these problems related to each other. (The only relation known is the obvious fact that Dense CSPs is harder than DENSEST k-SUBGRAPH with perfect completeness.) Second, it could be troublesome if ETH turns out to be false; since all hardness of approximation results in this line of

[1] To be clear, the QPTAS for an approximate version of graph isomorphism from [AFK02] is also similar to the bilinear optimization algorithm described earlier. However, Babai's algorithm for graph isomorphism is not.

work are based on ETH, it would not be clear anymore what we can even say about these problems. Note that it is totally possible that ETH turns out to be false, while LOGNP and LOGSNP are still not contained in P. As a result, it would be much more desirable if we can establish completeness results for hardness of approximation as well; at this point, however, it is not even clear which class these problems should be based on, and more works need to be done to achieve this goal.

Direction 3. *Prove Gap-ETH (assuming ETH)*

Given the applications of Gap-ETH in fine-grained hardness of approximation (as partly presented in this dissertation), an obvious open question is to determine whether Gap-ETH holds or not. A specific question here is whether we can prove Gap-ETH, if we assume ETH. As discussed earlier in Section 2.3, this would hold if a linear-size PCP exists, a well-known open problem in the area of PCP. We would like to stress here that this might not be the only way to prove Gap-ETH. For instance, the reduction could take exponential time and it could be (even adaptive) Turing reduction, which might be useful.

On the other hand, if one were to attempt to disprove Gap-ETH, we encourage one to disprove the following stronger conjecture, which we call Gap-SETH, first.

Conjecture 12.1 (Gap-SETH). *For every $\varepsilon > 0$, there exists $k = k(\varepsilon) \in \mathbb{N}$ and $\delta = \delta(\varepsilon) > 0$ such that no algorithm can distinguish between a satisfiable k-*SAT *formula and one that is not $(1-\delta)$ satisfiable in $O(2^{(1-\varepsilon)n})$ time where n is the number of variables.*

To the best of our knowledge, such a conjecture had never been studied before and it is hence unclear whether it should be true. In fact, it could also be that there is a simple algorithm that refutes the conjecture. However, we are not aware of such an algorithm either. It should be noted here that, if Gap-SETH holds, then it would immediately imply all the results that have so far been achieved via the Distributed PCP framework.

Direction 4. *Prove a "Parameterized Version of PCP Theorem"*

We next move on to the questions more specifically about parameterized complexity/approximability. On this front, the most obvious question is whether one can prove a "PCP-like Theorem" for parameterized complexity. Before we go forward, let us first point out that the distributed PCP framework is much different than the standard PCP (and this is the reason why we state it in terms of communication complexity instead of using the terminology from [ARW17a]). In fact, if one thinks about the framework in the PCP terminology, then the verifier can read *the whole proof*, unlike the traditional PCP where the verifier can only query a few bits/positions of the proof. The only restriction for distributed PCP however is that the accepting configuration for each randomness is small (i.e. "small left alphabet" of MAXCOV). Alternatively, one could think about distributed PCP as a PCP where the verifier query a few positions, similar to the typical PCP formulation, but when there are multiple provers and these provers are *honest*; this is indeed the original viewpoint in [ARW17a]. On the other hand, there is only one prover who is (possibly) *cheating* in the standard PCP Theorem, which is the whole point of the PCP Theorem. Both of these viewpoints demonstrate that distributed PCP is fundamentally different from the (standard) PCP Theorem.

So what should be the "parameterized PCP Theorem"? First, if we think of the PCP Theorem as a characterization of NP, then one might hope for a PCP characterization of some fundamental class in the area of parameterized complexity. In this case, W[1] is a natural candidate and, similar to the PCP characterization of NP, the characterization would be that every parameterized language in W[1] has a PCP verifier that uses a constant number of queries, $f(k)$ randomness (for some function f that can depends on the language), has completeness one and soundness $1/2$. We remark here that the verifier is now allowed to run in FPT time, and the alphabet size is allowed to be as large as $g(k) \cdot \text{poly}(n)$ for any function g.

Note here that, due to the clause-variable game transformation (see Definition 3.20) and parallel repetition [Raz98], we may assume without loss of generality that the number of queries is two. It is also simple to observe that all languages with such PCPs lie in W[1]. As a result, in the PCP language (cf. [MR10]), such a characterization would translate to the following:

Conjecture 12.2. W*[1] = PCP*$_{1,1/2}[f(k), 2]_{g(k)\cdot poly(n)}$ *(where the prover is allowed to run in FPT time).*

In terms of hardness of approximation, the above conjecture translates to the hardness of approximating 2CSP where the parameter is the number of variables, the exact same setting as in Chapter 9 except that here the gap is only some constant. (It should be noted, once again, that the gap can then be amplify to any constant factor via parallel repetition [Raz98]; however, this does not get super constant factor as in Chapter 9.) Such a conjecture was made before in by Lokshtanov et al. [Lok+17] under the name *Parameterized Inapproximability Hypothesis (PIH)*:

Conjecture 12.3 (Parameterized Inapproximability Hypothesis (PIH) [Lok+17])**.** *For some $\varepsilon > 0$, it is* W*[1]-hard to distinguish a satisfiable instance of parameterized 2-CSP from one which is not even $(1 - \varepsilon)$-satisfiable.*

Indeed, the hypothesis above can sometimes be used in place of Gap-ETH to obtain inapproximability results, but often fails to yield as stronger inapproximability factor as that from Gap-ETH. For instance, it is simple to see that PIH implies that k-Clique is hard to approximate to some constant factor, and this factor can be amplify to any constant factor. However, it is not known whether the inverse is true; that is, it is not known whether hardness of approximating k-Clique (to within some constant factor) implies PIH. This leaves us with another conjecture that is not known to be equivalent to PIH:

Conjecture 12.4. *For some $\varepsilon > 0$, it is* W*[1]-hard to approximate k-Clique to within a factor of $(1 - \varepsilon)$.*

It is now a good point to also note that, when taking the hardness of approximation perspective, the PCP Theorem is about NP-completeness of gap problems (e.g. Gap-3SAT). In this regards, Conjectures 12.3 and 12.4 can both be viewed as this form of "parameterized PCP Theorems".

For other parameterized complexity classes, such as W[t] for $t \geqslant 2$, it is not clear what their PCP characterization should be (in the sense of Conjecture 12.2 for W[1]); in fact, even machine-based definition for these classes are pretty complicated (see [CFG05] and references therein).

Nevertheless, if one takes the hardness of approximation perspective, then one can attempt to generalize conjectures similar to Conjectures 12.3, 12.4 to higher classes in the W hierarchy. Still, we have to be slightly careful here, since the "canonical" problems used to defined these classes are the weighted circuit satisfiability for weft-t circuits. However, when the circuits are not monotone or anti-monotone, it is not even clear at all what approximation even means.

When the circuits are monotone (resp. anti-monotone), we have a well-defined optimization problem: find an assignment with minimum (resp. maximum) weight that satisfies the circuit. Let WEIGHTED t-MONOTONE SATISFIABILITY (resp. WEIGHTED t-ANTIMONOTONE SATISFIABILITY) be this problem. The following is known for the *exact* version of the problem [DF95b; DF95a]. For even $t \geqslant 2$, WEIGHTED t-MONOTONE SATISFIABILITY and WEIGHTED $(t+1)$-MONOTONE SATISFIABILITY are W$[t]$-complete. For odd $t \geqslant 3$, WEIGHTED t-ANTIMONOTONE SATISFIABILITY and WEIGHTED $(t+1)$-ANTIMONOTONE SATISFIABILITY are W$[t]$-complete. Moreover, the variants of the problems with no bound on the weft, which are simply called WEIGHTED MONOTONE SATISFIABILITY and WEIGHTED ANTIMONOTONE SATISFIABILITY, are known to be W$[P]$-complete.

We can then ask for the approximability of these problems. This has been studied before in literature [AR08; EGG08; Mar13]. In particular, Marx [Mar13] showed that both WEIGHTED MONOTONE SATISFIABILITY and WEIGHTED ANTIMONOTONE SATISFIABILITY are totally FPT inapproximable, assuming W$[P] \neq$ FPT and W$[1] \neq$ FPT respectively. In fact, for the monotone case, he show a finer-grained result that, for even $t \geqslant 2$, WEIGHTED $(t+2)$-MONOTONE SATISFIABILITY and WEIGHTED $(t+3)$-MONOTONE SATISFIABILITY are W$[t]$-hard to approximate to within $f(k)$ ration for any function f. However, this does not give completeness result yet as the weft is "off" by additive factor of 2, and hence we can try to ask the following questions:

Open Question 13. *For even $t \geqslant 2$, are* WEIGHTED t-MONOTONE SATISFIABILITY *and* WEIGHTED $(t+1)$-MONOTONE SATISFIABILITY W$[t]$*-hard to approximate to within $f(k)$ ratio for any f?*

For odd $t \geqslant 3$, are WEIGHTED t-ANTIMONOTONE SATISFIABILITY *and* WEIGHTED $(t+1)$-ANTIMONOTONE SATISFIABILITY W$[t]$*-hard to approximate to within $f(k)$ ratio for any $f = o(k)$?*

Note that, if the above question is positively resolved for $t = 2$, then it would imply the total inapproximability of k-DOMSET under W$[2] \neq$ FPT (i.e. answer Question 1).

Direction 5. *Toward Tight Hardness of Approximation for Parameterized Problems?*

A recurring feature of the hardness of approximation results for parameterized problems throughout this thesis is that the problems we consider are so hard, that even parameterization and approximation together can hardly help beyond some straightforward algorithm. While one might view these as satisfactory "tight" results, they prompt the obvious question: what happens to the problems that parameterization *does* help achieve better approximation?

To be more precise, let us focus on the problems for which (i) are hard to solve exactly in FPT time, (ii) admit a "considerably better" approximation ratio in FPT time, but (iii) are not known to admit an FPT approximation scheme. Note that (i) and (iii) are here so that there is at least something to prove in terms of hardness of approximation. As for (ii), it not only serves to ensure that these are the problems with "non-trivial" parameterized approximation algorithms, but also helps to force us to think differently if we are to prove hardness of approximation for these problems. The latter is a technical motivation for this research direction, because so far[2] many of the parameterized hardness of approximation proofs borrow a lot of ideas from the theory of NP-hardness of approximation (with the exception of distributed PCP). It would be much more interesting if we can also develop some additional techniques that are completely different from those in the NP-hardness regime.

To the best of our knowledge, there are not too many such candidate problems that fit into these restrictions. Nonetheless, there are already interesting examples. In fact, Lokshtanov et al. [Lok+17] proposed the PIH conjecture partially to tackle one such problem: the DIRECTED ODD CYCLE TRANSVERSAL (DOCT) problem. In the non-parameterized regime, DOCT is UGC-hard to approximation to within any constant factor. However, Lokshtanov et al. gives an FPT time 2-approximation algorithm for the problem. Furthermore, they show that, for some $\varepsilon > 0$, no FPT algorithm achieves an approximation ratio of $(1 + \varepsilon)$, assuming PIH. Since then, several more problems of this kind are known. For instance, in [CFM18], Chitnis, Feldman and the author point out two additional problems with these properties: the STRONGLY CONNECTED STEINER SUBGRAPH (SCSS) and the DIRECTED STEINER NETWORK ON BIDIRECTED GRAPHS (BI-DSN).

Obtaining tight hardness of approximation results for such problems (i.e. $(2 - \delta)$-factor hardness for DOCT) is an interesting question. Drawing parallel to the theory of NP-hardness of approximation, the quest to obtain optimal hardness of approximation has led to many fascinating development, including the innovative use of long codes and fourier analysis (e.g. [BGS98; Hås01; ST00]) and later the unique games conjecture and its implications (e.g. [Kho02; Kho+07; MOO05; Rag08]). It is thus interesting to understand whether there is a similar depth to the theory of parameterized hardness of approximation in this sense. For instance, a crucial notion to obtain optimal inapproximability results in the NP regime is the notion of *dictatorship test/gadget* (see e.g. [Kho+07; Rag08]). Is there a similar notion in the parameterized regime that can be used to prove optimal parameterized hardness of approximation?

