

# Structured Representations for Goal-Directed Decision Making

*Vivek Myers*

Electrical Engineering and Computer Sciences  
University of California, Berkeley

Technical Report No. UCB/EECS-2024-208

<http://www2.eecs.berkeley.edu/Pubs/TechRpts/2024/EECS-2024-208.html>

December 8, 2024



Copyright © 2024, 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.



# Structured Representations for Goal-Directed Decision Making

by

Vivek Myers

A thesis submitted in partial satisfaction of the

requirements for the degree of

Master of Science

in

Computer Science

in the

Graduate Division

of the

University of California, Berkeley

Committee in charge:

Professor Sergey Levine, Co-chair

Professor Anca Dragan, Co-chair

Fall 2024

# TABLE OF CONTENTS

---

<b>Table of Contents</b>	<b>i</b>
<b>List of Figures</b>	<b>iv</b>
<b>List of Tables</b>	<b>xii</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Publications . . . . .	1
1.2 Bibliography . . . . .	2
<b>I REPRESENTING TASKS AND GOALS</b>	<b>3</b>
<b>2 Goal Representations for Instruction Following</b>	<b>4</b>
2.1 Related Work . . . . .	5
2.2 Problem Setup . . . . .	6
2.3 Goal Representations for Instruction Following . . . . .	7
2.4 Experiments . . . . .	9
2.5 Scaling of Annotation Supervision . . . . .	18
2.6 Conclusion . . . . .	18
<b>3 Policy Adaptation via Language Optimization</b>	<b>20</b>
3.1 Related Work . . . . .	22
3.2 Policy Adaptation via Language Optimization . . . . .	22
3.3 Regret Analysis . . . . .	26
3.4 Experiments . . . . .	33
3.5 Conclusion . . . . .	38
<b>II REPRESENTATIONS FOR COMPOSITIONAL DECISION MAKING</b>	<b>42</b>
<b>4 A Metric Structure for Successor Representations</b>	<b>43</b>
4.1 General distances for goal-reaching . . . . .	45
4.2 Using our Temporal Distance for RL . . . . .	49
4.3 One-step Metric Distillation (CMD-1): . . . . .	51
4.4 Two-step metric distillation (CMD-2) . . . . .	52
4.5 Experiments . . . . .	54
4.6 Hitting Times . . . . .	58
4.7 Proofs . . . . .	64
4.8 Didactic Examples . . . . .	65

4.9	Action-Invariance . . . . .	68
4.10	Related Work . . . . .	68
<b>5</b>	<b>Temporal Representation Alignment</b>	<b>71</b>
5.1	Temporal Representation Alignment . . . . .	72
5.2	Experiments . . . . .	76
5.3	Additional Visualizations . . . . .	83
5.4	Analysis of Compositionality . . . . .	83
5.5	Related Work . . . . .	85
5.6	Conclusions and Limitations . . . . .	87
<b>6</b>	<b>Planning with Contrastive Representations</b>	<b>88</b>
6.1	Preliminaries . . . . .	90
6.2	Contrastive Representations Make Inference Easy . . . . .	92
6.3	Proofs . . . . .	95
6.4	Numerical Simulation . . . . .	102
6.5	Additional Experiments . . . . .	105
6.6	Related Work . . . . .	106
6.7	Discussion . . . . .	108
<b>7</b>	<b>Invariance to Planning</b>	<b>110</b>
7.1	Preliminaries . . . . .	111
7.2	Planning Invariance and Horizon Generalization . . . . .	112
7.3	Methods for Planning Invariance: Old and New . . . . .	118
7.4	Experiments . . . . .	120
7.5	Definition of Path Relaxation . . . . .	123
7.6	Formalizing Planning Invariance . . . . .	125
7.7	New Methods for Planning Invariance . . . . .	131
7.8	Self-Consistent Models . . . . .	131
7.9	Evidence from Prior Work . . . . .	133
7.10	Conclusion . . . . .	134
<b>III</b>	<b>REPRESENTATIONS FOR TRACTABLE INTRINSIC MOTIVATION</b>	<b>136</b>
<b>8</b>	<b>Empowerment via Successor Representations</b>	<b>137</b>
8.1	The Information Geometry of Empowerment . . . . .	139
8.2	Maximizing Empowerment with Contrastive Representations . . . . .	147
8.3	Experiments . . . . .	152
8.4	Additional Ablations and Qualitative Results . . . . .	155
8.5	Discussion . . . . .	157
8.6	Related Work . . . . .	158

<b>9</b>	<b>The Representation Geometry of Intrinsic Motivation</b>	<b>160</b>
9.1	Binary NCE . . . . .	161
9.2	Symmetric infoNCE with a learned temperature . . . . .	162
9.3	Euclidean infoNCE . . . . .	164
	<b>Bibliography</b>	<b>168</b>
<b>A</b>	<b>Videos and Code</b>	<b>197</b>
<b>B</b>	<b>Language Model Prompting Details</b>	<b>198</b>
B.1	GRIF Instruction Augmentation . . . . .	198
B.2	PALO Prompting Details . . . . .	198
<b>C</b>	<b>Experiment Details</b>	<b>202</b>
C.1	PALO Evaluation Details . . . . .	202
C.2	TRA Implementation . . . . .	202
C.3	CMD Implementation . . . . .	209
C.4	ESR Details . . . . .	209
C.5	Planning Invariance and Horizon Generalization . . . . .	210

## LIST OF FIGURES

---

2.1	Our approach learns representations of instructions that are aligned to transitions from the initial state to the goal . . . . .	4
2.2	<b>Left:</b> We explicitly align representations between goal-conditioned and language-conditioned tasks on the labeled dataset $\mathcal{D}_L$ through contrastive learning. <b>Right:</b> Given the pre-trained task representations, we train a policy on both labeled and unlabeled datasets. . . . .	9
2.3	Comparison of success rates $\pm$ SE between the top three methods across all trials within the three scenes. Two other baselines LCBC and R3M (not shown) achieved 0.0 success in all evaluation tasks although they do succeed on tasks that are heavily covered in the training data. Statistical significance is starred. The initial observation and instructions of each scene are shown. . . . .	10
2.4	Success rates of ablations with one standard error. . . . .	13
2.5	<b>Left:</b> Comparison of the top-5 text to image retrieval accuracy of representations learned by different ablations. <b>Right:</b> Examples of retrieved image pairs given instructions. . . . .	14
2.6	Scaling of GRIF grounding capability by number of language annotations available. . . . .	18
3.1	An overview of the PALO algorithm for few-shot adaptation with language. (Left) We build off a pre-trained policy that has learned to follow low-level language instructions from a large dataset of expert demonstrations. (Middle) Given a new task and a few expert demonstrations, we use a VLM to propose candidate decompositions into subtasks. We optimize over these decompositions jointly with the partitions of trajectories into subtasks, selecting the subtask decomposition that minimizes the validation error of the learned policy. (Right) At test time, we condition the pre-trained policy on the selected decomposition to solve the task. . . . .	20
3.2	PALO enables pre-trained generalist policies to adapt new tasks with as few as five demonstrations by searching in language space instead of parameter space. . . . .	21
3.3	A visualization of an example execution of our method on the long-horizon task “put the beet toy in the drawer.” The VLM deconstructs $\ell$ into candidate high-level subtasks $c_{1:K}^H$ and low-level subtasks $c^{L1} : K$ and optimizes over the expert trajectories. The optimal $c_{1:K}^H$ and $c^{L1} : m$ are chosen and unrolled in real-world evaluations, which result in successful completion of the task (trajectory shown in gray). . . . .	25
3.4	Comparison of PALO with baseline methods on different scenes with one standard error. . . . .	35

3.5	An execution of our method on the task “pour the contents of the scoop into the bowl.” Full breakdown of task and instructions can be seen at Section 3.4. . . . .	36
3.6	Ablation study of PALO on different scenes, plotted with one standard error. . . . .	36
3.7	Performance of PALO with 5 demonstrations compared to finetuning Octo on different number of demonstrations, plotted with one standard error. . . . .	37
3.8	An execution of our method on the task “Pry out the pot using the ladle.”	38
3.9	An execution of our method on the task “pour the contents of the scoop into the bowl.”. Note that the high level instruction is $\ell$ itself, as the best-proposed language decomposition does not create additional subtasks. . . . .	39
3.10	Failure in execution: while the robot completed every subtask correctly up until the last subtask, it did not achieve it due to errors within the policy. . . . .	39
3.11	Spatial reasoning failure occurred when masking out low level instruction. The task was to “sweep the mints using the towel.” Due to the presence of the pot and the mushroom, being both strong priors within BridgeData, the policy chose not to follow the high level instruction. . . . .	40
3.12	Grounding failure occurs when high level instruction is masked out. While the low level instruction “move the gripper left” correctly predicts the next reasonable action, masking out the context of the subtask “put the mushroom in the bowl” causes the policy to overshoot its trajectory. . . . .	40
3.13	In this instance, we mask out the high level instructions, and the policy is only conditioned on the low-level instructions. We see that the low-level instruction “move the gripper forward and left.” causes the robot to overshoot its trajectory and causes failure in execution. . . . .	41
4.1	An overview of our theoretical distance construction as well as the concrete implementation with metric distillation . . . . .	44
4.2	<i>(Left)</i> We collect four types of trajectories on this 2D navigation task. The large gray arrows depict the direction of motion. Note that navigating between certain states requires piecing together trajectories of different colors. <i>(Right)</i> Our proposed temporal distance correctly pieces together trajectories, allowing an RL agent to successfully navigate between pairs of states that never occur on the same trajectory. This combinatorial generalization [1] or “stitching” [2] property is typically associated with bootstrapping with temporal difference learning, which our temporal distances do not require. . . . .	55
4.3	Metric distillation enables more efficient offline training and long-horizon compositional generalization . . . . .	56
4.4	A simple illustration of a metric over $\mathcal{S} \times \mathcal{A}$ . . . . .	65

5.1 Example rollouts of a task with TRA and GCBC to put all food items in the bowl. While TRA can implicitly decompose the task into steps and execute them one by one, GCBC is unable to do that and fails to ground to any relevant objects. GCBC+AWR on the other hand only grounds one object, failing to display any compositionality . . . . . 72

5.2 Aggregated performance on compositional generalization tasks, consisting of instruction-following and goal-reaching tasks. . . . . 77

5.3 Example rollouts of a task with TRA and LCBC. While TRA is able to successfully compose the steps to complete the task, LCBC fails to ground the instruction correctly. . . . . 81

5.4 Aggregated success rate of using AWR as an additional policy learning metric over all 4 scenes. . . . . 81

5.5 In these figures, we see that TRA is able to perform good compositional generalization over a variety of tasks seen within BridgeData . . . . . 83

5.6 Most of the failure cases came from the fact that a policy cannot learn depth reasoning, causing early grasping or late release, and it has trouble reconciling with multimodal behavior . . . . . 84

5.7 Visualizing the bound (Fig. 5.7 from Theorem 5.1) on the compositional generalization error. . . . . 86

6.1 We apply temporal contrastive learning to observation pairs to obtain representations  $(\psi(x_0), \psi(x_{t+k}))$  such that  $A\psi(x_0)$  is close to  $\psi(x_{t+k})$ . While inferring waypoints in the high-dimensional observation space is challenging, we show that the distribution over intermediate latent representations has a closed form solution corresponding to linear interpolation between the initial and final representations. . . . . 88

6.2 A parametrization for temporal contrastive learning. . . . . 92

6.3 Predicting representations of future states. . . . . 93

6.4 **Numerical simulation of our analysis.** (*Top Left*) Toy dataset of time-series data consisting of many outwardly-spiraling trajectories. We apply temporal contrastive learning to these data. (*Top Right*) For three initial observations (■), we use the learned representations to predict the distribution over future observations. Note that these distributions correctly capture the spiral structure. (*Bottom Left*) For three observations (★), we use the learned representations to predict the distribution over preceding observations. (*Bottom Right*) Given an initial and final observation, we plot the inferred posterior distribution over the waypoint (Section 6.2). The representations capture the shape of the distribution. . . . . 96

6.5 Using inferred paths over our contrastive representations for control boosts success rates by  $4.5\times$  on the most difficult goals (18%  $\rightarrow$  84%). Alternative representation learning techniques fail to improve performance when used for planning. . . . . 103

6.6 Planning for 39-dimensional robotic door opening. (*Top Left*) We use a dataset of trajectories demonstrating door opening from prior work [2] to learn representations. (*Top Right*) We use our method and three baselines to infer one intermediate waypoint between the first and last observation in a trajectory from a held-out validation set. Errors are measured using the mean squared error with the true waypoint observation; predicted representations are converted to observations using nearest neighbors on a validation set. (*Bottom*) We visualize a TSNE [3] of the states along the sampled trajectory as blue circles, with the transparency indicating the index along the trajectory. The inferred plan is shown as red circles connected by arrows. Our method generates better plans than alternative representation learning methods (PCA, VIP). . . . . 104

6.7 Our approach enables a goal-conditioned policy to reach farther targets (red) from the start (green) by planning over intermediate waypoints (orange). . . . . 105

6.8 Planning for 46-dimensional robotic hammering. (*Left*) A dataset of trajectories demonstrating a hammer knocking a nail into a board [2]. (*Center*) We visualize the learned representations as blue circles, with the transparency indicating the index of that observation along the trajectory. We also visualize the inferred plan (Section 6.2) as red circles connected by arrows. (*Right*) Representations learned by PCA on the same trajectory as (*a, left*). . . . . 106

6.9 **Stock Prediction.** We apply temporal contrastive learning to time series data of the stock market. Data are the opening prices for the 500 stocks in the S&P 500, over a four year window. We remove 30 stocks that are missing data. For evaluation, we choose a 100 day window from a validation set, and use Theorem 6.2 to perform “inpainting”, predicting the intermediate stock prices *jointly* for all stocks (orange), given the first and last stock price. The true stock prices are shown in blue. While we do not claim that this is a state-of-the-art model for stock prediction, this experiment demonstrates another potential application of our theoretical results. . . . . 107

7.1 **Horizon generalization.** A policy generalizes over the horizon if optimality over all start-goal pairs  $(s, s')$  a small temporal distance  $d(s, s') < c$  apart (say, in the training set) leads to optimality over all possible start-goal pairs. . . . . 110



7.2 **Visualizing planning invariance.** Planning invariance (Definition 7.1) means that a policy should take similar actions when directed towards a goal (purple arrow and purple star) as when directed towards an intermediate waypoint (brown arrow and brown star). We visualize a policy with (*Right*) and without (*Left*) this property via the misalignment and alignment of actions towards the waypoint and the goal, where the optimal path is tan and the suboptimal path is gray. . . . . 113

7.3 **Invariance to planning leads to horizon generalization.** (*Left*) Invariance to planning maps  $(s_0, \{s_1, s_2, s_4\})$  together in latent space, which results in a shared optimal action. (*Right*) After successfully reaching the closest waypoint  $s_1$  in 1 step, the next optimal action is also shared, meaning any trajectory of length 2 is optimal. We can repeat this argument for trajectories of length 4, 8, . . . until the entire reachable state space is covered. . . . . 114

7.4 **Quantifying horizon generalization and invariance to planning.** On a simple navigation task, we collect short trajectories and train two goal-conditioned policies, comparing both to a random policy. (*Left*) We evaluate on  $(s, g)$  pairs of varying distances, observing that metric regression with a quasimetric exhibits strong horizon generalization. (*Right*) In line with our analysis, the policy that has strong horizon generalization is also more invariant to planning: combining that policy with planning does not increase performance. Figure 7.7 shows a version of this plot that also includes the tabular setting. . . . . 120

7.5 **Measuring horizon generalization in a high-dimensional (27D observation, 8DoF control) task.** (*Left*) We use an enlarged version of the quadruped “ant” environment, training all goal-conditioned RL methods on (start, goal) pairs that are at most 10 meters apart. (*Right*) We evaluate several RL methods, measuring the horizon generalization of each. These results reveal that (*i*) some degree of horizon generalization is possible; (*ii*) the learning algorithm influences the degree of generalization; (*iii*) the value function architecture influences the degree of generalization; and (*iv*) no method achieves perfect generalization, suggesting room for improvement in future work. The ratio of success at 10m vs 5m and 20m vs 10m corresponds to  $\eta$  from Section 7.2. Results are plotted with standard errors across random seeds. . . . . 121

7.6 **Impact of horizon generalization on Bellman errors.** (*Left*) Two goal-reaching methods exhibit different horizon generalization. (*Right*) Despite neither method being trained with the Bellman loss, we observe that the method with stronger horizon generalization has a lower Bellman loss. Thus, understanding horizon generalization may be important even when using TD methods (which guarantee horizon generalization at convergence). . . . . 123

7.7 **Quantifying horizon generalization and invariance to planning.** On a simple navigation task, we collect short trajectories and train two goal-conditioned policies, comparing both to a random policy. (*Top Left*) We evaluate on  $(s, g)$  pairs of varying distances, observing that metric regression with a quasimetric exhibits strong horizon generalization. (*Top Right*) In line with our analysis, the policy that has strong horizon generalization is also more invariant to planning: combining that policy with planning does not increase performance. (*Bottom Row*) We repeat these experiments using function approximation (instead of a tabular model), observing similar trends. . . . . 124

7.8 Quantifying horizon generalization ( $x$ -axis) and planning invariance ( $y$ -axis). See text Section 7.4 for more details. . . . . 125

7.9 **Evidence of Horizon Generalization and Planning Invariance from Prior work.** (*a*) Prior work has observed that if policies are trained in an online setting and perform planning during exploration, then those policies see little benefit from doing planning during evaluation. This observation suggests that these policies may have learned to be planning invariant. While results are not stratified into training and testing tasks, we speculate that the faster learning of that method (relative to baselines, not shown) may be explained by the policy generalizing from easy tasks (which are learned more quickly) to more difficult tasks. (*b*) Prior work studies how data augmentation can facilitate combinatorial generalization. While the notion of combinatorial generalization studied there is slightly from horizon generalization, a method that performs combinatorial generalization would also achieve effective horizon generalization. . . . . 133

8.1 We propose an algorithm training assistive agents to empower human users—the assistant should take actions that enable human users to visit a wide range of future states, and the human’s actions should exert a high degree of influence over the future outcomes. Our algorithm scales to high-dimensional settings, opening the door to building assistive agents that need not directly reason about human intentions. . . . . 137

8.2 **The Information Geometry of Empowerment**, illustrating the analysis in Section 8.1. *(Left)* For a given state  $s_t$  and assistant policy  $\pi_{\mathbf{R}}$ , we plot the distribution over future states for 6 choices of the human policy  $\pi_{\mathbf{H}}$ . In a 3-state MDP, we can represent each policy as a vector lying on the 2-dimensional probability simplex. We refer to the set of all possible state distributions as the *state marginal polytope*. *(Center)* Mutual information corresponds to the distance between the center of the polytope and the vertices that are maximally far away. *(Right)* Empowerment corresponds to maximizing the size of this polytope. For example, when an assistive agent moves an obstacle out of a human user’s way, the human user can spend more time at desired state. . . . . 141

8.3 We evaluate our method with and without conditioning on the robot action  $a^{\mathbf{R}}$ . Conditioning aids learning significantly, which we theorize is because it removes uncertainty in the classification. . . . . 148

8.4 We compare a greedy policy ( $\gamma = 0$ ) against our standard policy ( $\gamma = 0.9$ ). 149

8.5 Visualizing training empowerment in a 5x5 Gridworld with 10 obstacles. Our empowerment objective maximizes the influence of the human’s actions on the future state, preferring the state where the human can reach the goal to the trapped state. This corresponds to maximizing the volume of the state marginal polytope, which is proportional to the number of states that the human can reach from their current position. To visualize the representations, we set the latent dimension to 3 instead of 100. . . . . 150

8.6 We apply our method to the benchmark proposed in prior work [4], visualized in Fig. 8.7a. The four subplots show variant tasks of increasing complexity (more blocks), ( $\pm 1$  SE). We compare against AvE [4], the Goal Inference baseline from [4] which assumes access to a world model, and Reward Inference [5] where we recover the reward from a learned q-value. These prior approaches fail on all except the easiest task, highlighting the importance of scalability. . . . . 153

8.7 (a) The modified environment from Du et al. [4] scaled to  $N = 7$  blocks, and (b, c) the two layouts of the Overcooked environment [6]. . . . . 154

8.8 In Coordination Ring, our ESR agent learns to wait for the human to add an onion to the pot, and then adds one itself. There is another pot at the top which is nearly full, but the empowerment agent takes actions to maximize the impact of the human’s actions, and so follows the lead of the human by filling the empty pot. . . . . 154

9.1 KL divergence between conditional and marginal goal distributions as a function of  $\|\phi(s)\|$  for different representation dimensions  $n$  with  $C = 5$  (up to a constant). The KL divergence is monotonically increasing with  $\|\phi(s)\|$ . . . . . 164

9.2	Mutual information $I(s; g)$ between the state and goal representations in the Euclidean infoNCE setting. Both the representation ratio $C$ and the marginal precision $\tau$ scale monotonically with the mutual information. The mutual information $I(s; g)$ is positive only when $C > 1$ , and so we can additionally interpret this as a constraint on the representation ratio.	167
C.1	Sample rollouts using PALO on unseen testing tasks. . . . .	206
C.2	S-shaped maze. . . . .	211

# LIST OF TABLES

---

2.1	Comparison of Approaches . . . . .	11
2.2	Comparison of Ablations . . . . .	12
2.3	Evaluation Scenes . . . . .	17
3.1	Method Comparisons . . . . .	34
3.2	Ablations . . . . .	35
4.1	Offline RL benchmarks . . . . .	55
5.1	Compositional Generalization Error of Methods . . . . .	77
5.2	Real-world Language Conditioned Evaluation . . . . .	79
5.3	Real-world Goal-Conditioned Evaluation . . . . .	80
7.1	Summary of methods and modifications tested . . . . .	119
8.1	Overcooked Results . . . . .	155
C.1	Task Instructions . . . . .	204
C.2	Task Breakdown . . . . .	206

## 1

INTRODUCTION

---

The last decade of advances in AI has been dominated by approaches to scaling deep learning across large datasets and compute. These advances have led to significant progress in natural language processing and computer vision, but have not yet translated to the same level of success in sequential decision-making settings.

One explanation for this discrepancy is that we have not yet figured out how to leverage the structure of decision-making problems to exploit these advances. In this thesis, we tackle the question: what is the right way to represent the world for sequential decision-making?

In *Part I*, I will discuss work on how we should represent tasks and goals in a way that lets us leverage the large-scale robotics datasets and pretrained models that have emerged in recent years. Then, *Part II* will focus on representations that enable compositional and long-horizon decision-making in more general settings. *Part III* begins to explore how representations can be structured to compute information-theoretic quantities that enable new *intrinsic motivation* capabilities.

## 1.1 PUBLICATIONS

The subsequent chapters are adapted from the following papers. A full list of collaborators can be found in the references within the *bibliography* (§1.2).

**Chapter 2:** Goal Representations for Instruction Following: A Semi-Supervised Language Interface to Control [1]

**Chapter 3:** Policy Adaptation via Language Optimization: Decomposing Tasks for Few-Shot Imitation [2]

**Chapter 4:** Learning Temporal Distances: Contrastive Successor Features Can Provide a Metric Structure for Decision-Making [3]

**Chapter 5:** Successor Representations Enable Emergent Compositional Instruction Following [4]

**Chapter 6:** Inference via Interpolation: Contrastive Representations Provably Enable Planning and Inference [5]

**Chapter 7:** Invariance to Planning in Goal-Conditioned RL [6]

**Chapter 8:** Learning to Assist Humans Without Inferring Rewards [7]

## 1.2 BIBLIOGRAPHY

- [1] Vivek Myers, Andre Wang He, Kuan Fang, Homer Rich Walke, Philippe Hansen-Estruch, Ching-An Cheng, Mihai Jalobeanu, Andrey Kolobov, Anca Dragan, and Sergey Levine. Goal Representations for Instruction Following: A Semi-Supervised Language Interface to Control. In *Conference on Robot Learning*, pp. 3894–3908. 2023.
- [2] Vivek Myers, Bill Chunyuan Zheng, Oier Mees, Sergey Levine, and Kuan Fang. Policy Adaptation via Language Optimization: Decomposing Tasks for Few-Shot Imitation. In *Conference on Robot Learning*. 2024.
- [3] Vivek Myers, Chongyi Zheng, Anca Dragan, Sergey Levine, and Benjamin Eysenbach. Learning Temporal Distances: Contrastive Successor Features Can Provide a Metric Structure for Decision-Making. In *International Conference on Machine Learning*. 2024.
- [4] Vivek Myers, Bill Chunyuan Zheng, Anca Dragan, Kuan Fang, and Sergey Levine. Successor Representations Enable Emergent Compositional Instruction Following. 2024.
- [5] Benjamin Eysenbach, Vivek Myers, Ruslan Salakhutdinov, and Sergey Levine. Inference via Interpolation: Contrastive Representations Provably Enable Planning and Inference. In *Neural Information Processing Systems*. 2024.
- [6] Vivek Myers, Cathy Ji, and Benjamin Eysenbach. Invariance to Planning in Goal-Conditioned RL. 2024.
- [7] Vivek Myers, Evan Ellis, Sergey Levine, Benjamin Eysenbach, and Anca Dragan. Learning to Assist Humans Without Inferring Rewards. In *Neural Information Processing Systems*. 2024.

# I

## REPRESENTING TASKS AND GOALS



# 2 GOAL REPRESENTATIONS FOR INSTRUCTION FOLLOWING

Visual goals (i.e., goal images), though less intuitive for humans, provide complementary benefits for task representation in policy learning. Goals benefit from being easy to ground since, as images, they can be directly compared with other states. More importantly, goal tasks provide additional supervision and enable learning from unstructured data through hindsight relabeling [14–16]. However, compared to language instructions, specifying visual goals is less practical for real-world applications, where users likely prefer to tell the robot what they want rather than having to show it.

Exposing an instruction-following interface for goal-conditioned policies could combine the strengths of both goal- and language- task specification to enable generalist robots that can be easily commanded. While goal-conditioned policy

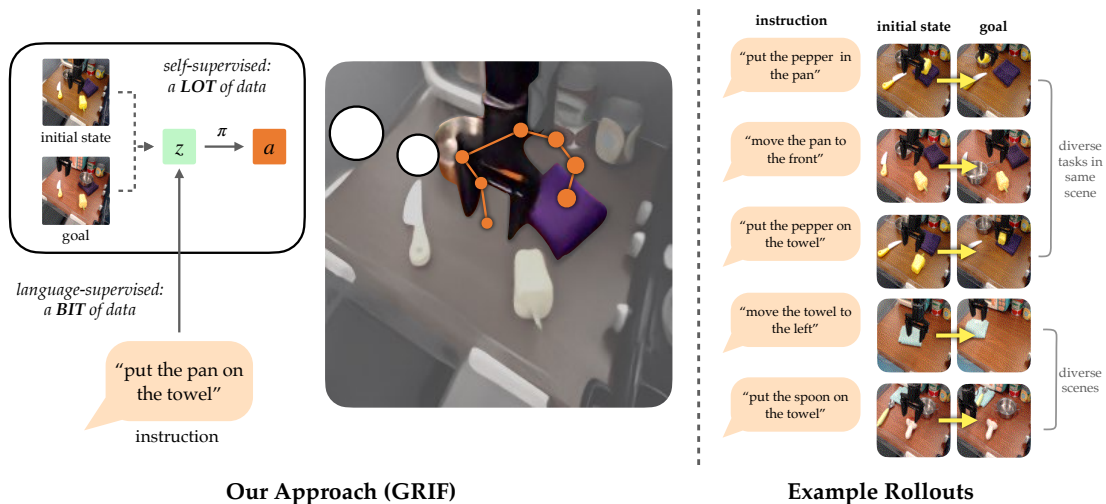


Figure 2.1: **Left:** Our approach learns representations of instructions that are aligned to transitions from the initial state to the goal. When commanded with instructions, the policy  $\pi$  computes the task representation  $z$  from the instruction and predicts the action  $a$  to solve the task. Our approach is trained with a small number of labeled demonstrations and large-scale unlabeled demonstrations. **Right:** Our approach can solve diverse tasks and generalize to vast environment variations.

learning can help digest unstructured data, non-robot vision-language data sources make it possible to connect language and visual goals for generalization to diverse instructions in the real world.

