

Generating Optimal Molecules with Synthesizability and 3D Equivariant Conformational Constraints

*Aditi Krishnapriyan, Ed.
Daniel Klein, Ed.*



Electrical Engineering and Computer Sciences
University of California, Berkeley

Technical Report No. UCB/EECS-2023-257

<http://www2.eecs.berkeley.edu/Pubs/TechRpts/2023/EECS-2023-257.html>

December 1, 2023

Copyright © 2023, by the author(s).
All rights reserved.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission.

Acknowledgement

I want to express my deepest gratitude and appreciation to all those who have supported me throughout my journey of completing this Master's thesis.

I want to thank my research advisor Professor Aditi Krishnapriyan and my Master's advisor Professor Dan Klein.

I would also like to thank Kevin Yang for his support and guidance throughout my journey into machine learning applications for small molecules.

Lastly, I would like to thank Professor Connor Coley for his insight and support for the synthesis project and its greater applications.

The work that is presented herein would not be possible without these key contributors as well as several others who have provided assistance in various ways from proofreading to helping formulate key insights and ideas.

**Generating Optimal Molecules with Synthesizability and 3D
Equivariant Conformational Constraints**

by Daniel Reidenbach

Research Project

Submitted to the Department of Electrical Engineering and Computer Sciences,
University of California at Berkeley, in partial satisfaction of the requirements for the
degree of **Master of Science, Plan II.**

Approval for the Report and Comprehensive Examination:

Committee:



Professor Dan Klein
MS Research Advisor

05/05/2023

(Date)



Professor Aditi Krishnapriyan
Research Advisor / Second Reader

05/06/2023

(Date)

Generating Optimal Molecules with Synthesizability and 3D Equivariant Conformational Constraints

Danny Reidenbach
UC Berkeley
dreidenbach@berkeley.edu

Abstract

This thesis examines two forms of constraint-driven machine learning-based molecule generation techniques. The first is BBO-SYN, a generative framework based on black-box optimization (BBO), which predicts diverse molecules with desired properties together with corresponding synthesis pathways. BBO-SYN uses recent advances in a Monte Carlo Tree Search-based latent search algorithm to locate promising reactants that produce high-scoring products when fed to a pretrained language model for chemical reaction prediction. BBO-SYN is empirically shown to produce high-scoring and diverse synthesis trees while operating over a large continuous reactant space. Similarly, after exploring synthesizability constraints, CoarsenConf was developed to generate optimal 3D low-energy conformers in an SE(3) equivariant fashion. CoarsenConf is a hierarchical graph variational autoencoder that coarsens input molecule graphs based on torsion angles to learn a subgraph level latent distribution that is used for an efficient autoregressive generation via aggregated attention. CoarsenConf predominantly outperforms state-of-the-art methods with significantly less data and training iterations on more robust benchmarks.

1 Acknowledgements

I would like to express my deepest gratitude and appreciation to all those who have supported me throughout my journey of completing this Master’s thesis. I would like to thank my research advisor Professor Aditi Krishnapriyan and my Master’s advisor Professor Dan Klein. I would also like to thank Kevin Yang for his support and guidance throughout my journey into machine learning applications for small molecules. Lastly, I would like to thank Professor Connor Coley for his insight and support for the synthesis project and its greater applications. The work that is presented herein would not be possible without these key contributors as well as several others who have provided assistance in various ways from proofreading to helping formulate key insights and ideas.

2 Synthesis Introduction

Robust automated methods for molecular design have the potential to greatly accelerate early-stage drug discovery and molecular optimization. However, while several current strategies can accelerate the filtering of 10^{60} possible drug-like molecules [Reymond and Awale, 2012] or generate novel molecules with desired properties [Kim et al., 2020b, Engkvist et al., 2021], many methods may overlook molecular synthesizability constraints, resulting in output molecules which are challenging or impossible to synthesize in practice. Despite recent work on automated retrosynthetic planners [Law et al., 2009, Segler et al., 2018a, Coley et al., 2018a, 2019a], finding viable and economically feasible synthesis pathways is still a labor- and time-intensive process.¹

¹Retrosynthesis: Given a product molecule, predict the reaction tree and reactants to synthesize it.

As many de novo generation methods can optimize for arbitrary properties given a scorer, some works have designed rule- or model-based heuristic synthesizability scores to guide optimization towards synthesizable molecules [Ertl and Schuffenhauer, 2009, Coley et al., 2018b, Segler et al., 2018a]. While such approaches are plausible in theory, generative models often exploit these heuristics in practice. Additionally, heuristics only address half the problem: even given a perfect heuristic for synthesizability, the corresponding chemical reaction steps would still be unknown.

In this work, we design a model for the task of *synthesizability-constrained molecular design*, which we define as generating not only synthesizable molecules optimized for desired chemical properties but also corresponding reaction pathways for actually creating those molecules [Gottipati et al., 2020]. In doing so, we can significantly reduce the difficulty of physically synthesizing the predicted molecules in practice.

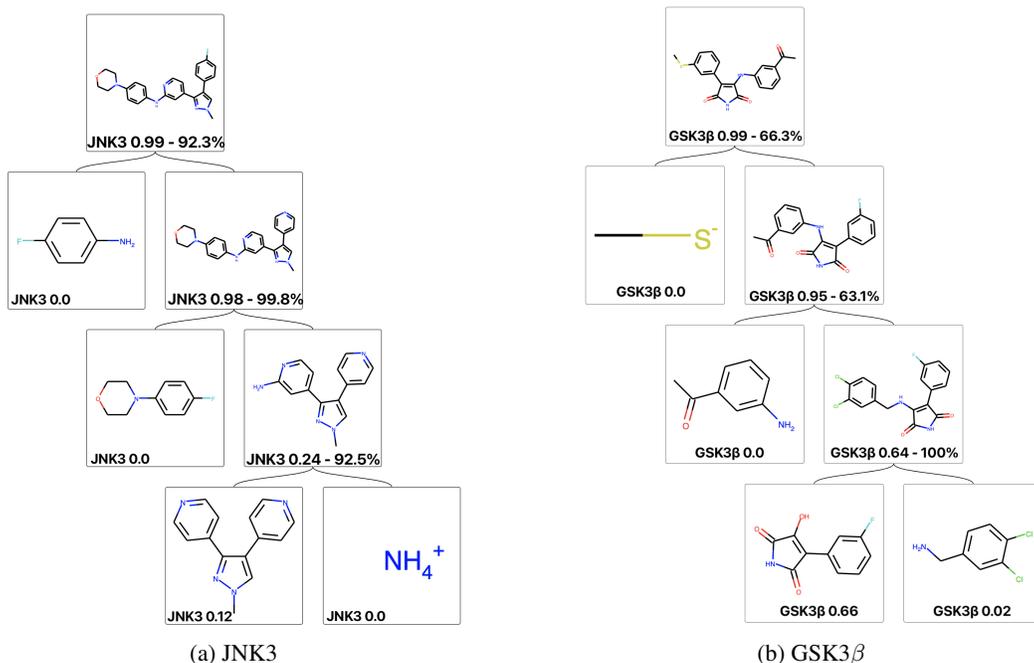


Figure 1: Example synthesis trees generated by BBO-SYN for JNK3 and GSK3 β properties. Left-side nodes are reactants found directly in the latent space via BBO-SYN. All products correspond to the top-1 Molecular Transformer prediction, with associated reaction probabilities shown.

We propose BBO-SYN, which leverages black-box optimization (BBO) to generate accurate synthesis trees with final products possessing high desired property scores. Fig. 2 illustrates the underlying workflow of BBO-SYN and how it is used to generate a synthesis tree, like in Fig. 1. BBO-SYN uses a Monte Carlo Tree Search (MCTS) based latent space partitioning algorithm, LaP³ [Yang et al., 2021], to find effective reactants for building synthesis trees. BBO-SYN improves over DAGs [Bradshaw et al., 2020], a template-free synthesis planner with discrete reactants, by converting the reactant selection problem to an optimization problem over a continuous latent space. In this way, BBO-SYN can easily handle any number of potential reactants and is agnostic to the chosen molecular representation. We use the same model-based chemical reaction predictor, Molecular Transformer [Schwaller et al., 2019], as DAGs to encourage the production of viable reaction steps, as depicted in Fig. 1. Fig. 1 illustrates the synthesis trees generated by BBO-SYN when optimizing for two distinct chemical properties. BBO-SYN progresses by adding more reactants with each step up the tree until reaching the final product. Finally, we show that BBO-SYN outperforms DAGs in four synthesis-aware property optimization tasks (QED, DRD2, GSK3 β , and JNK3), increasing the product diversity by 37% and the mean property score by 25% on our hardest JNK3 task.

3 Related Work

3.1 Black Box Optimization

BBO methods constitute a flexible class of approaches that optimize a given function with little to no assumptions on its internal structure. Classical approaches such as CEM [Rubinstein, 1999] and CMA-ES [Hansen, 2006] learn a local model around promising trajectories; however, both greedily focus on promising regions of the search space and may get trapped in local optima. Other recent approaches, such as VOOT [Kim et al., 2020a] and DOO [Munos, 2011], use a recursive region partitioning scheme to alleviate the aforementioned issues. LA-MCTS [Wang et al., 2020a] and LaP³ [Yang et al., 2021], a latent-space-based extension, further improve upon prior methods by adaptively partitioning the search regions based on sampled function values.

3.2 Chemical Reaction Prediction

A strong chemical reaction prediction system is critical for predicting viable synthesis pathways. There are two distinct methodologies for chemical reaction prediction: template-based and template-free. Template-based methods use chemical reaction rules based on subgroup pattern matching scraped from literature [Bøgevig et al., 2015, Szymkuć et al., 2016, Chen and Jung, 2021, Dai et al., 2019, Coley et al., 2017, Zhang et al., 2022b]. These methods provide approximations for feasible reactions but are limited by the availability and specificity of applicable chemical reaction templates. Template-free methods directly model chemical reactions to generalize to unseen reactions but can struggle with prediction accuracy compared to template-based methods especially when subject to out-of-domain data. Some methods rely on editing the graphical representation of molecules [Coley et al., 2019b, Sacha et al., 2021], others model the problem as a sequence-to-sequence generation problem [Schwaller et al., 2019, Lin et al., 2020, Duan et al., 2020], and more recent works leverage both representations for more efficient reaction prediction [Tu and Coley, 2022]. We employ template-free methods for reaction prediction because such methods can in principle generalize beyond a limited number of available reaction templates. BBO-SYN takes advantage of such generalizability to optimize over its continuous molecular latent space.

3.3 Continuous Molecule Representations

As it is computationally infeasible to enumerate every drug-like molecule for de novo generation and molecular design, many methods choose to optimize over a fixed-sized continuous vector molecule representation. Such representations unlock a wide array of complex and in some cases fully differentiable optimization techniques that are intractable over a discrete set of molecules [Gómez-Bombarelli et al., 2018]. For example, both SMILES-VAE [Gómez-Bombarelli et al., 2018] and MolMIM [Reidenbach et al., 2022] are models that generate novel molecules, optimized for molecular properties directly in their respective latent spaces.

Following this logic, to take advantage of recent advancements in BBO, BBO-SYN employs LaP³ over a continuous latent space. We chose to work with HierVAE’s [Jin et al., 2020b] molecular representation as it has been well-benchmarked on several property-guided optimization tasks. HierVAE breaks down input 2D molecule graphs into common subgraph motifs, building a hierarchical auto-encoder for autoregressive molecule generation. BBO-SYN expands on the prior idea of direct latent optimization by leveraging BBO in an iterative mechanism designed explicitly for synthesizability-constrained generation.

3.4 Synthesizability-Constrained Generation

As synthesizability-constrained generation is rooted in reaction prediction, there exist both template-based and template-free methods. Several early methods, such as SYNOPSIS [Vinkers et al., 2003] and DOGS [Hartenfeller et al., 2012], combine discrete synthetic building blocks for molecular design. RL methods such as PGFS [Gottipati et al., 2020], REACTOR [Horwood and Noutahi, 2020], and SynNet [Gao et al., 2022] use reaction templates to form a discrete action space for an actor-critic algorithm to generate optimal synthesis trees. Popular template-free methods include ChemBO [Korovina et al., 2020], which uses Bayesian optimization, and MoleculeChef [Bradshaw et al., 2019], which leverages latent gradients over fused reactant embeddings. Building on MoleculeChef, DAGs [Bradshaw et al., 2020] uses an iterative RL finetuning scheme over whole synthesis tree embeddings

for synthesizability-constrained molecular property optimization. While DAGs limits its entire generative process to a small discrete set of reactants, BBO-SYN uses LaP³, an MCTS-based latent search algorithm, to locate optimal reactants for building synthesis trees. Unlike DAGs, BBO-SYN can be scaled to handle extremely large reactant sets with little increase in computational cost due to its use of BBO and continuous molecular representations.

Due to the inherent tree structure of synthesizability-constrained generation, several methods use MCTS to generate plausible synthesis trees. AutoSynRoute [Lin et al., 2020] and SMC [Zhang et al., 2022b] use MCTS to explore pathways in template-free retrosynthesis and generate template-based reaction networks based on stacks of linear reactions. Several methods combine MCTS with RL to create dynamic synthesis solutions. Segler et al. [2018b] uses MCTS with an expansion policy network to guide retrosynthetic pathways toward buyable reactants. Similarly, Wang et al. [2020b] integrates a learned value function with MCTS to discover efficient and safe synthesis pathways. In contrast, BBO-SYN employs MCTS in a fundamentally different role in the *inner* optimization loop for reactant selection. We note that modeling entire synthesis trees via MCTS is orthogonal to our proposed method; combining BBO-SYN with existing MCTS approaches would likely further improve performance in exchange for higher compute costs.