We end by remarking that, for one of the aforementioned problems (SCSS), a tight hardness of approximation was shown [CFM18]. However, the gadget seems to be ad-hoc and it is unclear how to generalize this to work for other problems.

Direction 6. *Prove Hardness of Approximation in P Beyond "PSP-Style" Problems*

Finally, as described in Chapter 6, the distributed PCP framework, which is currently the main method to prove hardness of approximation in P, naturally starts with a PSP problem and produces MAXCOV instance. The latter can also be viewed as a PSP problem, with the distinction that now the predicate f is not boolean, but rather returns the number of left super-nodes covered by the selected labeling instead. The last direction we suggest is to try to come up with a different

[2] For instance, in this thesis, we have seen that the parameterized inapproximability for Dominating Set, Clique, Even Set and Shortest Vector Problem borrow ideas (to some degree) from their NP-hardness of approximation counterpart [Fei98; Fei+91; DMS03; Kho05].

gap-producing technique for problems in P that either (i) does not start with a PSP problem or (ii) does not produce a "PSP-style" problem. The hope for (i) would be to apply the techniques to other fine-grained complexity hypotheses that are not of the PSP form, such as the APSP hypothesis [WW18]. The goal for (ii) is of course to prove hardness of approximation for problems in P that are not of the PSP forms; for instance, there are many problems with dynamic programming algorithms for which tight running time lower bounds (in P) are known, and these problems are not of the PSP form. Perhaps the most prominent such example is the Longest Common Subsequent (LCS) problem, for which, despite a considerable amount of afford [AB17; AR18], no hardness of approximation is yet known under "standard" fine-grained complexity assumptions.

Bibliography

[Aar+09] Scott Aaronson, Salman Beigi, Andrew Drucker, Bill Fefferman, and Peter W. Shor. "The Power of Unentanglement". In: *Theory of Computing* 5.1 (2009), pp. 1–42.

[AB17] Amir Abboud and Arturs Backurs. "Towards Hardness of Approximation for Polynomial Time Problems". In: *ITCS*. 2017, 11:1–11:26.

[ABC09] Chrisil Arackaparambil, Joshua Brody, and Amit Chakrabarti. "Functional Monitoring without Monotonicity". In: *ICALP*. 2009, pp. 95–106.

[ABV01] Alexei E. Ashikhmin, Alexander Barg, and Serge G. Vladut. "Linear Codes with Exponentially Many Light Vectors". In: *J. Comb. Theory, Ser. A* 96.2 (2001), pp. 396–399.

[AC09] Reid Andersen and Kumar Chellapilla. "Finding Dense Subgraphs with Size Bounds". In: *WAW*. 2009, pp. 25–37.

[ACW16] Josh Alman, Timothy M. Chan, and R. Ryan Williams. "Polynomial Representations of Threshold Functions and Algorithmic Applications". In: *FOCS*. 2016, pp. 467–476.

[AD97] Miklós Ajtai and Cynthia Dwork. "A Public-Key Cryptosystem with Worst-Case/Average-Case Equivalence". In: *STOC*. 1997, pp. 284–293.

[AFK02] Sanjeev Arora, Alan M. Frieze, and Haim Kaplan. "A new rounding procedure for the assignment problem with applications to dense graph arrangement problems". In: *Math. Program.* 92.1 (2002), pp. 1–36.

[Aga+91] Pankaj K. Agarwal, Herbert Edelsbrunner, Otfried Schwarzkopf, and Emo Welzl. "Euclidean Minimum Spanning Trees and Bichromatic Closest Pairs". In: *Discrete & Computational Geometry* 6 (1991), pp. 407–422.

[AHI02] Yuichi Asahiro, Refael Hassin, and Kazuo Iwama. "Complexity of finding dense subgraphs". In: *Discrete Applied Mathematics* 121.1–3 (2002), pp. 15–26.

[AIM14] Scott Aaronson, Russell Impagliazzo, and Dana Moshkovitz. "AM with Multiple Merlins". In: *CCC*. 2014, pp. 44–55.

[Ajt96] Miklós Ajtai. "Generating Hard Instances of Lattice Problems (Extended Abstract)". In: *STOC*. 1996, pp. 99–108.

[Ajt98] Miklós Ajtai. "The Shortest Vector Problem in ℓ_2 is NP-hard for Randomized Reductions (Extended Abstract)". In: *STOC*. 1998, pp. 10–19.

[AK14] Per Austrin and Subhash Khot. "A Simple Deterministic Reduction for the Gap Minimum Distance of Code Problem". In: *IEEE Trans. Information Theory* 60.10 (2014), pp. 6636–6645.

[AKK95] Sanjeev Arora, David Karger, and Marek Karpinski. "Polynomial Time Approximation Schemes for Dense Instances of NP-hard Problems". In: *STOC*. Las Vegas, Nevada, USA: ACM, 1995, pp. 284–293.

[AL13] Amir Abboud and Kevin Lewi. "Exact Weight Subgraphs and the k-Sum Conjecture". In: *ICALP*. 2013, pp. 1–12.

[Alo+03] Noga Alon, Wenceslas Fernandez de la Vega, Ravi Kannan, and Marek Karpinski. "Random Sampling and Approximation of MAX-CSPs". In: *J. Comput. Syst. Sci.* 67.2 (Sept. 2003), pp. 212–243.

[Alo+11] Noga Alon, Sanjeev Arora, Rajsekar Manokaran, Dana Moshkovitz, and Omri Weinstein. "Inapproximabilty of Densest k-Subgraph from Average Case Hardness". Unpublished Manuscript. 2011.

[Alo02] Noga Alon. "Testing subgraphs in large graphs". In: *Random Struct. Algorithms* 21.3-4 (2002), pp. 359–370.

[Alp10] Ethem Alpaydin. *Introduction to Machine Learning*. 2nd. The MIT Press, 2010.

[ALW14] Amir Abboud, Kevin Lewi, and Ryan Williams. "Losing Weight by Gaining Edges". In: *ESA*. 2014, pp. 1–12.

[AM09] Per Austrin and Elchanan Mossel. "Approximation Resistant Predicates from Pairwise Independence". In: *Computational Complexity* 18.2 (2009), pp. 249–271.

[AMM17] Haris Angelidakis, Yury Makarychev, and Pasin Manurangsi. "An Improved Integrality Gap for the Călinescu-Karloff-Rabani Relaxation for Multiway Cut". In: *IPCO*. 2017, pp. 39–50.

[AMS06] Noga Alon, Dana Moshkovitz, and Shmuel Safra. "Algorithmic construction of sets for *k*-restrictions". In: *ACM Trans. Algorithms* 2.2 (2006), pp. 153–177.

[AMS07] Christoph Ambuhl, Monaldo Mastrolilli, and Ola Svensson. "Inapproximability Results for Sparsest Cut, Optimal Linear Arrangement, and Precedence Constrained Scheduling". In: *FOCS*. Oct. 2007, pp. 329–337.

[AMS12] Noga Alon, Ankur Moitra, and Benny Sudakov. "Nearly complete graphs decomposable into large induced matchings and their applications". In: *STOC*. 2012, pp. 1079–1090.

[AOW15] Sarah R. Allen, Ryan O'Donnell, and David Witmer. "How to Refute a Random CSP". In: *FOCS*. 2015, pp. 689–708.

[App17] Benny Applebaum. “Exponentially-Hard gap-CSP and local PRG via Local Hardcore Functions”. In: *ECCC* 24 (2017), p. 63.

[AR08] Michael Alekhnovich and Alexander A. Razborov. “Resolution Is Not Automatizable Unless W[P] Is Tractable”. In: *SIAM J. Comput.* 38.4 (2008), pp. 1347–1363.

[AR18] Amir Abboud and Aviad Rubinstein. “Fast and Deterministic Constant Factor Approximation Algorithms for LCS Imply New Circuit Lower Bounds”. In: *ITCS*. 2018, 35:1–35:14.

[Aro+12] Sanjeev Arora, Rong Ge, Sushant Sachdeva, and Grant Schoenebeck. “Finding overlapping communities in social networks: toward a rigorous approach”. In: *EC*. 2012, pp. 37–54.

[Aro+97] Sanjeev Arora, László Babai, Jacques Stern, and Z. Sweedyk. “The Hardness of Approximate Optima in Lattices, Codes, and Systems of Linear Equations”. In: *J. Comput. Syst. Sci.* 54.2 (1997), pp. 317–331.

[Aro+98] Sanjeev Arora, Carsten Lund, Rajeev Motwani, Madhu Sudan, and Mario Szegedy. “Proof Verification and the Hardness of Approximation Problems”. In: *J. ACM* 45.3 (May 1998), pp. 501–555.

[ARW17a] Amir Abboud, Aviad Rubinstein, and R. Ryan Williams. “Distributed PCP Theorems for Hardness of Approximation in P”. In: *FOCS*. 2017, pp. 25–36.

[ARW17b] Amir Abboud, Aviad Rubinstein, and Ryan Williams. “Distributed PCP Theorems for Hardness of Approximation in P”. In: *CoRR* abs/1706.06407 (2017). arXiv: `1706.06407`.

[AS03] Sanjeev Arora and Madhu Sudan. “Improved Low-Degree Testing and its Applications”. In: *Combinatorica* 23.3 (2003), pp. 365–426.

[AS18] Divesh Aggarwal and Noah Stephens-Davidowitz. “(Gap/S)ETH hardness of SVP”. In: *STOC*. 2018, pp. 228–238.

[AS98] Sanjeev Arora and Shmuel Safra. “Probabilistic Checking of Proofs: A New Characterization of NP”. In: *J. ACM* 45.1 (Jan. 1998), pp. 70–122.

[AW09] Scott Aaronson and Avi Wigderson. “Algebrization: A New Barrier in Complexity Theory”. In: *TOCT* 1.1 (2009), 2:1–2:54.

[AW13] Anna Adamaszek and Andreas Wiese. “Approximation Schemes for Maximum Weight Independent Set of Rectangles”. In: *FOCS*. 2013, pp. 400–409.

[AW15] Josh Alman and Ryan Williams. “Probabilistic Polynomials and Hamming Nearest Neighbors”. In: *FOCS*. 2015, pp. 136–150.

[Bab+03] László Babai, Anna Gál, Peter G. Kimmel, and Satyanarayana V. Lokam. “Communication Complexity of Simultaneous Messages”. In: *SIAM J. Comput.* 33.1 (2003), pp. 137–166.

[Bab16] László Babai. “Graph isomorphism in quasipolynomial time [extended abstract]”. In: *STOC*. 2016, pp. 684–697.

[Bab85] László Babai. “Trading Group Theory for Randomness”. In: *STOC*. Providence, Rhode Island, USA: ACM, 1985, pp. 421–429.

[Bar+04] Ziv Bar-Yossef, T. S. Jayram, Ravi Kumar, and D. Sivakumar. “An information statistics approach to data stream and communication complexity”. In: *J. Comput. Syst. Sci.* 68.4 (2004), pp. 702–732.

[Bar+11] Boaz Barak, Moritz Hardt, Thomas Holenstein, and David Steurer. “Subsampling Mathematical Relaxations and Average-case Complexity”. In: *SODA*. SODA ’11. San Francisco, California: SIAM, 2011, pp. 512–531.

[Bar+12] Boaz Barak, Fernando G. S. L. Brandão, Aram Wettroth Harrow, Jonathan A. Kelner, David Steurer, and Yuan Zhou. “Hypercontractivity, sum-of-squares proofs, and their applications”. In: *STOC*. 2012, pp. 307–326.

[Bar15] Siddharth Barman. “Approximating Nash Equilibria and Dense Bipartite Subgraphs via an Approximate Version of Caratheodory’s Theorem”. In: *STOC*. Portland, Oregon, USA: ACM, 2015, pp. 361–369.