To this end, we propose Goal Representations for Instruction Following (GRIF), an approach that jointly trains a language- and a goal- conditioned policy with aligned task representations. We term task representations *aligned* because our objective encourages learning similar representations for language instructions and state transitions that correspond to the same semantic task. GRIF learns this representation structure explicitly through a contrastive task alignment term. Since task representations across language and image goal modalities have similar semantics, this approach allows us to use robot data collected without annotations to improve performance by the agent on image goal tasks when viewed as a goal-conditioned policy, and thus indirectly improve language-conditioned performance in a semi-supervised manner. An overview of GRIF is shown in Figure 2.1.

We present an approach for learning a language interface for visuomotor control without extensive language labels. With this method, we demonstrate that the semantic knowledge from a pre-trained vision-language model (CLIP [17]) can be used to improved task representations and manipulation even though such models perform poorly at task understanding out-of-the-box. Our experiments show that aligning task representations to scene changes enables improved performance at grounding and following language instructions within diverse real-world scenes.

## 2.1 RELATED WORK

**Robotic control with language interfaces.** Early works in language-conditioned robotic control use hand-designed parse trees or probabilistic graphical models to convert instructions into symbolic states to configure the downstream planners and controllers [18–21]. To generalize beyond limited human specifications, a growing number of works have trained conditional policies end-to-end to follow natural language instructions [22–28, 26, 29–32]. Combining recent advances large language models (LLMs) [33] with learned language-conditioned policies as a low-level API has paved the way for broad downstream applications with improved planning and generalization [34–39]. However, most of these methods need high-capacity policy networks with massive, costly labeled demonstration data. As a result, the learned policies often generalize poorly to unseen scenarios or can only handle limited instructions in real-world scenes. Unlike past work, we learn low-level language-conditioned control from less annotated data.

**Vision-language pre-training.** Vision-language models (VLMs) enable textual descriptions to be associated with visual scenes [17, 40]. Through contrastive learning over internet-scale data, recent large-scale VLMs such as CLIP [17] have achieved unprecedented zero-shot and few-shot generalization capabilities, with a wide range of applications.

Despite these advances, applying pre-trained VLMs to robotic control is not straightforward since control requires grounding instructions in motions instead of static images. Through training from scratch or fine-tuning on human trajectories [41, 42], recent approaches learn representations for visuomotor control [43, 16]. These works use language labels to learn visual representations for control without directly using language as an interface to the policy. In CLIPort, Shridhar et al. [29] use pre-trained CLIP [17] to enable sample-efficient policy learning. Their approach selects actions from high-level skills through imitation, assuming access to predefined pick-and-place motion primitives with known camera parameters. In contrast, our approach learns to align the representation of the instruction and the representation of the transition from the initial state to the goal on labeled robot data, and uses these representations for control without assumptions about the observation and action spaces. Other approaches use VLMs to recover reward signals for reinforcement learning [44–48]. In contrast, our approach directly trains language-conditioned policy through imitation learning without the need for online interactions with the environment.

**Learning language-conditioned tasks by reaching goals.** Alternatively, language-conditioned policies can be constructed or learned through goal-conditioned policies [49, 50]. Lynch and Sermanet [51] propose an approach that facilitates language-conditioned imitation learning by sharing the policy network and aligning the representations of the two conditional tasks. Based on the same motivation, we propose an alternative approach which explicitly extends the alignment of VLMs to specify tasks as changes in the scene. By tuning a contrastive alignment objective, our method is able to exploit the knowledge of VLMs [17] pre-trained on broad data. This explicit alignment improves upon past approaching to connecting images and language [52, 53] by explicitly aligning tasks instead merely jointly training on conditional tasks. In Section 2.4, we show our approach significantly improves the performance of the learned policy and enhances generalization to new instructions.

## 2.2 PROBLEM SETUP

Our objective is to train robots to solve tasks specified by natural language from interactions with the environment. This problem can be formulated as a conditional Markov decision process (MDP) denoted by the tuple  $(\mathcal{S}, \mathcal{A}, \rho, P, \mathcal{W}, \gamma)$ , with state space  $\mathcal{S}$ , action space  $\mathcal{A}$ , initial state probability  $\rho$ , transition probability  $P$ , an instruction space  $\mathcal{W}$ , and discount  $\gamma$ . Given the instruction  $\ell \in \mathcal{W}$ , the robot takes action  $a_t \in \mathcal{A}$  given the state  $s_t$  at each time step  $t$  to achieve success.

To solve such tasks, we train a language-conditioned policy  $\pi(a|s, \ell)$  on a combination of human demonstrations and autonomously collected trajectories. Since high-quality natural language annotations are expensive and time-consuming to obtain, we assume that only a small portion of the trajectories are labeled with the corresponding instructions. The robot has access to a combination of two datasets—

a small-scale labeled dataset  $\mathcal{D}_L$  with annotated instructions and a large-scale unlabeled dataset  $\mathcal{D}_U$  consists of more diverse play data collected in an open-ended manner. Our goal is to train  $\pi(a|s, \ell)$  while taking advantage of both the labeled and unlabeled datasets. We formulate  $\pi(a|s, \ell)$  as a stochastic policy that predicts the Gaussian distribution  $\mathcal{N}(\mu_a, \Sigma_a)$ .

### 2.3 GOAL REPRESENTATIONS FOR INSTRUCTION FOLLOWING

We propose Goal Representations for Instruction Following (GRIF) to interface visuomotor policies with natural language instructions in a semi-supervised fashion (Fig. 2.2). Although the language-conditioned policy cannot be directly trained on the unlabeled dataset  $\mathcal{D}_U$ , we can facilitate the training through goal-conditioned tasks. Solving both types of tasks requires the policy to understand the human intent, ground it in the current observation, and predict necessary actions. Although the first steps involve understanding task specifications of different modalities (goal images and language), the remaining steps of such processes can be shared between the two settings. To this end, we decouple the language-conditioned policy  $\pi(a|s, \ell)$  into a policy network  $\pi_\theta(a|s, z)$  and a language-conditioned task encoder  $f_\phi(\ell)$ , where  $z = f_\phi(\ell)$  is the representation of the task specified by the instruction  $\ell$ . To solve the goal-conditioned task, we also introduce a goal-conditioned task encoder  $h_\psi$ . The policy network  $\pi_\theta$  is shared between the language-conditioned and goal-conditioned tasks.

This approach relies on the alignment of task representations. While most existing VLMs align text with static images, we argue that the representation of the goal-conditioned tasks should be computed from the state-goal pair  $(s_0, g)$ . This is because the instruction often focuses on the changing factors from the initial state to the goal rather than directly describing the entire goal image, e.g., “*move the metal pan to the left*”. Therefore, the representations of goal-conditioned tasks are computed as  $z = h_\psi(s_0, g)$  and we aim to train the encoders such that for  $(s_0, g, \ell)$  sampled from the same trajectory, the distance between  $f_\phi(\ell)$  and  $h_\psi(s_0, g)$  should be close and far apart otherwise. We illustrate our high-level approach in Figure 2.2.

#### Explicit Alignment through Contrastive Learning

We propose explicitly aligning the representations of goal-conditioned and language-conditioned tasks through contrastive learning [52]. Compared to implicitly aligning the task presentations through joint training of the two conditional policies, contrastive alignment requires that all relevant information for selecting actions be included in the shared task representation. This improves the transfer between the action prediction tasks for both goal and language modalities by preventing the policy from relying on features only present in one task modality in selecting actions. Using an InfoNCE objective [53], we train the two encoders  $f_\phi$  and  $h_\psi$  to represent instructions  $\ell$  and transitions  $(s_0, g)$  according to their task semantics.

More concretely, for  $(s_0, g)$  and  $\ell$  that correspond to the same task, we would like their embeddings  $z_\ell = f_\varphi(\ell)$  and  $z_g = h_\psi(s_0, g)$  to be close in the latent space, while  $z_\ell$  and  $z_g$  corresponding to different tasks to be far apart.

To compute the InfoNCE objective, we define  $\mathcal{C}(s, g, \ell) = \cos(f(\ell), h(s, g))$  with the cosine similarity  $\cos(\cdot, \cdot)$ . We sample positive data  $s^+, g^+, \ell^+ \sim \mathcal{D}_L$  by selecting the start state, end state, and language annotation of a random trajectory. We sample negative examples  $s^-, g^- \sim \mathcal{D}_L$  by selecting the start state and end state of a random trajectory, and sample  $\ell^- \sim \mathcal{D}_L$  by selecting the language annotation of another random trajectory. For each positive tuple, we sample  $k$  negative examples and denote them as  $\{s_i^-, g_i^-\}_{i=1}^k$  and  $\{\ell_i^-\}_{j=1}^k$ . Then we can define the InfoNCE  $\mathcal{L}_{\text{task}}$ :

$$\begin{aligned} \mathcal{L}_{\text{lang} \rightarrow \text{goal}} &= -\log \frac{\exp(\mathcal{C}(s^+, g^+, \ell^+)/\tau)}{\exp(\mathcal{C}(s^+, g^+, \ell^+)/\tau) + \sum_{i=1}^k \exp(\mathcal{C}(s_i^-, g_i^-, \ell^+)/\tau)} \\ \mathcal{L}_{\text{goal} \rightarrow \text{lang}} &= -\log \frac{\exp(\mathcal{C}(s^+, g^+, \ell^+)/\tau)}{\exp(\mathcal{C}(s^+, g^+, \ell^+)/\tau) + \sum_{j=1}^k \exp(\mathcal{C}(s^+, g^+, \ell_j^-)/\tau)} \\ \mathcal{L}_{\text{task}} &= \mathcal{L}_{\text{lang} \rightarrow \text{goal}} + \mathcal{L}_{\text{goal} \rightarrow \text{lang}} \end{aligned} \quad (2.1)$$

where  $\tau$  is a temperature hyperparameter.  $\mathcal{L}_{\text{lang} \rightarrow \text{goal}}$  and  $\mathcal{L}_{\text{goal} \rightarrow \text{lang}}$  represent the log classification accuracy of our alignment in predicting goals from language and language from goals respectively.

### Weight Initialization with Vision-Language Models

To handle tasks involving objects and instructions beyond those contained in the limited labeled dataset, we wish to incorporate prior knowledge from broader sources into the encoders  $f_\varphi$  and  $h_\psi$ . For this purpose, we investigate practical ways to incorporate Vision-Language Models (VLMs) [17] pre-trained on massive paired images and texts into our encoders. Pre-trained VLMs demonstrate effective zero-shot and few-shot generalization capability for vision-language tasks [17, 54]. However, they are originally designed for aligning a single static image with its caption without the ability to understand the *changes* in the environment that language tasks correspond to, and perform poorly on compositional generalization [55, 56], which is key to modeling changes in scene state. We wish to encode the change between images while still exploiting prior knowledge in pre-trained VLMs.

To address this issue, we devise a mechanism to accommodate and fine-tune the CLIP [17] model for aligning the transition  $(s_0, g)$  with the instruction  $\ell$ . Specifically, we duplicate and halve the weights of the first layer of the CLIP architecture so it can operate on pairs of stacked images rather than single images. Details on how we modify the pre-trained CLIP to accommodate encoding changes are presented in Section 2.4. In practice, we find this mechanism significantly improves the generalization capability of the learned policy  $\pi_\theta(a|s, g)$ .

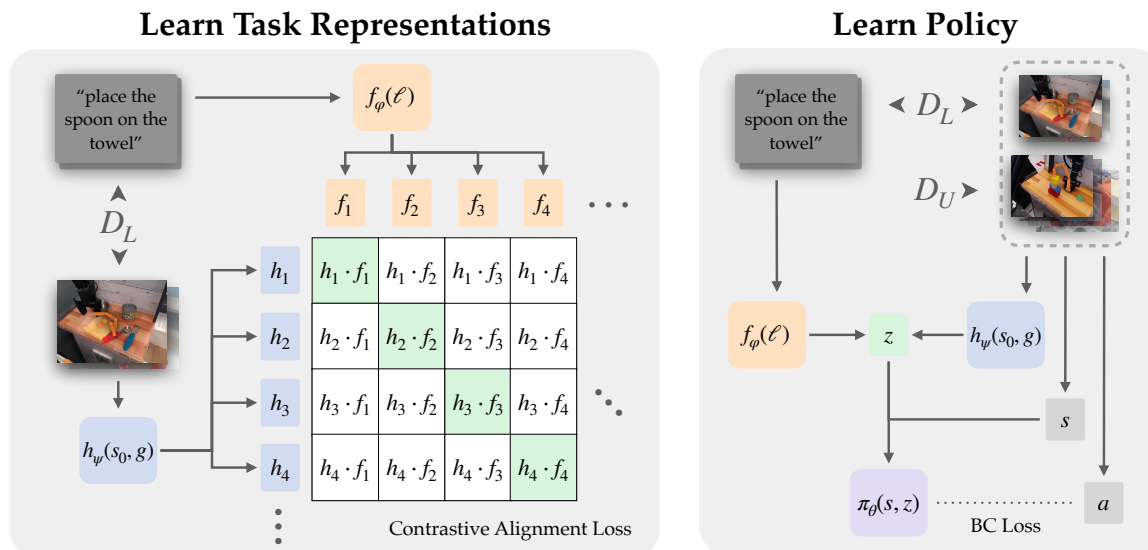


Figure 2.2: **Left:** We explicitly align representations between goal-conditioned and language-conditioned tasks on the labeled dataset  $\mathcal{D}_L$  through contrastive learning. **Right:** Given the pre-trained task representations, we train a policy on both labeled and unlabeled datasets.

### Policy Learning with Aligned Representations

We train the policy jointly on the two datasets  $\mathcal{D}_L$  and  $\mathcal{D}_U$  with the aligned task representations. By sampling  $(\ell, s_t, a_t)$  from  $\mathcal{D}_L$ , we train the policy network  $\pi_\theta(a|z)$  to solve language-conditioned tasks with  $z = f_\varphi(\ell)$ . And by sampling  $(s_0, g, s_t, a_t)$  from  $\mathcal{D}_L \cup \mathcal{D}_U$ ,  $\pi_\theta$  is trained to reach goals with  $z = h_\psi(s_0, g)$ . We train with behavioral cloning to maximize the likelihood of the actions  $a_t$ .

We investigate two ways to train the policy given the encoders  $f_\varphi$  and  $h_\psi$ . The straightforward way is to jointly train the policy network  $\pi_\theta$  and the two encoders end-to-end. This process adapts the encoders with the policy network to encourage them to incorporate information that facilitates downstream robotic control, but can also backfire if the policy learns to rely on visual-only features that are absent in the language conditioned setting. Alternatively, we can freeze the pre-trained weights of the two encoders and only train the shared policy network  $\pi_\theta$  on the two datasets. In Section 2.4, we evaluate and discuss the performances of both options.

## 2.4 EXPERIMENTS

Our work started with the premise of tapping into large, goal-conditioned datasets. To build a language interface for goal-conditioned policy learning, we proposed to learn explicitly aligned task representations, and to align instructions to state changes rather than static goals. Lastly, we advocated for the use of pre-trained



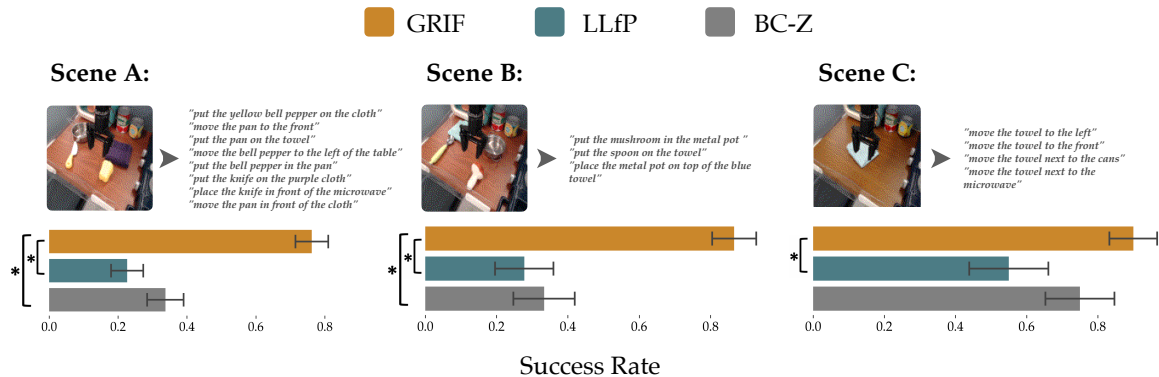


Figure 2.3: Comparison of success rates  $\pm$ SE between the top three methods across all trials within the three scenes. Two other baselines LCBC and R3M (not shown) achieved 0.0 success in all evaluation tasks although they do succeed on tasks that are heavily covered in the training data. Statistical significance is starred. The initial observation and instructions of each scene are shown.

VLMs to incorporate larger sources of vision-language knowledge. Therefore, we aim to test the following hypotheses in our experiments:

- H1:** Unlabeled trajectories will benefit the language-conditioned policy on new instructions.
- H2:** Explicitly aligning task representations improves upon the implicit alignment from LangLfP-style joint training [51].
- H3:** The prior knowledge in pre-trained VLMs can improve learned task representations.
- H4:** Aligning *transitions* with language enable better use of VLMs compared to conventional image-language contrastive methods [47, 57].

Our experiments are conducted in an table-top manipulation domain. For training, we use a labeled dataset  $\mathcal{D}_L$  containing 7k trajectories and an unlabeled  $\mathcal{D}_U$  containing 47k trajectories. Our approach learns to imitate the 6 DOF continuous gripper control actions in the data at 5Hz. The evaluation scenes and unseen instructions are shown in Fig. 2.3. Additional details about the environment, the dataset, and the breakdown of results are described in Section 2.4.

## Comparative Results

We compare the proposed GRIF with four baseline methods on a set of 15 unseen instructions from all 3 scenes and report the aggregated results in Figure 2.3, with GRIF attaining the best performance across all scenes. The per-task success rates can be found in Section 2.4. **LCBC** [22] uses a behavioral cloning objective to train a

Table 2.1: Comparison of Approaches

		Success Rate				
Scene	Task	GRIF	LCBC	LLfP	R3M	BC-Z
A	· put the yellow bell pepper on the cloth	<b>0.6</b>	0.0	0.0	0.0	<b>0.6</b>
	· move the pan to the front	<b>1.0</b>	0.0	0.6	0.0	0.0
	· put the pan on the towel	0.8	0.0	0.3	0.0	<b>0.9</b>
	· move the bell pepper to the left of the table	0.7	0.0	0.0	0.0	<b>0.8</b>
	· put the bell pepper in the pan	<b>0.8</b>	0.0	0.1	0.0	0.3
	· put the knife on the purple cloth	<b>0.7</b>	0.0	0.2	0.0	0.0
	· place the knife in front of the microwave	<b>0.7</b>	0.0	0.0	0.0	0.1
B	· move the pan in front of the cloth	<b>0.6</b>	0.0	0.3	0.0	0.0
	· put the mushroom in the metal pot	<b>0.9</b>	0.0	0.5	0.0	0.4
	· put the spoon on the towel	<b>0.9</b>	0.0	0.3	0.0	0.4
C	· place the metal pot on top of the blue towel	<b>0.8</b>	0.0	0.0	0.0	0.2
	· move the towel to the left	<b>1.0</b>	0.0	<b>1.0</b>	0.0	<b>1.0</b>
	· move the towel to the front	<b>1.0</b>	0.0	<b>1.0</b>	0.0	<b>1.0</b>
	· move the towel next to the cans	<b>0.6</b>	0.0	0.0	0.0	0.2
	· move the towel next to the microwave	<b>1.0</b>	0.0	0.2	0.0	0.8

policy conditioned on language from  $\mathcal{D}_L$ , similar to prior methods on instruction-conditioned imitation learning. **LLfP** [51] jointly trains a goal conditioned and language conditioned policy on partially labeled data, but does not learn aligned task representations. **R3M** [16] provides pre-trained state representations for robot manipulation that are predictive of language-conditioned rewards. We use this approach as a baseline by jointly training goal- and language-conditioned policies while using R3M state encodings as goal representations (i.e.,  $h_\psi(s_0, g) = \text{R3M}(g)$ ). **BC-Z** [23] jointly trains language- and video-conditioned policies and uses an additional cosine similarity term to align video and language embeddings. This approach does not transfer directly into our goal-conditioned setting, but we create a baseline that adapts it to our setting by jointly training goal- and language-conditioned policies while aligning task representations with a cosine distance loss. The architecture choices are standardized across all methods for fair comparisons. Unless stated otherwise, all baselines use a ResNet-18 as the goal encoder  $h_\psi(s_0, g)$ . In our preliminary experiments, this architecture was found to give good performance when used to train goal-conditioned policies in our setting. For the language encoder  $f_\phi(\ell)$ , all baselines use a pre-trained and frozen MUSE model [58], as in previous work [51, 23].

We find that language-conditioned policies must make use of unlabeled trajectories to achieve non-zero success rates when generalizing to new language



Table 2.2: Comparison of Ablations

		Success Rate			
Scene	Task	GRIF (Frozen)	GRIF (Joint)	GRIF (Labeled)	
A	put the yellow bell pepper on the cloth	0.6	0.8	1.0	
	move the pan to the front	1.0	1.0	0.7	
	put the pan on the towel	0.8	1.0	0.1	
	move the bell pepper to the left of the table	0.7	0.4	0.2	
	put the bell pepper in the pan	0.8	0.6	1.0	
	put the knife on the purple cloth	0.7	0.4	0.1	
	place the knife in front of the microwave	0.7	0.6	0.5	
	move the pan in front of the cloth	0.6	0.9	0.2	
			No Start	No Align	No CLIP
	A	put the yellow bell pepper on the cloth	0.3	0.5	0.0
move the pan to the front		0.6	0.8	0.0	
put the pan on the towel		0.6	0.6	0.0	
move the bell pepper to the left of the table		0.4	0.6	0.2	
put the bell pepper in the pan		0.7	0.6	0.1	
put the knife on the purple cloth		0.2	0.2	0.0	
place the knife in front of the microwave		0.1	0.0	0.0	
move the pan in front of the cloth		0.4	0.0	0.3	

instructions in support of **H1**. LCBC does not use unlabeled data and fails to complete any tasks. R3M jointly trains goal- and language-conditioned policies, but it also fails all tasks. This is likely due to its goal encodings being frozen and unable to be implicitly aligned to language during joint training. Methods that use implicit or explicit alignment (GRIF, LLfP, BC-Z), are able to exploit unlabeled goal data to follow instructions to varying degrees of success. These comparisons suggest that the combined effect of using pre-trained CLIP to align transitions with language significantly improves language-conditioned capabilities. Our model significantly outperformed all baselines on 8 out of 15 tasks, achieving high success rates on several tasks where the baselines almost completely fail (“*place the knife in front of the microwave*”, “*move the pan in front of the cloth*”, “*put the knife on the purple cloth*”), while

achieving similar performance to the next-best baseline on the remaining tasks. Where baselines failed, we often observed *grounding* failures. The robot reached for incorrect objects, placed them in incorrect locations, or was easily distracted by nearby objects into performing a different task.

### Ablation Study

We run a series of ablations to analyze the performance of GRIF and test the hypotheses. **No Align** ablates the effect of explicit alignment by removing the contrastive objective. We also unfreeze the task encoders so that they are implicitly aligned via joint training of the language- and goal-conditioned policies. **No CLIP** ablates the effect of using pre-trained CLIP by replacing the

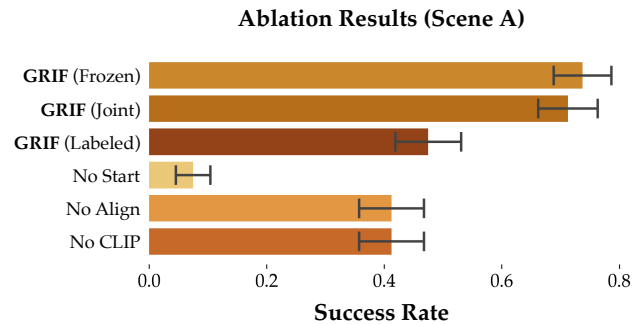


Figure 2.4: Success rates of ablations with one standard error.

image and text encoders with a ResNet-18 and pre-trained MUSE language encoder. In **No Start**, the image task representations only depend on goals as  $h_\psi(g)$ , instead of depending on transitions as  $h_\psi(s_0, g)$ . This is the conventional way to connect goals and language with CLIP that is often used in previous work [57, 47]. For **GRIF (Labeled)**, we exclude  $\mathcal{D}_U$  to study whether using unlabeled data is important for performance. **GRIF (Joint)** trains the task alignment and policy losses jointly, taking gradients through the image encoder and freezing the language encoder. This is the end-to-end approach discussed in Section 2.3. We refer to our full approach without joint training as **GRIF (Frozen)** in the remainder of this section.

As shown in Figure 2.4, explicit alignment, pre-trained CLIP, and transition-based task representations all play critical roles in achieving high success rates. Notably, the conventional approach of aligning static goals and instructions with CLIP (**No Start**) fails almost completely in our setting. This is in support of **H4** and confirms that transitions, and not goal images, should be aligned to language tasks. In **GRIF (Labeled)**, dropping  $\mathcal{D}_U$  significantly decreases success rates, further supporting **H1**. We observe that this is primarily due to a deterioration of manipulation skills rather than grounding, which is expected as grounding is mostly learned via explicit alignment on  $\mathcal{D}_L$ . Regarding **H2** and **H3**, we observe that removing either alignment or CLIP results in a large drop in performance. We also observed that **No Align** outperforms its counterpart *LLfP* by using the pre-trained CLIP model (after the modification in Section 2.3) in the task encoder. We hypothesize that this is because CLIP has already been explicitly aligned during pre-training, and some

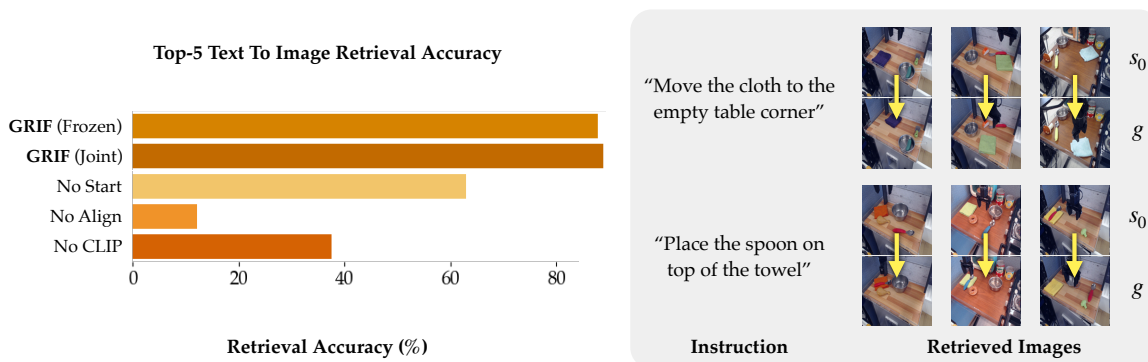


Figure 2.5: **Left:** Comparison of the top-5 text to image retrieval accuracy of representations learned by different ablations. **Right:** Examples of retrieved image pairs given instructions.

of its knowledge is retained during joint training with the policy even without GRIF’s task alignment loss. Lastly, deciding to freeze the task encoders during policy training does not appear to significantly affect our model’s performance. This is likely because the contrastive learning phase already learns representations that can represent the semantic task, so there is less to gain from further implicit alignment during joint training.

### Analysis on the Learned Task Representations

For additional analysis, we evaluate our model’s task grounding capabilities independently of the manipulation policy and compare it with ablations. Specifically, we evaluate how well our model can connect new language instructions to correct goals in a scene. This is important to downstream policy success: if the model is able to project the language to a representation  $f_\phi(l)$  that is close to that of the correct (but unprovided) goal  $h_\psi(s_0, g)$ , then the policy will likely be able to execute the task since it has been trained on a large amount of goal-conditioned data.