4 Methods

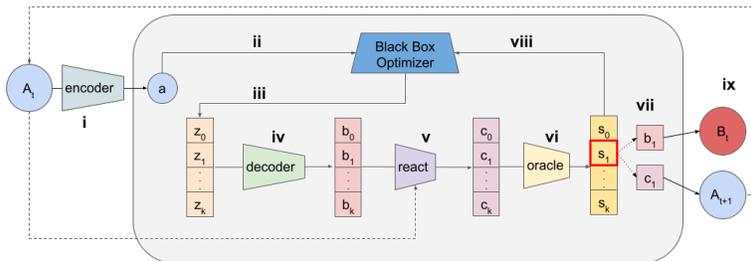


Figure 2: Internal diagram of BBO-SYN : (i) The input molecule A_t is encoded and then (ii) passed to the black-box optimizer. (iii) The optimizer generates $k+1$ latent points, which are then (iv) decoded into SMILES. (v) The input and latent reactants are reacted together where (vi) each product is scored by a given oracle function. (vii) The highest-scoring product and latent reactant are stored. (viii) The scores for all products are returned to the BBO to score the latent points generated in step iii to update for the next iteration. (ix) The optimal reactant B_t and product A_{t+1} are returned after all BBO iterations are complete and A_{t+1} becomes the input for the next synthesis iteration.

4.1 Outer Synthesizability Optimization Loop

We define our synthesis framework, BBO-SYN, as follows. BBO-SYN breaks down the task of molecule generation into two distinct optimization steps: an outer iterative loop and an inner BBO loop for latent reactant selection. The outer loop is where synthesis trees are built one node at a time, as shown in Fig. 1, to enforce strict model-based synthesizability constraints. BBO-SYN begins by encoding an input molecule and passing it to the black-box optimizer where LaP³ is used in the inner loop to generate reactant options directly in the latent space (Fig. 2 i-iii). Independent of the underlying BBO method, BBO-SYN scores proposed latent reactants z for each input molecule A according to: $\text{oracle}(\text{react}(A, \text{decode}(z)))$ (Fig. 2 iv-vi). Every potential latent solution is decoded and reacted with our input molecule via template-free reaction prediction. After each sub-iteration of the black-box optimization, the product and accompanying reactant yielding the highest property score are saved, and then all scores are returned to the optimizer to update and continue the next sub-iteration (Fig. 2 vii-viii).²

Once the outer procedure is complete, the best product is used as the input reactant for the next BBO-SYN outer iteration, and the associated best reactant partner is returned (Fig. 2 ix). We show that we can greedily build synthesis trees by reusing the intermediate products as new inputs to

²We used the default number of LaP³ iterations according to <https://github.com/yangkevin2/neurips2021-lap3>.

generate a final optimal product for the desired chemical property. While greedy optimization might not necessarily lead to truly optimal synthesis trees, we found it worked well enough in practice.

4.2 Inner Black-Box Optimization Loop

The inner optimization loop uses BBO to select reactants from a continuous latent space for the building of optimal synthesis trees. Specifically, BBO-SYN uses LaP³, which iteratively samples latent points to learn a recursive space partition focusing on good regions while also still exploring bad regions using an upper confidence bound. While prior methods utilize MCTS to model entire synthesis trees (corresponding to our outer loop), LaP³ uses MCTS to learn a space partitioning function which is used to produce optimal latent reactants for individual chemical reaction steps.

As synthesizability-constrained generation is driven by the choice of available reactants, we opted to use a continuous molecular representation to efficiently handle a variable number of candidate reactants without major changes to underlying methodology [Jin et al., 2020b]. As it is challenging to generate a smooth molecular latent space [Gómez-Bombarelli et al., 2018, Zhang et al., 2022a], we use BBO to alleviate the structural and optimization difficulties of working in a non-smooth space [Yang et al., 2021, Wang et al., 2020a]. BBO thus provides us with a powerful solution to the reactant selection component of template-free synthesizability-constrained generation. Furthermore, we stress BBO-SYN’s modularity as it does not depend on any one choice in reaction predictor, molecular latent space, or optimization goal. Discrete methods such as DAGs must be retrained from scratch for each alteration. BBO-SYN, on the other hand, can seamlessly take advantage of future advances in adjacent areas of BBO, chemical reaction prediction, and latent molecular representation.

5 Synthesis Experiments

5.1 Baselines and Task Setup

We compare BBO-SYN to various DAGs’ fine-tuned DoG-Gen models [Bradshaw et al., 2020], one per tested property holding fixed the initial set of starting molecules and the chemical reaction predictor. DAGs was chosen as it is the most expressive template-free forward synthesis planner to date that has been successfully applied to property-guided molecule generation. For the initial set of starting molecules, we use a subset of the starting molecules from DAGs’ published validation set of crafted synthesis trees.³ Both methods use DAGs’ pre-trained Molecular Transformer model weights for template-free chemical reaction prediction to ensure a fair comparison. Both methods also only consider the top-1 Molecular Transformer generated products with no threshold for model confidence. BBO-SYN limits the depth of generated synthesis trees to 4, as each optimization step is computationally expensive.⁴

Given that BBO-SYN’s generated synthesis trees are conditioned on a specified starting point, we filter the final DoG-Gen products to keep only the highest-scoring synthesis tree for each of the shared starting points.⁵ This alignment step is necessary as DoG-Gen is fine-tuned by repeatedly re-training on its top-k seen trees and can only return a sorted list of every synthesis tree encountered during its iterative refinement. In this way, we can condition the outputs of both methods on the same discrete set of starting molecules.

Due to the different definitions and design choices of the components of the synthesis-aware generation task, a true head-to-head comparison is difficult to create. As a result, DoG-Gen is only used as an anchor point to understand BBO-SYN’s performance. As BBO-SYN explores a continuous reactant latent space, it is not confined to the same set of discrete reactants as DoG-Gen. Our benchmarks are designed to mitigate the differences of continuous vs. discrete spaces as much as possible and provide extensive ablations to understand how BBO presents a robust and scalable solution to the synthesis-aware generation problem. Since the BBO-SYN latent spaces are trained to approximate the distribution of the discrete DoG-Gen reactants, and all synthesis trees start with one of the discrete reactants, we opt to use the initial starting molecule as the main equalizing criteria in the later comparisons.

³We use SMILES with length in [5,25], totaling 2246 molecules.

⁴99% of DoG-Gen trees had depth ≤ 4 with max depth of 10. Best DoG-Gen JNK3 trees were depth 4.

⁵Synthesis trees are scored by the property score of the final product molecule.

5.2 Reactant Latent Space Set Up

We trained two distinct HierVAE [Jin et al., 2020b] models with a 32-dimensional latent space. The first was trained solely on the DAGs published building blocks (4,343 molecules). The second latent space was trained on all unique reactant and product molecules in the USPTO_MIT data set [Jin et al., 2017] that was used to train the aforementioned Molecular Transformer. The second latent space also included the above building blocks filtered for SMILES length [3,45], resulting in a total of 404,898 molecules or potential reactants for forward synthesis. These two latent spaces are henceforth referred to as the small and large latent spaces, respectively. We applied BBO-SYN to optimize for various chemical properties over both latent spaces to understand how our framework would operate in various environments.

5.3 Metrics

We focus on each method’s ability to generate diverse final product molecules with high property scores. As such, we report the property score distributions of the final product molecules of all generated synthesis trees. For all experiments, we used the TDC package [Huang et al., 2021] for property oracle functions (QED, DRD2, JNK3, GSK3 β). We also report the internal diversity (IntDiv_p), defined as $1 - \sqrt[p]{\frac{1}{|G|^2} \sum_{m_1, m_2 \in G} T(m_1, m_2)}$ for a set of molecules G and Tanimoto similarity T taken from MOSES [Polykovskiy et al., 2020]. A low percentage of diverse molecules illustrates a method’s collapse to a select few solutions, i.e., a lack of generative robustness.

5.4 Results

Here we present the respective optimized product distributions for both DoG-Gen and BBO-SYN for DRD2, GSK3 β , and JNK3.

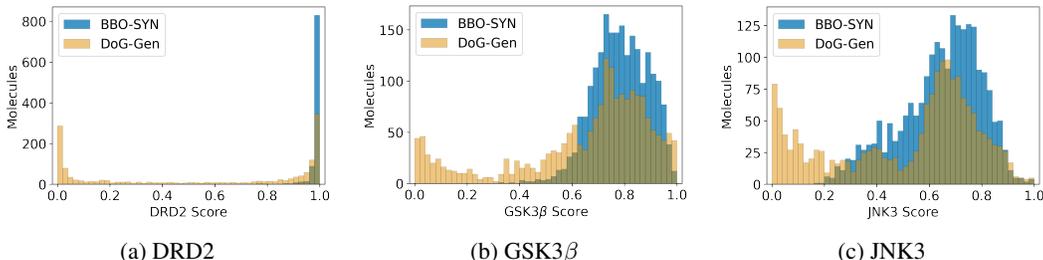


Figure 3: Property scores of final predicted molecules for BBO-SYN and DoG-Gen property-guided optimization. Both methods use the same initial building blocks and chemical reaction predictor.

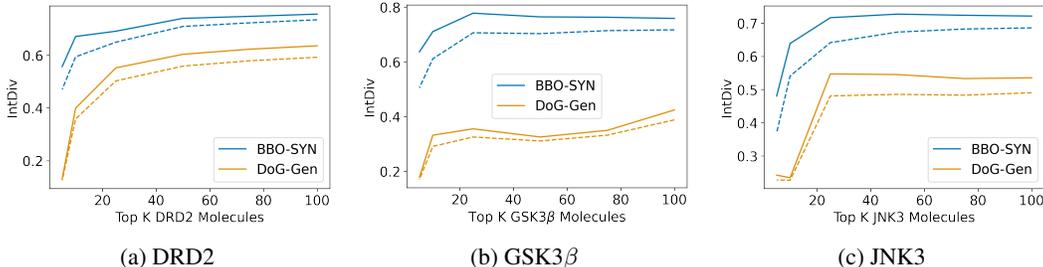


Figure 4: Internal diversity of top 5, 10, 25, 50, 75, 100 molecules. Solid and dashed lines correspond to IntDiv_1 and IntDiv_2 .

Fig. 3 shows that BBO-SYN significantly eliminates weaker products while maintaining top-end performance. We see in Fig. 4 BBO-SYN produces more unique high-scoring molecules for all properties. We hypothesize that the large increase in diversity (37% for top 100 JNK3 molecules) is due to BBO-SYN’s independent optimization for each input. Compared to DoG-Gen’s bulk

finetuning, BBO-SYN searches the reactant space in parallel to find the best reactants for each starting molecule.

BBO-SYN finds a unique optimal product for nearly every input which is desirable when developing novel molecules. By actively searching for promising reactants, BBO-SYN avoids converging to a small set of solutions, as seen in DoG-Gen. Although BBO-SYN has a higher computational cost due to the inner loop LaP³ optimization steps, we observed that giving DoG-Gen additional training iterations to equalize the property oracle budget of BBO-SYN resulted in no discernible difference in the resulting property distributions. We suspect that because DoG-Gen is repeatedly fine-tuned on its top encountered synthesis trees, increasing DoG-Gen’s oracle budget only further increases the apparent mode collapse. We also acknowledge the flaws of template-free reaction prediction as seen in the low model confidence reactions in Fig. 1. While it does not impact the comparison between BBO-SYN and DoG-Gen as they use the same reaction prediction, in the future, reaction confidence can be directly optimized by incorporating it into the BBO scoring function.

5.5 Analysis and Ablations

Below we analyze the effect of the choice of BBO-SYN’s BBO method as well as the latent space size on the property scores and diversity of the generated synthesis trees. Specifically, we compare two BBO methods, a simple CMA-ES, and LaP³, over both the small and large latent spaces for a series of property optimization tasks. We utilized a SMILES length penalty on all CMA-ES proposed reactants to prevent exploding sequence lengths due to sampling from non-smooth regions of the latent space.⁶ No length penalties were needed for LaP³.

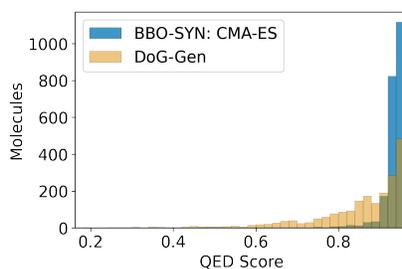


Figure 5: Unique QED Synthesis Product Distributions. Even when LaP³ is replaced with a weaker optimization method, CMA-ES, BBO-SYN outperforms DoG-Gen when using the same set of starting points and chemical reaction predictor.

Fig. 5 demonstrates that BBO-SYN needs only a simple CMA-ES to achieve strong performance for QED. Compared to DoG-Gen, BBO-SYN increases the diversity of the top 100 molecules by 5%. However, the story changes when we compare CMA-ES and LaP³ on the more challenging properties: DRD2, GSK3 β , and JNK3.

CMA-ES struggles to generate molecules with JNK3 greater than 0.6, whereas that is where most of the LaP³ optimized results are located (Fig. 6(a)). Similar behavior can be seen for GSK3 β and DRD2 in Appendix Fig. 12(a)- 13(a) as CMA-ES tends to generate more broad distribution whereas LaP³ is more concentrated around high scoring molecules. Fig. 6(b) shows the internal diversity for the analyzed generated products. We point to the significant gap in property optimization performance as the reason for CMA-ES achieving higher diversity, i.e., it is easy to generate diverse products when they are not strongly optimized for a specific property.