[BE98] Shai Ben-David and Nadav Eiron. “Self-Directed Learning and Its Relation to the VC-Dimension and to Teacher-Directed Learning”. In: *Machine Learning* 33.1 (1998), pp. 87–104.

[Bec+18] Luca Becchetti, Andrea E. F. Clementi, Pasin Manurangsi, Emanuele Natale, Francesco Pasquale, Prasad Raghavendra, and Luca Trevisan. “Average Whenever You Meet: Opportunistic Protocols for Community Detection”. In: *ESA*. 2018, 7:1–7:13.

[Bei+19] Xiaohui Bei, Xinhang Lu, Pasin Manurangsi, and Warut Suksompong. “The Price of Fairness for Indivisible Goods”. In: *IJCAI*. To appear. 2019.

[Bel+93] Mihir Bellare, Shafi Goldwasser, Carsten Lund, and Alexander Russell. “Efficient probabilistically checkable proofs and applications to approximations”. In: *STOC*. 1993, pp. 294–304.

[Ben+16a] Eli Ben-Sasson, Alessandro Chiesa, Ariel Gabizon, Michael Riabzev, and Nicholas Spooner. “Short Interactive Oracle Proofs with Constant Query Complexity, via Composition and Sumcheck”. In: *ECCC* 23 (2016), p. 46.

[Ben+16b] Eli Ben-Sasson, Yohay Kaplan, Swastik Kopparty, Or Meir, and Henning Stichtenoth. “Constant Rate PCPs for Circuit-SAT with Sublinear Query Complexity”. In: *J. ACM* 63.4 (2016), 32:1–32:57.

[Ben80] Jon Louis Bentley. “Multidimensional Divide-and-Conquer”. In: *Commun. ACM* 23.4 (1980), pp. 214–229.

[Ben83] Michael Ben-Or. “Lower Bounds for Algebraic Computation Trees (Preliminary Report)”. In: *STOC*. 1983, pp. 80–86.

[Ber+13] Piotr Berman, Arnab Bhattacharyya, Konstantin Makarychev, Sofya Raskhodnikova, and Grigory Yaroslavtsev. “Approximation algorithms for spanner problems and Directed Steiner Forest”. In: *Inf. Comput.* 222 (2013), pp. 93–107.

[BFS16] Cristina Bazgan, Florent Foucaud, and Florian Sikora. “On the Approximability of Partial VC Dimension”. In: *COCOA*. 2016, pp. 92–106.

[BG15] Mark Braverman and Ankit Garg. “Small Value Parallel Repetition for General Games”. In: *STOC*. Portland, Oregon, USA: ACM, 2015, pp. 335–340.

[BGS17] Huck Bennett, Alexander Golovnev, and Noah Stephens-Davidowitz. “On the Quantitative Hardness of CVP”. In: *FOCS*. 2017, pp. 13–24.

[BGS98] Mihir Bellare, Oded Goldreich, and Madhu Sudan. “Free Bits, PCPs, and Nonapproximability-Towards Tight Results”. In: *SIAM J. Comput.* 27.3 (1998), pp. 804–915.

[Bha+10] Aditya Bhaskara, Moses Charikar, Eden Chlamtac, Uriel Feige, and Aravindan Vijayaraghavan. “Detecting high log-densities: an $O(n^{1/4})$ approximation for densest k-subgraph”. In: *STOC*. 2010, pp. 201–210.

[Bha+12] Aditya Bhaskara, Moses Charikar, Aravindan Vijayaraghavan, Venkatesan Guruswami, and Yuan Zhou. “Polynomial Integrality Gaps for Strong SDP Relaxations of Densest k-subgraph”. In: *SODA*. 2012, pp. 388–405.

[Bha+16a] Amey Bhangale, Rajiv Gandhi, Mohammad Taghi Hajiaghayi, Rohit Khandekar, and Guy Kortsarz. “Bicovering: Covering Edges With Two Small Subsets of Vertices”. In: *ICALP*. 2016, 6:1–6:12.

[Bha+16b] Umang Bhaskar, Yu Cheng, Young Kun Ko, and Chaitanya Swamy. “Hardness Results for Signaling in Bayesian Zero-Sum and Network Routing Games”. In: *EC*. Maastricht, The Netherlands: ACM, 2016, pp. 479–496.

[Bha+16c] Arnab Bhattacharyya, Ameet Gadekar, Suprovat Ghoshal, and Rishi Saket. “On the Hardness of Learning Sparse Parities”. In: *ESA*. 2016, 11:1–11:17.

[Bha+18] Arnab Bhattacharyya, Suprovat Ghoshal, Karthik C. S., and Pasin Manurangsi. “Parameterized Intractability of Even Set and Shortest Vector Problem from Gap-ETH”. In: *ICALP*. 2018, 17:1–17:15.

[BKW15] Mark Braverman, Young Kun-Ko, and Omri Weinstein. “Approximating the best Nash Equilibrium in $n^{o(\log n)}$-time breaks the Exponential Time Hypothesis”. In: *SODA*. 2015, pp. 970–982.

[BL05] Yonatan Bilu and Nathan Linial. “Monotone maps, sphericity and bounded second eigenvalue”. In: *J. Comb. Theory, Ser. B* 95.2 (2005), pp. 283–299.

[Blu94] Avrim Blum. “Separating Distribution-Free and Mistake-Bound Learning Models over the Boolean Domain”. In: *SIAM J. Comput.* 23.5 (1994), pp. 990–1000.

[BMT78] Elwyn R. Berlekamp, Robert J. McEliece, and Henk C. A. van Tilborg. “On the inherent intractability of certain coding problems (Corresp.)” In: *IEEE Trans. Information Theory* 24.3 (1978), pp. 384–386.

[Bon+15] Edouard Bonnet, Bruno Escoffier, Eun Jung Kim, and Vangelis Th. Paschos. "On Subexponential and FPT-Time Inapproximability". In: *Algorithmica* 71.3 (2015), pp. 541–565.

[Bon+18] Édouard Bonnet, László Egri, Bingkai Lin, and Dániel Marx. "Fixed-parameter approximability of Boolean MinCSPs". In: *arXiv preprint arXiv:1601.04935* (2018).

[BPR16] Yakov Babichenko, Christos H. Papadimitriou, and Aviad Rubinstein. "Can Almost Everybody be Almost Happy?" In: *ITCS*. 2016, pp. 1–9.

[BPS09] Shai Ben-David, Dávid Pál, and Shai Shalev-Shwartz. "Agnostic Online Learning". In: *COLT*. 2009.

[BR60] R. C. Bose and Dwijendra K. Ray-Chaudhuri. "On A Class of Error Correcting Binary Group Codes". In: *Information and Control* 3.1 (1960), pp. 68–79.

[Bra+17] Mark Braverman, Young Kun-Ko, Aviad Rubinstein, and Omri Weinstein. "ETH Hardness for Densest-k-Subgraph with Perfect Completeness". In: *SODA*. 2017, pp. 1326–1341.

[BRS11] Boaz Barak, Prasad Raghavendra, and David Steurer. "Rounding Semidefinite Programming Hierarchies via Global Correlation". In: *FOCS*. 2011, pp. 472–481.

[BS08] Eli Ben-Sasson and Madhu Sudan. "Short PCPs with Polylog Query Complexity". In: *SIAM J. Comput.* 38.2 (2008), pp. 551–607.

[BS76] Jon Louis Bentley and Michael Ian Shamos. "Divide-and-Conquer in Multidimensional Space". In: *STOC*. 1976, pp. 220–230.

[BS92] Piotr Berman and Georg Schnitger. "On the Complexity of Approximating the Independent Set Problem". In: *Inf. Comput.* 96.1 (1992), pp. 77–94.

[BS94] Mihir Bellare and Madhu Sudan. "Improved non-approximability results". In: *STOC*. 1994, pp. 184–193.

[Cam89] Kathie Cameron. "Induced matchings". In: *Discrete Applied Mathematics* 24.1-3 (1989), pp. 97–102.

[CCF10] Amin Coja-Oghlan, Colin Cooper, and Alan Frieze. "An Efficient Sparse Regularity Concept". In: *SIAM J. Discrete Math.* 23.4 (2010), pp. 2000–2034.

[CCK09] Deeparnab Chakrabarty, Julia Chuzhoy, and Sanjeev Khanna. "On Allocating Goods to Maximize Fairness". In: *FOCS*. 2009, pp. 107–116.

[CDK12] Eden Chlamtac, Michael Dinitz, and Robert Krauthgamer. "Everywhere-Sparse Spanners via Dense Subgraphs". In: *FOCS*. 2012, pp. 758–767.

[CE16] Julia Chuzhoy and Alina Ene. "On Approximating Maximum Independent Set of Rectangles". In: *FOCS*. 2016, pp. 820–829.

[CFG05] Yijia Chen, Jörg Flum, and Martin Grohe. "Machine-based methods in parameterized complexity theory". In: *Theor. Comput. Sci.* 339.2-3 (2005), pp. 167–199.

[CFL83] Ashok K. Chandra, Merrick L. Furst, and Richard J. Lipton. “Multi-Party Protocols”. In: *STOC*. 1983, pp. 94–99.

[CFM18] Rajesh Chitnis, Andreas Emil Feldmann, and Pasin Manurangsi. “Parameterized Approximation Algorithms for Bidirected Steiner Network Problems”. In: *ESA*. 2018, 20:1–20:16.

[CGG06] Yijia Chen, Martin Grohe, and Magdalena Grüber. “On Parameterized Approximability”. In: *IWPEC*. 2006, pp. 109–120.

[CH10] Liming Cai and Xiuzhen Huang. “Fixed-Parameter Approximation: Conceptual Framework and Approximability Results”. In: *Algorithmica* 57.2 (2010), pp. 398–412.

[Cha+17] Parinya Chalermsook, Marek Cygan, Guy Kortsarz, Bundit Laekhanukit, Pasin Manurangsi, Danupon Nanongkai, and Luca Trevisan. “From Gap-ETH to FPT-Inapproximability: Clique, Dominating Set, and More”. In: *FOCS*. 2017, pp. 743–754.

[Cha+99] Moses Charikar, Chandra Chekuri, To-Yat Cheung, Zuo Dai, Ashish Goel, Sudipto Guha, and Ming Li. “Approximation Algorithms for Directed Steiner Problems”. In: *J. Algorithms* 33.1 (1999), pp. 73–91.

[Cha16] Siu On Chan. “Approximation Resistance from Pairwise-Independent Subgroups”. In: *J. ACM* 63.3 (2016), 27:1–27:32.

[Che+04] Jianer Chen, Xiuzhen Huang, Iyad A. Kanj, and Ge Xia. “Linear FPT reductions and computational lower bounds”. In: *STOC*. 2004, pp. 212–221.

[Che+06] Jianer Chen, Xiuzhen Huang, Iyad A. Kanj, and Ge Xia. “Strong computational lower bounds via parameterized complexity”. In: *J. Comput. Syst. Sci.* 72.8 (2006), pp. 1346–1367.

[Che+11] Chandra Chekuri, Guy Even, Anupam Gupta, and Danny Segev. “Set connectivity problems in undirected graphs and the directed steiner network problem”. In: *ACM Trans. Algorithms* 7.2 (2011), 18:1–18:17.

[Che+15a] Wei Chen, Fu Li, Tian Lin, and Aviad Rubinstein. “Combining Traditional Marketing and Viral Marketing with Amphibious Influence Maximization”. In: *EC*. Portland, Oregon, USA: ACM, 2015, pp. 779–796.

[Che+15b] Yu Cheng, Ho Yee Cheung, Shaddin Dughmi, Ehsan Emamjomeh-Zadeh, Li Han, and Shang-Hua Teng. “Mixture Selection, Mechanism Design, and Signaling”. In: *FOCS*. 2015, pp. 1426–1445.

[Che18a] Lijie Chen. “On The Hardness of Approximate and Exact (Bichromatic) Maximum Inner Product”. In: *CCC*. 2018, 14:1–14:45.