Our task representations are trained with a contrastive objective, offering a convenient way to compute alignment between language and goals. On an dataset of labeled held-out trajectories, we compute the similarities between all pairs of visual task representations  $h_\psi(s_0, g)$  and language task representations  $f_\phi(l)$ . For each language instruction, we retrieve the top  $k$  most similar  $(s_0, g)$  transitions and compute the accuracy for the correct transition being retrieved. We compute this metric in fixed batches of 256 examples and average over the validation set to report a text-to-image retrieval accuracy. We compute this metric for representations from each of our ablations and report the results in Figure 2.5 to analyze why GRIF outperforms other approaches in our main experiments. Our task representations show significantly better generalization compared to using a conventional image-language alignment (**No Start**), despite it being CLIP’s original pre-training

objective. The alignment accuracy is also more than 50% higher than when using non-VLM encoders (**No CLIP**), suggesting potentially large gains in grounding capability through using VLMs.

We also study the effect of the number of language annotations on our model’s grounding capability. Even at less than half the number of language annotations (3k), GRIF outperforms all the ablations in Figure 2.5, achieving a retrieval accuracy of 73%. Detailed results for this ablation are presented in Section 2.5, showing our approach is robust to lower amounts of language supervision.

## Environment Details

We provide more details on the real-world environment in this section.

### Robot

We use a 6DOF WidowX 250 robot with a 1DOF parallel-jaw gripper. We install the robot on a tabletop where it can reach and manipulate objects within an environment set up in front of it. The robot receives inputs from a Logitech C920 RGB camera installed in an over-the-shoulder view. The images are passed into the policy at a  $128 \times 128$ , and the control frequency is 5Hz. Teleoperation data is collected with a Meta Quest 2 VR headset that controls the robot.

### Dataset Details

The dataset consists of trajectories collected from 24 different environments, which includes kitchen-, sink-, and tabletop-themed manipulation environments. The dataset features around 100 objects, including containers, utensils, toy food items, towels, and other kitchen-themed objects. It includes demonstrations of 13 high-level skills (pick and place, sweep, etc.) applied to different objects. Out of the 54k total trajectories, 7k are annotated with language instructions. Around 44k of the trajectories are expert demonstrations and around 10k are collected by a scripted policy.

### Method Details

#### Policy Network

Our policy network  $\pi_\theta(a|s, z)$  uses a ResNet-34 architecture. To condition on the task embedding  $z$ , it is first passed through 2 fully connected layers. Then, the policy network is conditioned on the embedding using FiLM layers, which are applied at the end of every block throughout the ResNet. The image encoding is then passed into a fully connected network to predict the action distribution. The policy network predicts the action mean, and we use a fixed standard deviation.

### CLIP Model Surgery

Instead of separately encoding  $s_0$  and  $g$  inside  $f_\phi$ , we perform a “surgery” to the CLIP model to enable it to take  $(s_0, g)$  as inputs while keeping most of its pre-trained network weights as intact as possible. Specifically, we clone the weight matrix  $W_{\text{in}}$  of the first layer in the pre-trained CLIP and concatenate them along the channel dimension to be  $[W_{\text{in}}; W_{\text{in}}]$ , creating a model that can accept the stacked  $[s_0, g]$  as inputs. We also halve the values of this new weight matrix to make it  $W'_{\text{in}} = [W_{\text{in}}/2; W_{\text{in}}/2]$ , ensuring its output  $0.5(W_{\text{in}}s_0 + W_{\text{in}}g)$  will follow a distribution similar to the output by the original first layer  $W_{\text{in}}s_0$ . While this surgery alone cannot perfectly close the gap, the resultant modified encoder can serve as a capable initialization for the transition encoder  $h_\psi$ . We further fine-tune  $h_\psi$  on the labeled robot dataset  $\mathcal{D}_L$  using the aforementioned method to adapt it for instruction-following tasks.

### Negative Sampling

For training the contrastive objective on  $\mathcal{D}_L$ , our batch sampling strategy is non-standard. We use 2 dataloaders in parallel; the first samples from shuffled trajectories, while the second iterates through trajectories in the order that they are stored in the dataset. Each samples batches of 128 trajectories and they are concatenated to produce a batch size of 256. The reason for this is that if we were to use a standard sampling strategy, most examples in a batch would be from different scenes. This is not useful for the contrastive loss because the representations would just learn to distinguish tasks based on the set of objects that appear. The robot benefits from being able to distinguish different tasks in the same scene, so we try to include many trajectories from the same scene in each batch. Using an unshuffled dataloader is a convenient way to achieve this since trajectories from the same scene are stored together. This can be considered a form of negative mining for the contrastive learning stage.

### Goal Relabeling

For unlabeled trajectories in  $\mathcal{D}_U$ , we use a simple goal relabeling strategy: with 50% probability, we use the final achieved state as the goal, and with 50 % probability we uniformly sample an intermediate state in the trajectory to use as the goal. We do not relabel the annotated trajectories in  $\mathcal{D}_L$ .

### Hyperparameters

When training the task encoders using the contrastive learning objective, we use a batch size of 256. We reduce the batch size to 128 when we train the policy network. We use the Adam optimizer with a learning rate schedule that uses linear warmup and cosine decay. The peak learning rate is  $3e-4$  for all parameters except the CLIP ViT encoders, for which we use  $3e-5$ . We use 2000 warmup steps and  $2e6$  decay steps

Table 2.3: Evaluation Scenes

Scene	Objects
A	knife, pepper, towel, & pot
B	mushroom, towel, spoon, & pot
C	towel

for the learning rate schedule. When we jointly train the alignment and behavioral cloning losses, we use a weight of 1.0 on both terms. These hyperparameters were found through random search. We train our models for 150k steps, which takes around 13 hours on 2 Google Cloud TPU cores.

### Experimental Details

The scenes were constructed with the objects shown in Table 2.3 within a toy kitchen setup.

During evaluation, we roll out the policy given the instruction for 60 steps. Task success determined by a human according to the following criteria:

- Tasks that involve putting an object into or on top of a container (e.g. pot, pan, towel) are judged successes if any part of the object lies within or on top of the container.
- Tasks that involve moving an object toward a certain direction are judged successes if the object is moved sufficiently in the correct direction to be visually noticeable.
- Tasks that involve moving an object to a location relative to another object are judged successes if the object ends in the correct quadrant and are aligned with the reference object as instructed. For example, in “place the knife in front of the microwave,” the knife should be placed in the top-left quadrant, and be overlapping with the microwave in the horizontal axis.
- If the robot attempts to grasp any object other than the one instructed, and this results in a movement of the object, then the episode is judged a failure.

### Experimental Results

We show per-task success rates for our approaches, the baselines, and the ablations in this section. The tasks in scenes A and B were evaluated for 10 trials each, while those in C were evaluated for 5 trials.

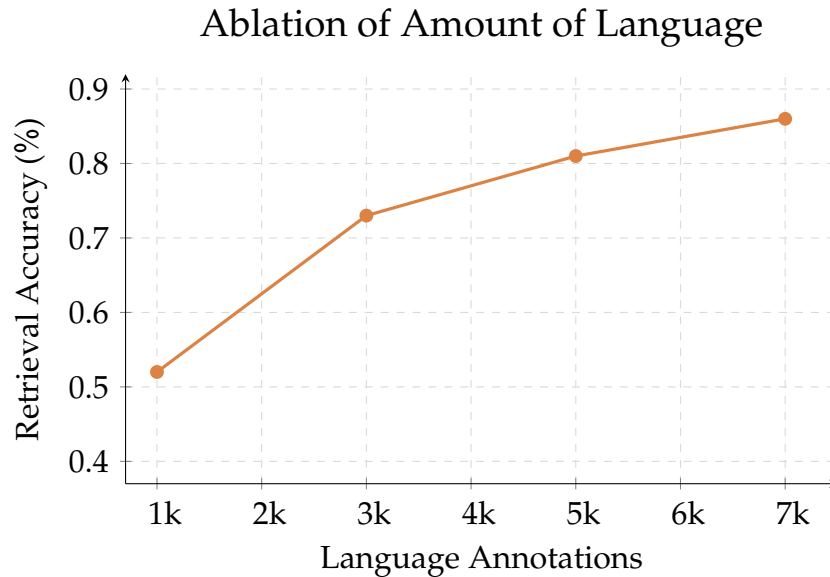


Figure 2.6: Scaling of GRIF grounding capability by number of language annotations available.

## 2.5 SCALING OF ANNOTATION SUPERVISION

We ablate effect of the amount of language supervision on GRIF’s grounding capabilities. We compute the (top-5) text-to-image retrieval accuracy of GRIF representations when trained on 7k, 5k, 3k, and 1k annotations, and find accuracies of 86%, 81%, 73%, and 52% respectively. These accuracies are plotted in Figure 2.6. By comparing these accuracies with the grounding performance of the ablations in Figure 2.5, we see GRIF enables more robust grounding with little language supervision.

## 2.6 CONCLUSION

Our approach to aligning image goals and language instructions enables a robot to utilize large amounts of unlabeled trajectory data to learn goal-conditioned policies, while providing a “language interface” to these policies via aligned language-goal representations. In contrast to prior language-image alignment methods, our representations align *changes* in state to language, which we show leads to significantly better performance than more commonly used CLIP-style image-language alignment objectives. Our experiments demonstrate that our approach can effectively leverage unlabeled robotic trajectories, with large improvements in performance over baselines and methods that only use the language-annotated data.

**Limitations and future work.** Our method has a number of limitations that could

be addressed in future. For instance, our method is not well-suited for tasks where instructions say more about *how* to do the task rather than *what* to do (e.g., “*pour the water slowly*”)—such qualitative instructions might require other types of alignment losses that more effectively consider the intermediate steps of task execution. Our approach also assumes that all language grounding comes from the portion of our dataset that is fully annotated or a pre-trained VLM. An exciting direction for future work would be to extend our alignment loss to utilize non-robot vision-language data, such as videos, to learn rich semantics from Internet-scale data. Such an approach could then use this data to improve grounding on language not in the robot dataset and enable broadly generalizable and powerful robotic policies that can follow user instructions.



# 3 POLICY ADAPTATION VIA LANGUAGE OPTIMIZATION

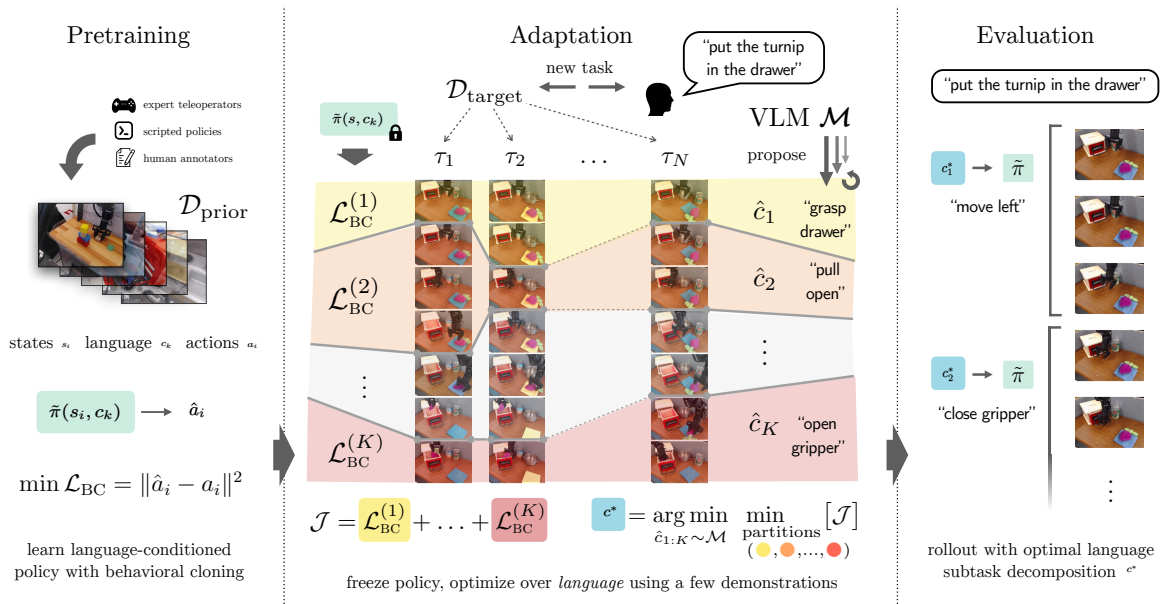


Figure 3.1: An overview of the PALO algorithm for few-shot adaptation with language. (Left) We build off a pre-trained policy that has learned to follow low-level language instructions from a large dataset of expert demonstrations. (Middle) Given a new task and a few expert demonstrations, we use a VLM to propose candidate decompositions into subtasks. We optimize over these decompositions jointly with the partitions of trajectories into subtasks, selecting the subtask decomposition that minimizes the validation error of the learned policy. (Right) At test time, we condition the pre-trained policy on the selected decomposition to solve the task.

Robot learning promises policies that can adapt and generalize to new behaviors. However, in practice, today’s robotic policies often struggle to effectively finetune for truly new tasks [59, 51, 60–62]. For example, consider the task of making a salad: while a person could likely follow a new recipe with only a few examples by remembering the key steps, a robot learning approach may need many more

demonstrations to achieve similar performance, and still recover a more brittle policy.

A key difference that allows humans to learn tasks so quickly is their semantic understanding of the world. Humans have a symbolic representation of the task, such as the names of the ingredients and the steps to prepare them, rather than a series of low-level actions. This representation enables them to understand the task at a higher level, mapping directly into low-level behaviors they are already familiar with [63, 64]. How can we enable robots to quickly learn new tasks through a semantic understanding of the world?

Language provides a potential bridge between these task semantics and low-level control [65]. Recent advances in large language models (LLMs) and vision-language models (VLMs) have shown promise in understanding and grounding language from a few demonstrations [33, 66]. We propose Policy Adaptation via Language Optimization (PALO), a method for exploiting the semantic understanding of VLMs in combination with a pre-trained robot policy to enable adaptation to new tasks with only a few demonstrations (Fig. 3.1).

Past approaches that fine-tune directly to new demonstrations are often overparameterized and sample-inefficient, due to the cost inherent in collecting teleoperated trajectories [67]. Instead, we use a few demonstrations as a calibration set to guide the decomposition of a new language task into a sequence of subtasks that can be used by a language-conditioned policy. Our approach samples possible decompositions of the task from a VLM and chooses the one that minimizes the validation error of the learned policy on the calibration set.

The key is that in the few-shot setting, a few demonstrations provide a better signal for adapting to new tasks when used to select the right sequence of language subtasks with the help of a VLM, rather than directly fine-tuning the policy parameters (Fig. 3.2). Unlike prior work, our approach can learn unseen, long-

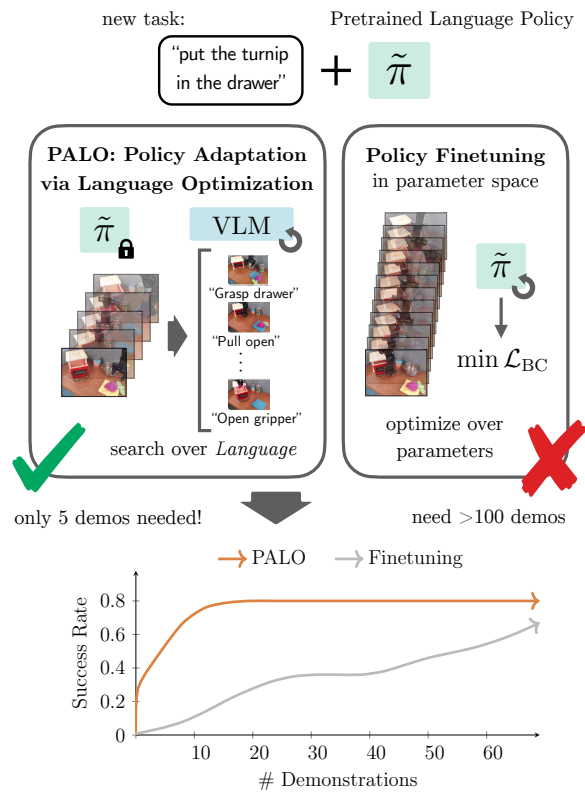


Figure 3.2: PALO enables pre-trained generalist policies to adapt new tasks with as few as five demonstrations by searching in language space instead of parameter space.

horizon behaviors with fewer than 10 demonstrations across a variety of tabletop manipulation tasks.

### 3.1 RELATED WORK

Our approach lies at the intersection of few-shot learning and approaches that leverage language and large pre-trained models for robotics.

**Few-shot learning.** Broadly speaking, few-shot learning approaches utilize diverse data to enable rapid test-time adaptation to a new task from a few examples. These techniques have been applied in various domains, including vision [68–70], natural language processing [33, 71], and reinforcement learning [72, 73]. Frameworks for few-shot learning include optimization-based meta-learning [74–76], where a model is trained to quickly fine-tune to new tasks, nonparametric methods based on particular modeling assumptions such as metric approaches and Gaussian processes [77–80], and in-context learning [33, 81–83], where a large model is conditioned on a context to adapt to a new task. Unlike past approaches to few-shot learning in robotics [84, 67], we show that language can be used to enable nonparametric adaptation without fine-tuning.

**Language-conditioned robotic control.** While early approaches to instruction-following in robotics relied on manually designed symbolic representations [18, 20, 21], recent work has focused on applying deep learning techniques to understand natural language instructions [85, 29]. These approaches use learned behavioral cloning policies on top of language [51, 86, 39], connect language representations to grounded representations of the environment [7, 87–89], or use the compositional structure of language to decompose tasks and plan [90, 35, 91–93]. Our approach is the first to enable few-shot adaptation to new demonstrations in robotics by leveraging the structure of language.

**Foundation models and robotics.** Large-scale internet pre-training has seen recent success in the domains of vision and natural language processing [33, 17, 94, 95, 33, 96]. Recent work has investigated if these models can be trained and/or fine-tuned for downstream robotics tasks [85, 26, 97, 67, 98, 99]. Other work has investigated if these models can be used to provide semantic knowledge for downstream robot learning pipelines [100–106]. Our approach falls into this latter category, but unlike the past works, we perform *few-shot adaptation* in language-conditioned robot control using the semantic knowledge in large pre-trained VLMs.

### 3.2 POLICY ADAPTATION VIA LANGUAGE OPTIMIZATION

Our goal is to enable a learned language-conditioned robot policy to perform new tasks with only a few demonstrations. The key insight is that the structure of language can be exploited to enable few-shot adaptation to new demonstrations in

robotics. Fundamentally, few-shot adaptation to new tasks depends on a policy’s ability to generalize its existing knowledge to correctly fit to new demonstrations. One approach for adapting a learned policy is to directly fine-tune to new demonstrations, but in robotics settings where expert data collection is costly, this is often infeasible due to overfitting.

We propose Policy Adaptation via Language Optimization (PALO), which instead uses demonstrations of a task that is outside the training distribution with the reasoning capabilities of a pre-trained vision-language model (VLM) to determine the correct sequence of decomposed subtasks that are in-distribution for the robot policy. Given a language instruction  $\ell$ , we compute a task decomposition  $c_{1:K}$  that is both semantically consistent with the instruction (determined by the VLM) and feasible in the environment (measured by policy validation loss on expert demonstrations).

### Notation

Formally, we assume a contextual Markov Decision Process (MDP) structure. We have a state space  $\mathcal{S}$ , continuous action space  $\mathcal{A} = (0, 1)^{d_A}$ , initial state distribution  $p_0$ , transition probabilities  $P$ , and free-form language instruction  $\ell \in \mathcal{L}$  chosen from the language instruction space  $\mathcal{L}$ . We use the notation  $\mathcal{P}(X)$  to denote the set of probability distributions over a space  $X$ .

The robot selects the action  $a_t \in \mathcal{A}$  based on the observed state  $s_t \in \mathcal{S}$  at each time step  $t \in \{1 \dots H\}$  over a finite horizon  $H$  to achieve states in  $\mathcal{S}_\ell$ . We denote a robot policy as a map  $\pi(\hat{a}_t | s_t, \ell)$ , which maps the state  $s_t$  and instruction  $\ell$  to a distribution over actions  $\hat{a}_t$ . For convenience, we assume actions are selected under a fixed isotropic Gaussian noise model unless otherwise specified, and will denote the mode of the distribution  $\pi(\hat{a} | s_t, \bullet)$  as  $\pi(s_t, \bullet)$ . A robot policy then yields a distribution over trajectories  $(\{(s_i, a_i)\}_{i=1}^H, \ell) \sim \mathcal{T}_\pi^\rho$  given a task distribution  $\rho \in \mathcal{P}(\mathcal{L})$ .

### Problem Statement

We want to solve out-of-distribution instruction-following tasks involving unseen objects and skills given only a few demonstrations. For (pre-)training the instruction-following policy we assume access to a dataset that has been generated using language tasks sampled from some distribution  $\rho_{\text{prior}} \in \mathcal{P}(\mathcal{L})$  with an expert policy  $\pi_\beta(\hat{a} | s, \ell)$ . For training an instruction-following policy  $\hat{\pi}(s, \ell)$ , we assume a prior dataset  $\mathcal{D}_{\text{prior}} = \{(\tau^{(i)}, c_{1:K}^{(i)}, \ell^{(i)})\}_{i=1}^{N_{\text{prior}}}$  for  $\tau^{(i)}, \ell \sim \mathcal{T}_{\pi_\beta}^{\rho_{\text{prior}}}$  and additional hierarchically-decomposed subtask instructions  $c_{1:K} \in \mathcal{L}^K$  that are distributed according to  $p(c_{1:K} | s_0, \ell)$  for decomposition size  $K < H$ .

A target task is sampled from a separate distribution  $\rho_{\text{target}} \in \mathcal{P}(\mathcal{L})$  which requires interacting with unseen objects in novel ways, so the policy trained on  $\mathcal{D}_{\text{prior}}$  performs poorly zero-shot. To solve this new task, we assume there exists

an additional dataset  $\mathcal{D}_{\text{target}} = (\{\tau_1 \dots \tau_n\}, \ell)$  for  $\tau_i \sim \mathcal{T}_{\pi_\beta}^{\delta_\ell} | s_0$  with  $s_0 \sim p_0$  and  $\ell \sim \rho_{\text{target}}$  collected by human experts. While a large  $\mathcal{D}_{\text{target}}$  can enable directly training  $\pi(s, \ell)$  to solve the target task, we are interested in challenging few-shot scenarios in which  $\mathcal{D}_{\text{target}}$  only contains a handful of demonstrations (e.g., 5). In this paper, we tackle this challenge by decomposing the novel target task into a sequence of subtasks that are solvable by the pre-trained  $\tilde{\pi}(s, c)$  using a VLM  $\mathcal{M}$ . Notably, we do not assume any ground truth labels for the task decomposition are given, and aim to generate the optimal language decomposition  $c_{1:K}$  based on the unlabeled demonstration dataset  $\mathcal{D}_{\text{target}}$  collected by human operators.

Our approach makes two assumptions about the structure of the target task.

**Assumption 3.1.** *The target task subtask annotations  $c_i$  locally match those of the prior dataset, i.e., are distributed identically for  $i \sim \text{Unif}(1 \dots H)$*

$$\mathbb{E}_{\ell \sim \rho_{\text{target}}, s_0 \sim p_0} p(c_i | s_0, \ell) \approx \mathbb{E}_{\ell \sim \rho_{\text{prior}}, s_0 \sim p_0} p(c_i | s_0, \ell). \quad (3.1)$$

Assumption 3.1 states that even if the overall target tasks in  $\rho_{\text{target}}$  are unseen, the low-level manipulation skills (e.g., “close the gripper,” “move the arm right”) will be represented in the policy training.

**Assumption 3.2.** *The VLM  $\mathcal{M}$  can approximate the distribution of the subtask annotations  $c_{1:K}$  in the target task, i.e.,*

$$p_{\mathcal{M}}(c_{1:K} | s_0, \ell) \approx p(c_{1:K} | s_0, \ell). \quad (3.2)$$

Assumption 3.2 states that the VLM can propose candidate task decompositions that are consistent with the instruction  $\ell$  in new scenes. Qualitatively, these assumptions are consistent with recent advances in robot manipulation training data [62, 98] and embodied reasoning with VLMs [107] and are empirically validated in our experiments in Section 3.4 using the BridgeDataV2 dataset [62] and GPT-4o [94] with prompting described in Appendix B.2.

In Section 3.3 we show that under these assumptions, our PALO algorithm can achieve low regret on out-of-distribution tasks, and discuss how violating these assumptions affects performance.

### Task Decomposition with Language

To guide the pre-trained policy  $\hat{\pi}$  to solve the unseen target task, we decompose the high-level language instruction  $\ell$  of the target task into a sequence of subtask instructions  $c_{1:K} = (c_1, \dots, c_K)$  for the  $K$  subtasks as a set of language decomposition. Instead of commanding  $\hat{\pi}$  with the original instruction  $\ell$ , we use a combination of  $\ell$  and the subtask instructions  $c_k$  as the input in each subtask to produce the action as  $a_t \leftarrow \tilde{\pi}(s_t, c_k)$ . In our methods, we used GPT-4o [94] as a backbone to generate

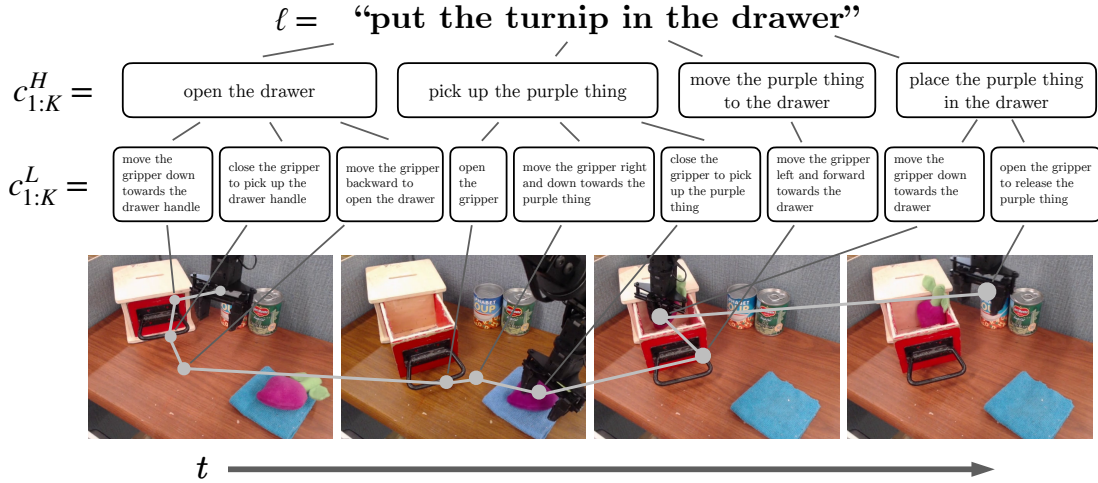


Figure 3.3: A visualization of an example execution of our method on the long-horizon task “put the beet toy in the drawer.” The VLM deconstructs  $\ell$  into candidate high-level subtasks  $c_{1:K}^H$  and low-level subtasks  $c_{1:K}^L$  and optimizes over the expert trajectories. The optimal  $c_{1:K}^H$  and  $c_{1:K}^L$  are chosen and unrolled in real-world evaluations, which result in successful completion of the task (trajectory shown in gray).

instruction sets. We denote by  $\mathcal{M}(s_0, \ell)$  the support of possible task decompositions sampled from this VLM (see details in Appendix B.2).

Aside from the sequential order of the subtasks, the robot needs to decide when to switch to the next subtask. For this purpose, we introduce an additional variable  $u = (u_1, \dots, u_K) \sim \text{Unif}(\mathcal{U})$  where  $\mathcal{U}$  is the space of ordered partitions of  $\{0 \dots H\}$ , so  $u_k$  denotes the time steps on which the robot is executing the  $k$ -th subtask. Notably, we assume the optimal solution to the target task follows a fixed structure, i.e., the same subtask sequence  $c$  can be used to solve the task, regardless of the initial state  $s_0$ . Meanwhile,  $u$  can be different in each episode, since the number of steps needed to complete each subtask depends on  $s_0$  as well as stochasticity in the environment and the policy.