For all three tested properties, the property scores and diversity of generated molecules also depend on the number of reactants considered or, in the case of BBO-SYN, the size of latent space used. Fig. 6(a) illustrates the impact of the number of available reactants on generating high-scoring synthesis trees. LaP³ achieves significantly better property scores when given the large latent space that was trained with 100x the molecules as the small. Interestingly CMA-ES seems to prefer the small latent space for top-end performance. Fig. 6(b) shows how the internal diversity is correlated with the width of the property distribution, and as a result, CMA-ES and LaP³ on the large latent space result in the

⁶Property scores of product SMILES of length ≥ 70 and reactant SMILES of length ≥ 55 were reduced by a factor of 10.

best and worst, respectively. Similar results can be found in Appendix Fig. 12-13 for GSK3 β and DRD2.

We note that a significant advantage of BBO-SYN over DoG-Gen is its ability to consider 100x more reactants by using a different latent space, with negligible increase in computational cost. It is infeasible to run DAGs with over 400k potential reactants.

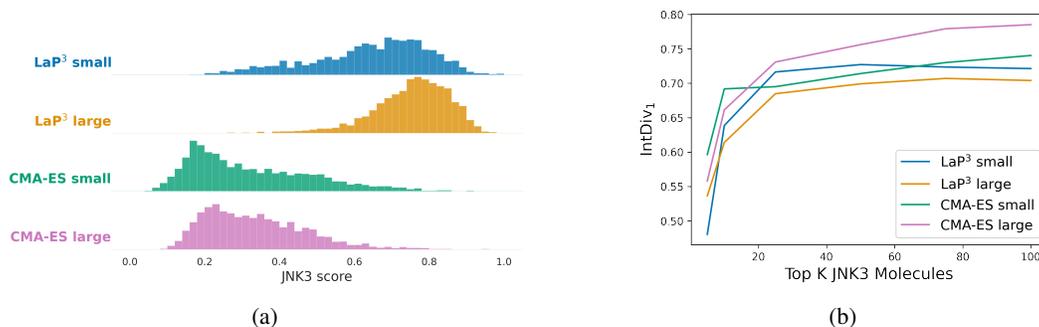


Figure 6: (a) JNK3 property score distributions of generated molecules for various latent space sizes and BBO methods. (b) Internal diversity (IntDiv_1) of top 5, 10, 25, 50, 75, 100 JNK3-optimized molecules.

However, LaP³'s improved performance on the large latent space may come at the cost of complete synthesizability. When using the large latent space, it is possible that chosen reactants may actually be intermediate products, resulting in *convergent synthesis*: molecule X can react with another intermediate Y, rather than requiring Y to be a starting material. Convergent synthesis poses new difficulties but may also increase the potential flexibility of the method. For example, by introducing simple RL as seen in DAGs and PGFS, one could at each step allow BBO-SYN to choose between adding to the tree directly (as we do currently) or picking two latent reactants to produce a convergent reactant [Gottipati et al., 2020]. While we do not explicitly attempt this, we expect that this procedure should be feasible because BBO-SYN's large latent space is trained on USPTO reactants and their single reaction step products.

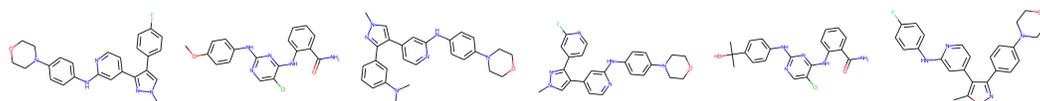


Figure 7: Top 6 BBO-SYN generated products optimized for JNK3.

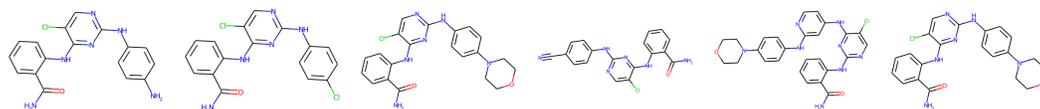


Figure 8: Top 6 DoG-Gen generated products optimized for JNK3.

Lastly, we provide a qualitative comparison of the diversity of the top 6 generated products for JNK3 in Fig. 7-8. Here it can be seen that DoG-Gen mostly makes minor updates to the end of the same underlying molecular scaffold, whereas BBO-SYN generates more geometrically different molecules. Similar results for GSK3 β and DRD2 can be found in Appendix Fig. 15 - 18.

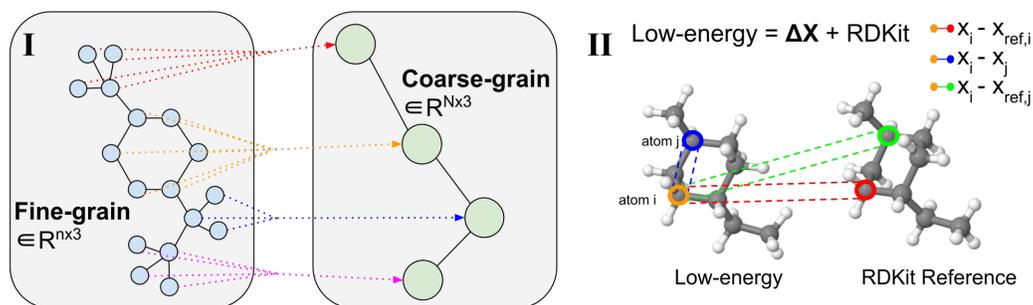


Figure 9: **Coarse-grained prediction task.** (I) Fine-grain molecules are split along rotatable bonds that define torsion angles. They are then coarse-grained to reduce the dimensionality and learn a subgraph-level latent distribution. (II) Illustration of the autoregressive decoder message passing parameterization and learning objective. We autoregressively learn the optimal distortion, *i.e.* distance from the reference molecule. Due to equivariance, the reference is not aligned during generation.

6 3D Constraints: Introduction

Molecular conformer generation (MCG) is a fundamental task in computational chemistry. The objective is to predict stable low-energy 3D molecular structures, known as conformers. Accurate molecular conformations are crucial for a wide range of applications that depend on precise spatial and geometric qualities, including drug discovery and protein docking. In traditional physics-based methods, there is a trade-off between speed and accuracy. Quantum mechanical methods, such as CREST [Pracht et al., 2020], are accurate but computationally slow, while stochastic cheminformatics-based methods like RDKit EKT DG Riniker and Landrum [2015] provide more efficient but less accurate results. As the difficulty of computing low-energy structures increases with the number of atoms and rotatable bonds in a molecule, there has been interest in developing machine learning (ML) methods to generate efficient and accurate conformer predictions.

Prior generative models have shown success in the current benchmarks used to evaluate the accuracy of conformer predictions. However, current benchmarks [Mansimov et al., 2019] are problematic as for a molecule with L known conformers, they measure the minimum error of the best conformer out of a model-generated set of size $2L$. We contend that this is an inherently flawed metric, as by failing to penalize a success rate or validity of $\frac{1}{2L}$, prior benchmarks do not accomplish their goal of assessing a model’s accuracy and ability to generalize. Poor model performance is essentially masked out as long as one good conformer exists. Furthermore, molecules could have *many* conformers. Only measuring the best error to a single conformer is not comprehensive, as not all conformers are likely to appear in the real world. Additionally, despite knowing the error associated with the optimal conformer, even determining the optimal conformer among the available $2L$ options without prior knowledge of the ideal structure remains a considerable challenge. Thus, these metrics are not fully informative as to whether current models will be useful for downstream tasks such as protein docking, which depends on highly accurate 3D structures. In this work, we further discuss the implications of these flaws and provide new comprehensive metrics as a step towards better-measuring model performance. We also show that by applying a built-in relatively inexpensive force field optimization, RDKit, an open-source cheminformatics software, predominantly outperforms prior state-of-the-art methods, especially on our new metrics.

Our improved metrics also serve as motivation for improving conformer generative modeling at an architectural level. To this end, we introduce CoarsenConf, an SE(3)-equivariant hierarchical VAE that pools information from fine-grain atomic coordinates to a coarse-grain subgraph level representation for efficient autoregressive conformer generation. Coarse-graining reduces the dimensionality of the problem allowing conditional autoregressive generation rather than generating all coordinates independently, as done in prior work. By directly conditioning on the 3D coordinates of prior generated subgraphs, our model better generalizes across chemically and spatially similar subgraphs. This mimics the underlying molecular synthesis process, in which small functional units bond together to form large drug-like molecules. Unlike prior methods, CoarsenConf generates low-energy conformers with the ability to model atomic coordinates, distances, and torsion angles directly.

Our main contributions are as follows:

- We present CoarsenConf, a novel conditional hierarchical VAE. CoarsenConf learns a coarse-grained subgraph-level latent distribution for efficient SE(3)-equivariant autoregressive conformer generation. Our coarse-graining strategy allows CoarsenConf to learn optimal torsional angles implicitly. To our knowledge, this is the first method to use coarse-graining in the context of MCG and the first to explore 3D autoregressive techniques for MCG.
- CoarsenConf is the first model capable of handling variable-length coarse-to-fine generation using an aggregated attention strategy. Prior coarse-to-fine methods [Wang et al., 2022] require the number of coarse-grain beads to be fixed for all input molecules, and focused on the task of coarse-to-fine backmapping.
- We predominantly outperform prior methods on GEOM-QM9 and GEOM-DRUGS evaluation benchmarks [Axelrod and Gómez-Bombarelli, 2022]. We do so while requiring less training time, 2% the total training iterations, and 59% and 92.7% less data compared to prior state-of-the-art for QM9 and DRUGS, respectively.
- We enhance the existing benchmarks for conformer generation by incorporating direct measurements of average and maximum root mean square deviation (RMSD) across the set of generated conformers for each molecule. These new metrics provide a more comprehensive assessment of a model’s ability to generate robust conformations, as they evaluate the quality of all generated samples, not just the best ones. Here, robust refers to the consistent generation of high-quality conformers.

7 Background

Notations. We represent each molecule as a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where \mathcal{V} is the set of vertices representing atoms and \mathcal{E} is the set of edges representing inter-atomic bonds. Each node v in \mathcal{V} describes the chosen atomic features such as element type, atomic charge, and hybridization state. Each edge e_{uv} in \mathcal{E} describes the corresponding chemical bond connecting u and v , and is labeled with its bond type. Following Simm and Hernandez-Lobato [2020], each molecular graph is expanded to incorporate auxiliary edges connecting all atoms within a 4Å radius to enhance long-range interactions in message passing. The spatial position of each atom in \mathcal{V} is represented by a 3D coordinate vector $\mathbf{r} \in \mathbb{R}^3$, such that the full molecule conformation is represented by the matrix $\mathbf{X} \in \mathbb{R}^{|\mathcal{V}| \times 3}$.

Problem Definition. *Molecular conformation generation* (MCG) is a conditional generative process that aims to model the conditional distribution of 3D molecular conformations \mathbf{X} , given the 2D molecule graph \mathcal{G} , i.e., $p(\mathbf{X}|\mathcal{G})$. While prior works have shown some success with learning 3D conformations starting with only the 2D information, many require complex and compute-intensive architectures [Zhu et al., 2022, Zhou et al., 2023a]. Recently, Jing et al. [2022] demonstrated good performance on the GEOM-DRUGS dataset by priming the method with easy-to-obtain 3D approximations via RDKit EKTDG [Riniker and Landrum, 2015]. Jing et al. [2022] showed that RDKit is highly effective at generating conformations with correct inter-atomic distances and, as a result, can constrain the problem to a diffusion process only over torsion angles.

We thus formalize MCG as modeling the conditional distribution $p(\mathbf{X}|\mathcal{R})$, where $\mathcal{R} = (\mathcal{V}, \mathcal{E}, \mathcal{X})$ and \mathcal{X} is the RDKit generated atomic coordinates. This is functionally the same underlying distribution as $p(\mathbf{X}|\mathcal{G})$, as we use RDKit as a building block to provide an approximation starting from only 2D information. We will show that more robust conformers are generated by conditioning on approximations without imposing explicit angular and distance constraints.

Classical Methods for Conformer Generation. A molecular conformer refers to the collection of 3D structures that are energetically favorable and correspond to local minima of the potential energy surface. CREST [Pracht et al., 2020] uses semi-empirical tight-binding density functional theory for energy calculations, which, while computationally less expensive than ab-initio quantum mechanical (QM) methods, still requires approximately 90 core hours per drug-like molecule [Axelrod and Gómez-Bombarelli, 2022]. Though CREST was used to generate the ground truth GEOM dataset, it is too slow for downstream applications such as high-throughput virtual screening.

Cheminformatics methods, such as RDKit EKTDG, are commonly used to quickly generate approximate low-energy conformations of molecules. These methods are less accurate than QM methods due to the sparse coverage of the conformational space resulting from stochastic sampling. Additionally, force field optimizations are inherently less accurate than the above QM methods. RDKit

EKTDG employs a genetic algorithm for Distance Geometry optimization that can be enhanced with a molecular mechanics force field optimization (MMFF). Zhou et al. [2023b] recently highlighted potential flaws in the current MCG benchmarks and demonstrated how EKTDG, in conjunction with a traditional sampling and clustering procedure, can outperform many recent deep learning methods. To overcome the accuracy-efficiency trade-off in classical methods, we combine easy-to-obtain classical approximations with efficient data-driven strategies to improve the generalization of robust MCG.