[Che18b] Lijie Chen. “Toward Super-Polynomial Size Lower Bounds for Depth-Two Threshold Circuits”. In: *CoRR* abs/1805.10698 (2018). arXiv: `1805.10698`.

[CHK11] Moses Charikar, MohammadTaghi Hajiaghayi, and Howard J. Karloff. “Improved Approximation Algorithms for Label Cover Problems”. In: *Algorithmica* 61.1 (2011), pp. 190–206.

[CHK13] Rajesh Hemant Chitnis, MohammadTaghi Hajiaghayi, and Guy Kortsarz. “Fixed-Parameter and Approximation Algorithms: A New Look”. In: *IPEC*. 2013, pp. 110–122.

[Chl+16] Eden Chlamtác, Michael Dinitz, Christian Konrad, Guy Kortsarz, and George Rabanca. “The Densest k-Subhypergraph Problem”. In: *APPROX/RANDOM*. 2016, 6:1–6:19.

[Chl+17a] Eden Chlamtác, Michael Dinitz, Guy Kortsarz, and Bundit Laekhanukit. “Approximating Spanners and Directed Steiner Forest: Upper and Lower Bounds”. In: *SODA*. 2017, pp. 534–553.

[Chl+17b] Eden Chlamtác, Pasin Manurangsi, Dana Moshkovitz, and Aravindan Vijayaraghavan. “Approximation Algorithms for Label Cover and The Log-Density Threshold”. In: *SODA*. 2017, pp. 900–919.

[Chu+15] Julia Chuzhoy, Yury Makarychev, Aravindan Vijayaraghavan, and Yuan Zhou. “Approximation Algorithms and Hardness of the k-Route Cut Problem”. In: *ACM Trans. Algorithms* 12.1 (Dec. 2015), 2:1–2:40.

[Chv79] Vasek Chvátal. “A Greedy Heuristic for the Set-Covering Problem”. In: *Math. Oper. Res.* 4.3 (1979), pp. 233–235.

[CL15] Chandra Chekuri and Shi Li. “A note on the hardness of approximating the k-WAY HYPERGRAPH CUT problem”. Unpublished Manuscript. 2015.

[CL16] Yijia Chen and Bingkai Lin. “The Constant Inapproximability of the Parameterized Dominating Set Problem”. In: *FOCS*. 2016, pp. 505–514.

[CL99] Edith Cohen and David D. Lewis. “Approximating Matrix Multiplication for Pattern Recognition Tasks”. In: *J. Algorithms* 30.2 (1999), pp. 211–252.

[CLN13] Parinya Chalermsook, Bundit Laekhanukit, and Danupon Nanongkai. “Independent Set, Induced Matching, and Pricing: Connections and Tight (Subexponential Time) Approximation Hardnesses”. In: *FOCS*. 2013, pp. 370–379.

[CM18] Eden Chlamtác and Pasin Manurangsi. “Sherali-Adams Integrality Gaps Matching the Log-Density Threshold”. In: *APPROX*. 2018, 10:1–10:19.

[CMY08] Graham Cormode, S. Muthukrishnan, and Ke Yi. “Algorithms for distributed functional monitoring”. In: *SODA*. 2008, pp. 1076–1085.

[CN99] Jin-yi Cai and Ajay Nerurkar. “Approximating the SVP to within a Factor $(1 + 1/\dim^{\xi})$ Is NP-Hard under Randomized Reductions”. In: *J. Comput. Syst. Sci.* 59.2 (1999), pp. 221–239.

[Coo71] Stephen A. Cook. “The Complexity of Theorem-Proving Procedures”. In: *STOC*. 1971, pp. 151–158.

[Coo88] Stephen A. Cook. “Short Propositional Formulas Represent Nondeterministic Computations”. In: *Inf. Process. Lett.* 26.5 (Jan. 1988), pp. 269–270.

[Cor+09] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. *Introduction to Algorithms, Third Edition*. 3rd. The MIT Press, 2009.

[CRR14] Arkadev Chattopadhyay, Jaikumar Radhakrishnan, and Atri Rudra. "Topology Matters in Communication". In: *FOCS*. 2014, pp. 631–640.

[CT12] Eden Chlamtác and Madhur Tulsiani. "Handbook on Semidefinite, Conic and Polynomial Optimization". In: ed. by F. Miguel Anjos and B. Jean Lasserre. Boston, MA: Springer US, 2012. Chap. Convex Relaxations and Integrality Gaps, pp. 139–169.

[CW12a] Qi Cheng and Daqing Wan. "A Deterministic Reduction for the Gap Minimum Distance Problem". In: *IEEE Trans. Information Theory* 58.11 (2012), pp. 6935–6941.

[CW12b] Qi Cheng and Daqing Wan. "A Deterministic Reduction for the Gap Minimum Distance Problem". In: *IEEE Trans. Information Theory* 58.11 (2012), pp. 6935–6941.

[CW19] Lijie Chen and Ryan Williams. "An Equivalence Class for Orthogonal Vectors". In: *SODA*. 2019, pp. 21–40.

[Cyg+14] Marek Cygan, Fedor Fomin, Bart MP Jansen, Lukasz Kowalik, Daniel Lokshtanov, Dániel Marx, Marcin Pilipczuk, and Saket Saurabh. "Open problems for fpt school 2014". 2014.

[Cyg+15] Marek Cygan, Fedor V. Fomin, Lukasz Kowalik, Daniel Lokshtanov, Dániel Marx, Marcin Pilipczuk, Michal Pilipczuk, and Saket Saurabh. *Parameterized Algorithms*. Springer, 2015.

[Cyg+17] Marek Cygan, Fedor V. Fomin, Danny Hermelin, and Magnus Wahlström. "Randomization in Parameterized Complexity (Dagstuhl Seminar 17041)". In: *Dagstuhl Reports* 7.1 (2017), pp. 103–128.

[CZ15] Stephen R. Chestnut and Rico Zenklusen. "Hardness and Approximation for Network Flow Interdiction". In: *CoRR* abs/1511.02486 (2015).

[Dan16] Amit Daniely. "Complexity theoretic limitations on learning halfspaces". In: *STOC*. 2016, pp. 105–117.

[Dem+07] Erik D. Demaine, Gregory Gutin, Dániel Marx, and Ulrike Stege. "07281 Open Problems – Structure Theory and FPT Algorithmcs for Graphs, Digraphs and Hypergraphs". In: *Structure Theory and FPT Algorithmics for Graphs, Digraphs and Hypergraphs, 08.07. - 13.07.2007*. 2007.

[DF13] Rodney G. Downey and Michael R. Fellows. *Fundamentals of Parameterized Complexity*. Texts in Computer Science. Springer, 2013.

[DF95a] Rodney G. Downey and Michael R. Fellows. "Fixed-Parameter Tractability and Completeness I: Basic Results". In: *SIAM J. Comput.* 24.4 (1995), pp. 873–921.

[DF95b] Rodney G. Downey and Michael R. Fellows. "Fixed-Parameter Tractability and Completeness II: On Completeness for W[1]". In: *Theor. Comput. Sci.* 141.1&2 (1995), pp. 109–131.

[DF99] Rodney G. Downey and Michael R. Fellows. *Parameterized Complexity*. Monographs in Computer Science. Springer, 1999.

[DFM06] Rodney G. Downey, Michael R. Fellows, and Catherine McCartin. “Parameterized Approximation Problems”. In: *IWPEC*. 2006, pp. 121–129.

[DFS16] Argyrios Deligkas, John Fearnley, and Rahul Savani. “Inapproximability Results for Approximate Nash Equilibria”. In: *WINE*. 2016, pp. 29–43.

[DG08] Irit Dinur and Elazar Goldenberg. “Locally Testing Direct Product in the Low Error Range”. In: *FOCS*. 2008, pp. 613–622.

[DHK15] Irit Dinur, Prahladh Harsha, and Guy Kindler. “Polynomially Low Error PCPs with polyloglog n Queries via Modular Composition”. In: *STOC*. 2015, pp. 267–276.

[Din+03] Irit Dinur, Guy Kindler, Ran Raz, and Shmuel Safra. “Approximating CVP to Within Almost-Polynomial Factors is NP-Hard”. In: *Combinatorica* 23.2 (2003), pp. 205–243.

[Din+11] Irit Dinur, Eldar Fischer, Guy Kindler, Ran Raz, and Shmuel Safra. “PCP Characterizations of NP: Toward a Polynomially-Small Error-Probability”. In: *Computational Complexity* 20.3 (2011), pp. 413–504.

[Din02] Irit Dinur. “Approximating SVP_∞ to within almost-polynomial factors is NP-hard”. In: *Theor. Comput. Sci.* 285.1 (2002), pp. 55–71.

[Din07] Irit Dinur. “The PCP theorem by gap amplification”. In: *J. ACM* 54.3 (2007), p. 12.

[Din16] Irit Dinur. “Mildly exponential reduction from gap 3SAT to polynomial-gap label-cover”. In: *ECCC* 23 (2016), p. 128.

[DK17] Irit Dinur and Tali Kaufman. “High dimensional expanders imply agreement expanders”. In: *ECCC* 24 (2017), p. 89.

[DK99] Yevgeniy Dodis and Sanjeev Khanna. “Design Networks with Bounded Pairwise Distance”. In: *STOC*. 1999, pp. 750–759.

[DKL18] Roee David, Karthik C. S., and Bundit Laekhanukit. “On the Complexity of Closest Pair via Polar-Pair of Point-Sets”. In: *SoCG*. 2018, 28:1–28:15.

[DM18] Irit Dinur and Pasin Manurangsi. “ETH-Hardness of Approximating 2-CSPs and Directed Steiner Network”. In: *ITCS*. 2018, 36:1–36:20.

[DMS03] Ilya Dumer, Daniele Micciancio, and Madhu Sudan. “Hardness of approximating the minimum distance of a linear code”. In: *IEEE Trans. Information Theory* 49.1 (2003), pp. 22–37.

[DMZ05] William Duckworth, David Manlove, and Michele Zito. “On the approximability of the maximum induced matching problem”. In: *J. Discrete Algorithms* 3.1 (2005), pp. 79–91.

[DN17] Irit Dinur and Inbal Livni Navon. “Exponentially Small Soundness for the Direct Product Z-Test”. In: *CCC*. 2017, 29:1–29:50.

[Dol+14] Thorsten Doliwa, Gaojian Fan, Hans Ulrich Simon, and Sandra Zilles. "Recursive teaching dimension, VC-dimension and sample compression". In: *Journal of Machine Learning Research* 15.1 (2014), pp. 3107–3131.

[Dow+08] Rodney G. Downey, Michael R. Fellows, Catherine McCartin, and Frances A. Rosamond. "Parameterized approximation of dominating set problems". In: *Inf. Process. Lett.* 109.1 (2008), pp. 68–70.

[Dow+99] Rodney G. Downey, Michael R. Fellows, Alexander Vardy, and Geoff Whittle. "The Parametrized Complexity of Some Fundamental Problems in Coding Theory". In: *SIAM J. Comput.* 29.2 (1999), pp. 545–570.

[DS14] Irit Dinur and David Steurer. "Analytical approach to parallel repetition". In: *STOC*. 2014, pp. 624–633.

[DS16] Amit Daniely and Shai Shalev-Shwartz. "Complexity Theoretic Limitations on Learning DNF's". In: *COLT*. 2016, pp. 815–830.

[Dug14] Shaddin Dughmi. "On the Hardness of Signaling". In: *FOCS*. 2014, pp. 354–363.

[EG04] Friedrich Eisenbrand and Fabrizio Grandoni. "On the complexity of fixed parameter clique and dominating set". In: *Theor. Comput. Sci.* 326.1-3 (2004), pp. 57–67.

[EGG08] Kord Eickmeyer, Martin Grohe, and Magdalena Grüber. "Approximation of Natural W[P]-Complete Minimisation Problems Is Hard". In: *CCC*. 2008, pp. 8–18.