### Few-Shot Adaptation through Language Decomposition

We design a simple sampling-based inference algorithm to find the best  $c^*$  for guiding the policy  $\tilde{\pi}$  to solve the target task. Since the resulting actions depend on both  $c$  and  $u$ , as discussed in Section 3.2, we jointly optimize  $c$  and  $u$  to minimize a cost function  $\mathcal{J}$  over all trajectories in  $\mathcal{D}_{\text{target}}$ :

$$\min_{c_{1:K} \in \mathcal{M}(s_0, \ell)} \sum_{\tau \in \mathcal{D}_{\text{target}}} \left( \min_{u_{1:K} \in \mathcal{U}} \mathcal{J}(c, u, \tau) \right). \quad (3.3)$$



To measure how well  $c$  and  $u$  enable the policy  $\tilde{\pi}$  to reproduce each  $\tau$ , the cost function is defined with the mean squared error between the predicted action  $\hat{a}_t$  and the ground truth  $a_t$  at each time step  $t$ . More specifically, we evaluate the policy  $\tilde{\pi}$  on the demonstration trajectory given  $c$  and  $u$  to compute  $\hat{a}_t \leftarrow \tilde{\pi}(s_t, c_{\min\{k:t \in u_k\}})$ . Then the cost function is defined as:

$$\mathcal{J}(c, u, \tau) = \sum_{n=1}^K \sum_{t \in u_n} \|a_t - \tilde{\pi}(s_t, c)\|^2. \quad (3.4)$$

By minimizing this cost across demonstrations, we compute a decomposition of the task  $c$  that would optimally perform the task by minimizing the loss between the action of the robot and the expert.

### Learning Composable Instruction-Following Primitives

We use language-conditioned behavior cloning [22] to learn a policy  $\hat{\pi}(s_t, \ell)$  based on the expert trajectories of  $\mathcal{D}_{\text{prior}}$ . To enable conditioning on fine-grained hierarchical language instructions, we factorize  $\hat{\pi}$  through  $c_{1:K}$ :

$$\hat{\pi}(\hat{a} | s_t, \ell) = \sum_{c_{1:K} \in \mathcal{L}} p(c_{1:K} | \ell) \sum_{k=1}^K \tilde{\pi}(\hat{a} | s_t, c_k) p(\mathfrak{k}_t = k) \quad (3.5)$$

for the subtask index at time  $t$ :  $\mathfrak{k}_t = \min\{k : t \in u_k, u_{1:K} \sim \text{Unif}(\mathcal{U})\}$ . We learn parameters  $\theta$  for  $\tilde{\pi}_\theta$  by minimizing the following behavioral cloning objective:

$$\mathcal{L}_{\text{BC}}(\theta) = \mathbb{E}_{(s_t, a_t, c_k, \ell) \sim \mathcal{D}_{\text{prior}}} \left[ \sum_{t=1}^H \|\tilde{\pi}_\theta(s_t, c_k) - a_t\|^2 \right]. \quad (3.6)$$

The training dataset  $\mathcal{D}_{\text{prior}}$  is an augmented version of BridgeData [62], a dataset containing a diverse set of manipulation tasks on common household objects. Details about how the subtask instructions are generated are discussed in Appendix C.2. Each  $c_i$  is further partitioned into a high-level component  $c_i^H$  and a low-level component  $c_i^L$ . Our full implementation is described in Appendix C.2.

### 3.3 REGRET ANALYSIS

Our theoretical results study the regret of this approach on out-of-distribution tasks in  $\rho_{\text{target}}$ , showing that it trades off the performance of the pre-trained policy on  $\rho_{\text{prior}}$  and the performance of the VLM  $\mathcal{M}$  in accurately modeling the hierarchical language decomposition  $p(c_{1:K})$  in  $\rho_{\text{target}}$ . We define regret with respect to the expert policy  $\pi_\beta$  and a given task distribution in terms of the MSE:

$$R_{\pi_\beta}(\pi; \rho) = \mathbb{E}_{\mathcal{T}_{\pi_\beta}^\rho} \left[ \frac{1}{H\sqrt{d_A}} \sum_{t=1}^H \|\pi(s_t, \ell) - \pi_\beta(s_t, \ell)\|^2 \right]. \quad (3.7)$$

**Algorithm 1:** Policy Adaptation via Language Optimization (PALO)

---

**Require:** a VLM  $\mathcal{M}$ , pre-trained instruction-following policy  $\pi(\hat{a} \mid s_t, c)$ , candidate decompositions to sample  $M$ , optimization steps  $N$

**Input:** new task described by  $\ell$  with  $n$  expert demonstrations  $\mathcal{D}_{\text{target}}$  collected manually

**Output:** policy  $\hat{\pi}(\cdot \mid s_t)$  adapted to the new task  $\ell$

- 1: **for**  $i = 1$  to  $M$  **do**
- 2:      $c_{1:K}^{(i)} \sim \mathcal{M}(s_0, \ell)$
- 3:     **for**  $j = 1$  to  $N$  **do**
- 4:          $u_{1:K}^{(i,j)} \sim \text{Unif}(\mathcal{U})$
- 5:  $\hat{c}_{1:K} \leftarrow \arg \min_{c_{1:K} \in \{c^{(i)}\}_{i=1}^M} \min_{u \in \{u^{(i,j)}\}_{j=1}^N} \mathcal{J}(c_{1:K}, u, \tau)$  (eq. 3.4)
- 6:  $\pi_{\text{PALO}}(\hat{a} \mid s_t, \ell) \leftarrow \pi(\hat{a} \mid s_t, \hat{c}_{\mathbf{k}_t})$
- 7: **return**  $\pi_{\text{PALO}}$ .

---

**Theorem 3.1.** *The (out-of-distribution) regret of PALO on  $\rho_{\text{target}}$  can be bounded as:*

$$R_{\pi_\beta}(\pi_{\text{PALO}}; \rho_{\text{target}}) \leq R_{\pi_\beta}(\hat{\pi}; \rho_{\text{prior}}) + \mathbb{E}[D_{\text{TV}}(p_{\text{target}}(c_{\mathbf{k}_t}), p_{\text{prior}}(c_{\mathbf{k}_t}))] \\ + (2D_{\text{KL}}[p(c_{1:K}), p_{\mathcal{M}}])^{1/2} + \frac{\sqrt{M} + \sqrt{n \log(Mn)}}{n} + 1/M + 1/K + N^{-2/K} \quad (3.8)$$

where  $\pi_{\text{PALO}}$  is from Algorithm 1,  $\hat{\pi}(s_t, \ell)$  is trained on  $\mathcal{D}_{\text{prior}}$  (Section 3.2), and  $t \sim \text{Unif}(1 \dots H)$ .

The proof is in Section 3.3. Theorem 3.1 shows that in the limit as  $N, M \rightarrow \infty$ , we can decompose the out-of-distribution regret of PALO into a sum of the in-distribution regret of the pre-trained policy, and the performance of the VLM in accurately decomposing language tasks:

$$R_{\pi_\beta}(\pi_{\text{PALO}}; \rho_{\text{target}}) \lesssim \underbrace{R_{\pi_\beta}(\hat{\pi}; \rho_{\text{prior}})}_{\text{pre-training MSE}} + \underbrace{(2 \mathbb{E}_{\rho_{\text{target}}} D_{\text{KL}}[p(c_{1:K}) \parallel p_{\mathcal{M}}(c_{1:K})])^{1/2}}_{\text{VLM accuracy}} \\ + \underbrace{\mathbb{E}[D_{\text{TV}}(p_{\text{target}}(c_{\mathbf{k}_t}), p_{\text{prior}}(c_{\mathbf{k}_t}))]}_{\text{local marginal conformity}}. \quad (3.9)$$

Viewing the **VLM accuracy** and **local marginal conformity** terms as the extent to which Assumption 3.1 and Assumption 3.2 are satisfied, we can see that under these conditions, Theorem 3.1 lets us directly relate the performance of the pre-trained policy  $\hat{\pi}$  on the training data  $\mathcal{D}_{\text{prior}}$  to the performance of the PALO algorithm on out-of-distribution tasks.

*Proof of Theorem 3.1.* We will first consider the empirical regret of the MLE estimate of  $c_{1:K}$ , and relate it to in-distribution regret of  $\tilde{\pi}$  using PAC techniques (see



Catoni [108], Alquier [109]). We will then bound the remaining error due to the approximations made by the PALO algorithm and this empirical regret.

Recall our definition of regret:

$$R_{\pi_\beta}(\pi; \rho) = \mathbb{E}_{\mathcal{T}_{\pi_\beta}^\rho} \left[ \frac{1}{H\sqrt{d_A}} \sum_{t=1}^H \|\pi(s_t, \ell) - \pi_\beta(s_t, \ell)\|^2 \right]. \quad (\text{from eq. 3.7})$$

We can also define an empirical target regret  $R_{\text{EMP}}$  measuring the fit of some  $c \in \mathcal{L}^K$  to the target distribution  $\rho_{\text{target}}$  in terms of Eq. (3.4):

$$R_{\text{EMP}}(c_{1:K}) = \mathbb{E}_{\mathcal{D}_{\text{target}} \sim \rho_{\text{target}}} \left[ \frac{1}{H\sqrt{d_A}} \sum_{\tau \in \mathcal{D}_{\text{target}}} \min_{u_{1:K}} \mathcal{J}(c_{1:K}, u_{1:K}, \tau) \right] \quad (3.10)$$

where  $\mathcal{J}$  is the cost function in Eq. (3.4). PALO selects  $c_{\text{PALO}} = \arg \min_{c_{1:K} \in \mathcal{L}^K} \widehat{R}_{\text{EMP}}(c_{1:K})$  to minimize an approximation of this quantity for samples  $u^{(1)}, \dots, u^{(N)} \sim \text{Unif}(\mathcal{U})$ :

$$\widehat{R}_{\text{EMP}}(c_{1:K}) = \mathbb{E}_{\mathcal{D}_{\text{target}} \sim \rho_{\text{target}}} \left[ \frac{1}{H\sqrt{d_A}} \sum_{\tau \in \mathcal{D}_{\text{target}}} \min_{i \in \{1 \dots N\}} \mathcal{J}(c_{1:K}, u^{(i)}, \tau) \right]. \quad (3.11)$$

We will also define a distributional notion of conditional regret for our analysis:

$$\widetilde{R}_{\pi_\beta}(\tilde{\pi} \mid s_0, \ell, c_{1:K}) = \mathbb{E}_{\tau \sim \mathcal{T}_{\pi_\beta}^{s_0}} \left[ \frac{1}{H\sqrt{d_A}} \min_{u_{1:K} \in \mathcal{U}} \mathcal{J}(c_{1:K}, u_{1:K}, \tau) \right]. \quad (3.12)$$

We now make use of the following PAC result [110, 109], which follows from Hoeffding's inequality:

**Lemma 3.2** (Alquier [109, Theorem 1.2]). *Let  $\mathcal{H}$  be a class of functions  $f : \mathcal{X} \rightarrow [0, 1]$  with  $|\mathcal{H}| = M$ , and let  $\rho \in \mathcal{P}(X)$  be an arbitrary data distribution. Further, suppose  $\mathcal{D}$  is a sample of size  $n$  drawn i.i.d. from  $\rho$ . Then, for any  $\varepsilon \in (0, 1)$ , we have*

$$\Pr \left( \underbrace{\mathbb{E}_{x \sim \rho}[f(x)]}_{\text{generalization risk}} \leq \underbrace{\mathbb{E}_{x \sim \mathcal{D}}[f(x)]}_{\text{empirical risk}} + \sqrt{\frac{\log M - \log \varepsilon}{2n}} \right) \geq 1 - \varepsilon. \quad (3.13)$$

Taking  $X$  to be the space of trajectories and  $\mathcal{H} = \mathcal{M}(s_0, \ell)$  for  $f(c) = \min_u \mathcal{J}(c, u, \tau)$ , we can apply Lemma 3.2 to the empirical regret  $R_{\text{EMP}}$  in Eq. (3.10) to obtain (for any  $\varepsilon \in (0, 1)$ )

$$\Pr \left( \forall c_{1:K} \in \mathcal{M}(s_0, \ell), R_{\pi_\beta}(\hat{\pi} \mid s_0, \ell, c_{1:K}) \leq R_{\text{EMP}}(c_{1:K}) + \sqrt{\frac{\log M - \log \varepsilon}{2n}} \right) \geq 1 - \varepsilon. \quad (3.14)$$

Taking  $c_{\text{PALO}}$  to be the output of the PALO algorithm, we can relate the true regret of PALO on the current task (left) to its empirical regret (right):

$$\Pr\left(R_{\pi_\beta}(\hat{\pi} \mid s_0, \ell, c_{\text{PALO}}) \leq R_{\text{EMP}}(c_{\text{PALO}}) + \sqrt{\frac{\log M - \log \varepsilon}{2n}}\right) \geq 1 - \varepsilon. \quad (3.15)$$

Since regret is bounded by 1, we can convert to an expectation:

$$\mathbb{E}_{\mathcal{D}_{\text{target}}}[R_{\pi_\beta}(\hat{\pi} \mid s_0, \ell, c_{\text{PALO}})] \leq \mathbb{E}_{\mathcal{D}_{\text{target}}}[R_{\text{EMP}}(c_{\text{PALO}})] + \sqrt{\frac{\log M - \log \varepsilon}{2n}} + \varepsilon.$$

**Lemma 3.3.** *Suppose  $u, u' \sim \text{Unif}(\mathcal{U})$  are i.i.d. samples from a uniform distribution over the ordered  $K$ -partitions  $\mathcal{U}$  of  $\{1 \dots H\}$ . For any  $\varepsilon \in [0, 1/K]$ , we have*

$$\Pr\left(\sum_{k=1}^K |u_k \cap u'_k| \leq H\varepsilon\right) \leq e^{-2H(\frac{1}{K} - \varepsilon)^2}.$$

**Lemma 3.4.** *There exists an  $\varepsilon \in [0, 1/K]$  such that*

$$\varepsilon + e^{-2H(\frac{1}{K} - \varepsilon)^2} \leq 1/K + N^{-2/K}. \quad (3.16)$$

Since Algorithm 1 (line 4) only samples  $N$  values for  $u$  instead of the full space for the min in Eq. (3.10), we must separately consider the degree of suboptimality in the decomposition  $c_{\text{PALO}}$  relative to the optimal  $c^* = \arg \min_{c \in \mathcal{M}(s_0, \ell)} R_{\text{EMP}}(c)$  that results from our approach to determine the effect of  $N$  on the final bound. Applying Lemma 3.3, we can say:

$$\begin{aligned} & \mathbb{E}_{\mathcal{D}_{\text{target}}}\left[\tilde{R}_{\pi_\beta}(\hat{\pi} \mid s_0, \ell, c_{\text{PALO}})\right] \\ & \leq \mathbb{E}_{\mathcal{D}_{\text{target}}}\left[R_{\text{EMP}}(c_{\text{PALO}})\right] + \sqrt{\frac{\log M - \log \varepsilon}{2n}} \\ & \leq \mathbb{E}_{\mathcal{D}_{\text{target}}}\left[R_{\text{EMP}}(c^*)\right] + \sqrt{\frac{\log M - \log \varepsilon}{2n}} + \varepsilon + e^{-2H(\frac{1}{K} - \varepsilon)^2} \\ & \leq \mathbb{E}_{\mathcal{D}_{\text{target}}}\left[R_{\text{EMP}}(c^*)\right] + \sqrt{\frac{\log M - \log \varepsilon}{2n}} + 1/K + N^{-2/K}. \end{aligned}$$

For  $\varepsilon = \sqrt{M}/n$ , we get

$$\begin{aligned} & \mathbb{E}_{\mathcal{D}_{\text{target}}}\left[R_{\pi_\beta}(\hat{\pi} \mid s_0, \ell, c_{\text{PALO}})\right] \\ & \leq \mathbb{E}_{\mathcal{D}_{\text{target}}}\left[R_{\text{EMP}}(c^*)\right] + \frac{\sqrt{M} + \sqrt{n \log(Mn)}}{n} + 1/K + N^{-2/K}. \quad (3.17) \end{aligned}$$

So, we have related the true regret of PALO on the current task (left) to its empirical regret in the limit of infinite samples (right). All that remains is to compute the empirical regret, for which we make use of the following lemmas.

**Lemma 3.5.** Denote the true (unobserved) target decomposition as  $c_{1:K}$ . We can relate the empirical regret of the optimal PALO solution  $c^*$  to the empirical regret of the true decomposition.

$$\mathbb{E}_{\mathcal{D}_{\text{target}}} [R_{\text{EMP}}(c^*)] \leq \mathbb{E}_{\mathcal{D}_{\text{target}}} [R_{\text{EMP}}(c_{1:K}) + D_{\text{TV}}(p(c_{1:K}), p_{\mathcal{M}}(c_{1:K}))] + 1/M$$

**Lemma 3.6.** The empirical regret of  $\tilde{\pi}$  can be bounded for  $i \sim \text{Unif}(1 \dots K)$  as

$$\mathbb{E}_{\mathcal{D}_{\text{target}}} [R_{\text{EMP}}(c_{1:K})] \leq R_{\pi_\beta}(\hat{\pi}; \rho_{\text{prior}}) + \mathbb{E} [D_{\text{TV}}(p_{\text{target}}(c_{\mathbb{t}_t}), p_{\text{prior}}(c_{\mathbb{t}_t}))].$$

Applying Lemma 3.5 and Lemma 3.6 to Eq. (3.17) yields a bound of the correct form.

$$\begin{aligned} \mathbb{E}_{\mathcal{D}_{\text{target}}} [R_{\text{EMP}}(c^*)] &\leq \mathbb{E}_{\mathcal{D}_{\text{target}}} [R_{\text{EMP}}(c_{1:K})] + D_{\text{TV}}(p(c_{1:K}), p_{\mathcal{M}}) + 1/M \\ &\leq \mathbb{E}_{\mathcal{D}_{\text{prior}}} [R_{\pi_\beta}(\hat{\pi}; \rho_{\text{prior}})] + \mathbb{E} [D_{\text{TV}}(p_{\text{target}}(c_{\mathbb{t}_t}), p_{\text{prior}}(c_{\mathbb{t}_t}))] \\ &\quad + D_{\text{TV}}(p(c_{1:K}), p_{\mathcal{M}}) + 1/M. \end{aligned}$$

To make the  $D_{\text{TV}}(p(c_{1:K}), p_{\mathcal{M}})$  term more interpretable as a VLM accuracy, we convert to a KL divergence with Pinsker's inequality [111]:

$$\mathbb{E}_{\mathcal{D}_{\text{target}}} [R_{\text{EMP}}(c_{\text{PALO}})] \leq \mathbb{E}_{\mathcal{D}_{\text{prior}}} [R_{\pi_\beta}(\hat{\pi}; \rho_{\text{prior}})] \mathbb{E} [D_{\text{TV}}(p_{\text{target}}(c_{\mathbb{t}_t}), p_{\text{prior}}(c_{\mathbb{t}_t}))] \quad (3.18)$$

$$+ \sqrt{2D_{\text{KL}}(p(c_{1:K}), p_{\mathcal{M}})} + 1/M. \quad (3.19)$$

Since  $\mathbb{E}_{\mathcal{D}_{\text{target}}} [R_{\pi_\beta}(\hat{\pi} \mid s_0, \ell, c_{\text{PALO}})] = R_{\pi_\beta}(\pi_{\text{PALO}}; \rho_{\text{target}})$ , plugging Eq. (3.19) into Eq. (3.17) gives the desired result:

$$\begin{aligned} R_{\pi_\beta}(\pi_{\text{PALO}}; \rho_{\text{target}}) &\leq [R_{\pi_\beta}(\hat{\pi}; \rho_{\text{prior}})] \\ + \mathbb{E} [D_{\text{TV}}(p_{\text{target}}(c_{\mathbb{t}_t}), p_{\text{prior}}(c_{\mathbb{t}_t}))] &\quad (3.20) \end{aligned}$$

$$\begin{aligned} &+ \sqrt{2D_{\text{KL}}(p(c_{1:K}), p_{\mathcal{M}})} + 1/M \\ &+ \frac{\sqrt{M} + \sqrt{n \log(Mn)}}{n} + 1/K + N^{-2/K}. \quad (3.21) \end{aligned}$$

□

*Proof of Lemma 3.3.* Define  $\{X_i\}_{i=1}^H$  to be the unique index  $k$  such that  $i \in u_k$ , and  $\{X'_i\}_{i=1}^H$  to be the unique index  $k$  such that  $i \in u'_k$ . We have

$$\begin{aligned} \Pr\left(\sum_{k=1}^K |u_k \cap u'_k| \geq H\varepsilon\right) &= \Pr\left(\sum_{i=1}^H \mathbf{1}\{X_i = X'_i\} \geq H\varepsilon\right) \\ &= \Pr\left(\sum_{i=1}^H \mathbf{1}\{X_i \neq X'_i\} \leq H(1 - \varepsilon)\right). \quad (3.22) \end{aligned}$$

Now, we observe

$$\Pr(X_i \neq X'_i) = \sum_{k=1}^K (1 - p_{X_i}(k)) p_{X'_i}(k) \quad (3.23)$$

$$= 1 - \sum_{k=1}^K p_{X_i}(k)^2. \quad (3.24)$$

Eq. (3.23) is concave in  $p_{X_i}$ , and so is maximized when for any  $\delta p_{X_i}$  and some  $\lambda$ ,

$$\lambda \delta p_{X_i}(k) = -2 \sum_{k=1}^K p_{X_i}(k) \delta p_{X_i}(k),$$

i.e., when  $p_{X_i}(k) = \text{const.} = 1/K$  for all  $k$ . Thus, we have

$$\mathbb{E}[\mathbf{1}\{X_i \neq X'_i\}] = \Pr(X_i \neq X'_i) \leq 1 - 1/K.$$

Continuing from Eq. (3.22) with  $\mu = \mathbb{E}[\sum_{i=1}^H \mathbf{1}\{X_i \neq X'_i\}]$ ,

$$\begin{aligned} \Pr\left(\sum_{i=1}^H \mathbf{1}\{X_i \neq X'_i\} \leq H(1 - \varepsilon)\right) &= 1 - \Pr\left(\sum_{i=1}^H \mathbf{1}\{X_i \neq X'_i\} \leq \mu + (H(1 - \varepsilon) - \mu)\right) \\ &\geq 1 - \exp\left(\frac{-2(H(1 - \varepsilon) - \mu)^2}{H}\right) \\ &\quad \text{(Hoeffding [112])} \\ &\geq 1 - \exp\left(\frac{-2H^2((1 - \varepsilon) - (1 - 1/K))^2}{H}\right) \\ &= 1 - \exp(-2H(1/K - \varepsilon)^2). \end{aligned} \quad (3.25)$$

Taking the complement of Eq. (3.25) yields the desired result:

$$\Pr\left(\sum_{k=1}^K |u_k \cap u'_k| \leq H\varepsilon\right) \leq e^{-2H(\frac{1}{K} - \varepsilon)^2}. \quad (3.26)$$

□

*Proof of Lemma 3.4.* The statement follows from the ansatz

$$\varepsilon = \frac{1}{K} - \sqrt{\frac{\log N}{NHK}}$$

Plugging in,

$$\begin{aligned} \varepsilon + e^{-2H(\frac{1}{K} - \varepsilon)^2} &= N^{-2/K} + \frac{1}{K} - \left(\frac{\log N}{HKN}\right)^{1/2} \\ &\leq 1/K + N^{-2/K}. \end{aligned}$$

□

*Proof of Lemma 3.5.* Recall the definition of the optimal PALO solution

$$c^* = \arg \min_{c \in \mathcal{M}(s_0, \ell)} R_{\text{EMP}}(c). \quad (3.27)$$

Now, noting regrets are bounded by 1 from Eq. (3.7), we have

$$\begin{aligned} \mathbb{E}_{\mathcal{D}_{\text{target}}} [R_{\text{EMP}}(c^*)] &= \mathbb{E}_{\mathcal{D}_{\text{target}}} \left[ \min_{c \in \mathcal{M}(s_0, \ell)} R_{\text{EMP}}(c) \right] \\ &= \mathbb{E}_{\mathcal{D}_{\text{target}}} \left[ \left( \frac{p(c)}{p_{\mathcal{M}}(c)} \right) \min_{\{c^{(i)}\}_{i=1}^M \sim p_{c_{1:K}}} [R_{\text{EMP}}(c^{(i)})] \right] \\ &= \mathbb{E}_{\mathcal{D}_{\text{target}}} \left[ \min_{\{c^{(i)}\}_{i=1}^M \sim p_{c_{1:K}}} [R_{\text{EMP}}(c^{(i)})] \right] \\ &\quad + \mathbb{E}_{\mathcal{D}_{\text{target}}} \left[ \left( \frac{p(c)}{p_{\mathcal{M}}(c)} - 1 \right) \min_{\{c^{(i)}\}_{i=1}^M \sim p_{c_{1:K}}} [R_{\text{EMP}}(c^{(i)})] \right] \\ &\leq \mathbb{E}_{\mathcal{D}_{\text{target}}} \left[ \min_{\{c^{(i)}\}_{i=1}^M \sim p_{c_{1:K}}} [R_{\text{EMP}}(c^{(i)})] \right] + \mathbb{E}_{\mathcal{D}_{\text{target}}} \left| \frac{p(c)}{p_{\mathcal{M}}(c)} - 1 \right| \\ &\leq \mathbb{E}_{\mathcal{D}_{\text{target}}} \left[ \min_{\{c^{(i)}\}_{i=1}^M \sim p_{c_{1:K}}} [R_{\text{EMP}}(c^{(i)})] + D_{\text{TV}}(p(c_{1:K}), p_{\mathcal{M}}(c_{1:K})) \right] \\ &= \mathbb{E}_{\mathcal{D}_{\text{target}}} \left[ \Pr(R_{\text{EMP}}(c_{1:K}) < c^{(i)} \text{ for } \{c^{(i)}\}_{i=1}^M \sim p_{c_{1:K}}) \right. \\ &\quad \left. + R_{\text{EMP}}(c_{1:K}) + D_{\text{TV}}(p(c_{1:K}), p_{\mathcal{M}}(c_{1:K})) \right] \\ &= \mathbb{E}_{\mathcal{D}_{\text{target}}} [R_{\text{EMP}}(c_{1:K}) + D_{\text{TV}}(p(c_{1:K}), p_{\mathcal{M}}(c_{1:K}))] + 1/M. \end{aligned}$$

□

*Proof of Lemma 3.6.* We consider the empirical regret of  $\tilde{\pi}$  using the true decomposition  $u_{1:K}, c_{1:K} \sim p_{\text{target}}$ , for  $t \sim \text{Unif}(1 \dots H)$  and  $\mathbf{k}_t$  defined as in Eq. (3.5):

$$\begin{aligned} \mathbb{E}_{\mathcal{D}_{\text{target}}} [R_{\text{EMP}}(c_{1:K})] &= \mathbb{E}_{\mathcal{D}_{\text{target}}} \left[ \frac{1}{H\sqrt{d_A}} \sum_{\tau \in \mathcal{D}_{\text{target}}} \min_{u_{1:K}} \mathcal{J}(c_{1:K}, u_{1:K}, \tau) \right] \\ &= \mathbb{E}_{\mathcal{D}_{\text{target}}} \left[ \frac{1}{H\sqrt{d_A}} \sum_{\tau \in \mathcal{D}_{\text{target}}} \min_{u_{1:K}} \sum_{n=1}^K \sum_{t \in u_n} \|a_t - \tilde{\pi}(s_t, c_n)\|^2 \right] \\ &\leq \mathbb{E}_{u_n, c_n \sim \mathcal{D}_{\text{target}}} \left[ \frac{1}{H\sqrt{d_A}} \sum_{\tau \in \mathcal{D}_{\text{target}}} \sum_{n=1}^K \sum_{t \in u_n} \|a_t - \tilde{\pi}(s_t, c_n)\|^2 \right] \end{aligned}$$

$$\begin{aligned}
&\leq \frac{1}{H\sqrt{d_A}} \mathbb{E}_{p_{\text{target}}} \left[ \sum_{n=1}^K \sum_{t \in u_n} \|a_t - \tilde{\pi}(s_t, c_n)\|^2 + D_{\text{TV}}(p_{\text{target}}(c_n), p_{\text{prior}}(c_n)) \right] \\
&= \mathbb{E}_{u_n, c_n \sim p_{\text{prior}}} \left[ \frac{1}{H\sqrt{d_A}} \sum_{n=1}^K \sum_{t \in u_n} \|a_t - \tilde{\pi}(s_t, c_n)\|^2 \right] + \mathbb{E} [D_{\text{TV}}(p_{\text{target}}(c_{\mathbb{t}_t}), p_{\text{prior}}(c_{\mathbb{t}_t}))] \\
&= R_{\pi_\beta}(\hat{\pi}; \rho_{\text{prior}}) \\
&\quad + \mathbb{E} [D_{\text{TV}}(p_{\text{target}}(c_{\mathbb{t}_t}), p_{\text{prior}}(c_{\mathbb{t}_t}))].
\end{aligned}$$

□

## System Details

We use a ResNet-34 [113] to model the policy  $\pi(a | s, c)$ , where  $c = (c^H, c^L)$  is a concatenation of high- and low-level instructions. The instruction  $c = (c^L, c^H)$  is passed through a frozen MUSE model [58] to obtain embeddings before being fused into the ResNet with FiLM layers [114]. Architecture details are presented in ??, and the overall algorithm is shown in Algorithm 1.