Deep Learning Methods for Conformer Generation. Several probabilistic deep learning methods for MCG have been developed, such as variational autoencoders in CVGAE [Mansimov et al., 2019] and ConfVAE [Xu et al., 2021b], normalizing flows in CGCF [Xu et al., 2021a], score-based generative models in ConfGF [Shi et al., 2021] and DGSM [Luo et al., 2021], and diffusion models in GeoDiff [Xu et al., 2022] and Torsional Diffusion [Jing et al., 2022]. GraphDG [Simm and Hernandez-Lobato, 2020] forgoes modeling coordinates and angles, relying solely on distance geometry. DMCG [Zhu et al., 2022] and Uni-Mol [Zhou et al., 2023a] present examples of effective large models, the first mimicking the architecture of AlphaFold [Jumper et al., 2021] and the second using large-scale SE(3)-equivariant transformer pretraining.

Molecular Coarse-graining. Molecular coarse-graining refers to the simplification of a molecule representation by grouping the fine-grained (FG) atoms in the original structure into individual coarse-grained (CG) beads \mathcal{B} with a rule-based mapping. Coarse-graining has been widely utilized in protein design [Kmieciak et al., 2016, Yang and Gomez-Bombarelli, 2023], and analogously fragment-level or subgraph-level generation has proven to be highly valuable in diverse 2D molecule design tasks [Chen et al., 2021]. Breaking down generative problems into smaller pieces is an approach that can be applied to several 3D molecule tasks. For instance, CGVAE [Wang et al., 2022] learns a fixed-sized latent distribution to back map or restore FG coordinates from CG coordinates effectively.

Autoregressive Molecule Generation. Autoregressive models provide control over the generative process by enabling direct conditioning on prior information, allowing for a more precise and targeted generation of output. Autoregressive generation has shown success in 2D molecule tasks in both atom-wise and subgraph-level techniques, as shown in GraphAF [Shi et al., 2020] and HierVAE [Jin et al., 2020a]. Similarly, 3DLinker [Huang et al., 2022] and SQUID [Adams and Coley, 2022] showcase the usefulness of 3D autoregressive molecule generation and their ability to leverage conditional information in both atom-wise and subgraph-level settings for 3D linkage and shape-conditioned generative tasks respectively. HERN [Jin et al., 2022] further demonstrates the power of hierarchical equivariant autoregressive methods in the task of computational 3D antibody design.

SE(3) Equivariance. Let \mathcal{X} and \mathcal{Y} be the input and output vector spaces, respectively, which possess a set of transformations $G: G \times \mathcal{X} \rightarrow \mathcal{X}$ and $G \times \mathcal{Y} \rightarrow \mathcal{Y}$. The function $\phi: \mathcal{X} \rightarrow \mathcal{Y}$ is called equivariant with respect to G if when we apply any transformation to the input, the output also changes via the same transformation or under a certain predictable behavior Han et al. [2022], *i.e.*,

Definition 1 *The function $\phi: \mathcal{X} \rightarrow \mathcal{Y}$ is G -equivariant if it commutes with any transformation in G ,*

$$\phi(\rho_{\mathcal{X}}(g)x) = \rho_{\mathcal{Y}}(g)\phi(x), \forall g \in G, \quad (1)$$

where $\rho_{\mathcal{X}}$ and $\rho_{\mathcal{Y}}$ are the group representations in the input and output space, respectively. Specifically, ϕ is called invariant if $\rho_{\mathcal{Y}}$ is the identity.

By enforcing SE(3)-equivariance in our probabilistic model, $p(\mathbf{X}|\mathcal{R})$ remains unchanged for any rototranslation of the approximate conformer \mathcal{R} . CoarsenConf’s architecture is inspired by recent equivariant graph neural network architectures, such as EGNN [Satorras et al., 2021] and PaiNN [Schütt et al., 2021], as well as Vector Neuron multi-layer perceptron (VN-MLP) [Deng et al., 2021].

8 Methods

8.1 Algorithmic Details

Coarse-graining Procedure. We first define a rotatable bond as any single bond between two non-terminal atoms, excluding amides and conjugated double bonds, where the torsion angle is the angle of rotation around the central bond. Formally, the torsion angle τ_{abcd} is defined about bond $(b, c) \in \mathcal{E}$ where (a, b) are a choice of reference neighbors s.t $a \in \mathcal{N}(b) \setminus c$ and $d \in \mathcal{N}(c) \setminus b$.

We coarsen molecules into a single bead for each connected component resulting from severing all rotatable bonds. This choice in CG procedure implicitly forces the model to learn over torsion angles

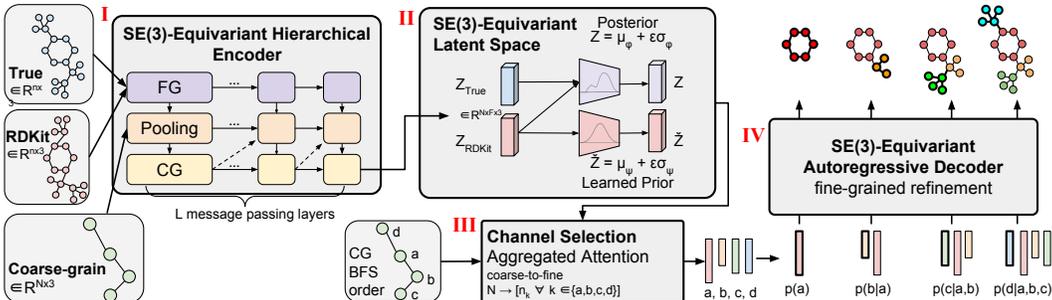


Figure 10: **CoarsenConf architecture:**(I) The encoder $q_\phi(z|X, \mathcal{R})$ takes ground truth conformer X , RDKit approximate conformer \mathcal{R} , and coarse-grained (CG) conformer \mathcal{C} as inputs (derived from X and predefined CG strategy), and outputs a variable-length equivariant CG representation via equivariant message passing and point convolutions. (II) Equivariant MLPs Deng et al. [2021] are applied to learn the mean and log variance of both the posterior and prior distributions. (III) The posterior (training) or prior (inference) is sampled and fed into the Channel Selection module, where an attention layer is used to learn the optimal pathway from CG to FG structure. (IV) Given the FG latent vector and the RDKit approximation, the decoder $p_\theta(X|\mathcal{R}, z)$ learns to recover the low-energy FG structure through autoregressive equivariant message passing. The entire model can be trained end-to-end by optimizing the KL divergence of latent distributions and reconstruction error of generated conformers.

as well as atomic coordinates and inter-atomic distances. We found that using a more physically constrained definition of torsional angles, as defined by Ganea et al. [2021], in the CG procedure led to a significant increase in performance compared to that used in Jing et al. [2022]. This is because the latter allows rotations around double and triple bonds, while the former does not. An example of the coarse-graining procedure is in Fig. 9(I).

Learning Framework. CoarsenConf is a conditional generative model that learns $p(X|\mathcal{R})$ where X is the low-energy 3D conformation, and \mathcal{R} is the RDKit best guess conformation. Specifically, we optimize $p(X|\mathcal{R})$ by maximizing its variational lower bound with an approximate posterior distribution $q_\phi(z|X, \mathcal{R})$ and learned prior $p_\psi(z|\mathcal{R})$:

$$\log p(X|\mathcal{R}) \geq \underbrace{\mathbb{E}_{q_\phi(z|X, \mathcal{R})} \log p_\theta(X|\mathcal{R}, z)}_{\mathcal{L}_{\text{reconstruction}}} + \underbrace{\mathbb{E}_{q_\phi(z|X, \mathcal{R})} \log \frac{p_\psi(z|\mathcal{R})}{q_\phi(z|X, \mathcal{R})}}_{\mathcal{L}_{\text{latent regularization}}}, \quad (2)$$

where $q_\phi(z|X, \mathcal{R})$ is the hierarchical equivariant encoder model, $p_\theta(X|\mathcal{R}, z)$ is the equivariant decoder model to recover X from \mathcal{R} and z , and $p_\psi(z|\mathcal{R})$ is the learned prior distribution. The reconstruction loss, $\mathcal{L}_{\text{recon.}}$, is implemented as $\text{MSE}(\mathcal{A}(X_{\text{true}}, X_{\text{model}}), X_{\text{model}})$, where \mathcal{A} is the Kabsch alignment function that provides an optimal rotation matrix and translation vector to minimize the mean squared error (MSE) [Kabsch, 1993]. The second term, $\mathcal{L}_{\text{reg.}}$, can be viewed as a regularization over the latent space and is implemented as $\beta D_{KL}(q_\phi(z|X, \mathcal{R}) \| p_\psi(z|\mathcal{R}))$ [Higgins et al., 2017]. In practice, we add an auxiliary loss function that measures the distance error between the coordinates of all 2-hop neighbors in the molecular graph.

Learning Optimal Distortions. As the goal of MCG is to model the conditional distribution $p(X|\mathcal{R})$, we can simplify the learning objective by setting $X = \mathcal{R} + \Delta X$ and learn the optimal distortion ΔX from the RDKit approximation. This greatly simplifies the learning objective, as we have ensured $\mathcal{L}_{\text{recon.}}$ is no worse than that of the RDKit approximation. This procedure is analogous to using multiple sequence alignments in protein structure prediction tasks to initialize the model in an approximate low-energy state [Jumper et al., 2021]. We stress that the RDKit approximate conformers are trivial to obtain compared to the cost of generative model training and inference.

8.2 Encoder Architecture

Hierarchical Modules. We describe the encoder, shown in Fig. 10(I). The model operates over SE(3)-invariant atom features $h \in R^{n \times D}$, and SE(3)-equivariant atomistic coordinates $x \in R^{n \times 3}$. A single encoder layer is composed of three modules: fine-grained, pooling, and coarse-grained.

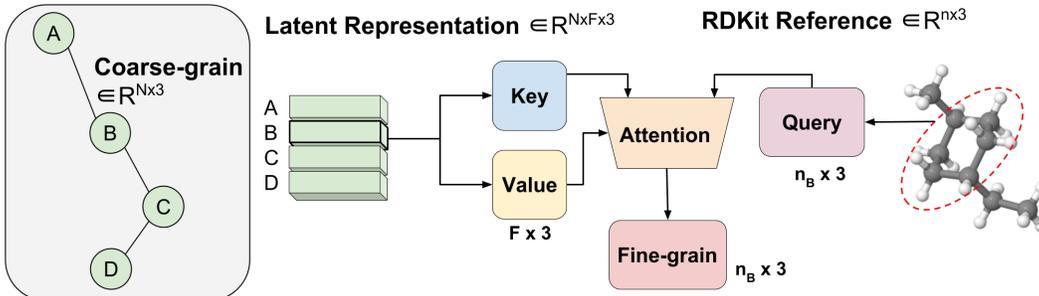


Figure 11: **Variable length coarse-to-fine backmapping via Aggregated Attention.** The latent vector of a single CG bead $Z_B \in R^{F \times 3}$ is used as the key and value of a single head attention operation with an embedding dimension of three to match the x,y,z coordinates. The query vector is the subset of the RDKit conformer corresponding to bead $B \in R^{n_B \times 3}$, where n_B is variable-length as we know a priori how many FG atoms correspond to a certain CG bead. Leveraging attention, we efficiently learn the optimal blending of latent features for FG reconstruction. We call this approach Aggregated Attention, as it aggregated 3D segments of FG information to form our latent query.

The fine-grained module is a graph-matching message-passing architecture. It differs from Stärk et al. [2022] by not having internal closed-form distance regularization and exclusively using unidirectional attention. Its purpose is to effectively match the approximate conformer and ground truth by updating attention from the former to the latter.

The pooling module takes in the updated representations (h and x) of both the ground truth molecule and the RDKit reference from the FG module. The pooling module is similar to the FG module, except it no longer uses attention and operates over a pooling graph. Given a molecule with n atoms and N CG beads, the pooling graph consists of $n + N$ nodes. There is a single directional edge from all atoms to their respective beads (see Fig. 9(I)). This allows message passing to propagate information through the predefined coarsening strategy.

The coarse-grained module uses the updated CG representations ($H \in R^{N \times D}$ and $X \in R^{N \times 3}$) from the pooling module to learn equivariant CG features (Z and $\tilde{Z} \in R^{N \times F \times 3}$) for the ground truth molecule and the RDKit reference. F is fixed as a hyperparameter for latent space size. N is allowed to be variable-length to handle molecules resulting from any coarsening procedure. The CG features are learned using a graph-matching point convolution [Thomas et al., 2018] with similar unidirectional attention as the FG module. Prior to the main message-passing operations, the input features undergo equivariant mixing [Huang et al., 2022] to further distill geometric information into the learned CG representation.

The modules in each encoder layer communicate with the respective module of the previous layer. This hierarchical message-passing scheme results in an informative and geometrically grounded final CG latent representation. We note that the pooling module of layer ℓ uses the updated invariant features H from the CG module of layer $\ell - 1$, as shown by the dashed lines in Fig. 10(I).

Equivariant Latent Space. As Z holds a mixture of equivariant spatial information, we maintain equivariance through the reparametrization trick of the VAE (Fig. 10(II)). Specifically, we define the posterior and prior means (μ_ϕ, μ_ψ) and standard deviations (σ_ϕ, σ_ψ), as follows:

$$\begin{aligned} \mu_\phi &= \text{VN-MLP}(Z, \tilde{Z}), & \log(\sigma_\phi^2) &= \text{MLP}(Z, \tilde{Z}), \\ \mu_\psi &= \text{VN-MLP}(\tilde{Z}), & \log(\sigma_\psi^2) &= \text{MLP}(\tilde{Z}). \end{aligned} \quad (3)$$

We use an invariant MLP to learn the variance and apply it to the x, y, and z directions to enforce equivariance. We found that setting $\sigma_\psi = 1$ results in smoother optimization rather than requiring an additional latent regularization to the standard normal as seen in Mansimov et al. [2019]. We note the conditional posterior is parameterized with both the ground truth and RDKit approximation whereas the learned conditional prior only uses the approximation.