[EH00] Lars Engebretsen and Jonas Holmerin. "Clique Is Hard to Approximate within $n^{1-o(1)}$". In: *ICALP*. 2000, pp. 2–12.

[Elb+09] Khaled M. Elbassioni, Rajiv Raman, Saurabh Ray, and René Sitters. "On the approximability of the maximum feasible subsystem problem with 0/1-coefficients". In: *SODA*. 2009, pp. 1210–1219.

[Emd81] Peter van Emde-Boas. *Another NP-complete partition problem and the complexity of computing short vectors in a lattice*. Report. Department of Mathematics. University of Amsterdam. Department, Univ., 1981.

[Fei+91] Uriel Feige, Shafi Goldwasser, László Lovász, Shmuel Safra, and Mario Szegedy. "Approximating Clique is Almost NP-complete (Preliminary Version)". In: *SFCS*. San Juan, Puerto Rico: IEEE Computer Society, 1991, pp. 2–12.

[Fei02] Uriel Feige. "Relations Between Average Case Complexity and Approximation Complexity". In: *STOC*. Montreal, Quebec, Canada, 2002, pp. 534–543.

[Fei04] Uriel Feige. "Approximating Maximum Clique by Removing Subgraphs". In: *SIAM J. Discrete Math.* 18.2 (2004), pp. 219–225.

[Fei98] Uriel Feige. "A Threshold of ln *n* for Approximating Set Cover". In: *J. ACM* 45.4 (1998), pp. 634–652.

[Fel+06] Vitaly Feldman, Parikshit Gopalan, Subhash Khot, and Ashok Kumar Ponnuswami. "New Results for Learning Noisy Parities and Halfspaces". In: *FOCS*. 2006, pp. 563–574.

[Fel+12] Michael R. Fellows, Jiong Guo, Dániel Marx, and Saket Saurabh. "Data Reduction and Problem Kernels (Dagstuhl Seminar 12241)". In: *Dagstuhl Reports* 2.6 (2012), pp. 26–50.

[FG06] Jörg Flum and Martin Grohe. *Parameterized Complexity Theory*. Texts in Theoretical Computer Science. An EATCS Series. Springer, 2006.

[FK00] Uriel Feige and Joe Kilian. "Two-Prover Protocols - Low Error at Affordable Rates". In: *SIAM J. Comput.* 30.1 (2000), pp. 324–346.

[FK05] Uriel Feige and Shimon Kogan. "The hardness of approximating hereditary properties". Technical Report. 2005.

[FK96] Alan M. Frieze and Ravi Kannan. "The Regularity Lemma and Approximation Schemes for Dense Problems". In: *FOCS*. 1996, pp. 12–20.

[FKN12] Moran Feldman, Guy Kortsarz, and Zeev Nutov. "Improved approximation algorithms for Directed Steiner Forest". In: *J. Comput. Syst. Sci.* 78.1 (2012), pp. 279–292.

[FKP01] Uriel Feige, Guy Kortsarz, and David Peleg. "The Dense k-Subgraph Problem". In: *Algorithmica* 29.3 (2001), pp. 410–421.

[FL01] Uriel Feige and Michael Langberg. "Approximation Algorithms for Maximization Problems Arising in Graph Partitioning". In: *J. Algorithms* 41.2 (Nov. 2001), pp. 174–211.

[FL98] Moti Frances and Ami Litman. "Optimal Mistake Bound Learning is Hard". In: *Inf. Comput.* 144.1 (1998), pp. 66–82.

[FLP16] Dimitris Fotakis, Michael Lampis, and Vangelis Th. Paschos. "Sub-exponential Approximation Schemes for CSPs: From Dense to Almost Sparse". In: *STACS*. 2016, 37:1–37:14.

[FM12] Fedor V. Fomin and Dániel Marx. "FPT Suspects and Tough Customers: Open Problems of Downey and Fellows". In: *The Multivariate Algorithmic Revolution and Beyond - Essays Dedicated to Michael R. Fellows on the Occasion of His 60th Birthday*. Ed. by Hans L. Bodlaender, Rod Downey, Fedor V. Fomin, and Dániel Marx. Vol. 7370. Lecture Notes in Computer Science. Springer, 2012, pp. 457–468.

[FM86] Peter Frankl and Hiroshi Maehara. "Embedding the n-cube in Lower Dimensions". In: *Eur. J. Comb.* 7.3 (1986), pp. 221–225.

[FM88] Peter Frankl and Hiroshi Maehara. "On the Contact Dimensions of Graphs". In: *Discrete & Computational Geometry* 3 (1988), pp. 89–96.

[FOZ16] Orr Fischer, Rotem Oshman, and Uri Zwick. "Public vs. Private Randomness in Simultaneous Multi-party Communication Complexity". In: *SIROCCO*. 2016, pp. 60–74.

[FS97] Uriel Feige and Michael Seltser. *On the densest k-subgraph problem*. Tech. rep. Weizmann Institute of Science, Rehovot, Israel, 1997.

[FSM19] Piotr Faliszewski, Krzysztof Sornat, and Pasin Manurangsi. "Approximation and Hardness of Shift-Bribery". In: *AAAI*. To appear. 2019.

[Gal14] François Le Gall. "Powers of tensors and fast matrix multiplication". In: *ISSAC*. 2014, pp. 296–303.

[Gil52] E. N. Gilbert. "A comparison of signalling alphabets". In: *Bell System Technical Journal* 31 (1952), pp. 504–522.

[GJ79] Michael R. Garey and David S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. New York, NY, USA: W. H. Freeman & Co., 1979.

[GK18] Elazar Goldenberg and Karthik C. S. "Towards a General Direct Product Testing Theorem". In: *FSTTCS*. 2018, 11:1–11:17.

[GKS12] Petr A. Golovach, Jan Kratochvíl, and Ondrej Suchý. "Parameterized complexity of generalized domination problems". In: *Discrete Applied Mathematics* 160.6 (2012), pp. 780–792.

[GL09] Doron Goldstein and Michael Langberg. "The Dense k Subgraph problem". In: *CoRR* abs/0912.5327 (2009).

[GO95] Anka Gajentaan and Mark H. Overmars. "On a Class of O(n^2) Problems in Computational Geometry". In: *Comput. Geom.* 5 (1995), pp. 165–185.

[Gol+99] Oded Goldreich, Daniele Micciancio, Shmuel Safra, and Jean-Pierre Seifert. "Approximating Shortest Lattice Vectors is not Harder than Approximating Closest Lattice Vectors". In: *Inf. Process. Lett.* 71.2 (1999), pp. 55–61.

[Gol08] Oded Goldreich. *Computational Complexity: A Conceptual Perspective*. 1st ed. New York, NY, USA: Cambridge University Press, 2008.

[GR18] Oded Goldreich and Guy N. Rothblum. "Simple Doubly-Efficient Interactive Proof Systems for Locally-Characterizable Sets". In: *ITCS*. 2018, 18:1–18:19.

[Gri01] Dima Grigoriev. "Linear lower bound on degrees of Positivstellensatz calculus proofs for the parity". In: *Theor. Comput. Sci.* 259.1-2 (2001), pp. 613–622.

[GS11] Venkatesan Guruswami and Ali Kemal Sinop. "Lasserre Hierarchy, Higher Eigenvalues, and Approximation Schemes for Graph Partitioning and Quadratic Integer Programming with PSD Objectives". In: *FOCS*. IEEE Computer Society, 2011, pp. 482–491.

[GS17] Omer Gold and Micha Sharir. "Dominance Product and High-Dimensional Closest Pair under L_infty". In: *ISAAC*. 2017, 39:1–39:12.

[GS96] Arnaldo Garcia and Henning Stichtenoth. "On the Asymptotic Behaviour of Some Towers of Function Fields over Finite Fields". In: *Journal of Number Theory* 61.2 (1996), pp. 248–273.

[Gup+19] Anupam Gupta, Euiwoong Lee, Jason Li, Pasin Manurangsi, and Michal Wlodarczyk. "Losing Treewidth by Separating Subsets". In: *SODA*. 2019, pp. 1731–1749.

[Haj+06] Mohammad Taghi Hajiaghayi, Kamal Jain, Lap Chi Lau, Ion I. Mandoiu, Alexander Russell, and Vijay V. Vazirani. "Minimum Multicolored Subgraph Problem in Multiplex PCR Primer Set Selection and Population Haplotyping". In: *ICCS*. 2006, pp. 758–766.

[Hal00] Magnús M. Halldórsson. "Approximations of Weighted Independent Set and Hereditary Subset Problems". In: *J. Graph Algorithms Appl.* 4.1 (2000).

[Hås01] Johan Håstad. "Some optimal inapproximability results". In: *J. ACM* 48.4 (2001), pp. 798–859.

[Hås96] Johan Håstad. "Clique is Hard to Approximate Within $n^{1-\varepsilon}$". In: *FOCS*. 1996, pp. 627–636.

[Hen06] Tomislav Hengl. "Finding the right pixel size". In: *Computers & Geosciences* 32.9 (2006), pp. 1283–1298.

[HIM11] Koki Hamada, Kazuo Iwama, and Shuichi Miyazaki. "The Hospitals/Residents Problem with Quota Lower Bounds". In: *ESA*. 2011, pp. 180–191.

[HJ06] Mohammad Taghi Hajiaghayi and Kamal Jain. "The Prize-collecting Generalized Steiner Tree Problem via a New Approach of Primal-dual Schema". In: *SODA*. Miami, Florida: Society for Industrial and Applied Mathematics, 2006, pp. 631–640.

[HK11] Elad Hazan and Robert Krauthgamer. "How Hard Is It to Approximate the Best Nash Equilibrium?" In: *SIAM Journal on Computing* 40.1 (2011), pp. 79–91.

[HKK13] Mohammad Taghi Hajiaghayi, Rohit Khandekar, and Guy Kortsarz. "The Foundations of Fixed Parameter Inapproximability". In: *CoRR* abs/1310.2711 (2013).

[HM13] Aram W. Harrow and Ashley Montanaro. "Testing Product States, Quantum Merlin-Arthur Games and Tensor Optimization". In: *J. ACM* 60.1 (Feb. 2013), 3:1–3:43.

[HNS88] Klaus H. Hinrichs, Jürg Nievergelt, and Peter Schorn. "Plane-Sweep Solves the Closest Pair Problem Elegantly". In: *Inf. Process. Lett.* 26.5 (1988), pp. 255–261.

[Hoc59] Alexis Hocquenghem. "Codes correcteurs d'erreurs". In: *Chiffres* 2 (Sept. 1959), pp. 147–156.

[Hol09] Thomas Holenstein. "Parallel Repetition: Simplification and the No-Signaling Case". In: *Theory of Computing* 5.1 (2009), pp. 141–172.

[HR07] Ishay Haviv and Oded Regev. "Tensor-based hardness of the shortest vector problem to within almost polynomial factors". In: *STOC*. 2007, pp. 469–477.

[IJK09] Russell Impagliazzo, Ragesh Jaiswal, and Valentine Kabanets. "Approximate List-Decoding of Direct Product Codes and Uniform Hardness Amplification". In: *SIAM J. Comput.* 39.2 (2009), pp. 564–605.

[IKW12] Russell Impagliazzo, Valentine Kabanets, and Avi Wigderson. "New Direct-Product Testers and 2-Query PCPs". In: *SIAM J. Comput.* 41.6 (2012), pp. 1722–1768.

[IM98] Piotr Indyk and Rajeev Motwani. “Approximate Nearest Neighbors: Towards Removing the Curse of Dimensionality”. In: *STOC*. 1998, pp. 604–613.

[Imp+10] Russell Impagliazzo, Ragesh Jaiswal, Valentine Kabanets, and Avi Wigderson. “Uniform Direct Product Theorems: Simplified, Optimized, and Derandomized”. In: *SIAM J. Comput.* 39.4 (2010), pp. 1637–1665.

[Ind+04] Piotr Indyk, Moshe Lewenstein, Ohad Lipsky, and Ely Porat. “Closest Pair Problems in Very High Dimensions”. In: *ICALP*. 2004, pp. 782–792.