## 3.4 EXPERIMENTS

In this section, we show that PALO can better adapt to long-horizon and out-of-distribution tasks from a few expert demonstrations than existing learned language-conditioned manipulation policies (both zero-shot and when finetuned to demonstrations), as well as a nonparametric few-shot adaptation method. Ablation studies also show all components of PALO are necessary.

### Experimental Setup

We evaluate on a variety of long-horizon and/or unseen tasks across four scenes from the Bridge tabletop manipulation setup [62]. These involve manipulating new combinations of objects and behaviors unseen in the training data to accomplish long-horizon tasks, such as making a salad or pouring into a bowl. For each task, we collect a set of five expert demonstrations  $\mathcal{D}_{\text{target}}$  for few-shot learning. Besides separating by scenes, we can also separate the tasks into 4 long-horizon tasks (put in, salad, sweep mints, sweep skittles) and 4 unseen-skills tasks (pry away, pour spoon, rotate marker, rotate spoon). Experimental details and example rollouts are presented in Appendix C.2.

### Baselines

We compare against the following baselines trained on BridgeData:

**Octo [67]:** A general transformer-based robot manipulation policy with diffusion action head.

Table 3.1: Method Comparisons

Scene	Task	PALO	RT-2-X	FT-Octo	Octo	GRIF	VINN	FT-LCBC	LCBC
Drawer	put in	<b>0.7</b>	0.0	0.0	0.2	0.1	0.0	0.3	0.1
	pry away	<b>0.6</b>	0.2	0.2	0.1	0.0	0.1	0.0	0.0
Bowl	salad	<b>0.7</b>	0.5	0.0	0.3	0.4	0.0	0.6	0.0
	pour	<b>0.5</b>	0.1	0.2	0.3	0.0	0.0	0.0	0.0
Sweep	mints	<b>0.7</b>	0.3	0.1	0.2	0.0	0.0	0.2	0.0
	skittles	<b>0.8</b>	0.4	0.0	0.4	0.3	0.0	0.3	0.2
Rotation	marker	<b>0.9</b>	0.4	0.0	0.1	0.3	0.4	0.4	0.0
	spoon	<b>0.8</b>	0.2	0.1	0.1	0.1	0.5	0.2	0.0
<b>Average</b>		<b>0.71</b>	0.26	0.10	0.21	0.15	0.13	0.25	0.08

**GRIF [7]:** A language-conditioned robot control method that uses pre-trained CLIP [17] representations to connect language instructions to goals for the policy to reach.

**RT-2-X [85]:** A language-conditioned robot control model with 55B parameters that transfers knowledge from internet-scale pre-training to manipulation zero-shot.

**LCBC [22]:** Language imitation with a ResNet and pretrained MUSE [58] embeddings.

**VINN [115]:** Using k-Nearest Neighbor to select actions from the training data based on similarity between the task representations of the observation and training data. We used GRIF’s CLIP encoder for the representations used to calculate similarity scores.

**FT-Octo:** Octo transformer finetuned on the few-shot demonstration (see Appendix C.2 for details).

**FT-LCBC:** Similar to FT-Octo, but fine-tuning LCBC on the few-shot demonstrations.

Results from the experiments are shown in Fig. 3.4, with detailed per-task breakdowns in Table 3.1.

Across eight different tasks, our PALO method yielded a success rate of 71.3%, while the best zero-shot policies only resulted in a success rate of 26.3%. While most of the zero-shot methods degrade when the task became increasingly more out-of-distribution for the pretrained policy (for example, tasks in the “salad” scene achieved a 30% overall performance across the 4 baseline models while pouring

Table 3.2: Ablations

Scene	Task	PALO	No $c^H$	No $c^L$	Fixed Times	Zero-shot	No VLM
Drawer	put in	<b>0.7</b>	0.2	0.4	0.4	0.3	0.0
	pry open	<b>0.6</b>	0.4	0.2	0.1	0.4	0.1
Bowl	salad	<b>0.7</b>	0.4	0.5	0.4	0.2	0.0
	pour scoop	<b>0.5</b>	0.1	0.4	0.4	0.2	0.0
Sweep	mints	<b>0.7</b>	0.5	0.3	0.5	0.0	0.0
	skittles	<b>0.8</b>	0.7	0.2	0.5	0.4	0.2
Rotation	marker	<b>0.9</b>	0.6	0.3	0.3	0.1	0.3
	spoon	<b>0.8</b>	0.6	0.1	0.2	0.3	0.2

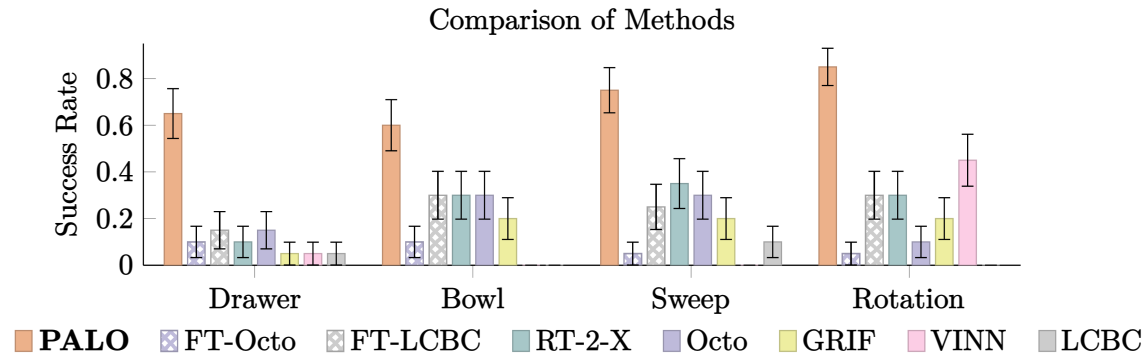


Figure 3.4: Comparison of PALO with baseline methods on different scenes with one standard error.

from scoop only achieved 12% performance across the models), our method remained effective, with all 8 tasks performing at a success rate of 50% or better.

The **FT-Octo** and **FT-LCBC** baselines allow us to compare the nonparametric adaptation of PALO to conventional parametric finetuning. While Octo trained only on BridgeData achieved moderate zero-shot success, finetuning on only five demonstrations overfit and worsened performance. The FT-LCBC baseline did benefit from finetuning, but still failed to ever exceed 30% success rate across all tasks. We observe that the small size of trajectories made these datasets an unfavorable candidate for finetuning, as any variance brought by the human controller may be amplified and cause unfavorable movements during evaluation. The nonparametric **VINN** baseline performed well on the rotation tasks (45% success rate), but failed to achieve greater than 5% success rate on the other tasks.

### Ablations

We ablate the following components of our method in Fig. 3.6:





Figure 3.5: An execution of our method on the task “pour the contents of the scoop into the bowl.” Full breakdown of task and instructions can be seen at Section 3.4.

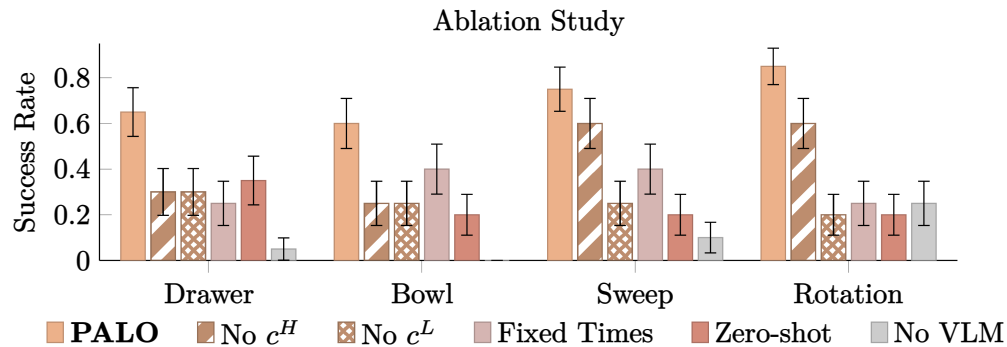


Figure 3.6: Ablation study of PALO on different scenes, plotted with one standard error.

**Ours:** Our full PALO approach

**No  $c^H$ :** No high-level  $c^H$  conditioning for the learned policy via masking.

**No  $c^L$ :** No low-level  $c^L$  instruction conditioning via masking.

**Fixed Times:** Use fixed  $u = [\frac{H}{k}, \frac{2H}{k}, \dots, \frac{(k-1)H}{k}]$  in each trajectory to evaluate Eq. (3.3).

**Zero-Shot Decomposition:** Generate  $c$  zero-shot without expert demonstrations.

**No VLM:** No VLM decomposition proposals by using only  $\ell$  with our policy.

While the sweeping and rotation scenes gave comparable performance with masked high level instructions (**No  $c^H$** ), the performance deteriorated in Drawer and Bowl, which involved more unfamiliar items for the pretrained policy. The remaining ablations (**No  $c^L$** , **Fixed Times**, **Zero-Shot Decomposition**, **No VLM**) decreased performance across all scenes. These approaches are discussed in more depth in Appendix C.1. Overall results in Fig. 3.6 show that all components of the PALO method are needed. Full evaluations across 10 trials are in Table 3.2.

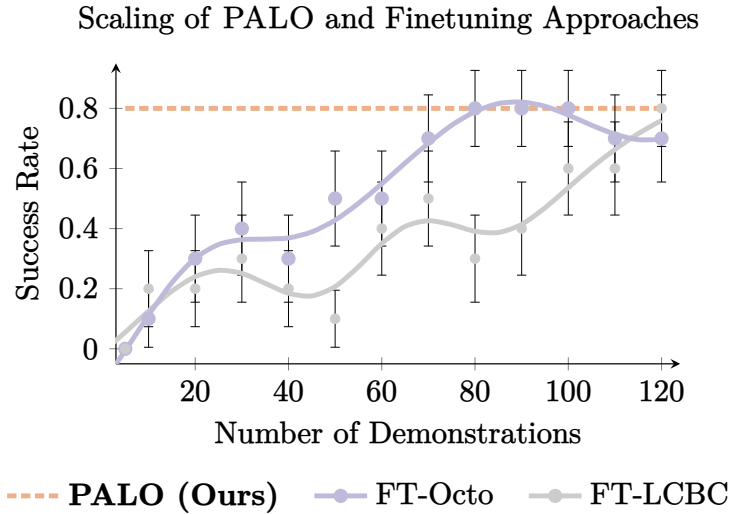


Figure 3.7: Performance of PALO with 5 demonstrations compared to finetuning Octo on different number of demonstrations, plotted with one standard error.

### Scaling with Demonstrations

We study the scaling of our nonparametric method and a parametric finetuning approach with  $> 5$  demonstrations of the skittle sweeping task in Fig. 3.7. We observe that while Policy Adaptation via Language Optimization achieves the best performance (80%) using any number of demonstrations, the Octo finetuning baseline needs at least 80 expert demonstrations to achieve comparable performance, while LCBC needs at least 120 demonstrations.

### Qualitative Results

We show successful task executions in Figs. 3.3 and 3.5. While the full method is robust to logically unsound instructions generated by the VLM, failures in reasoning and execution occur when we ablate our methods. Fig. 3.11 and Fig. 3.12 are two examples in which reasoning break down in ablations.

### Inference Details

During inference, we chunk each low-level instruction into length 8 intervals, switching to the new set of low-level (and high-level, if applicable) after these 8 steps. We chose a fixed interval instead of a dynamically allocated one due to the policy choosing to mostly stay put after finishing the action prescribed by the low-level instruction.

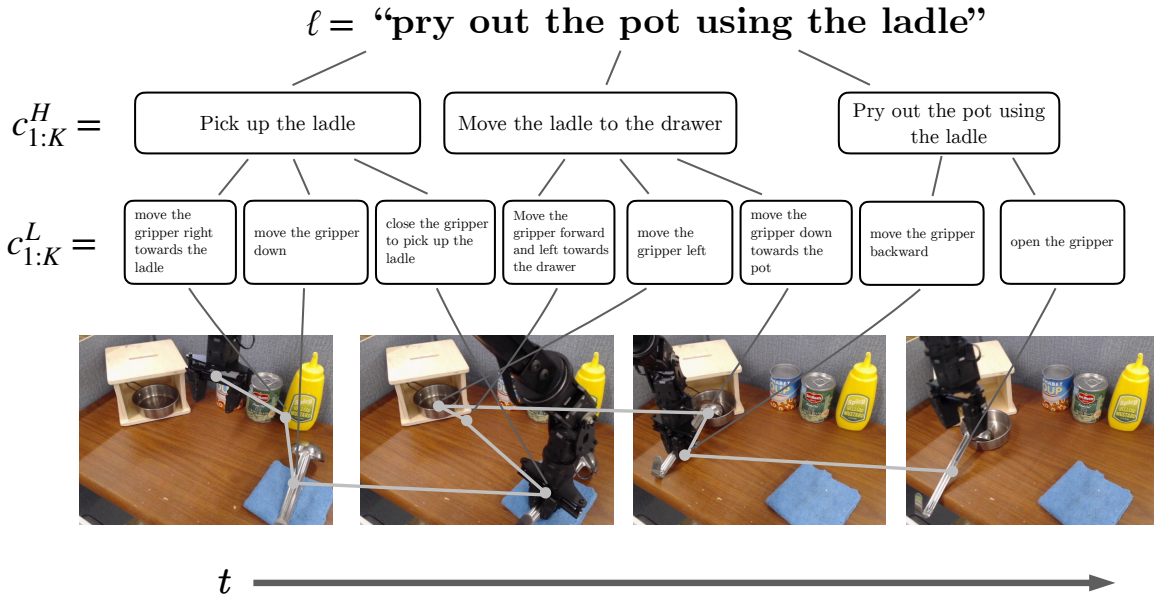


Figure 3.8: An execution of our method on the task “Pry out the pot using the ladle.”

### Success Cases

We show the full breakdowns of success cases here. Fig. 3.8 and Fig. 3.9 gives detailed description of the robot’s action primitives generated by PALO during inference.

### Full PALO Failure

While PALO is robust in generating language primitives that help achieve the task, it does not guarantee a successful execution of the policy as shown in Fig. 3.10. PALO can fail when the underlying policy fails to execute a low-level motion, after which the robot may not be able to recover and complete the task.

### Ablation Failures

When we ablate the components of PALO, we begin to see more critical failures. Fig. 3.13 demonstrates a case of grounding failure when  $c_H$  is masked out, i.e., when PALO loses half of the optimized task decomposition.

## 3.5 CONCLUSION

PALO is an approach for few-shot adaptation to unseen tasks that exploits the semantic understanding of task decomposition provided by vision-language models. In extensive real world experiments, we find that PALO is able to use language

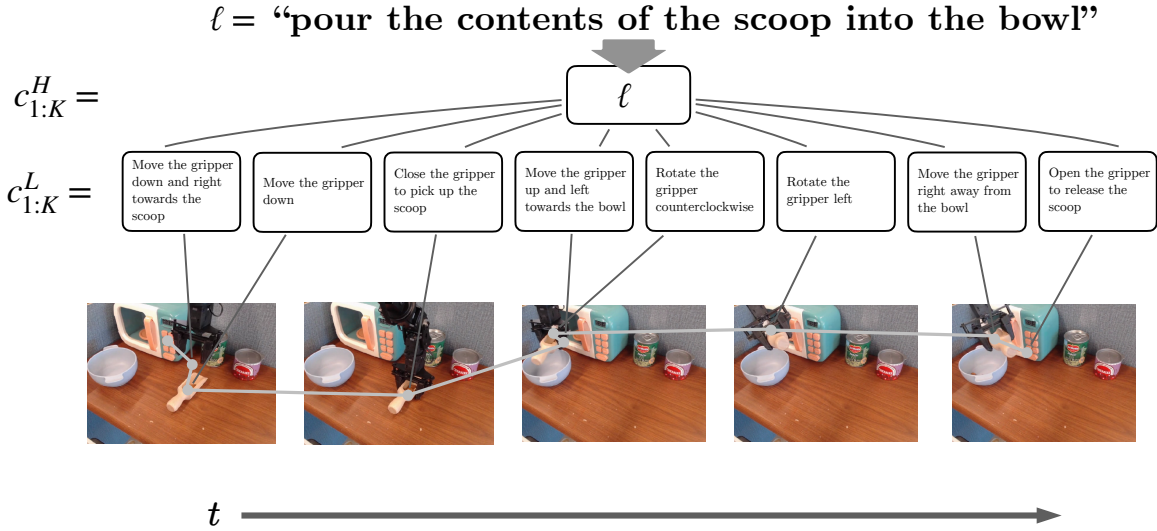


Figure 3.9: An execution of our method on the task “pour the contents of the scoop into the bowl.”. Note that the high level instruction is  $\ell$  itself, as the best-proposed language decomposition does not create additional subtasks.

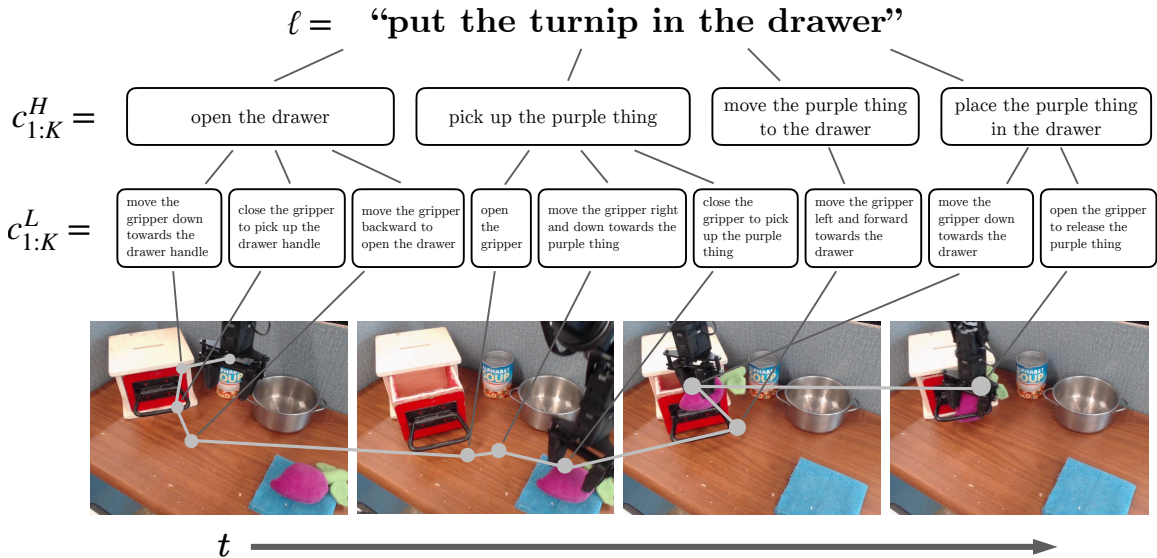


Figure 3.10: Failure in execution: while the robot completed every subtask correctly up until the last subtask, it did not achieve it due to errors within the policy.



Figure 3.11: Spatial reasoning failure occurred when masking out low level instruction. The task was to “sweep the mints using the towel.” Due to the presence of the pot and the mushroom, being both strong priors within BridgeData, the policy chose not to follow the high level instruction.



Figure 3.12: Grounding failure occurs when high level instruction is masked out. While the low level instruction “move the gripper left” correctly predicts the next reasonable action, masking out the context of the subtask “put the mushroom in the bowl” causes the policy to overshoot its trajectory.

to adapt to unseen long-horizon robot manipulation tasks across a wide range of tabletop setups.

**Limitations and Future Work.** We assume the dataset has a consistent format of high-level language labels and proprioception, making it more challenging to generalize our low-level heuristic generation on drastically different embodiments. The discrete optimization over subtask time steps may also scale poorly with the number of subtasks and time steps. Future work could explore more efficient optimization methods for this problem.



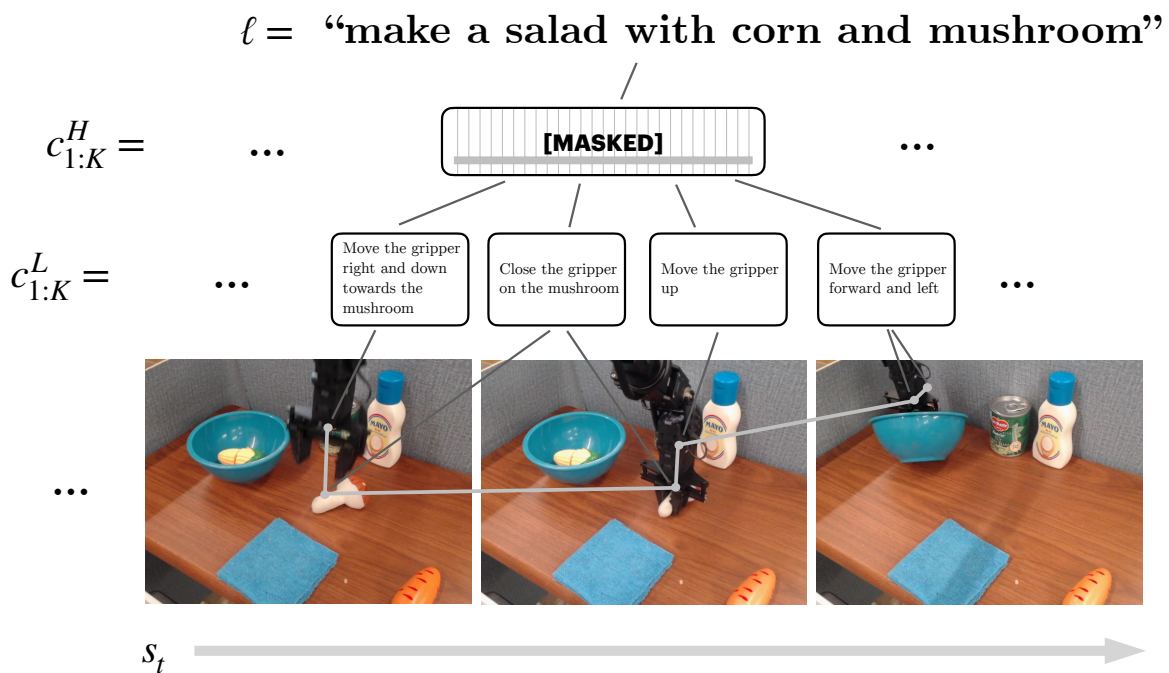


Figure 3.13: In this instance, we mask out the high level instructions, and the policy is only conditioned on the low-level instructions. We see that the low-level instruction “move the gripper forward and left.” causes the robot to overshoot its trajectory and causes failure in execution.

# II

## REPRESENTATIONS FOR COMPOSITIONAL DECISION MAKING

## 4

A METRIC STRUCTURE FOR  
SUCCESSOR REPRESENTATIONS

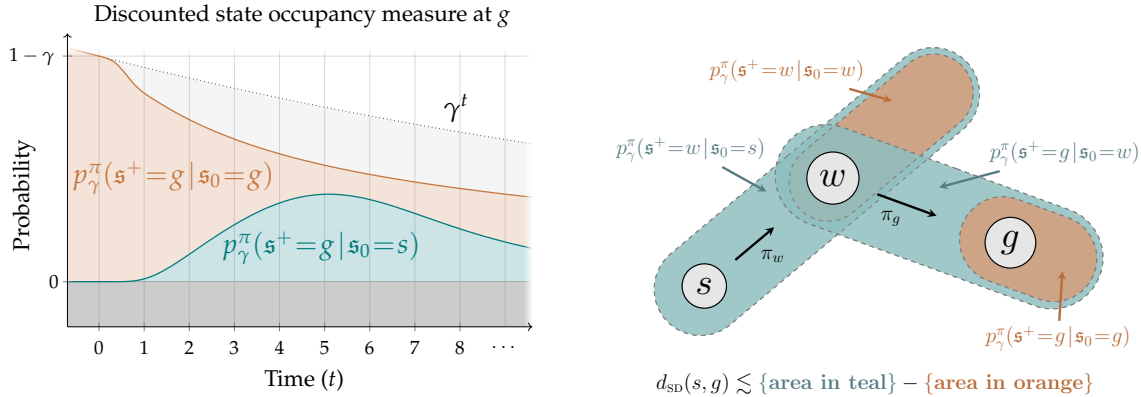
Graph search is one of the most important ideas in CS, being introduced in almost every introductory CS class. However, classes often overlook a key assumption: that transitions be deterministic. With deterministic transitions, shortest-path lengths obey the triangle inequality. This property, encoded into dynamic programming algorithms, allows one to search over an exponential number of paths and find the shortest in polynomial time. This property also allows for generalization, finding new paths unseen in the data. However, in graphs (or, more generally, Markov processes) with stochastic transitions, it is unclear how to define the distance between two states such that this distance obeys the triangle inequality.

A reasonable solution for goal-reaching is to learn *temporal distances*, which reflect some notion of transit time between states [116–120]. However, simply defining distances as hitting times breaks down in stochastic settings, as shown in prior work [46]. Stochastic settings are ubiquitous in real-world problems: from autonomous vehicles navigating around drunk bar-goers, to healthcare systems rife with unobservable features. Indeed, many advances in ML over the last decade have been predicated on probabilistic models (e.g., diffusion models, VAEs), so it seems rather anachronistic that an important control primitive (the notion of distances) is not well defined in a probabilistic sense.

The key challenge is that the prior notions of temporal distance break down in stochastic settings. Nonetheless, the triangle inequality holds great appeal as a strong inductive bias for learning temporal distances: the distance between two states should be less than the length of a path that goes through a particular waypoint state. Indeed, prior work has aimed to exploit this notion by learning “temporal distance metrics” that can broadly generalize from less data.

The starting point for our work is to think about distances probabilistically. Because the dynamics may be stochastic, the number of steps it takes to traverse between two states is not a definite quantity, but rather a random variable. To estimate the (long-term) probabilities of transiting between two states, we will build on prior temporal contrastive learning [53, 121], a popular and stable class of time series representation learning methods. Intuitively, these methods learn representations from time series data so that observations that occur nearby in time are given similar representations. Importantly, contrastive methods based





(a) Starting at state  $s$ , we visualize the (discounted) probability of reaching state  $g$  after exactly  $t$  steps (teal). The sum of these probabilities (■ area) is the probability of reaching state  $g$  at some point in the future. Our method defines the distance between states  $s$  and  $g$  as the difference in these shaded areas (■ area - ■ area), which we prove is non-negative (Lemma 4.3). For a fixed policy  $\pi$  and state  $s$ , we can view the  $\gamma$ -time-discounted distribution of future states as a distribution over reachable goals  $p_\gamma^\pi(\mathfrak{s}^+ = g | \mathfrak{s}_0 = s)$ . For a fixed goal  $g$ , it is always easier to stay in  $g$  than it is to reach  $g$  from  $s$ , reflected in the fact that the total mass under the orange curve is greater than the blue curve (see Lemma 4.3).