8.3 Decoder Architecture

We sample from the learned posterior (training) and learned prior (inference) to get $Z = \mu + \epsilon\sigma$, where ϵ is noise sampled from a standard Gaussian distribution as the input to the decoder. We note

the role of the decoder is two-fold. The first is to convert the latent coarsened representation back into FG space through a process we call channel selection. The second is to refine the fine-grain representation autoregressively to generate the final low-energy coordinates.

Channel Selection. To improve conformer generation using coarse-graining, we need to ensure that our model can handle variable-length backmapping. This aspect is crucial because every molecule can be coarsened into a different number of beads, and there is no explicit limit to the number of atoms a single bead can represent. Unlike CGVAE [Wang et al., 2022], which requires training a separate model for each choice in granularity N , CoarsenConf is capable of reconstructing FG coordinates from any N (illustrated in Fig. 10(III)).

CGVAE defines the process of channel selection as selecting the top k latent channels, where k is the number of atoms in a CG bead of interest. Instead of discarding all learned information in the remaining $F - k$ channels in the latent representation, we use a novel aggregated attention mechanism. This mechanism learns the optimal mixing of channels to reconstruct the FG coordinates and is illustrated in Fig. 11. The attention operation allows us to actively query our latent representation for the number of atoms we need, and draw upon similarities to the learned RDKit approximation that has been distilled into the latent space through the encoding process. Channel selection translates the CG latent tensor $Z \in R^{N \times F \times 3}$ into FG coordinates $X_{CC} \in R^{n \times 3}$.

Autoregressive Refinement. Once channel selection is complete, we have effectively translated the variable-length CG representation back into the desired FG form. From here, X_{CC} is grouped into its corresponding CG beads, but left in FG coordinates to do a bead-wise autoregressive generation of final low-energy coordinates (Fig. 10(IV)). As there is no intrinsic ordering of subgraphs, we use a breadth-first search that prioritizes larger subgraphs with large out-degrees. In other words, we generate a linear order that focuses on the largest, most connected subgraphs and works outward. We believe that by focusing on the most central component first, which occupies the most 3D volume, we can reduce the propagation of error that is typically observed in autoregressive approaches. We stress that by coarse-graining by torsion angle connectivity, our model learns the optimal torsion angles in an unsupervised manner as the conditional input to the decoder is not aligned. CoarsenConf ensures each next generated subgraph is rotated properly to achieve a low coordinate and distance error.

Learning the Optimal Distortion. The decoder architecture is similar to the EGNN-based FG layer in the encoder. However, it differs in two important ways. First, we mix the conditional coordinates with the invariant atom features using a similar procedure as in the CG layer instead of typical graph matching. Second, we learn to predict the difference between the RDKit reference and ground truth conformations. This provides an upper error bound and also enables us to leverage easy-to-obtain approximations more effectively. More formally, a single decoder layer is defined as follows:

$$\boldsymbol{\mu}^{(l)} = \frac{1}{|\mathcal{V}_{prev}|} \sum_{k \in \mathcal{V}_{prev}} x_k, \quad (4a)$$

$$\tilde{\mathbf{h}}_i = \phi^m(\mathbf{h}_i^{(l)}, \mathbf{x}_i^{(l)}, \boldsymbol{\mu}^{(l)}, \|\mathbf{x}_i^{(l)} - \boldsymbol{\mu}^{(l)}\|^2), \forall i \in \mathcal{V}_{cur}, \quad (4b)$$

$$\mathbf{m}_{j \rightarrow i} = \phi^e(\tilde{\mathbf{h}}_i^{(l)}, \tilde{\mathbf{h}}_j^{(l)}, \|\mathbf{x}_i^{(l)} - \mathbf{x}_j^{(l)}\|^2, \|\mathbf{x}_i^{(l)} - \mathbf{x}_{ref,j}^{(l)}\|^2, \|\mathbf{x}_i^{(l)} - \mathbf{x}_{ref,i}^{(l)}\|^2), \forall (i, j) \in \mathcal{E}_{cur}, \quad (4c)$$

$$\mathbf{m}_i = \frac{1}{|\mathcal{N}(i)|} \sum_{j \in \mathcal{N}(i)} \mathbf{m}_{j \rightarrow i}, \forall i \in \mathcal{V}_{cur}, \quad (4d)$$

$$\mathbf{u}_{j' \rightarrow i} = a_{j' \rightarrow i} \mathbf{W} \mathbf{h}_{j'}^{(l)}, \forall i \in \mathcal{V}_{cur}, j' \in \mathcal{V}_{prev}, \quad (4e)$$

$$\mathbf{u}_i = \sum_{j' \in \mathcal{V}_{prev}} \mathbf{u}_{j' \rightarrow i}, \forall i \in \mathcal{V}_{cur}, \quad (4f)$$

$$\mathbf{x}_i^{(l+1)} = \mathbf{x}_{ref,i}^{(l)} + \sum_{j \in \mathcal{N}(i)} (\mathbf{x}_i^{(l)} - \mathbf{x}_j^{(l)}) \phi^x(\mathbf{m}_{j \rightarrow i}), \quad (4g)$$

$$\mathbf{h}_i^{(l+1)} = (1 - \beta) \cdot \mathbf{h}_i^{(l)} + \beta \cdot \phi^h(\tilde{\mathbf{h}}_i^{(l)}, \mathbf{m}_i, \mathbf{u}_i, \mathbf{f}_i), \forall i \in \mathcal{V}_{cur}, \quad (4h)$$

where $(\mathcal{V}_{cur}, \mathcal{E}_{cur})$ and $(\mathcal{V}_{prev}, \mathcal{E}_{prev})$ refer to the subgraph currently being generated and the set of all previously generated subgraphs, *i.e.*, the current state of the molecule. ϕ^m , ϕ^e , ϕ^x , and ϕ^h refer to separate shallow MLPs for the feature mixing, edge message calculation, coordinate update, and invariant feature update, respectively. Eq. 4(a-b) creates a mixed feature for each atom comprised

Table 1: Quality of generated conformer ensembles for the GEOM-QM9 test set ($\delta = 0.5\text{\AA}$) in terms of Coverage (%) and Average RMSD (\AA). Torsional Diffusion (TD) was benchmarked using its evaluation code and available generated molecules, per their public instructions. Note that CoarsenConf (5 epochs) was restricted to using 41% of the data used by TD (250 epochs) to exemplify a low-compute and data-constrained setting.

Method	Metric	Recall				Precision			
		Coverage \uparrow		AR \downarrow		Coverage \uparrow		AR \downarrow	
		Mean	Med	Mean	Med	Mean	Med	Mean	Med
RDKit + MMFF	min	75.2	100.0	0.219	0.173	82.1	100.0	0.157	0.119
Torsional Diffusion	min	82.2	100.0	0.179	0.148	78.4	100.0	0.222	0.197
CoarsenConf	min	76.9	100.0	0.246	0.211	80.2	100.0	0.227	0.186
RDKit + MMFF	mean	47.6	50.0	0.476	0.441	48.3	40.0	0.476	0.441
Torsional Diffusion	mean	44.6	25.0	0.516	0.482	43.9	20.5	0.520	0.482
CoarsenConf	mean	49.1	50.0	0.464	0.431	50.1	50.0	0.464	0.431
RDKit + MMFF	max	33.4	0.00	0.665	0.634	30.9	0.00	0.698	0.662
Torsional Diffusion	max	25.3	0.00	0.786	0.729	28.1	0.00	0.736	0.683
CoarsenConf	max	35.8	0.00	0.639	0.606	34.0	0.00	0.647	0.624

of the current FG invariant feature and 3D position vectors (h and x), and the previous centroid μ and respective centroid distances. Eq. 4(c-d) defines the message passing operation that uses the aforementioned mixed features \tilde{h} and a series of important distances between the model-based conformer and RDKit reference (see Fig. 9(II) for a helpful visualization). Eq. 4(e-f) apply the same unidirectional attention updates seen in the encoder architecture. Eq. 4(g-h) update the position and feature vector for each atom using the above messages and attention coefficients, with f representing the original invariant node features $h^{\ell=0}$. We emphasize that Eq. 4(g) formulates the overall objective as learning the optimal distortion of the RDKit reference to achieve the low-energy position *i.e.* $x^* = x_{ref} + \Delta x$. The CG autoregressive strategy allows CoarsenConf to handle extremely large molecules efficiently, as the max number of time steps is equal to the max number of CG beads. CoarsenConf is trained using teacher forcing [Williams and Zipser, 1989], which enables an explicit mixing of low-energy coordinates with the current FG positions from channel selection Eq. 4(a-b).

9 Experiments

Data. We use the GEOM dataset [Axelrod and Gómez-Bombarelli, 2022], consisting of QM9 (average 11 atoms) and DRUGS (average 44 atoms), to train and evaluate our model. We use the same train/val/test molecule splits, 106586/13323/1000 for QM9 and 243473/30433/1000 for DRUGS, from Ganea et al. [2021]. Jing et al. [2022] demonstrated that these splits are significantly more challenging than those used in Xu et al. [2022], more accurately representing the challenges of MCG.

Baselines. We compare with Torsional Diffusion [Jing et al., 2022], which has outperformed all previous models on the previous evaluation metrics, and RDKit EKTDG + MMFF. We focus on these two methods as they depend on RDKit-generated structures directly or as input. For reference, Torsional Diffusion (TD) takes in the RDKit approximation and is trained to generate optimal torsion angle updates instead of explicit coordinates. We evaluate all methods over an expanded set of evaluation metrics to test the robustness of the generated conformations.

Data-constrained setting. In real-world applications like polymer design, the availability of data is frequently limited, and accompanied by a scarcity of conformers for each molecule. The current datasets, QM9 and DRUGS, do not mimic this setting very well. For example, on average QM9 has 15 conformers per molecule and DRUGS has 104 per molecule—both datasets have significantly more conformers than in an experimental drug design setting. Given this, rather than training on the first 30 conformers as done in TD, we train on the first five and two conformers (typically those with the largest Boltzmann weight) for QM9 and DRUGS, respectively. This corresponds to 41% of TD’s training set or 23% of the overall dataset for QM9, and 7.3% of TD’s training set or 1.7% of DRUGS overall. Even with less data, we outperform models trained on significantly larger training sets.

Table 2: Quality of generated conformer ensembles for the GEOM-DRUGS test set ($\delta = 0.75\text{\AA}$) in terms of Coverage (%) and Average RMSD (\AA). CoarsenConf (5 epochs) was restricted to using 7.3% of the data used by Torsional Diffusion (250 epochs) to exemplify a low-compute and data-constrained regime.

Method	Metric	Recall				Precision			
		Coverage \uparrow		AR \downarrow		Coverage \uparrow		AR \downarrow	
		Mean	Med	Mean	Med	Mean	Med	Mean	Med
RDKit + MMFF	min	49.0	46.2	0.896	0.825	55.3	55.1	0.840	0.715
Torsional Diffusion	min	73.5	80.6	0.574	0.552	55.9	57.6	0.770	0.720
CoarsenConf	min	45.7	42.4	0.888	0.848	47.0	41.8	0.915	0.823
RDKit + MMFF	mean	8.87	0.00	1.903	1.887	8.34	0.00	1.903	1.887
Torsional Diffusion	mean	7.19	0.00	1.793	1.864	7.90	0.00	1.793	1.864
CoarsenConf	mean	8.51	0.00	1.791	1.825	7.83	0.00	1.791	1.825
RDKit + MMFF	max	4.16	0.00	2.589	2.551	3.95	0.00	2.734	2.774
Torsional Diffusion	max	3.44	0.00	2.813	2.930	3.96	0.00	2.615	2.726
CoarsenConf	max	3.97	0.00	2.481	2.505	3.85	0.00	2.521	2.602

Metrics. We report both RMSD (AR) and Coverage for Recall (R) and Precision (P). Recall measures the number of ground truth conformers that are correctly predicted, and Precision measures the accuracy of the generated conformers. Following Jing et al. [2022], we generate two times the number of ground truth conformers for each molecule. More formally, for $K = 2L$ let $\{C_l^*\}_{l \in [1, L]}$ and $\{C_k\}_{k \in [1, K]}$ be respectively the sets of ground truth and generated conformers:

$$\begin{aligned} \text{COV}(f)\text{-R} &:= \frac{1}{L} \left| \{l \in [1..L] : f_{k \in [1..K]} \text{RMSD}(C_k, C_l^*) < \delta \} \right| \\ \text{AR}(f)\text{-R} &:= \frac{1}{L} \sum_{l \in [1..L]} f_{k \in [1..K]} \text{RMSD}(C_k, C_l^*), \\ f() &:= \{\text{min}, \text{mean}, \text{max}\}, \end{aligned} \quad (5)$$

where δ is the coverage threshold. The precision metrics are obtained by swapping ground truth and generated conformers. During a true test-time evaluation, the ground truth conformation is unknown. Therefore, we cannot compare it to all $2L$ generated structures to determine the most accurate one for the downstream application. Thus, the additions of average (mean) and worst-case (max) evaluation allow us to measure the robustness of the generative model by providing an error bound on each sample. As most downstream tasks that use molecular conformers possess expensive oracle functions, *i.e.* protein docking, it is important to be able to generate optimal conformers, as testing all $2L$ samples for each molecule may be infeasible. We note that AR is the central task-agnostic metric. Coverage is derived from AR and its threshold can be chosen arbitrarily, with its value being task-dependent.