[Ind00] Piotr Indyk. “Dimensionality reduction techniques for proximity problems”. In: *SODA*. 2000, pp. 371–378.

[IP01] Russell Impagliazzo and Ramamohan Paturi. “On the Complexity of k-SAT”. In: *J. Comput. Syst. Sci.* 62.2 (2001), pp. 367–375.

[IPZ01] Russell Impagliazzo, Ramamohan Paturi, and Francis Zane. “Which Problems Have Strongly Exponential Complexity?” In: *J. Comput. Syst. Sci.* 63.4 (2001), pp. 512–530.

[Jer92] Mark Jerrum. “Large Cliques Elude the Metropolis Process”. In: *Random Structures & Algorithms* 3.4 (1992), pp. 347–359.

[JKR19] Vishesh Jain, Frederic Koehler, and Andrej Risteski. “Mean-field approximation, convex hierarchies, and the optimality of correlation rounding: a unified perspective”. In: *STOC*. to appear. 2019.

[JL84] William Johnson and Joram Lindenstrauss. “Extensions of Lipschitz mappings into a Hilbert space”. In: *Conference in modern analysis and probability (New Haven, Conn., 1982)*. Vol. 26. Contemporary Mathematics. American Mathematical Society, 1984, pp. 189–206.

[Joh74] David S. Johnson. “Approximation Algorithms for Combinatorial Problems”. In: *J. Comput. Syst. Sci.* 9.3 (1974), pp. 256–278.

[Joh87] David S. Johnson. “The NP-completeness Column: An Ongoing Guide”. In: *J. Algorithms* 8.5 (Sept. 1987), pp. 438–448.

[Jus72] Jørn Justesen. “Class of constructive asymptotically good algebraic codes”. In: *IEEE Trans. Information Theory* 18.5 (1972), pp. 652–656.

[Kal+08] Adam Tauman Kalai, Adam R. Klivans, Yishay Mansour, and Rocco A. Servedio. “Agnostically Learning Halfspaces”. In: *SIAM J. Comput.* 37.6 (2008), pp. 1777–1805.

[Kan92] Viggo Kann. “On the Approximability of NP-complete Optimization Problems”. PhD thesis. Royal Institute of Technology, 1992.

[Kar72] Richard M. Karp. “Reducibility Among Combinatorial Problems”. In: *Proceedings of a symposium on the Complexity of Computer Computations, held March 20-22, 1972, at the IBM Thomas J. Watson Research Center, Yorktown Heights, New York.* 1972, pp. 85–103.

[Kay14] Neeraj Kayal. *Solvability of Systems of Polynomial Equations over Finite Fields*. A talk given by Neeraj Kayal at the Simons Institute for the Theory of Computing, Berkeley, CA [Accessed: 2017/20/7]. Oct. 2014.

[Kha93] Michael Kharitonov. “Cryptographic hardness of distribution-specific learning”. In: *STOC*. 1993, pp. 372–381.

[Kha95] Michael Kharitonov. “Cryptographic Lower Bounds for Learnability of Boolean Functions on the Uniform Distribution”. In: *J. Comput. Syst. Sci.* 50.3 (1995), pp. 600–610.

[Kho+07] Subhash Khot, Guy Kindler, Elchanan Mossel, and Ryan O’Donnell. “Optimal Inapproximability Results for MAX-CUT and Other 2-Variable CSPs?” In: *SIAM J. Comput.* 37.1 (2007), pp. 319–357.

[Kho01] Subhash Khot. “Improved Inaproximability Results for MaxClique, Chromatic Number and Approximate Graph Coloring”. In: *FOCS*. 2001, pp. 600–609.

[Kho02] Subhash Khot. “On the power of unique 2-prover 1-round games”. In: *STOC*. 2002, pp. 767–775.

[Kho05] Subhash Khot. “Hardness of approximating the shortest vector problem in lattices”. In: *J. ACM* 52.5 (2005), pp. 789–808.

[Kho06] Subhash Khot. “Ruling Out PTAS for Graph Min-Bisection, Dense k-Subgraph, and Bipartite Clique”. In: *SIAM J. Comput.* 36.4 (2006), pp. 1025–1071.

[KKT16] Guy Kindler, Alexandra Kolla, and Luca Trevisan. “Approximation of non-boolean 2CSP”. In: *SODA*. 2016, pp. 1705–1714.

[Kle97] Jon M. Kleinberg. “Two Algorithms for Nearest-Neighbor Search in High Dimensions”. In: *STOC*. 1997, pp. 599–608.

[Kli16] Adam R. Klivans. “Cryptographic Hardness of Learning”. In: *Encyclopedia of Algorithms*. 2016, pp. 475–477.

[KLM18] Karthik C. S., Bundit Laekhanukit, and Pasin Manurangsi. “On the parameterized complexity of approximating dominating set”. In: *STOC*. 2018, pp. 1283–1296.

[KLN99] Drago Krznaric, Christos Levcopoulos, and Bengt J. Nilsson. “Minimum Spanning Trees in d Dimensions”. In: *Nord. J. Comput.* 6.4 (1999), pp. 446–461.

[KM19] Karthik C. S. and Pasin Manurangsi. “On Closest Pair in Euclidean Metric: Monochromatic is as Hard as Bichromatic”. In: *ITCS*. 2019, 17:1–17:16.

[KM95] Samir Khuller and Yossi Matias. “A Simple Randomized Sieve Algorithm for the Closest-Pair Problem”. In: *Inf. Comput.* 118.1 (1995), pp. 34–37.

[KN97] Eyal Kushilevitz and Noam Nisan. *Communication Complexity*. New York, NY, USA: Cambridge University Press, 1997.

[Ko18] Young Kun Ko. “On Symmetric Parallel Repetition : Towards Equivalence of MAX-CUT and UG”. In: *ECCC* 25 (2018), p. 34.

[Kop13] Swastik Kopparty. *Lecture 5: k-wise independent hashing and applications*. Lecture notes for Topics in Complexity Theory and Pseudorandomness. Rutgers University, 2013.

[Kor+11] Guy Kortsarz, Vahab S. Mirrokni, Zeev Nutov, and Elena Tsanko. "Approximating Minimum-Power Degree and Connectivity Problems". In: *Algorithmica* 60.4 (2011), pp. 735–742.

[KP06] Subhash Khot and Ashok Kumar Ponnuswami. "Better Inapproximability Results for MaxClique, Chromatic Number and Min-3Lin-Deletion". In: *ICALP*. 2006, pp. 226–237.

[KP93] Guy Kortsarz and David Peleg. "On Choosing a Dense Subgraph (Extended Abstract)". In: *FOCS*. 1993, pp. 692–701.

[KR00] Subhash Khot and Venkatesh Raman. "Parameterized Complexity of Finding Subgraphs with Hereditary Properties". In: *COCOON*. 2000, pp. 137–147.

[KS07] Stavros G. Kolliopoulos and George Steiner. "Partially ordered knapsack and applications to scheduling". In: *Discrete Applied Mathematics* 155.8 (2007), pp. 889–897.

[KS09] Adam R. Klivans and Alexander A. Sherstov. "Cryptographic hardness for learning intersections of halfspaces". In: *J. Comput. Syst. Sci.* 75.1 (2009), pp. 2–12.

[KS16] Subhash Khot and Igor Shinkar. "On Hardness of Approximating the Parameterized Clique Problem". In: *ITCS*. 2016, pp. 37–45.

[KS92] Bala Kalyanasundaram and Georg Schnitger. "The Probabilistic Communication Complexity of Set Intersection". In: *SIAM J. Discrete Math.* 5.4 (1992), pp. 545–557.

[KST54] Tamás Kővári, Vera T. Sós, and Pál Turán. "On a problem of K. Zarankiewicz". eng. In: *Colloquium Mathematicae* 3.1 (1954), pp. 50–57.

[KT05] Jon Kleinberg and Éva Tardos. *Algorithm Design*. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 2005.

[Kuč95] Luděk Kučera. "Expected Complexity of Graph Partitioning Problems". In: *Discrete Appl. Math.* 57.2-3 (Feb. 1995), pp. 193–212.

[KV94] Michael J. Kearns and Leslie G. Valiant. "Cryptographic Limitations on Learning Boolean Formulae and Finite Automata". In: *J. ACM* 41.1 (1994), pp. 67–95.

[Las00] Jean B. Lasserre. "Global Optimization with Polynomials and the Problem of Moments". In: *SIAM J. on Optimization* 11.3 (Mar. 2000), pp. 796–817.

[Lau03] Monique Laurent. "A Comparison of the Sherali-Adams, Lovász-Schrijver, and Lasserre Relaxations for 0–1 Programming". In: *Math. Oper. Res.* 28.3 (July 2003), pp. 470–496.

[Lee16] Euiwoong Lee. "Partitioning a Graph into Small Pieces with Applications to Path Transversal". In: *CoRR* abs/1607.05122 (2016).

[Len83] Hendrik Willem Lenstra. “Integer Programming with a Fixed Number of Variables”. In: *Math. Oper. Res.* 8.4 (1983), pp. 538–548.

[Leo73] Leonid A. Levin. “Universal Sequential Search Problems”. In: *Probl. Peredachi Inf.* 9.3 (1973), pp. 115–116.

[Lin15] Bingkai Lin. “The Parameterized Complexity of *k*-Biclique”. In: *SODA*. 2015, pp. 605–615.

[Lin19] Bingkai Lin. “A Simple Gap-producing Reduction for the Parameterized Set Cover Problem”. In: *CoRR* abs/1902.03702 (2019). arXiv: `1902.03702`.

[Lit88] Nick Littlestone. “Learning Quickly When Irrelevant Attributes Abound: A New Linear-Threshold Algorithm”. In: *Mach. Learn.* 2.4 (Apr. 1988), pp. 285–318.

[LLL82] Arjen Klaas Lenstra, Hendrik Willem Lenstra, and László Lovász. “Factoring polynomials with rational coefficients”. In: *Mathematische Annalen* 261.4 (1982), pp. 515–534.

[LMM03] Richard J. Lipton, Evangelos Markakis, and Aranyak Mehta. “Playing large games using simple strategies”. In: *EC*. 2003, pp. 36–41.

[LMR91] Nathan Linial, Yishay Mansour, and Ronald L. Rivest. “Results on Learnability and the Vapnik-Chervonenkis Dimension”. In: *Inf. Comput.* 90.1 (1991), pp. 33–49.

[LMS11] Daniel Lokshtanov, Dániel Marx, and Saket Saurabh. “Lower bounds based on the Exponential Time Hypothesis”. In: *Bulletin of the EATCS* 105 (2011), pp. 41–72.

[LNV14] Zhentao Li, Manikandan Narayanan, and Adrian Vetta. “The Complexity of the Simultaneous Cluster Problem”. In: *Journal of Graph Algorithms and Applications* 18.1 (2014), pp. 1–34.

[Lok+17] Daniel Lokshtanov, M. S. Ramanujan, Saket Saurabh, and Meirav Zehavi. “Parameterized Complexity and Approximability of Directed Odd Cycle Transversal”. In: *CoRR* abs/1704.04249 (2017). arXiv: `1704.04249`.

[Lov75] L. Lovász. “On the ratio of optimal integral and fractional covers”. In: *Discrete Mathematics* 13.4 (1975), pp. 383–390.

[LRS15] James R. Lee, Prasad Raghavendra, and David Steurer. “Lower Bounds on the Size of Semidefinite Programming Relaxations”. In: *STOC*. 2015, pp. 567–576.

[LS09] Troy Lee and Adi Shraibman. “Lower Bounds in Communication Complexity”. In: *Foundations and Trends in Theoretical Computer Science* 3.4 (2009), pp. 263–398.

[Lue09] George S. Lueker. “Improved bounds on the average length of longest common subsequences”. In: *J. ACM* 56.3 (2009), 17:1–17:38.

[LV11] Guanfeng Liang and Nitin H. Vaidya. “Multiparty Equality Function Computation in Networks with Point-to-Point Links”. In: *SIROCCO*. 2011, pp. 258–269.