(b) Our proposed distance obeys the triangle inequality. Starting at state  $s$ , we look at the distribution over future states (■ area) and subtract off those states that the policy would reach starting from  $w$  (■ area). Our distance is defined as the difference in these areas,  $d_{\text{SD}}(s, w) \triangleq \blacksquare - \blacksquare$ . As seen in (a),  $p_\gamma^\pi(\mathfrak{s}^+ = g | \mathfrak{s}_0 = s) \leq p_\gamma^\pi(\mathfrak{s}^+ = g | \mathfrak{s}_0 = g)$ , so we can view their log ratio as a distance (see Theorem 4.4). We provide geometric intuition here. The blue area not covered by the orange corresponds to the distance. Subtracting the orange area allows distances to be added to stitch together behaviors.

Figure 4.1: An overview of our theoretical distance construction as well as the concrete implementation with metric distillation.

on NCE and infoNCE have a probabilistic interpretation, making them ripe for application to stochastic environments. Like prior work [121, 122], we account for the arrow of time [123] by using asymmetric representations, allowing the learned representations reflect the fact that (say) climbing up a mountain is more difficult from sliding back down. The representations learned by these temporal contrastive learning methods do not themselves satisfy the triangle inequality. However, we prove that a simple change of variables results in representations that do satisfy the triangle inequality. Intuitively, this change of variables corresponds to subtracting off the “distance” between a state and itself. Note that because the representations are asymmetric (see above), this extra “distance” is not zero, but rather corresponds to the likelihood of returning to the current state at some point in the future.

The main contribution is to propose a notion of temporal distance that provably satisfies the triangle inequality, even in stochastic settings. Our constructed temporal distance is easy to learn – simply take the features from (temporal) contrastive learning and perform a change of variables – no additional training required! After introducing and analyzing our proposed temporal distance, we demonstrate an application of our temporal distance to goal-conditioned reinforcement learning, using the distance function as a value function. We use a carefully controlled synthetic benchmark to test properties such as combinatorial generalization, temporal generalization, and finding shortest paths; our results here show that the proposed distance has appealing properties that prior methods lack. We also show that the RL method based on our distances can scale to 111-dimensional locomotion tasks, where it is competitive with prior methods on a parameter-adjusted basis.

#### 4.1 GENERAL DISTANCES FOR GOAL-REACHING

In this section, we introduce a novel distance metric for goal-reaching in controlled Markov processes. We show that this distance is a quasimetric, i.e., a metric that relaxes the assumption of symmetry. In the subsequent section (4.2), we show that this distance construction can enable additional generalization capabilities through a choice of model parameterization for temporal contrastive learning.

##### Preliminaries

We consider a discrete *controlled Markov process*  $M$  consisting of states  $s \in \mathcal{S}$ , actions  $a \in \mathcal{A}$ , dynamics  $P(s' | s, a)$ , initial state distribution  $p_0(\mathbf{s}_0 = s_0)$ , and a discount  $\gamma \in (0, 1)$ .

By augmenting  $M$  with the reward for any fixed goal  $g \in \mathcal{S}$ , which we define as

$$r_g(s) = (1 - \gamma) \delta_g(s),$$

where

$$\delta_g(s) = \begin{cases} 1 & \text{if } s=g \\ 0 & \text{otherwise} \end{cases}$$

is the Kronecker delta, we can extend  $M$  to a goal-dependent Markov decision process  $M_g$ . Denote by  $\Pi$  the (compact) set of stationary of policies  $\pi(a | s)$  on  $M$ . We also define  $\Pi_{\text{NM}} \supset \Pi$  to be the set of non-Markovian policies  $\pi(a_t | s_0 \dots s_t)$ . We can then derive the optimal goal-conditioned value function,

$$V_g^*(s) = \max_{\pi \in \Pi} p_\gamma^\pi(\mathbf{s}^+ = g | \mathbf{s}_0 = s), \quad (4.1)$$

where the *discounted state occupancy measure*  $p_\gamma^\pi$  is defined as the discounted distribution over future states  $\mathbf{s}^+$ ,

$$p_\gamma^\pi(\mathbf{s}^+ = s' | \mathbf{s}_0 = s) = (1 - \gamma) \sum_{k=0}^{\infty} \gamma^k p^\pi(\mathbf{s}_k = s' | \mathbf{s}_0 = s)$$

$$\text{where } p^\pi(\mathfrak{s}_{t+1}=s' | \mathfrak{s}_t = s) = \sum_{a \in \mathcal{A}} \pi(a|s)P(s'|s, a). \quad (4.2)$$

i.e., the distribution of  $\mathfrak{s}^+ \triangleq \mathfrak{s}_K$  for  $K \sim \text{Geom}(1 - \gamma)$ .

Here,  $\mathfrak{s}_t$  denotes the state at time  $t$  as a random variable, and  $\mathfrak{s}^+$  denotes the state at a geometrically distributed time in the future. When needed, under a policy  $\pi$ , we will additionally use the notation  $\mathfrak{a}_t$  and  $\mathfrak{a}^+$  to denote actions as random variables, defined analogously to  $\mathfrak{s}_t$  and  $\mathfrak{s}^+$ .

Since (4.1) is the optimal value function corresponding to the reward  $r_g$ , there will always be a stationary optimal goal-reaching policy  $\pi^g \in \Pi$  that attains the max in (4.1).

We can additionally view the setting of an *uncontrolled Markov process* (i.e., a Markov chain) as a special case of controlled Markov processes where there is a single action  $\mathcal{A} = \{a\}$  with a fixed policy  $\Pi = \{\pi\}$ .

To reason about the effects of actions, we can also consider the natural generalization of the *successor state-action distribution*, which is the distribution over future states and actions  $s', a'$  given that action  $a$  is taken in state  $s$  under  $\pi$ :

$$\begin{aligned} p_\gamma^\pi(\mathfrak{s}^+ = g, \mathfrak{a}^+ = a' | \mathfrak{s}_0 = s, \mathfrak{a}_0 = a) = \\ (1 - \gamma)\delta_{s,a}(g, a') + (1 - \gamma)\gamma \left[ \sum_{k=0}^{\infty} \sum_{s' \in \mathcal{S}} \right. \\ \left. \gamma^k p^\pi(\mathfrak{s}_k = g | \mathfrak{s}_0 = s') \pi(a' | g) P(s' | s, a) \right]. \end{aligned} \quad (4.3)$$

Finally, we recall the definition of a quasimetric space:

**Definition 4.1.** A quasimetric on  $S$  is a function  $d : S \times S \rightarrow \mathbb{R}$  satisfying the following for any  $x, y, z \in S$ .

**Positivity:**  $d(x, y) \geq 0$

**Identity:**  $d(x, y) = 0 \iff x = y$

**Triangle inequality:**  $d(x, z) \leq d(x, y) + d(y, z)$

### Our Proposed Temporal Distance

With these definitions in place, we can now define the proposed temporal distance. We will start by describing a “strawman” approach, and then proceed with the full method.

Motivated by prior work on successor representations [124] and self-predictive representations [125], a candidate temporal distance is to directly use the critic function from temporal contrastive learning. When positive examples are sampled from the discounted state occupancy measure, this critic has the following form:

$$-d(s, g) = \log \left( \frac{p_\gamma^\pi(\mathfrak{s}^+ = g | \mathfrak{s}_0 = s)}{p(g)} \right). \quad (\text{not a quasimetric})$$

However, a distance defined in this way does not satisfy the identity property of the quasimetric; namely, the distance between a state and itself can be non-zero. Our solution is to subtract off the “extra distance” between a state and itself,  $\tilde{d}(s, g) = d(s, g) - d(g, g)$ .

We now proceed with our main definition, which is a temporal distance that obeys the triangle inequality (and is a quasimetric) even in stochastic settings. We provide two definitions, one for controlled Markov processes and one for (uncontrolled) Markov processes:

**Definition 4.2.** We define the *successor distance* for a controlled Markov processes by:

$$d_{\text{SD}}(s, g) \triangleq \min_{\pi \in \Pi} \log \left( \frac{p_{\gamma}^{\pi}(\mathfrak{s}^+ = g | \mathfrak{s}_0 = g)}{p_{\gamma}^{\pi}(\mathfrak{s}^+ = g | \mathfrak{s}_0 = s)} \right), \quad (4.4)$$

As a special case for an uncontrolled Markov process, we can define:

$$d_{\text{SD}}(s, g) \triangleq \log \left( \frac{p_{\gamma}^{\pi}(\mathfrak{s}^+ = g | \mathfrak{s}_0 = g)}{p_{\gamma}^{\pi}(\mathfrak{s}^+ = g | \mathfrak{s}_0 = s)} \right). \quad (4.5)$$

To use these distances for control in model-free settings, we can extend this notion to include actions, yielding a distance over  $\mathcal{S} \times \mathcal{A}$ .

**Definition 4.3.** We define the *successor distance with actions* for a controlled Markov process by:

$$d_{\text{SD}}((s, a), (g, a')) \triangleq \min_{\pi \in \Pi} \left( \log \frac{p_{\gamma}^{\pi}(\mathfrak{s}^+ = g, \mathfrak{a}^+ = a' | \mathfrak{s}_0 = g, \mathfrak{a}_0 = a')}{p_{\gamma}^{\pi}(\mathfrak{s}^+ = g, \mathfrak{a}^+ = a' | \mathfrak{s}_0 = s, \mathfrak{a}_0 = a)} \right). \quad (4.6)$$

We make two brief lemmas about this definition; the proofs can be found in Section 4.6. Within  $\mathcal{S} \times \mathcal{A}$ , we can also say:

**Lemma 4.1.**  $d_{\text{SD}}((s, a), (s', a'))$  is independent of  $a'$  when  $s \neq s'$ .

In light of this independence, we denote  $d_{\text{SD}}(s, a, s') \triangleq d_{\text{SD}}((s, a), (s', a'))$  where applicable. Selecting actions that minimizes this distance corresponds to policy improvement:

**Lemma 4.2.** Selecting actions to minimize the successor distance is equivalent to selecting actions to maximize the (scaled and shifted) Q-function:

$$\begin{aligned} -d_{\text{SD}}(s, a, g) &= \frac{1}{p_g(g)} Q(s, a, g) + c_{\psi}(g) \\ \implies \arg \max_a d_{\text{SD}}(s, a, g) &= \arg \max_a Q(s, a, g). \end{aligned}$$

**Geometric interpretation.** Before proceeding to prove that this distance construction obeys the triangle inequality and the other quasimetric properties (Section 4.1), we provide intuition for this distance. We visualize this distance construction in Fig. 4.1 (b). The distribution over states visited starting at  $s$  ( $p_\gamma^\pi(\mathfrak{s}^+ = w | \mathfrak{s}_0 = s)$ ) is shown as the **teal region**; while states visited starting at  $w$  ( $p_\gamma^\pi(\mathfrak{s}^+ = w | \mathfrak{s}_0 = w)$ ) is shown as the **orange region**. Our proposed distance metric is the difference in the areas of these regions ( $\blacksquare - \blacksquare$ ). The theoretical results in the next section prove that this difference is always non-negative. Zooming out to look at the  $s$ ,  $w$ , and  $g$  together, we see that these set differences obey the triangle inequality – the area between  $s$  and  $g$  is smaller than the areas between  $s$  and  $w$  and between  $w$  and  $g$ . Concrete examples to build intuition for these definitions and results are presented in Section 4.8.

**Hitting times as a special case.** To provide additional intuition into our construction, we consider a special case; the subsequent section shows that the proposed distance is a valid quasimetric in much broader settings. In this special case, consider a controlled Markov process where the agent can remain at a state indefinitely. This assumption means that the  $p_\gamma^\pi(\mathfrak{s}^+ = g | \mathfrak{s}_0 = g) = 1$ , so the proposed distance metric can be simplified to  $d(s, g) = -\log p_\gamma^\pi(\mathfrak{s}^+ = g | \mathfrak{s}_0 = s)$ . This assumption also means that the hitting time of  $g$  from  $s$  has a deterministic value, which we will call  $H(s, g)$ . Thus, we can write the discounted state occupancy measure as  $p_\gamma^\pi(\mathfrak{s}^+ = g | \mathfrak{s}_0 = s) = \gamma^{H(s, g)}$ , so the proposed distance metric is equivalent to the hitting time:  $d(s, g) = H(s, g)$ . Importantly, and unlike prior work, our proposed distance continues to be a quasimetric outside of this special case, as we prove in the following section.

Theoretical results

Before proving this distance is a quasimetric over  $\mathcal{S}$ , we provide a helper lemma relating the difficulty of reaching a goal through a waypoint to the difficulty without the waypoint. The key insight we use here is that the notion of a hitting time can be generalized to represent distances in terms of discounted state occupancies.

**Lemma 4.3.** *For any  $s, w, g \in \mathcal{S}$ ,  $\pi \in \Pi$ ,*

$$\begin{aligned} \max_{\pi' \in \Pi} \left[ \frac{p_\gamma^\pi[\pi'](\mathfrak{s}^+ = g | \mathfrak{s}_0 = w) p_\gamma^\pi[\pi](\mathfrak{s}^+ = w | \mathfrak{s}_0 = s)}{p_\gamma^\pi[\pi](\mathfrak{s}^+ = w | \mathfrak{s}_0 = w)} \right] \\ \leq \max_{\pi' \in \Pi} p_\gamma^\pi[\pi'](\mathfrak{s}^+ = g | \mathfrak{s}_0 = s). \end{aligned}$$

The proof is in Section 4.6. This lemma is the key to proving our main result:

**Theorem 4.4** (Quasimetric).  $d_{\text{SD}}$  is a quasimetric over  $\mathcal{S}$ , satisfying the triangle inequality and other properties from Definition 4.1.

The proof is in Section 4.7. Compared with prior work [126], our result extends to stochastic settings; we will empirically compare to this and other prior methods in Section 4.5.

To make this result applicable to settings with unknown dynamics or without actions, we note the following corollaries:

**Corollary 4.4.1.**  $d_{\text{SD}}$  is a quasimetric over  $\mathcal{S} \times \mathcal{A}$ .

**Corollary 4.4.2.**  $d_{\text{SD}}$  is a quasimetric over an uncontrolled Markov process as in Eq. (4.5).

See Section 4.7 for discussion of these results.

## 4.2 USING OUR TEMPORAL DISTANCE FOR RL

In this section we describe an application of our proposed temporal distance to goal-conditioned reinforcement learning. The main challenge in doing this will be (1) estimating the successor distance defined in Eq. (4.4), and (2) doing so with an architecture that respects the quasimetric properties. Once learned, we will use the successor distance as a value function for training a policy.

To introduce our methods, Section 4.2 will first discuss how contrastive learning *almost* estimates the successor distance. We will then introduce two variants of our method, Contrastive Metric Distillation (CMD). The first method (CMD 1-step, Section 4.3) will acquire the successor distance by applying contrastive learning with an energy function that is the difference of two other functions. The second method (CMD 2-step, Section 4.4) will acquire the successor distance by taking the features from contrastive learning and distilling those features into a quasimetric architecture. In both cases, we then use the learned successor distance to train a goal-conditioned policy.

We emphasize that the key contribution here is the mathematical construct of *what* constitutes a temporal distance, not that we use a certain architecture to represent this temporal distance. Practically, we will use the Metric Residual Network (MRN) architecture [127] in our implementation. Pseudocode for the full algorithms (both one-step and two-step) is provided in Algorithms 2 and 3. We highlight the differences between the two methods in orange for clarity.

Building block: contrastive learning

Both of our proposed methods will use contrastive learning as a core primitive, so we start by discussing how we use contrastive learning to learn an energy

**Algorithm 2:** 1-step Contrastive Metric Distillation (CMD-1)

- 
- 1: **input:** batch size  $B$ , number of iterations  $T$
  - 2: **initialize potential**  $\psi$ , **quasimetric**  $\phi$ , and **policy**  $\mu$  parameters
  - 3: **define**  $f_\theta(s, a, g) \triangleq c_\psi(g) - d_\phi(s, a, g)$
  - 4: **for**  $t = 1 \dots T$  **do**
  - 5:     sample  $\{(s_i, a_i) \sim p_s\}_{i=1}^B$
  - 6:     sample  $\{(g_i, a'_i) \sim p_\gamma^\pi(\mathfrak{s}^+ = g_i | \mathfrak{s}_0 = s_i, a_i)\}_{i=1}^B$
  - 7:      $\phi \leftarrow \phi - \alpha \nabla_\phi [\mathcal{L}_{\phi, \psi}^c(\{s_i, a_i\}, \{g_i\})]$  (4.8, 4.12)
  - 8:      $\psi \leftarrow \psi - \alpha \nabla_\psi [\mathcal{L}_{\phi, \psi}^c(\{s_i, a_i\}, \{g_i\})]$  (4.8, 4.12)
  - 9:      $\mu \leftarrow \mu - \alpha \nabla_\mu [\mathcal{L}_\mu^\pi(\{s_i, a_i\}, \{g_i, a'_i\})]$  (4.20)
  - 10: **output**  $\pi_\mu$
- 

**Algorithm 3:** 2-step Contrastive Metric Distillation (CMD-2)

- 
- 1: **input:** : batch size  $B$ , number of iterations  $T$
  - 2: **initialize representations**  $\phi, \psi$ , and **policy parameters**  $\mu$
  - 3: **initialize quasimetric**  $\hat{\theta}$ , **margin**  $\lambda$
  - 4: **define**  $f_\theta(s, a, g) \triangleq \phi(s, a)^T \psi(g)$
  - 5: **for**  $t = 1 \dots T$  **do**
  - 6:     sample  $\{(s_i, a_i) \sim p_s\}_{i=1}^B$
  - 7:     sample  $\{(g_i, a'_i) \sim p_\gamma^\pi(\mathfrak{s}^+ = g_i | \mathfrak{s}_0 = s_i, a_i)\}_{i=1}^B$
  - 8:      $\phi \leftarrow \phi - \alpha \nabla_\phi [\mathcal{L}_{\phi, \psi}^c(\{s_i, a_i\}, \{g_i\})]$  (4.8, 4.16)
  - 9:      $\psi \leftarrow \psi - \alpha \nabla_\psi [\mathcal{L}_{\phi, \psi}^c(\{s_i, a_i\}, \{g_i\})]$  (4.8, 4.16)
  - 10:      $\mu \leftarrow \mu - \alpha \nabla_\mu [\mathcal{L}_\mu^\pi(\{s_i, a_i\}, \{g_i, a'_i\})]$  (4.20)
  - 11:      $\hat{\theta} \leftarrow \theta - \alpha \nabla_{\hat{\theta}} [\mathcal{L}_{\hat{\theta}, \phi, \psi}^d(\{s_i, a_i\}, \{g_i, a'_i\})]$  (4.17)
  - 12:      $\lambda \leftarrow \lambda + \alpha (\mathcal{C}_{\hat{\theta}}(\{s_i, a_i\}, \{g_i, a'_i\}) - \varepsilon^2)$  (4.17)
  - 13: **output**  $\pi_\mu$
- 

function  $f_\theta(s, a, g)$ , and the relationship between that energy function and the desired successor distance.

Following prior work [121], we will apply contrastive learning to learn an energy function  $f_\theta(s, a, g)$  that assigns high scores to  $(s, a, g)$  triplets from the same trajectory, and low scores to triplets where the goal  $g$  is unlikely to be visited at some point after the state-action  $(s, a)$  pair. Let  $p_{sa}(s, a)$  be a marginal distribution over state-action pairs, and let  $p_g(g) = \sum_{s \in \mathcal{S}} p_s(s) p_\gamma^\pi(\mathfrak{s}^+ = g | \mathfrak{s}_0 = s)$  be the corresponding marginal distribution over future states. Contrastive learning learns the energy function by sampling pairs of state-action  $(s, a)$  and goals  $g$  from the joint distribution  $s_i, a_i, g_i \sim p_\gamma^\pi(\mathfrak{s}^+ = g_i | \mathfrak{s}_0 = s_i, a_i) p_{sa}(s_i, a_i)$ . We will use the symmetrized infoNCE loss function (without resubstitution) [53, 128, 122], which provides the



following objective:

$$\min_{\theta} \mathbb{E}_{\{s_i, a_i, g_i\}_{i=1}^B} \mathcal{L}_{\theta}^c(\{s_i, a_i\}, \{g_i\}). \quad (4.7)$$

given the forward and backward classification losses:

$$\begin{aligned} \mathcal{L}_{\theta}^c &= \mathcal{L}_{\theta}^{\text{fwd}} + \mathcal{L}_{\theta}^{\text{bwd}} \quad (4.8) \\ \mathcal{L}_{\theta}^{\text{fwd}}(\{s_i, a_i\}, \{g_i\}) &= \sum_{i=1}^B \log \left( \frac{e^{f_{\theta}(s_i, a_i, g_i)}}{\sum_{j=1}^B e^{f_{\theta}(s_i, a_i, g_j)}} \right) \\ \mathcal{L}_{\theta}^{\text{bwd}}(\{s_i, a_i\}, \{g_i\}) &= \sum_{i=1}^B \log \left( \frac{e^{f_{\theta}(s_i, a_i, g_i)}}{\sum_{j=1}^B e^{f_{\theta}(s_j, a_i, g_i)}} \right). \end{aligned}$$

We highlight the indices  $i$  and  $j$  for clarity. As the batch size  $B$  becomes large, the optimal critic parameters  $\theta^*$  then satisfy [52, 129]

$$f_{\theta^*}(s, a, g) = \log \left( \frac{p_{\gamma}^{\pi}(\mathfrak{s}^+ = g | \mathfrak{s}_0 = s, a)}{C \cdot p_g(g)} \right), \quad (4.9)$$

where  $C$  is a free parameter. Finally, note that we can represent the successor distance (4.4) as the *difference* of this optimal critic evaluated on two different inputs:

$$f_{\theta^*}(g, a, g) - f_{\theta^*}(s, a, g) = \frac{p_{\gamma}^{\pi}(\mathfrak{s}^+ = g | \mathfrak{s}_0 = g, a)}{p_{\gamma}^{\pi}(\mathfrak{s}^+ = g | \mathfrak{s}_0 = s, a)}. \quad (4.10)$$

The next two section present practical methods for representing this difference, either via (1) a special parametrization of this critic (Section 4.3) or (2) distillation (Section 4.4).

### 4.3 ONE-STEP METRIC DISTILLATION (CMD-1):

In this section, we describe how to *directly* learn the successor distance using an architecture that is guaranteed to satisfy the triangle inequality and other quasimetric properties.

The key idea is to apply the contrastive learning discussed in the prior section to a particular parametrization of the energy function, so that the difference in Eq. (4.10) is represented as a single quasimetric network. We start by noting that the function learned by contrastive learning (Eq. (4.9)) can be decomposed into the successor distance plus an additional function that depends only on the future state  $g$ :

$$f_{\theta^*}(s, a, g) = \log \left( \frac{p_{\gamma}^{\pi}(\mathfrak{s}^+ = g | \mathfrak{s}_0 = s, a)}{C \cdot p_g(g)} \right) \quad (4.11)$$



$$= \underbrace{\log\left(\frac{p_\gamma^\pi(\mathfrak{s}^+ = g | \mathfrak{s}_0 = s, a)}{p_\gamma^\pi(\mathfrak{s}^+ = g | \mathfrak{s}_0 = g)}\right)}_{-d_\phi(s, a, g)} - \underbrace{\log\left(\frac{p_\gamma^\pi(\mathfrak{s}^+ = g | \mathfrak{s}_0 = g)}{C \cdot p_g(g)}\right)}_{-c_\psi(g)}.$$

Thus, we will apply the contrastive objective from Eq. (4.8) to an energy function  $f_{\theta=(\phi, \psi)}(s, a, g)$  parametrized as the difference of a quasimetric network  $d_\phi(s, a, g)$  an another learned function  $c_\psi : \mathbb{R} \rightarrow \mathbb{R}$ :

$$f_{\phi, \psi}(s, a, g) = c_\psi(g) - d_\phi(s, a, g). \quad (4.12)$$

The term  $c_\psi(g)$  is important for allowing  $f_\theta(s, a, g)$  to represent positive numbers, as  $-d_\phi(s, a, g)$  is non-positive because it is a quasimetric network. With this parametrization, we can use Eq. (4.10) to obtain the successor distance as

$$\begin{aligned} f_{\theta^*}(g, a, g) - f_{\theta^*}(s, a, g) \\ = \cancel{-d_\phi(g, a, g)} + \overset{0}{\cancel{c_\psi(g)}} + d_\phi(s, a, g) - \cancel{c_\psi(g)}. \end{aligned} \quad (4.13)$$

After contrastive learning, we will discard  $c_\psi(g)$  and use  $d_\phi(s, a, g)$  as our successor distance. We conclude by providing the formal result that this approach recovers the successor distance:

**Lemma 4.5.** *For  $s \neq g$ , the unique solution to the the loss function in Eq. (4.8) with the parametrization in Eq. (4.12) is*

$$d_{\phi^*}(s, a, g) = \log \frac{p_\gamma^\pi(\mathfrak{s}^+ = g | \mathfrak{s}_0 = s, \mathfrak{a}_0 = a)}{p_\gamma^\pi(\mathfrak{s}^+ = g | \mathfrak{s}_0 = g)}. \quad (4.14)$$

See Section 4.7 for the proof.

One appealing aspect of this approach is that it only involves one learning step. The next section provides an alternative approach that proceeds in two steps.

#### 4.4 TWO-STEP METRIC DISTILLATION (CMD-2)

In this section we present an alternative approach to estimating the successor distance with a quasimetric network. While the first approach (CMD 1-step) is appealing because of its simplicity, this approach may be appealing in settings where pre-trained contrastive features are already available, but users want to boost performance by capitalizing on the inductive biases of quasimetric networks.

The key idea behind our approach is that that optimal critic from contrastive learning (Eq. (4.9)) can be used to estimate the successor distance by performing a change of variables:

$$f_\theta(g, a, g) - f_\theta(s, a, g)$$

$$\begin{aligned}
&= \log \frac{p_\gamma^\pi(s^+ = g | s_0 = g, a)}{C p_g(g)} - \log \frac{p_\gamma^\pi(s^+ = g | s_0 = s, a)}{C p_g(g)} \\
&= \log \frac{p_\gamma^\pi(s^+ = g | s_0 = g, a)}{p_\gamma^\pi(s^+ = g | s_0 = s, a)}. \tag{4.15}
\end{aligned}$$

This final expression is the successor distance; Section 4.9 will discuss why the action  $a$  in the numerator can be ignored. Because the successor distance obeys the triangle inequality (and the other quasimetric properties), we will distill this difference into a quasimetric network. We will call this method CMD 2-Step.

### Distilling to a quasimetric architecture

The representations in Eq. (4.15) already form a quasimetric on  $S \times A$ , and could directly be used for action selection. However, because we know that these representations satisfy the triangle inequality, distilling them into a network that is architecturally-constrained to obey the triangle inequality serves as a very strong prior: a way of potentially combating overfitting and improving generalization. To do this, we distill the bound into a distance  $d_\phi$  parameterized by an MRN quasimetric [127].