9.1 Results

QM9. Our results, shown in Tab. 1, demonstrate a significant improvement over both RDKit+MMFF and Torsional Diffusion (TD) on the mean and max metrics. We have effectively shown that our model can improve upon RDKit initializations, resulting in state-of-the-art generalization. CoarsenConf does so using 59% of the TD’s data and for only 5 epochs compared to 250.

DRUGS. Tab. 2 demonstrates CoarsenConf’s ability to best generate robust low-error conformations for large drug-like molecules. With even further constraints on the data (only two conformers per molecule) and total iterations (5 epochs), CoarsenConf achieves superior AR results for the more comprehensive mean and max metrics. For comparison, TD uses 30 conformers for 250 epochs.

Takeaways. Both CoarsenConf and TD optimize $p(X|\mathcal{R})$ but utilize the RDKit approximations \mathcal{R} in different ways. TD learns to update the torsion angles of \mathcal{R} while CoarsenConf learns which geometric updates (coordinates, distances, and torsion angles) are optimal to translate \mathcal{R} to X . Unlike TD, which uses a preprocessing optimization procedure to generate substitute ground truth conformers that mimic $p(\mathcal{R})$, CoarsenConf directly learns from both X and \mathcal{R} through its hierarchical graph matching procedure. We thus directly address the distributional shift problem via our learning

framework. We hypothesize that this, along with our torsion angle-based CG strategy, leads to our observed improvements while only using 2% of the training iterations and 92.7% less data compared to TD in the case of GEOM-DRUGS. Although initialized with RDKit, CoarsenConf and TD are slightly worse on some of the min metrics. We hypothesize that by querying the force field optimization $2L$ times, eventually, RDKit generates a single accurate structure whose identity is only known in this idealistic evaluation setting where the ground truth is known. CoarsenConf, by contrast, is trained with a severely constrained number of conformations to generate $2L$ low-error structures over the entire ensemble. We also note that min performance is not correlated with robust results in the mean and max metrics. Overall, CoarsenConf provides a comprehensive learning framework for robust conformer generation.

10 Conclusion

In this work, we introduce BBO-SYN, a synthesizability-constrained generative framework that leverages template-free chemical reaction prediction to build property-guided synthesis trees. BBO-SYN uses LaP³ over a latent space of viable reactant molecules to select optimal reactants to produce products with high desired property scores. We show that BBO-SYN achieves state-of-the-art performance on QED, DRD2, JNK3, and GSK3 β guided synthesis tasks by substantially increasing product diversity while maintaining high property scores. BBO-SYN can uniquely handle any number of reactants with relative ease compared to discrete reactant systems. Furthermore, BBO-SYN’s latent space, black-box optimization method, and reaction predictor can be easily swapped out for orthogonal future advancements. The sampling efficiency of black-box optimization methods could also be further explored and is a key component to extending BBO-SYN to more complex tasks such as protein docking.

We also present CoarsenConf, a novel approach for robust molecular conformer generation that combines an SE(3)-equivariant hierarchical VAE with geometric coarse-graining techniques for autoregressive conformer generation. By utilizing easy-to-obtain approximate conformations, our model effectively learns the optimal distortion to generate low-energy conformers. Furthermore, CoarsenConf possesses unrestricted degrees of freedom, as it can adjust atomic coordinates, distances, and torsion angles freely. Experiments on QM9 and DRUGS demonstrate the effectiveness of CoarsenConf compared to existing methods, while requiring less training time and data. Moreover, our study expands upon existing conformer generation benchmarks, providing valuable insights into robust generation.

References

- Keir Adams and Connor W. Coley. Equivariant shape-conditioned generation of 3d molecules for ligand-based drug design. *ArXiv*, abs/2210.04893, 2022.
- Simon Axelrod and Rafael Gómez-Bombarelli. Geom, energy-annotated molecular conformations for property prediction and molecular generation. *Scientific Data*, 9(1):185, 2022. doi: 10.1038/s41597-022-01288-4. URL <https://doi.org/10.1038/s41597-022-01288-4>.
- Anders Bøgevig, Hans-Jürgen Federsel, Fernando Huerta, Michael G. Hutchings, Hans Kraut, Thomas Langer, Peter Löw, Christoph Oppawsky, Tobias Rein, and Heinz Saller. Route design in the 21st century: The icsynth software tool as an idea generator for synthesis prediction. *Organic Process Research & Development*, 19(2):357–368, 02 2015. doi: 10.1021/op500373e. URL <https://doi.org/10.1021/op500373e>.
- John Bradshaw, Brooks Paige, Matt J. Kusner, Marwin H. S. Segler, and José Miguel Hernández-Lobato. A model to search for synthesizable molecules, 2019.
- John Bradshaw, Brooks Paige, Matt J Kusner, Marwin Segler, and José Miguel Hernández-Lobato. Barking up the right tree: an approach to search over molecule synthesis dags. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 6852–6866. Curran Associates, Inc., 2020. URL <https://proceedings.neurips.cc/paper/2020/file/4cc05b35c2f937c5bd9e7d41d3686fff-Paper.pdf>.
- Benson Chen, Xiang Fu, Regina Barzilay, and T. Jaakkola. Fragment-based sequential translation for molecular optimization. *ArXiv*, abs/2111.01009, 2021.

- Shuan Chen and Yousung Jung. Deep retrosynthetic reaction prediction using local reactivity and global attention. *JACS Au*, 1(10):1612–1620, 10 2021. doi: 10.1021/jacsau.1c00246. URL <https://doi.org/10.1021/jacsau.1c00246>.
- Connor W. Coley, Luke Rogers, William H. Green, and Klavs F. Jensen. Computer-assisted retrosynthesis based on molecular similarity. *ACS Central Science*, 3(12):1237–1245, 12 2017. doi: 10.1021/acscentsci.7b00355. URL <https://doi.org/10.1021/acscentsci.7b00355>.
- Connor W. Coley, William H. Green, and Klavs F. Jensen. Machine learning in computer-aided synthesis planning. *Accounts of Chemical Research*, 51(5):1281–1289, 2018a. doi: 10.1021/acs.accounts.8b00087. URL <https://doi.org/10.1021/acs.accounts.8b00087>. PMID: 29715002.
- Connor W. Coley, Luke Rogers, William H. Green, and Klavs F. Jensen. Scscore: Synthetic complexity learned from a reaction corpus. *Journal of Chemical Information and Modeling*, 58(2): 252–261, 02 2018b. doi: 10.1021/acs.jcim.7b00622. URL <https://doi.org/10.1021/acs.jcim.7b00622>.
- Connor W. Coley, Dale A. Thomas, Justin A. M. Lummiss, Jonathan N. Jaworski, Christopher P. Breen, Victor Schultz, Travis Hart, Joshua S. Fishman, Luke Rogers, Hanyu Gao, Robert W. Hicklin, Pieter P. Plehiers, Joshua Byington, John S. Piotti, William H. Green, A. John Hart, Timothy F. Jamison, and Klavs F. Jensen. A robotic platform for flow synthesis of organic compounds informed by ai planning. *Science*, 365(6453):eaax1566, 2019a. doi: 10.1126/science.aax1566. URL <https://www.science.org/doi/abs/10.1126/science.aax1566>.
- Connor W. Coley, Wengong Jin, Luke Rogers, Timothy F. Jamison, Tommi S. Jaakkola, William H. Green, Regina Barzilay, and Klavs F. Jensen. A graph-convolutional neural network model for the prediction of chemical reactivity. *Chem. Sci.*, 10:370–377, 2019b. doi: 10.1039/C8SC04228D. URL <http://dx.doi.org/10.1039/C8SC04228D>.
- Hanjun Dai, Chengtao Li, Connor Coley, Bo Dai, and Le Song. Retrosynthesis prediction with conditional graph logic network. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL <https://proceedings.neurips.cc/paper/2019/file/0d2b2061826a5df3221116a5085a6052-Paper.pdf>.
- Congyue Deng, Or Litany, Yueqi Duan, Adrien Poulénard, Andrea Tagliasacchi, and Leonidas J. Guibas. Vector neurons: A general framework for so(3)-equivariant networks. *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 12180–12189, 2021.
- Hongliang Duan, Ling Wang, Chengyun Zhang, Lin Guo, and Jianjun Li. Retrosynthesis with attention-based nmt model and chemical analysis of “wrong” predictions. *RSC Adv.*, 10:1371–1378, 2020. doi: 10.1039/C9RA08535A. URL <http://dx.doi.org/10.1039/C9RA08535A>.
- Ola Engkvist, Josep Arús-Pous, Esben Jannik Bjerrum, and Hongming Chen. Chapter 13 molecular de novo design through deep generative models. In *Artificial Intelligence in Drug Discovery*, pages 272–300. The Royal Society of Chemistry, 2021. ISBN 978-1-78801-547-9. doi: 10.1039/9781788016841-00272. URL <http://dx.doi.org/10.1039/9781788016841-00272>.
- Peter Ertl and Ansgar Schuffenhauer. Estimation of synthetic accessibility score of drug-like molecules based on molecular complexity and fragment contributions. *Journal of Cheminformatics*, 1(1):8, Jun 2009. ISSN 1758-2946.
- Octavian Ganea, Lagnajit Pattanaik, Connor Coley, Regina Barzilay, Klavs Jensen, William Green, and Tommi Jaakkola. Geomol: Torsional geometric generation of molecular 3d conformer ensembles. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 13757–13769. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper_files/paper/2021/file/725215ed82ab6306919b485b81ff9615-Paper.pdf.
- Wenhao Gao, Rocío Mercado, and Connor W. Coley. Amortized tree generation for bottom-up synthesis planning and synthesizable molecular design. In *International Conference on Learning Representations*, 2022. URL <https://openreview.net/forum?id=FRxhHdnxt1>.

- Sai Krishna Gottipati, Boris Sattarov, Sufeng Niu, Yashaswi Pathak, Haoran Wei, Shengchao Liu, Karam J. Thomas, Simon Blackburn, Connor W. Coley, Jian Tang, Sarath Chandar, and Yoshua Bengio. Learning to navigate the synthetically accessible chemical space using reinforcement learning. In *Proceedings of the 37th International Conference on Machine Learning, ICML'20*. JMLR.org, 2020.
- Rafael Gómez-Bombarelli, Jennifer N. Wei, David Duvenaud, José Miguel Hernández-Lobato, Benjamín Sánchez-Lengeling, Dennis Sheberla, Jorge Aguilera-Iparraguirre, Timothy D. Hirzel, Ryan P. Adams, and Alán Aspuru-Guzik. Automatic chemical design using a data-driven continuous representation of molecules. *ACS Central Science*, 4(2):268–276, 2018. doi: 10.1021/acscentsci.7b00572. URL <https://doi.org/10.1021/acscentsci.7b00572>. PMID: 29532027.
- Jiaqi Han, Yu Rong, Tingyang Xu, and Wen bing Huang. Geometrically equivariant graph neural networks: A survey. *ArXiv*, abs/2202.07230, 2022.
- Nikolaus Hansen. The cma evolution strategy: A comparing review, 2006.
- Markus Hartenfeller, Heiko Zettl, Miriam Walter, Matthias Rupp, Felix Reisen, Ewgenij Proschak, Sascha Weggen, Holger Stark, and Gisbert Schneider. DOGS: reaction-driven de novo design of bioactive compounds. *PLoS Comput Biol*, 8(2):e1002380, February 2012.
- Irina Higgins, Loic Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner. beta-VAE: Learning basic visual concepts with a constrained variational framework. In *International Conference on Learning Representations*, 2017. URL <https://openreview.net/forum?id=Sy2fzU9g1>.
- Julien Horwood and Emmanuel Noutahi. Molecular design in synthetically accessible chemical space via deep reinforcement learning. *ACS Omega*, 5(51):32984–32994, December 2020.
- Kexin Huang, Tianfan Fu, Wenhao Gao, Yue Zhao, Yusuf Roohani, Jure Leskovec, Connor W Coley, Cao Xiao, Jimeng Sun, and Marinka Zitnik. Therapeutics data commons: Machine learning datasets and tasks for drug discovery and development. *NeurIPS Datasets and Benchmarks*, 2021.
- Yinan Huang, Xing Peng, Jianzhu Ma, and Muhan Zhang. 3dlinker: An e(3) equivariant variational autoencoder for molecular linker design. In *International Conference on Machine Learning*, 2022.
- Wengong Jin, Connor W. Coley, Regina Barzilay, and Tommi Jaakkola. Predicting organic reaction outcomes with weisfeiler-lehman network. In *Proceedings of the 31st International Conference on Neural Information Processing Systems, NIPS'17*, page 2604–2613, Red Hook, NY, USA, 2017. Curran Associates Inc. ISBN 9781510860964.
- Wengong Jin, Regina Barzilay, and T. Jaakkola. Hierarchical generation of molecular graphs using structural motifs. In *International Conference on Machine Learning*, 2020a.
- Wengong Jin, Regina Barzilay, and Tommi S. Jaakkola. Hierarchical generation of molecular graphs using structural motifs. *CoRR*, abs/2002.03230, 2020b. URL <https://arxiv.org/abs/2002.03230>.
- Wengong Jin, Regina Barzilay, and Tommi Jaakkola. Antibody-antigen docking and design via hierarchical structure refinement. In *International Conference on Machine Learning*, pages 10217–10227. PMLR, 2022.
- Bowen Jing, Gabriele Corso, Regina Barzilay, and Tommi S. Jaakkola. Torsional diffusion for molecular conformer generation. In *ICLR2022 Machine Learning for Drug Discovery*, 2022. URL <https://openreview.net/forum?id=D9IxPlXPJJS>.
- John Jumper, Richard Evans, Alexander Pritzel, Tim Green, Michael Figurnov, Olaf Ronneberger, Kathryn Tunyasuvunakool, Russ Bates, Augustin Žídek, Anna Potapenko, Alex Bridgland, Clemens Meyer, Simon A. A. Kohl, Andrew J. Ballard, Andrew Cowie, Bernardino Romera-Paredes, Stanislav Nikolov, Rishub Jain, Jonas Adler, Trevor Back, Stig Petersen, David Reiman, Ellen Clancy, Michal Zielinski, Martin Steinegger, Michalina Pacholska, Tamas Berghammer, Sebastian Bodenstern, David Silver, Oriol Vinyals, Andrew W. Senior, Koray Kavukcuoglu,