[LY80] John M. Lewis and Mihalis Yannakakis. “The Node-Deletion Problem for Hereditary Properties is NP-Complete”. In: *J. Comput. Syst. Sci.* 20.2 (1980), pp. 219–230.

[LY93] Carsten Lund and Mihalis Yannakakis. “The Approximation of Maximum Subgraph Problems”. In: *ICALP*. 1993, pp. 40–51.

[LY94] Carsten Lund and Mihalis Yannakakis. “On the Hardness of Approximating Minimization Problems”. In: *J. ACM* 41.5 (1994), pp. 960–981.

[Mae85] Hiroshi Maehara. “Contact patterns of equal nonoverlapping spheres”. In: *Graphs and Combinatorics* 1.1 (1985), pp. 271–282.

[Mae91] Hiroshi Maehara. “Dispersed Points and Geometric Embedding of Complete Bipartite Graphs”. In: *Discrete & Computational Geometry* 6 (1991), pp. 57–67.

[Maj17] Ruhollah Majdoddin. “Parameterized Complexity of CSP for Infinite Constraint Languages”. In: *CoRR* abs/1706.10153 (2017).

[Man15] Pasin Manurangsi. “On Approximating Projection Games”. MA thesis. Massachusetts Institute of Technology, Jan. 2015.

[Man17a] Pasin Manurangsi. “Almost-polynomial ratio ETH-hardness of approximating densest k-subgraph”. In: *STOC*. 2017, pp. 954–961.

[Man17b] Pasin Manurangsi. “Inapproximability of Maximum Edge Biclique, Maximum Balanced Biclique and Minimum k-Cut from the Small Set Expansion Hypothesis”. In: *ICALP*. 2017, 79:1–79:14.

[Man19a] Pasin Manurangsi. “A note on degree vs gap of Min-Rep Label Cover and improved inapproximability for connectivity problems”. In: *Inf. Process. Lett.* 145 (2019), pp. 24–29.

[Man19b] Pasin Manurangsi. “A Note on Max k-Vertex Cover: Faster FPT-AS, Smaller Approximate Kernel and Improved Approximation”. In: *SOSA*. 2019, 15:1–15:21.

[Man89] Udi Manber. *Introduction to Algorithms: A Creative Approach*. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 1989.

[Mar13] Dániel Marx. “Completely inapproximable monotone and antimonotone parameterized problems”. In: *J. Comput. Syst. Sci.* 79.1 (2013), pp. 144–151.

[McD89] Colin McDiarmid. *On the method of bounded differences*. Ed. by J.Editor Siemons. London Mathematical Society Lecture Note Series. Surveys in Combinatorics: Invited Papers at the Twelfth British Combinatorial Conference, Cambridge University Press, 1989, pp. 148–188.

[Mei13] Or Meir. “IP = PSPACE Using Error-Correcting Codes”. In: *SIAM J. Comput.* 42.1 (2013), pp. 380–403.

[Mic00] Daniele Micciancio. “The Shortest Vector in a Lattice is Hard to Approximate to within Some Constant”. In: *SIAM J. Comput.* 30.6 (2000), pp. 2008–2035.

[Mic01] Daniele Micciancio. “The hardness of the closest vector problem with preprocessing”. In: *IEEE Trans. Information Theory* 47.3 (2001), pp. 1212–1215.

[Mic12] Daniele Micciancio. “Inapproximability of the Shortest Vector Problem: Toward a Deterministic Reduction”. In: *Theory of Computing* 8.1 (2012), pp. 487–512.

[Mic14] Daniele Micciancio. “Locally Dense Codes”. In: *CCC*. 2014, pp. 90–97.

[MM15] Pasin Manurangsi and Dana Moshkovitz. “Approximating Dense Max 2-CSPs”. In: *APPROX*. 2015, pp. 396–415.

[MM17] Pasin Manurangsi and Dana Moshkovitz. “Improved Approximation Algorithms for Projection Games”. In: *Algorithmica* 77.2 (2017), pp. 555–594.

[MNP07] Rajeev Motwani, Assaf Naor, and Rina Panigrahy. “Lower Bounds on Locality Sensitive Hashing”. In: *SIAM J. Discrete Math.* 21.4 (2007), pp. 930–935.

[MNT16] Pasin Manurangsi, Preetum Nakkiran, and Luca Trevisan. “Near-Optimal UGC-hardness of Approximating Max k-CSP_R”. In: *APPROX*. 2016, 15:1–15:28.

[MOO05] Elchanan Mossel, Ryan O’Donnell, and Krzysztof Oleszkiewicz. “Noise stability of functions with low influences invariance and optimality”. In: *FOCS*. 2005, pp. 21–30.

[Mor+15] Shay Moran, Amir Shpilka, Avi Wigderson, and Amir Yehudayoff. “Compressing and Teaching for Low VC-Dimension”. In: *FOCS*. 2015, pp. 40–51.

[Mos12] Dana Moshkovitz. “The Projection Games Conjecture and The NP-Hardness of $\ln n$-Approximating Set-Cover”. In: *APPROX*. Vol. 7408. 2012, pp. 276–287.

[Mos14] Dana Moshkovitz. “Parallel Repetition from Fortification”. In: *FOCS*. 2014, pp. 414–423.

[MR09] Daniele Micciancio and Oded Regev. “Lattice-based cryptography”. In: *Post-quantum cryptography*. Springer, 2009, pp. 147–191.

[MR10] Dana Moshkovitz and Ran Raz. “Two-query PCP with subconstant error”. In: *J. ACM* 57.5 (2010), 29:1–29:29.

[MR17a] Pasin Manurangsi and Prasad Raghavendra. “A Birthday Repetition Theorem and Complexity of Approximating Dense CSPs”. In: *ICALP*. 2017, 78:1–78:15.

[MR17b] Pasin Manurangsi and Aviad Rubinstein. “Inapproximability of VC Dimension and Littlestone’s Dimension”. In: *COLT*. 2017, pp. 1432–1460.

[MS08] Claire Mathieu and Warren Schudy. “Yet another algorithm for dense max cut: go greedy”. In: *SODA*. 2008, pp. 176–182.

[MS09] Hannes Moser and Somnath Sikdar. “The parameterized complexity of the induced matching problem”. In: *Discrete Applied Mathematics* 157.4 (2009), pp. 715–727.

[MS17a] Pasin Manurangsi and Warut Suksompong. “Asymptotic existence of fair divisions for groups”. In: *Mathematical Social Sciences* 89 (2017), pp. 100–108.

[MS17b] Pasin Manurangsi and Warut Suksompong. “Computing an Approximately Optimal Agreeable Set of Items”. In: *IJCAI*. 2017, pp. 338–344.

[MS19a] Pasin Manurangsi and Warut Suksompong. "Computing a small agreeable set of indivisible items". In: *Artif. Intell.* 268 (2019), pp. 96–114.

[MS19b] Pasin Manurangsi and Warut Suksompong. "When does Envy-free Allocation Exists?" In: *AAAI*. To appear. 2019.

[MS77] F. J. MacWilliams and N. J. A. Sloane. *The theory of error correcting codes.* North-Holland mathematical library: v. 16. Amsterdam ; New York : North-Holland Pub. Co. ; New York : sole distributors for the U.S.A. and Canada, Elsevier/North Holland, 1977., 1977.

[MT09] Hannes Moser and Dimitrios M. Thilikos. "Parameterized complexity of finding regular induced subgraphs". In: *J. Discrete Algorithms* 7.2 (2009), pp. 181–190.

[MT18] Pasin Manurangsi and Luca Trevisan. "Mildly Exponential Time Approximation Algorithms for Vertex Cover, Balanced Separator and Uniform Sparsest Cut". In: *APPROX*. 2018, 20:1–20:17.

[MU02] Elchanan Mossel and Christopher Umans. "On the complexity of approximating the VC dimension". In: *J. Comput. Syst. Sci.* 65.4 (2002), pp. 660–671.

[Nis94] Noam Nisan. "The Communication Complexity of Threshold Gates". In: *Proceedings of "Combinatorics, Paul Erdos is Eighty*. 1994, pp. 301–315.

[NV10] Phong Q. Nguyen and Brigitte Vallée, eds. *The LLL Algorithm - Survey and Applications*. Information Security and Cryptography. Springer, 2010.

[OWZ14] Ryan O'Donnell, Yi Wu, and Yuan Zhou. "Optimal Lower Bounds for Locality-Sensitive Hashing (Except When q is Tiny)". In: *TOCT* 6.1 (2014), 5:1–5:13.

[Pac80] Janos Pach. "Decomposition of multiple packing and covering". In: *Diskrete Geometrie* 2 Kolloq. Math. Inst. Univ. Salzburg (1980), pp. 169–178.

[Pat10] Mihai Patrascu. "Towards polynomial lower bounds for dynamic problems". In: *STOC*. 2010, pp. 603–610.

[Pel07] David Peleg. "Approximation algorithms for the Label-Cover$_{\mathrm{MAX}}$ and Red-Blue Set Cover problems". In: *J. Discrete Algorithms* 5.1 (2007), pp. 55–64.

[Pis07] David Pisinger. "The quadratic knapsack problem—a survey". In: *Discrete Applied Mathematics* 155.5 (2007), pp. 623–648.

[PS85] Franco P. Preparata and Michael I. Shamos. *Computational Geometry: An Introduction*. New York, NY, USA: Springer-Verlag New York, Inc., 1985.

[PVZ12] Jeff M. Phillips, Elad Verbin, and Qin Zhang. "Lower bounds for number-in-hand multiparty communication complexity, made easy". In: *SODA*. 2012, pp. 486–501.

[PW10] Mihai Patrascu and Ryan Williams. "On the Possibility of Faster SAT Algorithms". In: *SODA*. 2010, pp. 1065–1075.

[PY91] Christos H. Papadimitriou and Mihalis Yannakakis. "Optimization, Approximation, and Complexity Classes". In: *J. Comput. Syst. Sci.* 43.3 (1991), pp. 425–440.

[PY96] Christos H. Papadimitriou and Mihalis Yannakakis. “On Limited Nondeterminism and the Complexity of the V-C Dimension”. In: *J. Comput. Syst. Sci.* 53.2 (1996), pp. 161–170.

[Rab76] Michael O. Rabin. “Probabilistic Algorithms”. In: *Proceedings of a Symposium on New Directions and Recent Results in Algorithms and Complexity, Computer Science Department, Carnegie-Mellon University, April 7-9, 1976*. 1976, pp. 21–39.

[Rag08] Prasad Raghavendra. “Optimal algorithms and inapproximability results for every CSP?” In: *STOC*. 2008, pp. 245–254.

[Rao11] Anup Rao. “Parallel Repetition in Projection Games and a Concentration Bound”. In: *SIAM J. Comput.* 40.6 (2011), pp. 1871–1891.

[Raz17] Ilya Razenshteyn. “High-Dimensional Similarity Search and Sketching: Algorithms and Hardness”. In: *PhD Thesis, MIT* (2017).

[Raz92] Alexander A. Razborov. “On the Distributional Complexity of Disjointness”. In: *Theor. Comput. Sci.* 106.2 (1992), pp. 385–390.

[Raz98] Ran Raz. “A Parallel Repetition Theorem”. In: *SIAM J. Comput.* 27.3 (1998), pp. 763–803.

[Reg03] Oded Regev. “New lattice based cryptographic constructions”. In: *STOC*. 2003, pp. 407–416.

[Reg05] Oded Regev. “On lattices, learning with errors, random linear codes, and cryptography”. In: *STOC*. 2005, pp. 84–93.

[Reg06] Oded Regev. “Lattice-Based Cryptography”. In: *CRYPTO*. 2006, pp. 131–141.

[Reg10] Oded Regev. “The Learning with Errors Problem (Invited Survey)”. In: *CCC*. 2010, pp. 191–204.

[RR06] Oded Regev and Ricky Rosen. “Lattice problems and norm embeddings”. In: *STOC*. 2006, pp. 447–456.