CMD 2-Step works by applying contrastive learning (Eq. 4.8). Following prior work [121], we will parametrize the energy function as the inner product between learned representations:  $f_{\phi, \psi}(s, a, g) = \phi(s, a)^T \psi(g)$ . The critic parameters are thus  $\theta = (\phi, \psi)$ . We then distill the quasimetric architecture using Eq. (4.15) as a constraint. We enforce the constraint with a Lagrange multiplier  $\lambda$  to ensure that the margin  $C_{\hat{\theta}}(\{s_i, a_i\}, \{g_i, a'_i\})$  for Eq. (4.15) satisfies  $C_{\hat{\theta}}(\{s_i, a_i\}, \{g_i, a'_i\}) \leq \varepsilon^2$  on pairs of states and future goals sampled from the data:

$$\begin{aligned}
&C_{\hat{\theta}}(\{s_i, a_i\}, \{g_i, a'_i\}) \\
&\triangleq \sum_{i,j=1}^B \max(0, d_{\hat{\theta}}((s_i, a_i), (g_i, a'_i)) - f_{\phi, \psi}(s_i, a_i, g_i))^2 \\
&\text{where } f_{\phi, \psi}(s, a, g) \triangleq (\phi(g, a) - \phi(s, a))^T \psi(g). \tag{4.16}
\end{aligned}$$

When distilling a distance  $d_{SD}$ , subject to the constraint above, we want to be maximally conservative in determining which goals we can reach. We assume Eq. (4.15) as a prior, and use dual descent to perform a constrained minimization of the objective

$$\begin{aligned}
&\mathcal{L}_{\hat{\theta}, \phi, \psi}^d(\{s_i, a_i\}, \{g_i, a'_i\}) \triangleq \\
&\sum_{i,j=1}^B \max(0, f_{\phi, \psi}(s_i, a_i, g_j) - d_{\hat{\theta}}(s_i, a_i, g_j))^2, \tag{4.17}
\end{aligned}$$

yielding an overall optimization

$$\min_{\hat{\theta}} \max_{\lambda \geq 0} \sum_{\{s_i, a_i, g_i, a'_i\}_{i=1}^B} \left[ \mathcal{L}_{\hat{\theta}, \phi, \psi}^d(\{s_i, a_i\}, \{g_i, a'_i\}) + \lambda (\mathcal{C}_{\hat{\theta}}(\{s_i, a_i\}, \{g_i, a'_i\}) - \varepsilon^2) \right]. \quad (4.18)$$

Parameterizing the quasimetric

For both methods in Sections 4.3 and 4.4, we learn a distance  $d_\theta : (\mathcal{S} \times \mathcal{A})^2 \rightarrow \mathbb{R}$  parametrized with the Metric Residual Network (MRN) architecture [127]. We apply the square root correction noted by Wang and Isola [130, Appendix C.2] to ensure that the distance satisfies the triangle inequality. This parameterization can be expressed using learned representations  $h_\theta, g_\theta : \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}^d$ :

$$d_\theta(x, y) = \Delta(h_\theta(x) - h_\theta(y)) + \|g_\theta(x) - g_\theta(y)\|$$

where  $\Delta(x) = \max_{i=1}^d [\max(0, x_i)]$ . (4.19)

Policy extraction

Once we extract distance  $d_{\text{SD}}$ , we learn a goal-conditioned policy  $\pi_\mu$  to select actions that minimize the distance successor between states and random goals [50]:

$$\min_{\mu} \mathbb{E}_{p_s(s) p_g(g, a') \pi_\mu(\hat{a} | s, g)} [\mathcal{L}_\mu^\pi(\{s_i, \hat{a}_i\}, \{g_i, a'_i\})] \quad (4.20)$$

To prevent the policy from sampling out-of-distribution actions for offline RL [131–133], we adopt another goal-conditioned behavioral cloning regularization from Zheng et al. [134] or use advantage weighted regression [135].

With the behavior cloning regularization, the policy extraction loss becomes:

$$\mathcal{L}_\mu^\pi(\{s_i, a_i\}, \{g_i, a'_i\}) = \sum_{i,j=1}^B \mathbb{E}_{\hat{a} \sim \pi_\mu(\hat{a} | s_i, g_j)} [d_\phi((s_i, \hat{a}), (g_j, a'_j)) + \log \pi_\mu(a_i | s_i, g_i)]. \quad (4.21)$$

## 4.5 EXPERIMENTS

Our experiments study a synthetic 2D navigation task to see whether our proposed temporal distance can learn meaningful distances of pairs of states unseen together during training (i.e., *combinatorial generalization*). We also study the efficacy of extracting policies from this learned distance function, both in this 2D navigation setting and in a 29-dim robotic locomotion problem from the AntMaze benchmark suite. As discussed below, for the latter experiment our comparison will be restricted to small neural network sizes. Additional implementation details are provided in Appendix C.3.

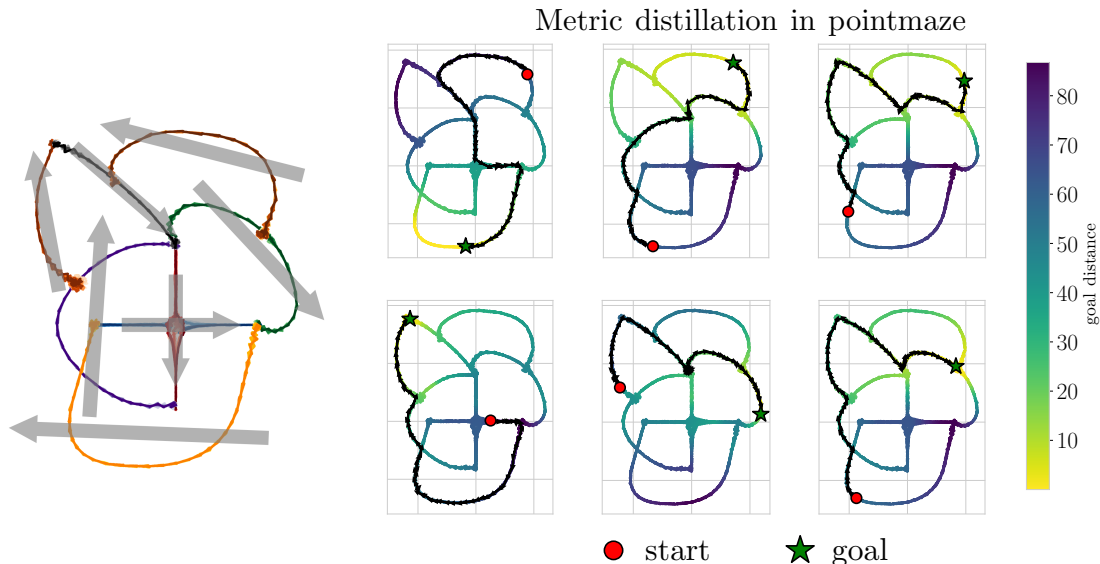


Figure 4.2: (Left) We collect four types of trajectories on this 2D navigation task. The large gray arrows depict the direction of motion. Note that navigating between certain states requires piecing together trajectories of different colors. (Right) Our proposed temporal distance correctly pieces together trajectories, allowing an RL agent to successfully navigate between pairs of states that never occur on the same trajectory. This combinatorial generalization [1] or “stitching” [2] property is typically associated with bootstrapping with temporal difference learning, which our temporal distances do not require.

Table 4.1: Offline RL benchmarks

	CMD 1-step	CMD 2-step	QRL	CRL (CPC)	GCBC	IQL <sup>1</sup>
umaze	90.3 ± 4.2	<b>97.0 ± 0.4</b>	76.8 ± 2.3	79.8 ± 1.6	65.4 ± 87.5	87.5
umaze-diverse	<b>90.3 ± 4.6</b>	<b>90.5 ± 1.4</b>	80.1 ± 1.3	77.6 ± 2.8	60.9 ± 62.2	62.2
medium-play	<b>78.0 ± 4.0</b>	72.3 ± 2.6	<b>76.5 ± 2.1</b>	72.6 ± 2.9	58.1 ± 71.2	71.2
medium-diverse	<b>83.0 ± 3.1</b>	71.8 ± 1.0	73.4 ± 1.9	71.5 ± 1.3	67.3 ± 70.0	70.0
large-play	<b>68.0 ± 2.1</b>	59.2 ± 1.8	52.9 ± 2.8	48.6 ± 4.4	32.4 ± 39.6	39.6
large-diverse	<b>74.5 ± 2.3</b>	63.6 ± 1.9	51.5 ± 3.8	54.1 ± 5.5	36.9 ± 47.5	47.5

### Controlled experiments on synthetic data

We first present results in a simple 2D navigation environment to illustrate how our approach can recombine pieces of data to navigate between pairs of states unseen together during training (i.e., combinatorial generalization).

We start by collecting four types of trajectories, identified in Fig. 4.2 (left). We will be primarily interested in what distances our method assigns to pairs of states that occur on different types of trajectories. Our hypothesis is that, by virtue of the

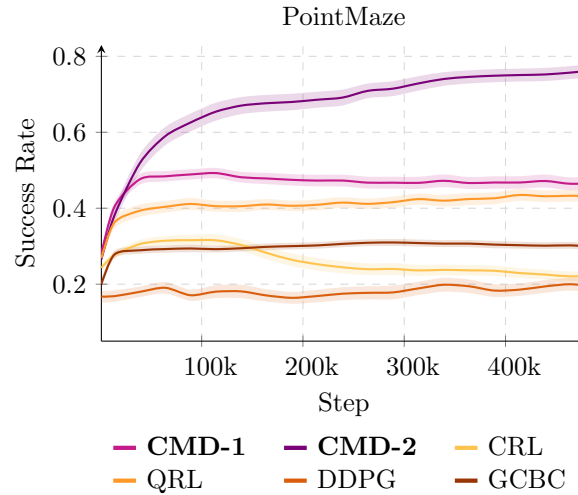


Figure 4.3: Metric distillation enables more efficient offline training and long-horizon compositional generalization. Results are plotted with one standard error.

triangle inequality, our method will correctly reason about *global distances*, despite only being trained on *locally* on individual trajectories. Note that the collected data is directed, so we will also be test whether our learned distance obeys the arrow of time.

**Visualizing the paths.** Using these data, we learn the contrastive representations and distill them into a quasimetric architecture, as described in Section 4.2. In the subfigures in Fig. 4.2 (right), we visualize these distances using the colormap, with the goal set to the state identified with the  $\star$ . This figure also visualizes paths created using the learned distances. Starting at the state identified as  $\bullet$ , greedily select a next state within an L2 ball that has minimal temporal distance to the goal. We repeat this process until arriving at the goal. These planned paths demonstrate that the learned temporal distances perform combinatorial generalization; each of the subfigures in Fig. 4.2 show examples of inferred paths that require correctly assigning distances to pairs of states that were unseen together during training. Note, too, that these paths follow the arrow of time: the small arrows depicting the paths go in the same direction that the data was collected (large gray arrows in the left subplot).

**Control performance.** We next study whether these learned distances can be used for control, using the same synthetic dataset as above. We will compare with four baselines. **DDPG** learns distances using Q-learning with a reward that is  $-1$  at every transition until the goal is reached [136, 137, 49]; at least in deterministic settings, these distances should correspond to hitting times. Quasimetric RL [126] is an extension of this baseline that uses a quasimetric architecture to represent these

distances. Contrastive RL [121] estimates distances directly using the contrastive features (the same as used for our method), but without the metric distillation step. For all these methods as well as our method, a policy is learned using advantage-weighted maximum likelihood [138, 139]. We also compare with a behavioral cloning baseline, which predicts the action that was most likely to occur in the dataset conditioned on state and goal.

We measure performance by evaluating the success rate of each these approaches at reaching randomly sampled goals. In Fig. 4.3, we plot this success rate over the course of training. Note that this experiment is done in the offline setting, so the X axis corresponds to the number of gradient steps. We observe that our temporal distance can successfully navigate to approximately 80% of goals, while the best prior method has a success rate of around 50%. Because our method starts with the same contrastive features as the contrastive RL baseline, the better performance of ours highlights the importance of the quasimetric architecture (i.e., of imposing the triangle inequality as an inductive bias). While both our method and quasimetric RL use a quasimetric architecture to represent a distance, we aim to represent the proposed distance metric from Section 4.2 while quasimetric RL aims to represent a hitting time; the better performance of our method highlights the need to use a temporal distance that is well defined in stochastic settings such as this.

### Scaling to higher-dimensional tasks

To study whether our temporal distance learning approach is applicable to higher-dimensional tasks, we apply it to a 111-dimensional robotic control task (AntMaze [2]). In this problem setting we additionally condition the temporal distance on the action and use the learned distance as a value function for selecting actions.

We compare our approach to three competitive baselines. **GCBC** is a conditional imitation learning method that learns a goal-conditioned policy directly, without a value function or distance function [140, 51, 141, 15]. Both our method and Contrastive RL (**CRL**) [121] learn representations in the same way (Section 4.2); the difference is that our method additionally distills these representations into a quasimetric architecture. Thus, comparing our method to CRL tests the importance of the triangle inequality as an inductive bias. We consider two variants of CRL using either rank-based NCE [53, 134] or binary-NCE [142], namely CRL (CPC) and CRL (NCE). Finally, Quasimetric RL (**QRL**) [143] represents a different type of temporal distance with the same quasimetric architecture as our method; it is unclear whether the temporal distance from QRL obeys the triangle inequality in stochastic settings. Thus, comparing our method to QRL tests the importance of using a temporal distance that is well defined in stochastic settings. Prior work [134] has shown that these baselines are more competitive than other recent alternatives, including IQL [144] with HER [14] and decision transformer [141].

Comparisons across random seeds are shown in Table 4.1.

## 4.6 HITTING TIMES

In this section, we show several lemmas relating the discounted state occupancy measure (defined in Eqs. (4.2) and (4.3)) to the hitting times of states and goals. We start by defining a notion of hitting time:

**Definition 4.4.** For  $\pi \in \Pi$  and  $s, g \in \mathcal{S}$ , define the random variable  $H_s^\pi(g)$  by

$$H_s^\pi(g) = \min\{t \geq 0 : E_t\} \quad (4.22)$$

where  $E_t$  is the event that  $\mathfrak{s}_t = g$  given  $\mathfrak{s}_0 = s$ .

In other words,  $H_s^\pi(g)$  is the smallest  $t$  such that  $\mathfrak{s}_t = g$  starting in  $\mathfrak{s}_0 = s$ , i.e., the hitting time of  $g$ .

Now, we can relate the discounted state occupancy measure to the hitting time of a goal.

**Lemma 4.6.** For  $H_s^\pi(g)$  defined as (4.22),

$$p_\gamma^\pi(\mathfrak{s}^+ = g \mid \mathfrak{s}_0 = s) = \mathbb{E}[\gamma^{H_s^\pi(g)}] p_\gamma^\pi(\mathfrak{s}^+ = g \mid \mathfrak{s}_0 = g).$$

*Proof.* Let  $p^\pi(\mathfrak{s}_t = g \mid \mathfrak{s}_0 = s, H_s^\pi(g) = h)$  be the probability of reaching goal  $g$  at time step  $t$  when starting at state  $s$  given hitting time  $H_s^\pi(g) = h$ . By the definition of  $H_s^\pi(g)$ , we have

$$p^\pi(\mathfrak{s}_t = g \mid \mathfrak{s}_0 = s, H_s^\pi(g) = h) = \begin{cases} 0 & t < h \\ p^\pi(\mathfrak{s}_t = g \mid \mathfrak{s}_h = g) & t \geq h \end{cases} \quad (4.23)$$

Thus,

$$\begin{aligned} p_\gamma^\pi(\mathfrak{s}^+ = g \mid \mathfrak{s}_0 = s) &= (1 - \gamma) \sum_{t=0}^{\infty} \gamma^t p^\pi(\mathfrak{s}_t = g \mid \mathfrak{s}_0 = s) \\ &= (1 - \gamma) \sum_{t=0}^{\infty} \sum_{h=0}^{\infty} \gamma^t p^\pi(\mathfrak{s}_t = g, H_s^\pi(g) = h \mid \mathfrak{s}_0 = s) \\ &= \sum_{h=0}^{\infty} p(H_s^\pi(g) = h) \left( (1 - \gamma) \sum_{t=0}^{\infty} \gamma^t p^\pi(\mathfrak{s}_t = g \mid \mathfrak{s}_0 = s, H_s^\pi(g) = h) \right) \\ &= \sum_{h=0}^{\infty} p(H_s^\pi(g) = h) \left( (1 - \gamma) \sum_{t=h}^{\infty} \gamma^t p^\pi(\mathfrak{s}_t = g \mid \mathfrak{s}_h = g) \right) \end{aligned}$$

(Plug in Eq. 4.23)

<sup>1</sup>IQL results are taken from Kostrikov et al. [144] which does not report standard errors.

$$\begin{aligned}
&= \sum_{h=0}^{\infty} \gamma^h p(H_s^\pi(g) = h) \left( (1 - \gamma) \sum_{t=h}^{\infty} \gamma^{t-h} p^\pi(\mathfrak{s}_{t-h} = g \mid \mathfrak{s}_0 = g) \right) \\
&\hspace{15em} \text{(Stationary property of MDP)} \\
&= \sum_{h=0}^{\infty} \gamma^h p(H_s^\pi(g) = h) \left( (1 - \gamma) \sum_{t=0}^{\infty} \gamma^t p^\pi(\mathfrak{s}_t = g \mid \mathfrak{s}_0 = g) \right) \\
&\hspace{15em} \text{(Change of variables)} \\
&= \mathbb{E}[\gamma^{H_s^\pi(g)}] p_\gamma^\pi(\mathfrak{s}^+ = g \mid \mathfrak{s}_0 = g),
\end{aligned}$$

as desired.  $\square$

We can generalize this result to account for actions as well.

**Definition 4.5.** For  $\pi \in \Pi$ ,  $s, g \in \mathcal{S}$ , and  $a, a' \in \mathcal{A}$ , we define the following additional hitting time random variables

$$H_{s,a}^\pi(g, a') = \min\{t \geq 0 : E_t\} \quad (4.24)$$

where  $E_t$  is the event that  $\mathfrak{s}_t = g, \mathfrak{a}_t = a'$  given  $\mathfrak{s}_0 = s, \mathfrak{a}_0 = a$

$$H_{s,a}^\pi(g) = \min\{t \geq 0 : E_t\} \quad (4.25)$$

where  $E_t$  is the event that  $\mathfrak{s}_t = g$  given  $\mathfrak{s}_0 = s, \mathfrak{a}_0 = a$ .

We now show an analogous result for the discounted state-action occupancy measure.

**Lemma 4.7.** For  $H_{s,a}^\pi(g, a)$  defined as (4.24) and  $s \neq g$ ,

$$p_\gamma^\pi(\mathfrak{s}^+ = g, \mathfrak{a}^+ = a' \mid \mathfrak{s}_0 = s, \mathfrak{a}_0 = a) = \mathbb{E}[\gamma^{H_{s,a}^\pi(g, a')}] p_\gamma^\pi(\mathfrak{s}^+ = g, \mathfrak{a}^+ = a' \mid \mathfrak{s}_0 = g, \mathfrak{a}_0 = a').$$

*Proof.* Let  $p^\pi(\mathfrak{s}_t = g, \mathfrak{a}_t = a' \mid \mathfrak{s}_0 = s, \mathfrak{a}_0 = a, H_{s,a}^\pi(g, a') = h)$  be the probability of reaching goal  $g$  at time step  $t$  then taking action  $a'$ , when starting at state  $s$  given the hitting time  $H_{s,a}^\pi(g) = h$  and  $\pi$  takes action  $a'$  at time  $h$ . By the definition of  $H_{s,a}^\pi(g)$ , we have

$$\begin{aligned}
&p^\pi(\mathfrak{s}_t = g, \mathfrak{a}_t = a' \mid \mathfrak{s}_0 = s, \mathfrak{a}_0 = a, H_{s,a}^\pi(g, a') = h) \\
&= \begin{cases} 0 & t < h \\ 1 & t = h \\ \pi(a' \mid g) p^\pi(\mathfrak{s}_t = g \mid \mathfrak{s}_h = g, \mathfrak{a}_h = a') & t > h. \end{cases} \quad (4.26)
\end{aligned}$$

Thus,



$$\begin{aligned}
& p_\gamma^\pi(\mathfrak{s}^+ = g, \mathfrak{a}^+ = a' \mid \mathfrak{s}_0 = s, \mathfrak{a}_0 = a) \\
&= (1 - \gamma) \sum_{t=0}^{\infty} \gamma^t p^\pi(\mathfrak{s}_t = g, \mathfrak{a}_t = a' \mid \mathfrak{s}_0 = s, \mathfrak{a}_0 = a) \\
&= (1 - \gamma) \sum_{t=0}^{\infty} \sum_{h=0}^{\infty} \gamma^t p^\pi(\mathfrak{s}_t = g, \mathfrak{a}_t = a', H_{s,a}^\pi(g, a') = h \mid \mathfrak{s}_0 = s, \mathfrak{a}_0 = a) \\
&= \sum_{h=0}^{\infty} p(H_{s,a}^\pi(g, a') = h) \left( (1 - \gamma) \sum_{t=0}^{\infty} \gamma^t p^\pi(\mathfrak{s}_t = g, \mathfrak{a}_t = a' \mid \mathfrak{s}_0 = s, \mathfrak{a}_0 = a, H_{s,a}^\pi(g, a') = h) \right) \\
&= \sum_{h=0}^{\infty} p(H_{s,a}^\pi(g, a') = h) (1 - \gamma) \left( 1 + \sum_{t=h+1}^{\infty} \gamma^t \pi(a' \mid g) p^\pi(\mathfrak{s}_t = g \mid \mathfrak{s}_h = g, \mathfrak{a}_h = a') \right) \\
&\hspace{15em} \text{(Plug in Eq. (4.26))} \\
&= \sum_{h=0}^{\infty} \gamma^h p(H_{s,a}^\pi(g, a') = h) (1 - \gamma) \left( 1 + \sum_{t=h+1}^{\infty} \gamma^t \pi(a' \mid g) p^\pi(\mathfrak{s}_{t-h} = g \mid \mathfrak{s}_h = g, \mathfrak{a}_h = a') \right) \\
&\hspace{15em} \text{(Stationary property of MDP)} \\
&= \sum_{h=0}^{\infty} \gamma^h p(H_{s,a}^\pi(g, a') = h) (1 - \gamma) \left( 1 + \sum_{t=1}^{\infty} \gamma^t \pi(a' \mid g) p^\pi(\mathfrak{s}_t = g \mid \mathfrak{s}_0 = g, \mathfrak{a}_0 = a') \right) \\
&\hspace{15em} \text{(Change of variables)} \\
&= \sum_{h=0}^{\infty} \gamma^h p(H_{s,a}^\pi(g, a') = h) (1 - \gamma) \left( 1 + \sum_{t=1}^{\infty} \gamma^t \pi(a' \mid g) p^\pi(\mathfrak{s}_t = g \mid \mathfrak{s}_1 = s') P(s' \mid s, a) \right) \\
&= \sum_{h=0}^{\infty} \gamma^h p(H_{s,a}^\pi(g, a') = h) (1 - \gamma) \left( 1 + \gamma \sum_{t=0}^{\infty} \gamma^t p^\pi(\mathfrak{s}_t = g \mid \mathfrak{s}_0 = s') \pi(a' \mid g) P(s' \mid s, a) \right) \\
&\hspace{15em} \text{(Change of variables)} \\
&= \sum_{h=0}^{\infty} \gamma^h p(H_{s,a}^\pi(g, a') = h) (1 - \gamma) \left( \delta_{g,a'}(g, a') + \gamma \sum_{t=0}^{\infty} \gamma^t p^\pi(\mathfrak{s}_t = g \mid \mathfrak{s}_0 = s') \pi(a' \mid g) P(s' \mid s, a) \right) \\
&= \sum_{h=0}^{\infty} \gamma^h p(H_{s,a}^\pi(g, a') = h) p_\gamma^\pi(\mathfrak{s}^+ = g, \mathfrak{a}^+ = a' \mid \mathfrak{s}_0 = g, \mathfrak{a}_0 = a') \\
&= \mathbb{E} [\gamma^{H_{s,a}^\pi(g, a')}] p_\gamma^\pi(\mathfrak{s}^+ = g, \mathfrak{a}^+ = a' \mid \mathfrak{s}_0 = g, \mathfrak{a}_0 = a'),
\end{aligned}$$

□

**Remark 4.8.** The hitting times  $H_s^\pi(g)$  and  $H_{s,a}^\pi(g)$  are independent of the distribution  $\pi(\cdot \mid g)$ .

**Remark 4.9.** We can write

$$H_{s,a}^\pi(g, a') = H_s^\pi(g) + \mathbb{E}_{\pi(\hat{a} \mid g)} [H_{g,\hat{a}}^\pi(g, a')].$$

These remarks follow from the definitions in Eqs. (4.24) and (4.25) and the conditional independence of the states before  $g$  is reached and the action taken at  $g$ .

**Lemma 4.1.**  $d_{\text{SD}}((s, a), (s', a'))$  is independent of  $a'$  when  $s \neq s'$ .

*Proof.* Suppose  $s \neq g$ . We have from Eq. (4.6) that

$$\begin{aligned}
d_{\text{SD}}((s, a), (g, a')) &= \min_{\pi \in \Pi} \left[ \log \frac{p_{\gamma}^{\pi}(\mathfrak{s}^+ = g, \mathfrak{a}^+ = a' | \mathfrak{s}_0 = g, \mathfrak{a}_0 = a')}{p_{\gamma}^{\pi}(\mathfrak{s}^+ = g, \mathfrak{a}^+ = a' | \mathfrak{s}_0 = s, \mathfrak{a}_0 = a)} \right] \\
&= - \max_{\pi \in \Pi} \left[ \log \mathbb{E} \left[ \gamma^{H_{s,a}^{\pi}(g, a')} \right] \right] && \text{(Lemma 4.7)} \\
&= - \max_{\pi \in \Pi} \log \mathbb{E} \left[ \gamma^{H_{s,a}^{\pi}(g) + \mathbb{E}_{\pi(\hat{a}|g)}[H_{g,\hat{a}}^{\pi}(g, a')]} \right]. && \text{(Remark 4.9)}
\end{aligned}$$

Now, from Remark 4.8, the first term  $H_{s,a}^{\pi}(g)$  is independent of  $\pi(\cdot | g)$ . Meanwhile, the second term  $\mathbb{E}_{\pi(\hat{a}|g)}[H_{g,\hat{a}}^{\pi}(g, a')]$  is minimized when  $\pi(\hat{a} | g) = \delta_{a'}(\hat{a})$ , i.e., when the action taken at  $g$  is  $a'$ . Thus, at the maximum  $\pi(\cdot | g) = \delta_{a'}(\cdot)$ ; continuing, we see

$$\begin{aligned}
d_{\text{SD}}((s, a), (g, a')) &= - \max_{\pi \in \Pi} \log \mathbb{E} \left[ \gamma^{H_{s,a}^{\pi}(g) + \mathbb{E}_{\pi(\hat{a}|g)}[H_{g,\hat{a}}^{\pi}(g, a')]} \right] \\
&= - \max_{\pi \in \Pi} \log \mathbb{E} \left[ \gamma^{H_{s,a}^{\pi}(g)} \right].
\end{aligned}$$

From this last expression we see that  $d_{\text{SD}}((s, a), (g, a'))$  is independent of the action at the goal  $a'$ , as desired.  $\square$

**Lemma 4.2.** *Selecting actions to minimize the successor distance is equivalent to selecting actions to maximize the (scaled and shifted) Q-function:*

$$\begin{aligned}
- d_{\text{SD}}(s, a, g) &= \frac{1}{p_g(g)} Q(s, a, g) + c_{\psi}(g) \\
\implies \arg \max_a d_{\text{SD}}(s, a, g) &= \arg \max_a Q(s, a, g).
\end{aligned}$$

*Proof.* As noted in prior work Eysenbach et al. [121, Lemma 4.1], the optimal critic (Eq. (4.9)) is equivalent to a scaled Q function:

$$e^{f_{\theta^*}(s,a,g)} = \frac{1}{C \cdot p_g(g)} \underbrace{p_{\gamma}^{\pi}(\mathfrak{s}^+ = g | \mathfrak{s}_0 = s, a)}_{Q(s,a,g)}.$$

Eq. (4.11) then tells us that the successor distance differs from  $f_{\theta^*}(s, a, g)$  by a term that depends only on  $g$ , so taking the argmin of the successor distance is the same as taking the argmax of this scaled Q function.  $\square$

Now, we will prove Lemma 4.3.