- Pushmeet Kohli, and Demis Hassabis. Highly accurate protein structure prediction with alphafold. *Nature*, 596(7873):583–589, 2021. doi: 10.1038/s41586-021-03819-2. URL <https://doi.org/10.1038/s41586-021-03819-2>.
- W. Kabsch. Automatic processing of rotation diffraction data from crystals of initially unknown symmetry and cell constants. *Journal of Applied Crystallography*, 26(6):795–800, Dec 1993. doi: 10.1107/S0021889893005588. URL <https://doi.org/10.1107/S0021889893005588>.
- Beomjoon Kim, Kyungjae Lee, Sungbin Lim, Leslie Kaelbling, and Tomas Lozano-Perez. Monte carlo tree search in continuous spaces using voronoi optimistic optimization with regret bounds. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(06):9916–9924, Apr. 2020a. doi: 10.1609/aaai.v34i06.6546. URL <https://ojs.aaai.org/index.php/AAAI/article/view/6546>.
- Hyunho Kim, Eunyoung Kim, Ingo Lee, Bongsung Bae, Minsu Park, and Hojung Nam. Artificial intelligence in drug discovery: A comprehensive review of data-driven and machine learning approaches. *Biotechnol. Bioprocess Eng.*, 25(6):895–930, 2020b.
- Sebastian Kmiecik, Dominik Gront, Michal Kolinski, Lukasz Wieteska, Aleksandra Elzbieta Dawid, and Andrzej Kolinski. Coarse-grained protein models and their applications. *Chemical Reviews*, 116(14):7898–7936, 07 2016. doi: 10.1021/acs.chemrev.6b00163. URL <https://doi.org/10.1021/acs.chemrev.6b00163>.
- Ksenia Korovina, Sailun Xu, Kirthevasan Kandasamy, Willie Neiswanger, Barnabás Póczos, Jeff Schneider, and Eric P. Xing. Chembo: Bayesian optimization of small organic molecules with synthesizable recommendations. In *AISTATS*, pages 3393–3403, 2020. URL <http://proceedings.mlr.press/v108/korovina20a.html>.
- James Law, Zsolt Zsoldos, Aniko Simon, Darryl Reid, Yang Liu, Sing Yoong Khew, A. Peter Johnson, Sarah Major, Robert A. Wade, and Howard Y. Ando. Route designer: A retrosynthetic analysis tool utilizing automated retrosynthetic rule generation. *Journal of Chemical Information and Modeling*, 49(3):593–602, 03 2009. doi: 10.1021/ci800228y. URL <https://doi.org/10.1021/ci800228y>.
- Kangjie Lin, Youjun Xu, Jianfeng Pei, and Luhua Lai. Automatic retrosynthetic route planning using template-free models. *Chem. Sci.*, 11:3355–3364, 2020. doi: 10.1039/C9SC03666K. URL <http://dx.doi.org/10.1039/C9SC03666K>.
- Shitong Luo, Chence Shi, Minkai Xu, and Jian Tang. Predicting molecular conformation via dynamic graph score matching. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 19784–19795. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper_files/paper/2021/file/a45a1d12ee0fb7f1f872ab91da18f899-Paper.pdf.
- Elman Mansimov, Omar Mahmood, Seokho Kang, and Kyunghyun Cho. Molecular geometry prediction using a deep generative graph neural network. *Scientific Reports*, 9, 2019.
- Rémi Munos. Optimistic optimization of a deterministic function without the knowledge of its smoothness. In J. Shawe-Taylor, R. Zemel, P. Bartlett, F. Pereira, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 24. Curran Associates, Inc., 2011. URL <https://proceedings.neurips.cc/paper/2011/file/7e889fb76e0e07c11733550f2a6c7a5a-Paper.pdf>.
- Daniil Polykovskiy, Alexander Zhebrak, Benjamin Sanchez-Lengeling, Sergey Golovanov, Oktai Tatanov, Stanislav Belyaev, Rauf Kurbanov, Aleksey Artamonov, Vladimir Aladinskiy, Mark Veselov, Artur Kadurin, Simon Johansson, Hongming Chen, Sergey Nikolenko, Alan Aspuru-Guzik, and Alex Zhavoronkov. Molecular Sets (MOSES): A Benchmarking Platform for Molecular Generation Models. *Frontiers in Pharmacology*, 2020.
- Philipp Pracht, Fabian Bohle, and Stefan Grimme. Automated exploration of the low-energy chemical space with fast quantum chemical methods. *Phys. Chem. Chem. Phys.*, 22:7169–7192, 2020. doi: 10.1039/C9CP06869D. URL <http://dx.doi.org/10.1039/C9CP06869D>.

- Danny Reidenbach, Micha Livne, Rajesh K. Ilango, Michelle Gill, and Johnny Israeli. Improving small molecule generation using mutual information machine, 2022. URL <https://arxiv.org/abs/2208.09016>.
- Jean-Louis Reymond and Mahendra Awale. Exploring chemical space for drug discovery using the chemical universe database. *ACS Chemical Neuroscience*, 3(9):649–657, 09 2012. doi: 10.1021/cn3000422. URL <https://doi.org/10.1021/cn3000422>.
- Sereina Riniker and Gregory A. Landrum. Better informed distance geometry: Using what we know to improve conformation generation. *Journal of Chemical Information and Modeling*, 55(12): 2562–2574, 12 2015. doi: 10.1021/acs.jcim.5b00654. URL <https://doi.org/10.1021/acs.jcim.5b00654>.
- Reuven Rubinstein. The cross-entropy method for combinatorial and continuous optimization. *Methodology And Computing In Applied Probability*, 1(2):127–190, 1999. doi: 10.1023/A: 1010091220143. URL <https://doi.org/10.1023/A:1010091220143>.
- Mikołaj Sacha, Mikołaj Błaż, Piotr Byrski, Paweł Dąbrowski-Tumański, Mikołaj Chromiński, Rafał Loska, Paweł Włodarczyk-Pruszyński, and Stanisław Jastrzębski. Molecule edit graph attention network: Modeling chemical reactions as sequences of graph edits. *Journal of Chemical Information and Modeling*, 61(7):3273–3284, 2021. doi: 10.1021/acs.jcim.1c00537. URL <https://doi.org/10.1021/acs.jcim.1c00537>. PMID: 34251814.
- Victor Garcia Satorras, Emiel Hooeboom, and Max Welling. E(n) equivariant graph neural networks. In *International Conference on Machine Learning*, 2021.
- Kristof T. Schütt, Oliver T. Unke, and Michael Gastegger. Equivariant message passing for the prediction of tensorial properties and molecular spectra. In *International Conference on Machine Learning*, 2021.
- Philippe Schwaller, Teodoro Laino, Théophile Gaudin, Peter Bolgar, Christopher A. Hunter, Costas Bekas, and Alpha A. Lee. Molecular transformer: A model for uncertainty-calibrated chemical reaction prediction. *ACS Central Science*, 5(9):1572–1583, 09 2019. doi: 10.1021/acscentsci.9b00576. URL <https://doi.org/10.1021/acscentsci.9b00576>.
- Marwin H. S. Segler, Mike Preuss, and Mark P. Waller. Planning chemical syntheses with deep neural networks and symbolic ai. *Nature*, 555(7698):604–610, 2018a. doi: 10.1038/nature25978. URL <https://doi.org/10.1038/nature25978>.
- Marwin H. S. Segler, Mike Preuss, and Mark P. Waller. Planning chemical syntheses with deep neural networks and symbolic ai. *Nature*, 555(7698):604–610, 2018b. doi: 10.1038/nature25978. URL <https://doi.org/10.1038/nature25978>.
- Chence Shi, Minkai Xu, Zhaocheng Zhu, Weinan Zhang, Ming Zhang, and Jian Tang. Graphaf: a flow-based autoregressive model for molecular graph generation. *ArXiv*, abs/2001.09382, 2020.
- Chence Shi, Shitong Luo, Minkai Xu, and Jian Tang. Learning gradient fields for molecular conformation generation. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 9558–9568. PMLR, 18–24 Jul 2021. URL <https://proceedings.mlr.press/v139/shi21b.html>.
- Gregor Simm and Jose Miguel Hernandez-Lobato. A generative model for molecular distance geometry. In Hal Daumé III and Aarti Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 8949–8958. PMLR, 13–18 Jul 2020.
- Hannes Stärk, Octavian Ganea, Lagnajit Pattanaik, Regina Barzilay, and Tommi Jaakkola. Equibind: Geometric deep learning for drug binding structure prediction. In *International Conference on Machine Learning*, pages 20503–20521. PMLR, 2022.

- Sara Szymkuć, Ewa P. Gajewska, Tomasz Klucznik, Karol Molga, Piotr Dittwald, Michał Startek, Michał Bajczyk, and Bartosz A. Grzybowski. Computer-assisted synthetic planning: The end of the beginning. *Angewandte Chemie International Edition*, 55(20):5904–5937, 2016. doi: <https://doi.org/10.1002/anie.201506101>. URL <https://onlinelibrary.wiley.com/doi/abs/10.1002/anie.201506101>.
- Nathaniel Thomas, Tess Smidt, Steven Kearnes, Lusann Yang, Li Li, Kai Kohlhoff, and Patrick Riley. Tensor field networks: Rotation-and translation-equivariant neural networks for 3d point clouds. *arXiv preprint arXiv:1802.08219*, 2018.
- Zhengkai Tu and Connor W. Coley. Permutation invariant graph-to-sequence model for template-free retrosynthesis and reaction prediction, 2022. URL <https://openreview.net/forum?id=LLHwQh9zEb>.
- H Maarten Vinkers, Marc R de Jonge, Frederik F D Daeyaert, Jan Heeres, Lucien M H Koymans, Joop H van Lenthe, Paul J Lewi, Henk Timmerman, Koen Van Aken, and Paul A J Janssen. SYNOPSIS: SYNthesize and OPTimize system in silico. *J Med Chem*, 46(13):2765–2773, June 2003.
- Linnan Wang, Rodrigo Fonseca, and Yuandong Tian. Learning search space partition for black-box optimization using monte carlo tree search. In *Proceedings of the 34th International Conference on Neural Information Processing Systems, NIPS'20*, Red Hook, NY, USA, 2020a. Curran Associates Inc. ISBN 9781713829546.
- Wujie Wang, Minkai Xu, Chen Cai, Benjamin Kurt Miller, Tess E. Smidt, Yusu Wang, Jian Tang, and Rafael Gomez-Bombarelli. Generative coarse-graining of molecular conformations. In *International Conference on Machine Learning*, 2022.
- Xiaoxue Wang, Yujie Qian, Hanyu Gao, Connor W. Coley, Yiming Mo, Regina Barzilay, and Klavs F. Jensen. Towards efficient discovery of green synthetic pathways with monte carlo tree search and reinforcement learning. *Chem. Sci.*, 11:10959–10972, 2020b. doi: 10.1039/D0SC04184J. URL <http://dx.doi.org/10.1039/D0SC04184J>.
- Ronald J. Williams and David Zipser. A Learning Algorithm for Continually Running Fully Recurrent Neural Networks. *Neural Computation*, 1(2):270–280, 06 1989. ISSN 0899-7667. doi: 10.1162/neco.1989.1.2.270. URL <https://doi.org/10.1162/neco.1989.1.2.270>.
- Minkai Xu, Shitong Luo, Yoshua Bengio, Jian Peng, and Jian Tang. Learning neural generative dynamics for molecular conformation generation. In *International Conference on Learning Representations*, 2021a. URL <https://openreview.net/forum?id=pAbm1qfheGk>.
- Minkai Xu, Wujie Wang, Shitong Luo, Chence Shi, Yoshua Bengio, Rafael Gomez-Bombarelli, and Jian Tang. An end-to-end framework for molecular conformation generation via bilevel programming. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 11537–11547. PMLR, 18–24 Jul 2021b. URL <https://proceedings.mlr.press/v139/xu21f.html>.
- Minkai Xu, Lantao Yu, Yang Song, Chence Shi, Stefano Ermon, and Jian Tang. Geodiff: A geometric diffusion model for molecular conformation generation. In *International Conference on Learning Representations*, 2022. URL <https://openreview.net/forum?id=PzcvxEMzvQC>.
- Kevin Yang, Tianjun Zhang, Chris Cummins, Brandon Cui, Benoit Steiner, Linnan Wang, Joseph E. Gonzalez, Dan Klein, and Yuandong Tian. Learning space partitions for path planning. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan, editors, *NeurIPS*, 2021.
- Soojung Yang and Rafael Gomez-Bombarelli. Chemically transferable generative backmapping of coarse-grained proteins. *ArXiv*, abs/2303.01569, 2023.
- Jiyang Zhang, Xi Xiao, Long-Kai Huang, Yu Rong, and Yatao Bian. Fine-tuning graph neural networks via graph topology induced optimal transport, 2022a.
- Qi Zhang, Chang Liu, Stephen Wu, and Ryo Yoshida. Bayesian sequential stacking algorithm for concurrently designing molecules and synthetic reaction networks, 2022b.