[RRS89] Jan Reiterman, Vojtech Rödl, and Edita Sinajová. “Embeddings of Graphs in Euclidean Spaces”. In: *Discrete & Computational Geometry* 4 (1989), pp. 349–364.

[RS10] Prasad Raghavendra and David Steurer. “Graph Expansion and the Unique Games Conjecture”. In: *STOC*. Cambridge, Massachusetts, USA, 2010, pp. 755–764.

[RS60] Irving S. Reed and Gustave Solomon. “Polynomial Codes over Certain Finite Fields”. In: *Journal of the Society for Industrial and Applied Mathematics (SIAM)* 8.2 (1960), pp. 300–304.

[RS96] Ronitt Rubinfeld and Madhu Sudan. “Robust Characterizations of Polynomials with Applications to Program Testing”. In: *SIAM J. Comput.* 25.2 (1996), pp. 252–271.

[RS97] Ran Raz and Shmuel Safra. “A Sub-constant Error-probability Low-degree Test, and a Sub-constant Error-probability PCP Characterization of NP”. In: *STOC*. El Paso, Texas, USA: ACM, 1997, pp. 475–484.

[RST12] Prasad Raghavendra, David Steurer, and Madhur Tulsiani. “Reductions between Expansion Problems”. In: *CCC*. 2012, pp. 64–73.

[RT12] Prasad Raghavendra and Ning Tan. “Approximating CSPs with Global Cardinality Constraints Using SDP Hierarchies”. In: *SODA*. Kyoto, Japan: SIAM, 2012, pp. 373–387.

[Rub16] Aviad Rubinstein. “Settling the Complexity of Computing Approximate Two-Player Nash Equilibria”. In: *FOCS*. 2016, pp. 258–265.

[Rub17a] Aviad Rubinstein. “Detecting communities is Hard (And Counting Them is Even Harder)”. In: *ITCS*. 2017, 42:1–42:13.

[Rub17b] Aviad Rubinstein. “Honest Signaling in Zero-Sum Games Is Hard, and Lying Is Even Harder”. In: *ICALP*. 2017, 77:1–77:13.

[Rub18] Aviad Rubinstein. “Hardness of approximate nearest neighbor search”. In: *STOC*. 2018, pp. 1260–1268.

[SA90] Hanif D. Sherali and Warren P. Adams. “A Hierarchy of Relaxation Between the Continuous and Convex Hull Representations”. In: *SIAM J. Discret. Math.* 3.3 (May 1990), pp. 411–430.

[Sak10] Rishi Saket. “Quasi-Random PCP and Hardness of 2-Catalog Segmentation”. In: *FSTTCS*. 2010, pp. 447–458.

[Sch00] Marcus Schaefer. “Deciding the K-Dimension is PSPACE-Complete”. In: *CCC*. 2000, pp. 198–203.

[Sch08] Grant Schoenebeck. “Linear Level Lasserre Lower Bounds for Certain k-CSPs”. In: *FOCS*. 2008, pp. 593–602.

[Sch99] Marcus Schaefer. “Deciding the Vapnik-Cervonenkis Dimension in Σ_3^P-Complete”. In: *J. Comput. Syst. Sci.* 58.1 (1999), pp. 177–182.

[SFL15] Piotr Skowron, Piotr Faliszewski, and Jerome Lang. “Finding a Collective Set of Items: From Proportional Multirepresentation to Group Recommendation”. In: *AAAI*. Austin, Texas: AAAI Press, 2015, pp. 2131–2137.

[SH75] Michael Ian Shamos and Dan Hoey. “Closest-Point Problems”. In: *FOCS*. 1975, pp. 151–162.

[Shu+01] Kenneth W. Shum, Ilia Aleshnikov, P. Vijay Kumar, Henning Stichtenoth, and Vinay Deolalikar. “A low-complexity algorithm for the construction of algebraic-geometric codes better than the Gilbert-Varshamov bound”. In: *IEEE Trans. Information Theory* 47.6 (2001), pp. 2225–2241.

[Sin64] Richard C. Singleton. “Maximum distance q -nary codes”. In: *IEEE Trans. Information Theory* 10.2 (1964), pp. 116–118.

[Sla96] Petr Slavík. “A Tight Analysis of the Greedy Algorithm for Set Cover”. In: *STOC*. 1996, pp. 435–441.

[Sri95] Aravind Srinivasan. “Improved approximations of packing and covering problems”. In: *STOC*. 1995, pp. 268–276.

[SS96] Michael Sipser and Daniel A. Spielman. “Expander codes”. In: *IEEE Trans. Information Theory* 42.6 (1996), pp. 1710–1722.

[ST00] Alex Samorodnitsky and Luca Trevisan. “A PCP characterization of NP with optimal amortized query complexity”. In: *STOC*. 2000, pp. 191–199.

[ST08] Akiko Suzuki and Takeshi Tokuyama. “Dense Subgraph Problems with Output-density Conditions”. In: *ACM Trans. Algorithms* 4.4 (Aug. 2008), 43:1–43:18.

[Ste93] Jacques Stern. “Approximating the Number of Error Locations within a Constant Ratio is NP-complete”. In: *AAECC*. 1993, pp. 325–331.

[Sti08] Henning Stichtenoth. *Algebraic Function Fields and Codes*. 2nd. Springer Publishing Company, Incorporated, 2008.

[SV82] Larry J. Stockmeyer and Vijay V. Vazirani. “NP-Completeness of Some Generalizations of the Maximum Matching Problem”. In: *Inf. Process. Lett.* 15.1 (1982), pp. 14–19.

[SW98] Anand Srivastav and Katja Wolf. “Finding Dense Subgraphs with Semidefinite Programming”. In: *APPROX*. London, UK, UK: Springer-Verlag, 1998, pp. 181–191.

[Tov84] Craig A. Tovey. “A simplified NP-complete satisfiability problem”. In: *Discrete Applied Mathematics* 8.1 (1984), pp. 85–89.

[Tre01] Luca Trevisan. “Non-approximability results for optimization problems on bounded degree instances”. In: *STOC*. 2001, pp. 453–461.

[Tul09] Madhur Tulsiani. “CSP gaps and reductions in the lasserre hierarchy”. In: *STOC*. 2009, pp. 303–312.

[Tur41] Pál Turán. “On an extremal problem in graph theory (in Hungarian)”. In: *Matematikai és Fizikai Lapok* 48 (1941), pp. 436–452.

[TV15] Sumedh Tirodkar and Sundar Vishwanathan. “On the Approximability of the Minimum Rainbow Subgraph Problem and Other Related Problems”. In: *ISAAC*. 2015, pp. 106–115.

[Vad12] Salil P. Vadhan. “Pseudorandomness”. In: *Foundations and Trends in Theoretical Computer Science* 7.1-3 (2012), pp. 1–336.

[Val15] Gregory Valiant. “Finding Correlations in Subquadratic Time, with Applications to Learning Parities and the Closest Pair Problem”. In: *J. ACM* 62.2 (2015), 13:1–13:45.

[Var57] R. R. Varshamov. “Estimate of the number of signals in error correcting codes”. In: *Dokl. Akad. Nauk SSSR* 117 (1957), pp. 739–741.

[Var97a] Alexander Vardy. “Algorithmic Complexity in Coding Theory and the Minimum Distance Problem”. In: *STOC*. 1997, pp. 92–109.

[Var97b] Alexander Vardy. “The intractability of computing the minimum distance of a code”. In: *IEEE Trans. Information Theory* 43.6 (1997), pp. 1757–1766.

[VC71] Vladimir N. Vapnik and Alexey Ya. Chervonenkis. “On the Uniform Convergence of Relative Frequencies of Events to Their Probabilities”. In: *Theory of Probability & Its Applications* 16.2 (1971), pp. 264–280.

[Veg+05] Wenceslas Fernandez de la Vega, Marek Karpinski, Ravi Kannan, and Santosh Vempala. “Tensor Decomposition and Approximation Schemes for Constraint Satisfaction Problems”. In: *STOC*. Baltimore, MD, USA: ACM, 2005, pp. 747–754.

[Vio15] Emanuele Viola. “The communication complexity of addition”. In: *Combinatorica* 35.6 (2015), pp. 703–747.

[VK07] Wenceslas Fernandez de la Vega and Claire Kenyon-Mathieu. “Linear Programming Relaxations of Maxcut”. In: *SODA*. New Orleans, Louisiana: Society for Industrial and Applied Mathematics, 2007, pp. 53–61.

[Vlă18] Serge Vlăduţ. “Lattices with exponentially large kissing numbers”. In: *arXiv preprint arXiv:1802.00886* (2018).

[VNT07] Serge Vladut, Dmitry Nogin, and Michael Tsfasman. *Algebraic Geometric Codes: Basic Notions*. Boston, MA, USA: American Mathematical Society, 2007.

[Wil05] Ryan Williams. “A new algorithm for optimal 2-constraint satisfaction and its implications”. In: *Theor. Comput. Sci.* 348.2-3 (2005), pp. 357–365.

[Wil18a] Ryan Williams. “On the Difference Between Closest, Furthest, and Orthogonal Pairs: Nearly-Linear vs Barely-Subquadratic Complexity”. In: *SODA*. 2018, pp. 1207–1215.

[Wil18b] Virginia Vassilevska Williams. “ON SOME FINE-GRAINED QUESTIONS IN ALGORITHMS AND COMPLEXITY”. In: *Proc. Int. Cong. of Math.* Vol. 3. 2018, pp. 3431–3472.

[Won+07] Raymond Chi-Wing Wong, Yufei Tao, Ada Wai-Chee Fu, and Xiaokui Xiao. “On Efficient Spatial Matching”. In: *VLDB*. 2007, pp. 579–590.

[WW15] Omri Weinstein and David P. Woodruff. “The Simultaneous Communication of Disjointness with Applications to Data Streams”. In: *ICALP*. 2015, pp. 1082–1093.

[WW18] Virginia Vassilevska Williams and R. Ryan Williams. “Subcubic Equivalences Between Path, Matrix, and Triangle Problems”. In: *J. ACM* 65.5 (2018), 27:1–27:38.

[Yao79] Andrew Chi-Chih Yao. “Some Complexity Questions Related to Distributive Computing (Preliminary Report)”. In: *STOC*. 1979, pp. 209–213.

[Yao91] Andrew Chi-Chih Yao. “Lower Bounds for Algebraic Computation Trees with Integer Inputs”. In: *SIAM J. Comput.* 20.4 (1991), pp. 655–668.

[Yar14] Grigory Yaroslavtsev. “Going for Speed: Sublinear Algorithms for Dense r-CSPs”. In: *CoRR* abs/1407.7887 (2014).

[Yar16] Grigory Yaroslavtsev. personal communication. Mar. 2016.

[YZ14] Yuichi Yoshida and Yuan Zhou. “Approximation schemes via Sherali-Adams hierarchy for dense constraint satisfaction problems and assignment problems”. In: *Innovations in Theoretical Computer Science, ITCS’14, Princeton, NJ, USA, January 12-14, 2014*. 2014, pp. 423–438.

[Zah71] Charles T. Zahn. “Graph-Theoretical Methods for Detecting and Describing Gestalt Clusters”. In: *IEEE Trans. Computers* 20.1 (1971), pp. 68–86.

[Zar51] Kazimierz Zarankiewicz. “Problem P101 (in French)”. In: *Colloquium Mathematicum* 2 (1951), p. 301.

[Zuc07] David Zuckerman. “Linear Degree Extractors and the Inapproximability of Max Clique and Chromatic Number”. In: *Theory of Computing* 3.1 (2007), pp. 103–128.

[Zuc96a] David Zuckerman. “On Unapproximable Versions of NP-Complete Problems”. In: *SIAM J. Comput.* 25.6 (1996), pp. 1293–1304.

[Zuc96b] David Zuckerman. “Simulating BPP Using a General Weak Random Source”. In: *Algorithmica* 16.4/5 (1996), pp. 367–391.