Gengmo Zhou, Zhifeng Gao, Qiankun Ding, Hang Zheng, Hongteng Xu, Zhewei Wei, Linfeng Zhang, and Guolin Ke. Uni-mol: A universal 3d molecular representation learning framework. In *The Eleventh International Conference on Learning Representations, 2023a*. URL <https://openreview.net/forum?id=6K2RM6wVqKu>.

Gengmo Zhou, Zhifeng Gao, Zhewei Wei, Hang Zheng, and Guolin Ke. Do deep learning methods really perform better in molecular conformation generation? *ArXiv*, abs/2302.07061, 2023b.

Jinhua Zhu, Yingce Xia, Chang Liu, Lijun Wu, Shufang Xie, Yusong Wang, Tong Wang, Tao Qin, Wengang Zhou, Houqiang Li, Haiguang Liu, and Tie-Yan Liu. Direct molecular conformation generation. *Transactions on Machine Learning Research*, 2022. URL <https://openreview.net/forum?id=1CPOHiztuw>.

A BBO Supplemental Experiments

Below we present additional figures for the GSK3 β and DRD2 black-box method and latent space size ablations.

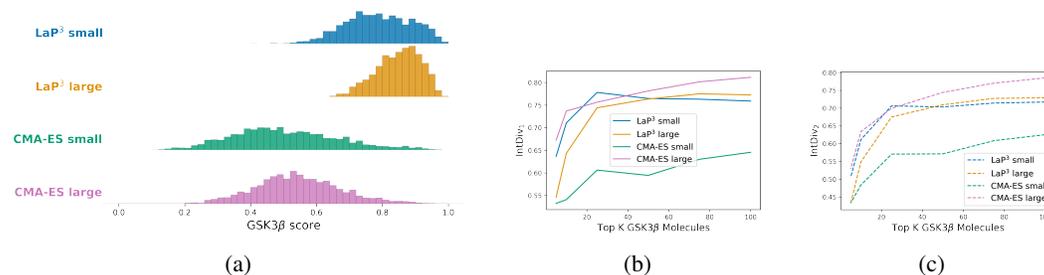


Figure 12: Optimization of GSK3 β for various latent space sizes and BBO methods. All use the same initial starting molecules. (bc) Internal Diversity of top 5, 10, 25, 50, 75, 100 GSK3 β optimized molecules. Solid and dashed lines correspond to IntDiv₁ and IntDiv₂.

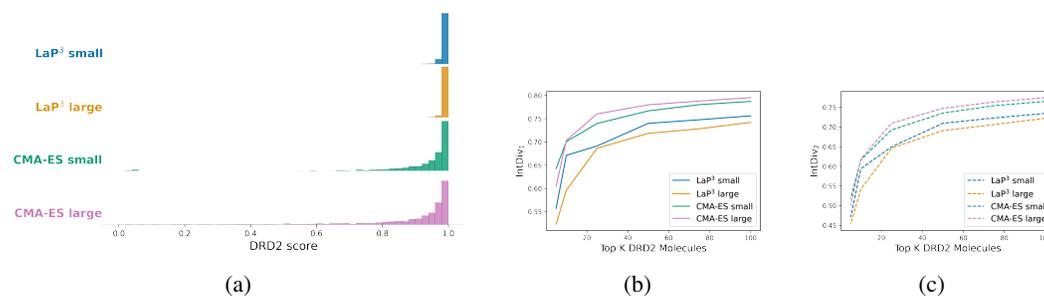


Figure 13: (a) Optimization of DRD2 for various latent space sizes and BBO methods. All use the same initial starting molecules. (bc) Internal Diversity of top 5, 10, 25, 50, 75, 100 DRD2 optimized molecules. Solid and dashed lines correspond to IntDiv₁ and IntDiv₂.

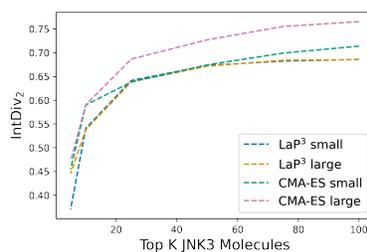


Figure 14: Internal Diversity of top 5, 10, 25, 50, 75, 100 JNK3 optimized molecules. Dashed lines correspond to IntDiv₂.



Figure 17: Top 6 BBO-SYN generated products optimized for DRD2.

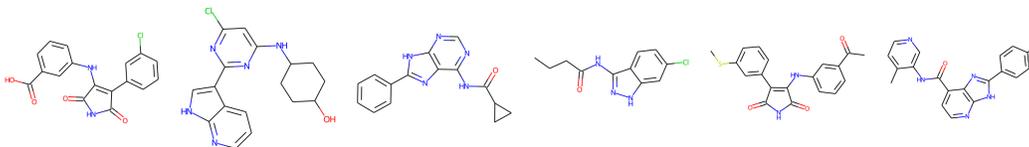


Figure 15: Top 6 BBO-SYN generated products optimized for GSK3 β .

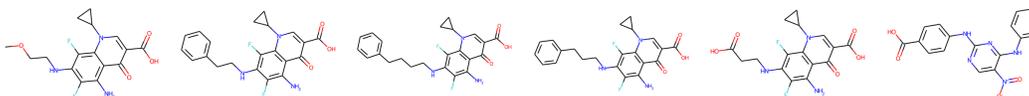


Figure 16: Top 6 DoG-Gen generated products optimized for GSK3 β .

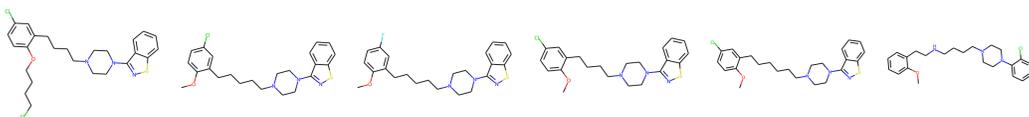


Figure 18: Top 6 DoG-Gen generated products optimized for DRD2.

B CoarsenConf Encoder Equations

B.1 Fine-grain Module

More formally, the fine-grain module is defined as follows:

$$\begin{aligned}
 \mathbf{m}_{j \rightarrow i} &= \phi^e(\mathbf{h}_i^{(l)}, \mathbf{h}_j^{(l)}, \|\mathbf{x}_i^{(l)} - \mathbf{x}_j^{(l)}\|^2, \mathbf{f}_{j \rightarrow i}), \forall (I, J) \in \mathcal{E} \cup \mathcal{E}' \\
 \mathbf{u}_{j' \rightarrow i} &= a_{j' \rightarrow i} \mathbf{W} \mathbf{h}_{j'}^{(l)}, \forall i \in \mathcal{V}, j' \in \mathcal{V}' \\
 \mathbf{m}_i &= \frac{1}{|\mathcal{N}(i)|} \sum_{j \in \mathcal{N}(i)} \mathbf{m}_{j \rightarrow i}, \forall i \in \mathcal{V} \cup \mathcal{V}' \\
 \mathbf{u}_i &= \sum_{j' \in \mathcal{V}'} \mathbf{u}_{j' \rightarrow i}, \forall i \in \mathcal{V}, \quad \text{and} \quad \mathbf{u}'_i = 0 \\
 \mathbf{x}_i^{(l+1)} &= \eta \cdot \mathbf{x}_i^{(0)} + (1 - \eta) \cdot \mathbf{x}_i^{(l)} + \sum_{j \in \mathcal{N}(i)} (\mathbf{x}_i^{(l)} - \mathbf{x}_j^{(l)}) \phi^x(\mathbf{m}_{j \rightarrow i}) \\
 \mathbf{h}_i^{(l+1)} &= (1 - \beta) \cdot \mathbf{h}_i^{(l)} + \beta \cdot \phi^h(\mathbf{h}_i^{(l)}, \mathbf{m}_i, \mathbf{u}_i, \mathbf{f}_i), \forall i \in \mathcal{V} \cup \mathcal{V}'
 \end{aligned} \tag{6}$$

where $a_{j \rightarrow i}$ are SE(3)-invariant attention coefficients derived from h embeddings, $\mathcal{N}(i)$ are the graph neighbors of node i , and W is a parameter matrix. The various ϕ functions are modeled using shallow neural networks, with ϕ^x outputting a scalar and ϕ^e and ϕ^h returning a D-dimensional vector. We note that the attention flows in a single direction from RDKit approximation to ground truth to prevent leakage in the parameterization of the learned prior distribution.

B.2 Pooling Module

More formally, the pooling module is defined as follows:

$$\begin{aligned}
\mathbf{m}_{j \rightarrow I} &= \phi^e(\mathbf{H}_I^{(l)}, \mathbf{h}_j^{(l)}, \|\mathbf{X}_I^{(l)} - \mathbf{x}_j^{(l)}\|^2, \mathbf{f}_{j \rightarrow I}), \forall (I, J) \in \mathcal{E} \cup \mathcal{E}' \\
\mathbf{m}_I &= \frac{1}{|\mathcal{N}(I)|} \sum_{j \in \mathcal{N}(I)} \mathbf{m}_{j \rightarrow I}, \forall I \in \mathcal{V} \cup \mathcal{V}' \\
\mathbf{X}_I^{(l+1)} &= \eta \cdot \mathbf{X}_I^{(0)} + (1 - \eta) \cdot \mathbf{X}_I^{(l)} + \sum_{j \in \mathcal{N}(I)} (\mathbf{X}_I^{(l)} - \mathbf{x}_j) \phi^x(\mathbf{m}_{j \rightarrow I}) \\
\mathbf{H}_I^{(l+1)} &= (1 - \beta) \cdot \mathbf{H}_I^{(l)} + \beta \cdot \phi^h(\mathbf{H}_I^{(l)}, \mathbf{m}_I, \mathbf{f}_I), \forall I \in \mathcal{V} \cup \mathcal{V}'
\end{aligned} \tag{7}$$

where capital letters refer to the CG representation of the pooling graph. As the pooling graph only contains edges from fine to coarse nodes, the fine-grain coordinates and features remain unchanged.

B.3 Coarse-grain Module

In the first step, invariant CG features H and equivariant features $\mathbf{v} \in \mathbb{R}^{F \times 3}$ are transformed and mixed to construct new expressive intermediate features H', H'', \mathbf{v}' by

$$H'_I = \phi_1(h_I, \|\text{VN-MLP}_1(\mathbf{v}_I)\|) \in \mathbb{R}^D \tag{8a}$$

$$H''_I = \phi_2(h_j, \|\text{VN-MLP}_2(\mathbf{v}_j)\|) \in \mathbb{R}^F \tag{8b}$$

$$\mathbf{v}'_I = \text{diag}\{\phi_3(H_I)\} \cdot \text{VN-MLP}_3(\mathbf{v}_I) \in \mathbb{R}^{F \times 3} \tag{8c}$$

Next, a point convolution [Thomas et al., 2018, Schütt et al., 2021, Huang et al., 2022] is applied to linearly transform the mixed features H', H'', \mathbf{v}' into messages:

$$\mathbf{m}_{I \leftarrow J}^H = \text{Ker}_1(\|\mathbf{r}_{I,J}\|) \odot H'_J \tag{9a}$$

$$\mathbf{m}_{I \leftarrow J}^v = \text{diag}\{\text{Ker}_2(\|\mathbf{r}_{I,J}\|)\} \cdot \mathbf{v}'_J + (\text{Ker}_3(\|\mathbf{r}_{I,J}\|) \odot h''_J) \cdot \mathbf{r}_{I,J}^\top \tag{9b}$$

$$\mathbf{u}_{J' \rightarrow I} = a_{J' \rightarrow I} \mathbf{W} \mathbf{H}_{J'}^{(l)}, \forall I \in \mathcal{V}, J' \in \mathcal{V}' \tag{9c}$$

$$\mathbf{u}_I = \sum_{J' \in \mathcal{V}'} \mathbf{u}_{J' \rightarrow I}, \forall I \in \mathcal{V}, \quad \text{and} \quad \mathbf{u}'_I = 0 \tag{9d}$$

$$\mathbf{H}_I^{l+1} = (1 - \delta_H) \cdot v_I + \delta_H \cdot \text{MLP}(H'_I, \sum_{J \in \mathcal{N}(I)} \mathbf{m}_{I \leftarrow J}^H, \mathbf{u}_I), \forall I \in \mathcal{V} \cup \mathcal{V}' \tag{9e}$$

$$\mathbf{v}_I = (1 - \delta_v) \cdot v_I + \delta_v \cdot \text{VN-MLP}_4(\mathbf{v}_I, \sum_{J \in \mathcal{N}(I)} \mathbf{m}_{I \leftarrow J}^v), \forall I \in \mathcal{V} \cup \mathcal{V}' \tag{9f}$$

We note that for $l > 0$ the H_I from the CG module are used in the next layer’s pooling module as seen in the dashed lines in Fig. 10 (I), creating a cyclic dependency to learn an information-rich and ground CG representation. All equivariant CG features v are initialized as zero. As point convolutions and VN operations are strictly SO(3)-equivariant, we subtract the molecule’s centroid from the atomic coordinates prior to encoding, making it effectively SE(3)-equivariant.