**Lemma 4.3.** For any  $s, w, g \in \mathcal{S}$ ,  $\pi \in \Pi$ ,

$$\begin{aligned} \max_{\pi' \in \Pi} \left[ \frac{p_\gamma^\pi[\pi'](\mathfrak{s}^+ = g \mid \mathfrak{s}_0 = w) p_\gamma^\pi[\pi](\mathfrak{s}^+ = w \mid \mathfrak{s}_0 = s)}{p_\gamma^\pi[\pi](\mathfrak{s}^+ = w \mid \mathfrak{s}_0 = w)} \right] \\ \leq \max_{\pi' \in \Pi} p_\gamma^\pi[\pi'](\mathfrak{s}^+ = g \mid \mathfrak{s}_0 = s). \end{aligned}$$

*Proof.* Define  $\tilde{\pi} \in \Pi_{\text{NM}}$  to be the non-Markovian policy that starts executing  $\pi'$  and switches to  $\pi$  after reaching  $w$ :

$$\tilde{\pi}(a_t \mid s_t) = \begin{cases} \pi(a_t \mid s_t) & w \in \{s_0, s_1, \dots, s_t\} \\ \pi'(a_t \mid s_t) & \text{otherwise.} \end{cases}$$

We take  $\pi' \in \Pi$  to be an arbitrary policy. Let  $E_1$  be the event where the hitting time of waypoint  $w$  is less than the hitting time of goal  $g$  starting from state  $s$ , i.e.,  $E_1 = \{H_s^{\tilde{\pi}}(w) < H_s^{\tilde{\pi}}(g)\}$ . Complementary, let  $E_2$  be the event where the hitting time of waypoint  $w$  is greater than or equal to the hitting time of goal  $g$  starting from state  $s$ , i.e.,  $E_2 = \{H_s^{\tilde{\pi}}(w) \geq H_s^{\tilde{\pi}}(g)\}$ . We note that  $E_1$  and  $E_2$  are mutually exclusive.

We start by rewriting  $p_\gamma^\pi[\tilde{\pi}](\mathfrak{s}^+ = g \mid \mathfrak{s}_0 = s)$ :

$$\begin{aligned} p_\gamma^\pi[\tilde{\pi}](\mathfrak{s}^+ = g \mid \mathfrak{s}_0 = s) &= \sum_{h=0}^{\infty} p(H_s^{\tilde{\pi}}(w) = h) \left( (1 - \gamma) \sum_{t=0}^{\infty} \gamma^t p^{\tilde{\pi}}(\mathfrak{s}_t = g \mid \mathfrak{s}_0 = s, H_s^{\tilde{\pi}}(w) = h) \right) \\ &= \sum_{h=0}^{\infty} p(H_s^{\pi'}(w) = h) \left( (1 - \gamma) \sum_{t=0}^{\infty} \gamma^t p^{\tilde{\pi}}(\mathfrak{s}_t = g \mid \mathfrak{s}_0 = s, H_s^{\tilde{\pi}}(w) = h) \right). \end{aligned} \tag{4.27}$$

Now,  $p^{\tilde{\pi}}(\mathfrak{s}_t = g \mid \mathfrak{s}_0 = s, H_s^{\tilde{\pi}}(w) = h)$  can be written as

$$\begin{aligned} p^{\tilde{\pi}}(\mathfrak{s}_t = g \mid \mathfrak{s}_0 = s, H_s^{\tilde{\pi}}(w) = h) &= \\ &\begin{cases} 0 & t < h, \text{ under } E_1 \\ p^{\tilde{\pi}}(\mathfrak{s}_t = g \mid \mathfrak{s}_0 = s, H_s^{\tilde{\pi}}(w) = h, H_s^{\tilde{\pi}}(g) \leq h) & t < h, \text{ under } E_2. \\ p^\pi(\mathfrak{s}_t = g \mid \mathfrak{s}_h = w) & t \geq h \end{cases} \end{aligned} \tag{4.28}$$

Dropping the first  $h$  terms (which are all non-negative), we get

$$\begin{aligned} \sum_{t=0}^{\infty} \gamma^t p^{\tilde{\pi}}(\mathfrak{s}_t = g \mid \mathfrak{s}_0 = s, H_s^{\tilde{\pi}}(w) = h) &\geq \sum_{t=h}^{\infty} \gamma^t p^{\tilde{\pi}}(\mathfrak{s}_t = g \mid \mathfrak{s}_0 = s, H_s^{\tilde{\pi}}(w) = h) \\ &= \sum_{t=h}^{\infty} \gamma^t p^\pi(\mathfrak{s}_t = g \mid \mathfrak{s}_h = w) \end{aligned}$$

Plugging this inequality into Eq. (4.27), we have

$$\begin{aligned}
p_\gamma^\pi[\tilde{\pi}](\mathfrak{s}^+ = g | \mathfrak{s}_0 = s) &\geq \sum_{h=0}^{\infty} p(H_s^{\pi'}(w) = h) \left( (1 - \gamma) \sum_{t=h}^{\infty} \gamma^t p^\pi(\mathfrak{s}_t = g | \mathfrak{s}_h = w) \right) \\
&= \sum_{h=0}^{\infty} \gamma^h p(H_s^{\pi'}(w) = h) \left( (1 - \gamma) \sum_{t=h}^{\infty} \gamma^{t-h} p^\pi(\mathfrak{s}_{t-h} = g | \mathfrak{s}_0 = w) \right) \\
&\hspace{15em} \text{(Stationary property of MDP)} \\
&= \sum_{h=0}^{\infty} \gamma^h p(H_s^{\pi'}(w) = h) \left( (1 - \gamma) \sum_{t=0}^{\infty} \gamma^t p^\pi(\mathfrak{s}_t = g | \mathfrak{s}_0 = w) \right) \\
&\hspace{15em} \text{(Change of variables)} \\
&= \mathbb{E}[\gamma^{H_s^{\pi'}(w)}] p_\gamma^\pi(\mathfrak{s}^+ = g | \mathfrak{s}_0 = w).
\end{aligned}$$

Applying Lemma 4.6 to the last step, we see

$$\begin{aligned}
p_\gamma^\pi[\tilde{\pi}](\mathfrak{s}^+ = g | \mathfrak{s}_0 = s) &\geq \mathbb{E}[\gamma^{H_s^{\pi'}(w)}] p_\gamma^\pi[\pi'](\mathfrak{s}^+ = g | \mathfrak{s}_0 = w) \\
&= \frac{p_\gamma^\pi[\pi'](\mathfrak{s}^+ = g | \mathfrak{s}_0 = w) p_\gamma^\pi(\mathfrak{s}^+ = w | \mathfrak{s}_0 = s)}{p_\gamma^\pi(\mathfrak{s}^+ = w | \mathfrak{s}_0 = w)}.
\end{aligned}$$

Since there is a stationary Markovian optimal policy  $\pi^*$  for  $r_g$  in  $M$ , we know from Lemma 4.2 that

$$p_\gamma^\pi[\tilde{\pi}](\mathfrak{s}^+ = g | \mathfrak{s}_0 = s) \leq \max_{\pi' \in \Pi} p_\gamma^\pi[\pi'](\mathfrak{s}^+ = g | \mathfrak{s}_0 = s),$$

so we have

$$\max_{\pi' \in \Pi} p_\gamma^\pi[\pi'](\mathfrak{s}^+ = g | \mathfrak{s}_0 = s) \geq \frac{p_\gamma^\pi[\pi'](\mathfrak{s}^+ = g | \mathfrak{s}_0 = w) p_\gamma^\pi(\mathfrak{s}^+ = w | \mathfrak{s}_0 = s)}{p_\gamma^\pi(\mathfrak{s}^+ = w | \mathfrak{s}_0 = w)}.$$

Since  $\pi'$  on the RHS was arbitrary, we conclude

$$\max_{\pi' \in \Pi} \left[ \frac{p_\gamma^\pi[\pi'](\mathfrak{s}^+ = g | \mathfrak{s}_0 = w) p_\gamma^\pi(\mathfrak{s}^+ = w | \mathfrak{s}_0 = s)}{p_\gamma^\pi(\mathfrak{s}^+ = w | \mathfrak{s}_0 = w)} \right] \leq \max_{\pi' \in \Pi} p_\gamma^\pi[\pi'](\mathfrak{s}^+ = g | \mathfrak{s}_0 = s).$$

□

**Lemma 4.10.** *For any  $s, w, g \in \mathcal{S}$ ,  $a_s, a_w, a_g \in \mathcal{A}$ , and  $\pi \in \Pi$ , we have*

$$\begin{aligned}
\max_{\pi' \in \Pi} \left[ \frac{p_\gamma^\pi[\pi'](\mathfrak{s}^+ = g, \mathfrak{a}^+ = a_g | \mathfrak{s}_0 = w, \mathfrak{a}_0 = a_w) p_\gamma^\pi[\pi'](\mathfrak{s}^+ = w, \mathfrak{a}^+ = a_w | \mathfrak{s}_0 = s, \mathfrak{a}_0 = a_s)}{p_\gamma^\pi[\pi'](\mathfrak{s}^+ = w, \mathfrak{a}^+ = a_w | \mathfrak{s}_0 = w, \mathfrak{a}_0 = a_w)} \right] \\
\leq \max_{\pi' \in \Pi} [p_\gamma^\pi[\pi'](\mathfrak{s}^+ = g, \mathfrak{a}^+ = a_g | \mathfrak{s}_0 = s, \mathfrak{a}_0 = a_s)].
\end{aligned}$$

The proof follows from the same argument as in Lemma 4.3 but applying Lemma 4.7 instead of Lemma 4.6.

## 4.7 PROOFS

**Theorem 4.4** (Quasimetric).  $d_{\text{SD}}$  is a quasimetric over  $\mathcal{S}$ , satisfying the triangle inequality and other properties from Definition 4.1.

*Proof.* We check the conditions of Definition 4.1:

**Positivity:** Applying Lemma 4.6, we see

$$\begin{aligned} d_{\text{SD}}(s, g) &= \min_{\pi \in \Pi} \log p_{\gamma}^{\pi}(\mathfrak{s}^+ = g \mid \mathfrak{s}_0 = g) - \log p_{\gamma}^{\pi}(\mathfrak{s}^+ = g \mid \mathfrak{s}_0 = s) \\ &= \min_{\pi \in \Pi} \log p_{\gamma}^{\pi}(\mathfrak{s}^+ = g \mid \mathfrak{s}_0 = g) - \log \mathbb{E}[\gamma^{H_s^{\pi}(g)}] p_{\gamma}^{\pi}(\mathfrak{s}^+ = g \mid \mathfrak{s}_0 = g) \\ &\geq \min_{\pi \in \Pi} \log p_{\gamma}^{\pi}(\mathfrak{s}^+ = g \mid \mathfrak{s}_0 = g) - \log p_{\gamma}^{\pi}(\mathfrak{s}^+ = g \mid \mathfrak{s}_0 = g) \\ &= 0. \end{aligned}$$

**Identity:** We see  $d_{\text{SD}}(s, g) = 0$  precisely iff  $p_{\gamma}^{\pi}(\mathfrak{s}^+ = g \mid \mathfrak{s}_0 = g) = p_{\gamma}^{\pi}(\mathfrak{s}^+ = g \mid \mathfrak{s}_0 = s)$  for some  $\pi \in \Pi$ . This holds when  $s = g$ . For  $s \neq g$ , we have  $p_{\gamma}^{\pi}(\mathfrak{s}^+ = g \mid \mathfrak{s}_0 = s) \leq \gamma p_{\gamma}^{\pi}(\mathfrak{s}^+ = g \mid \mathfrak{s}_0 = g)$ . Since  $p_{\gamma}^{\pi}(\mathfrak{s}^+ = g \mid \mathfrak{s}_0 = g) \geq 1 - \gamma$  by construction,  $d_{\text{SD}}(s, g) \neq 0$ .

**Triangle inequality:** We see:

$$\begin{aligned} d_{\text{SD}}(s, g) &= \min_{\pi \in \Pi} \log p_{\gamma}^{\pi}(\mathfrak{s}^+ = g \mid \mathfrak{s}_0 = g) - \log p_{\gamma}^{\pi}(\mathfrak{s}^+ = g \mid \mathfrak{s}_0 = s) \\ &\leq \min_{\pi \in \Pi} \log p_{\gamma}^{\pi}(\mathfrak{s}^+ = g \mid \mathfrak{s}_0 = g) - \log \left( \max_{\pi' \in \Pi} \left[ \frac{p_{\gamma}^{\pi}(\mathfrak{s}^+ = g \mid \mathfrak{s}_0 = w) p_{\gamma}^{\pi'}(\mathfrak{s}^+ = w \mid \mathfrak{s}_0 = s)}{p_{\gamma}^{\pi'}(\mathfrak{s}^+ = w \mid \mathfrak{s}_0 = w)} \right] \right) \\ &\hspace{15em} \text{(Lemma 4.3)} \\ &= \min_{\pi \in \Pi} \log p_{\gamma}^{\pi}(\mathfrak{s}^+ = g \mid \mathfrak{s}_0 = g) - \max_{\pi' \in \Pi} \log \left( \frac{p_{\gamma}^{\pi}(\mathfrak{s}^+ = g \mid \mathfrak{s}_0 = w) p_{\gamma}^{\pi'}(\mathfrak{s}^+ = w \mid \mathfrak{s}_0 = s)}{p_{\gamma}^{\pi'}(\mathfrak{s}^+ = w \mid \mathfrak{s}_0 = w)} \right) \\ &= \left( \min_{\pi \in \Pi} \log \frac{p_{\gamma}^{\pi}(\mathfrak{s}^+ = g \mid \mathfrak{s}_0 = g)}{p_{\gamma}^{\pi}(\mathfrak{s}^+ = g \mid \mathfrak{s}_0 = w)} \right) - \left( \max_{\pi' \in \Pi} \log \frac{p_{\gamma}^{\pi'}(\mathfrak{s}^+ = w \mid \mathfrak{s}_0 = s)}{p_{\gamma}^{\pi'}(\mathfrak{s}^+ = w \mid \mathfrak{s}_0 = w)} \right) \\ &= \left( \min_{\pi \in \Pi} \log \frac{p_{\gamma}^{\pi}(\mathfrak{s}^+ = g \mid \mathfrak{s}_0 = g)}{p_{\gamma}^{\pi}(\mathfrak{s}^+ = g \mid \mathfrak{s}_0 = w)} \right) + \left( \min_{\pi' \in \Pi} \log \frac{p_{\gamma}^{\pi'}(\mathfrak{s}^+ = w \mid \mathfrak{s}_0 = w)}{p_{\gamma}^{\pi'}(\mathfrak{s}^+ = w \mid \mathfrak{s}_0 = s)} \right) \\ &= d_{\text{SD}}(w, g) + d_{\text{SD}}(s, w) \tag{4.29} \end{aligned}$$

as desired. □

Consider the following didactic example for why we might want to extend the successor distance to the state-action space  $\mathcal{S} \times \mathcal{A}$  (Fig. 4.4).

$$\begin{array}{rcccl}
 \pi_1 \text{ data :} & (s, \rightarrow) & \xrightarrow{\mathbf{1}} & (w, \circ) & \curvearrowright \mathbf{1} & (g, \circ) & \curvearrowright \mathbf{1} \\
 & & & \vdots & & & \\
 & & & \approx & & & \\
 \pi_2 \text{ data :} & (s, \circ) & & (w, \rightarrow) & \xrightarrow{\mathbf{1}} & (g, \circ) & \curvearrowright \mathbf{1} \\
 & \uparrow \mathbf{1} & & & & & \\
 & \curvearrowright & & & & & \\
 \pi' \text{ behavior :} & (s, \rightarrow) & \xrightarrow{\mathbf{1}} & (w, \rightarrow) & \xrightarrow{\mathbf{1}} & (g, \circ) & \curvearrowright \mathbf{1}
 \end{array}$$

Figure 4.4: A simple illustration of a metric over  $\mathcal{S} \times \mathcal{A}$ . To stitch the behavior  $s \rightarrow w$  from  $\pi_1$  and  $w \rightarrow g$  from  $\pi_2$  to the behavior  $s \rightarrow g$  that is possible under some policy  $\pi'$ , we enforce an additional constraint that distances to  $(w, \rightarrow)$  are the same as distances to  $(w, \circ)$ .

**Corollary 4.4.1.**  $d_{\text{SD}}$  is a quasimetric over  $\mathcal{S} \times \mathcal{A}$ .

This statement follows from the same argument as in Theorem 4.4 but applying Lemma 4.10 instead of Lemma 4.3 to the triangle inequality.

**Corollary 4.4.2.**  $d_{\text{SD}}$  is a quasimetric over an uncontrolled Markov process as in Eq. (4.5).

This statement follows from Theorem 4.4 by taking  $\mathcal{A} = \{a\}$  so  $\Pi = \{\pi\}$ .

Analysis of CMD-1

**Lemma 4.5.** For  $s \neq g$ , the unique solution to the the loss function in Eq. (4.8) with the parametrization in Eq. (4.12) is

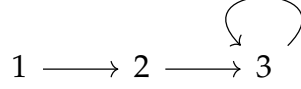
$$d_{\phi^*}(s, a, g) = \log \frac{p_{\gamma}^{\pi}(\mathbf{s}^+ = g \mid \mathbf{s}_0 = s, \mathbf{a}_0 = a)}{p_{\gamma}^{\pi}(\mathbf{s}^+ = g \mid \mathbf{s}_0 = g)}. \quad (4.14)$$

*Proof.* Eq. (4.10) together with Eq. (4.13) tell us that, if  $f(s, a, g)$  satisfies Eq. (4.9), then the learned  $d_{\phi}(s, a, g)$  is the successor distance. What remains is to show that the parametrization in Eq. (4.12) is sufficient to represent Eq. (4.9): Eq. (4.11) tells us that it is sufficient for (1) we use a universal quasimetric network for  $d_{\phi}(s, a, g)$  [127, 130], and (2) use a universal network  $c_{\psi}(g)$  (e.g., sufficient layers in a neural network [145]).  $\square$

## 4.8 DIDACTIC EXAMPLES

We present a few examples of how the successor distance defined in Eq. (4.5) yields a valid quasimetric.

Example 1: 3-state Markov Process.



Eq. (4.2) Assume that the initial state is “1”, a discount factor of  $\gamma$ , and that state “3” is absorbing. We assume that the discounted state occupancy measure states at  $t = 0$ , so that it includes the current time step.

$$p(3 | 3) = 1$$

$$p(2 | 2) = 1 - \gamma$$

$$p(2 | 1) = \gamma(1 - \gamma)$$

$$p(3 | 2) = \gamma$$

$$p(3 | a) = \gamma^2$$

$$d(1,3) = \log p(3 | 3) - \log p(3 | 1) = \log 1 - \log \gamma^2 = 0 + 2 \log \frac{1}{\gamma}$$

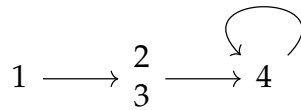
$$d(1,2) = \log p(2 | 2) - \log p(2 | 1) = \log(1 - \gamma) - \log \gamma(1 - \gamma) = \log \frac{1}{\gamma}$$

$$d(2,3) = \log p(3 | 3) - \log p(3 | 2) = \log 1 - \log \gamma = \log \frac{1}{\gamma}$$

$$d(1,2) + d(2,3) = 2 \log \frac{1}{\gamma} \geq d(1,3) = 2 \log \frac{1}{\gamma}. \checkmark$$

In this example, note that the triangle inequality is tight. This is because there is a single state that we are guaranteed to visit between states “1” and “3.”

Example 2: 4-state Markov Process.



From state “1”, states “2” and “3” each occur with probability 0.5.

$$p(4 | 4) = 1$$

$$p(2 | 2) = 1 - \gamma$$

$$p(2 | 1) = \frac{1}{2}(1 - \gamma)\gamma$$

$$p(4 | 1) = \gamma^2$$

$$p(4 | 2) = \gamma$$

$$d(1,2) = \log p(2 | 2) - \log p(2 | 1)$$

$$= \log(1 - \gamma) - \log \frac{1}{2}(1 - \gamma)\gamma = \log \frac{1}{\gamma} + \log 2$$

$$d(2,4) = \log p(4 | 4) - \log p(4 | 2)$$

$$\begin{aligned} &= \log 1 - \log \gamma = \log \frac{1}{\gamma} \\ d(1,4) &= \log p(4 | 4) - \log p(1 | 4) \\ &= \log 1 - \log \gamma^2 = 2 \log \frac{1}{\gamma} \\ d(1,2) + d(2,4) &= 2 \log \frac{1}{\gamma} + \log 2 \geq d(1,4) = 2 \log \frac{1}{\gamma}. \checkmark \end{aligned}$$

In this example, the triangle inequality is loose. This is because we have uncertainty over which states we will visit between “1” and “4.” One way to resolve this uncertainty is to aggregate states “2” and “3” together; if we did this, we’d be back at example 1, where the triangle inequality is tight.



## 4.9 ACTION-INVARIANCE

Let's assume that data are collected with a Markovian policy, so  $p(s', a' | s, a) = \beta(a' | s')p(s' | s, a)$ . Then CRL will learn

$$e^{f(s,a,s',a')} = \frac{p(s', a' | s, a)}{p(s', a')} \quad (4.30)$$

$$= \frac{\beta(a' | s')p(s' | s, a)}{\beta(a' | s')p(s')} \quad (4.31)$$

Thus, if data are collected with a Markovian policy, then the optimal critic will not depend on the future actions. Note that this remains true for any parametrization of the critic (including MRN) that can represent the optimal critic.

However, the assumption on a Markovian data collection policy can be violated in a few ways:

1. In the online setting, data are collected from policies at different iterations. In this setting, conditioning on a previous state and action can give you a better prediction of  $a'$  (violating the Markov assumption) because it can allow you to infer which policy you're using.
2. In goal-conditioned settings, the data collection policy is conditioned on the goal. Conditioning on a previous state and action can leak information about the desired goal.

One way of fixing this is to apply CRL to a different data distribution. Let  $p(s', a' | s, a)$  be given, and let  $\beta(a)$  be some distribution over actions (in practice, we might use the marginal distribution over actions in the dataset). Define

$$\tilde{p}(s', a' | s, a) \triangleq p(s' | s, a)\beta(a'), \quad \tilde{p}(s', a') \triangleq p(s')\beta(a'). \quad (4.32)$$

In practice, this corresponds to augmenting the CRL training examples  $(s, a, s', a') \rightarrow (s, a, s', \tilde{a}')$  by resampling the future actions. Now, consider applying CRL to this new distribution:

$$e^{f(s,a,s',a')} = \frac{\tilde{p}(s', a' | s, a)}{\tilde{p}(s', a')} \quad (4.33)$$

$$= \frac{\beta(a' | s')\tilde{p}(s' | s, a)}{\beta(a' | s')\tilde{p}(s')} \quad (4.34)$$

Thus, if we apply CRL to data augmented in this way, we're guaranteed to learn a critic function  $f(s, a, s', a')$  that is invariant to  $a'$ .

## 4.10 RELATED WORK

Our work builds on prior work in learning temporal distances and contrastive representation learning.

## Learning distances

Within any Markov decision process (MDP), there is an intuitive notion of “distance” between states as the difficulty of transitioning between them. There are many seemingly reasonable definitions for distance a priori: likelihood of reaching the goal at a particular time, expected time to reach the goal, likelihood of ever reaching the goal, etc. (under some policy). The key mathematical structure for a distance to be useful for reaching goals is that it must satisfy the triangle inequality  $d(a, c) \leq d(a, b) + d(b, c)$ : being able to go from  $a \rightarrow b$  and from  $b \rightarrow c$  means going from  $a \rightarrow c$  can be no harder than both of the aforementioned steps. Such a distance is called a *metric* over the state space if it is symmetric and more generally a *quasimetric* [146].

While prior work on bisimulation [147, 148] use a reward function to construct such a distance, our aim will be to define a notion of distance that does not require a reward function.

For the correct choice of distance, learning a goal-conditioned value function will correspond to selecting a distance metric that best enables goal reaching. Such a distance can then be learned with an architecture that directly enforces metric properties, e.g., Euclidean distance, metric residual network (MRN), interval quasimetric estimator (IQE), etc. [130, 143, 127]. Since the space of value (quasi)metrics imposes a strong induction bias over value functions, using the right metric architecture can enable better combinatorial and temporal generalization *without* requiring additional samples [126].

In deterministic MDPs, these notions of distance all coincide with distance  $d(s, g)$  being proportional to the (minimum) amount of time needed to reach the goal  $g$  when starting in state  $s$ . Approaches like Quasimetric RL [126, 127] learn this notion of distance, allowing optimal goal reaching in deterministic MDPs. In general MDPs, alternative notions of distance are required [46, 119, 149, 119, 149–151]. Existing approaches are often limited by assumptions such as symmetry or fail to satisfy metric properties. Our contribution is to construct a general formulation for a quasimetric over MDPs that can be easily learned from discounted state occupancy measures.

## Contrastive Representations

Contrastive learning has seen widespread adoption for learning to represent time series [152, 153]. These representations can be trained to approximate mutual information without requiring labels or reconstruction [142, 154, 155, 53, 156], and are useful for learning self-supervised representations across broad application areas [17, 157–161].

Within RL, contrastive learning can be used for goal-conditioned control as successor features [162, 121, 122]. Approaches that use contrastive representations for control are typically limited in combinatorial and temporal generalization since they

do not bootstrap value functions [163]. Unlike past approaches that use contrastive learning for decision-making, we show that these generalization capabilities *can* be obtained from contrastive successor features by imposing an additional metric structure.

### Goal-conditioned reinforcement learning (GCRL)

Goal-reaching presents an attractive formulation for learning useful behaviors in unsupervised RL settings [164, 49]. Recent advances in deep reinforcement learning have renewed interest in this problem as many real-world offline and online RL problems lack clear reward signals [14, 121, 165, 166, 15]. GCRL methods can learn goal-conditioned policies [167, 15], value functions [168, 169], and/or representations that enable goal-reaching [121, 134, 119]. Approaches that recover goal-conditioned policies can also enable additional capabilities like planning [170, 171], skill discovery [172, 173] and interface with other forms of task specification like language [46, 7, 97, 87, 174].

These GCRL techniques typically require bootstrapping with a learned value function, which can be costly and unstable, or struggle with long-horizon combinatorial and temporal generalization [1]. Our approach avoids both of these shortcomings by learning a distance metric that can implicitly combine behaviors without bootstrapping or making any assumptions about the environment dynamics.

# 5 TEMPORAL REPRESENTATION ALIGNMENT FOR COMPOSITIONAL INSTRUCTION FOLLOWING

---

Compositionality is a core aspect of intelligent behavior, describing the ability to sequence previously learned capabilities and solve new tasks [175]. In domains involving long-horizon decision-making like robotics, various learning approaches have been proposed to enable this property, including hierarchical learning [176], explicit subtask planning [177, 170, 90], and dynamic-programming-based “stitching” [1, 144]. In practice, these techniques are often unstable and/or data-inefficient in real-world robotics settings, making them difficult to scale [178].

By contrast, biological learners are adept at quickly composing behaviors to reach new goals [175]. Possible explanations for these capabilities have been proposed, including the ability to perform transitive inference [179], learn successor representations and causal models [124, 180], and plan with world models [181]. In common among these theories is the idea of learning structured representations of the world, which inference about which actions will lead to certain goals.

How might these concepts translate to algorithms for robot learning? In this work, we study how adding an auxiliary successor representation learning objective affects compositional behavior in a real-world tabletop manipulation setting. We show that learning this representation structure improves the ability of the robot to perform long-horizon, compositionally-new tasks, specified either through goal images or natural language instructions. Perhaps surprisingly, we found that this temporal alignment does not need to be used for training the policy or test-time inference, as long as it is used as an auxiliary loss over the same representations used for the tasks. An example of this can be seen in Fig. 5.1.

We evaluate our method, **Temporal Representation Alignment (TRA)**, on a set of challenging multi-step manipulation tasks in the BridgeData setup [62]. These tasks specifically test the compositional capabilities of the robot policies: as a whole, the tasks are out-of-distribution, but each distinct subtask can be described through a goal image that lies in the training distribution. Adding a simple time-contrastive alignment loss improves compositional performance on these tasks by >40% across 13 tasks in 4 scenes.

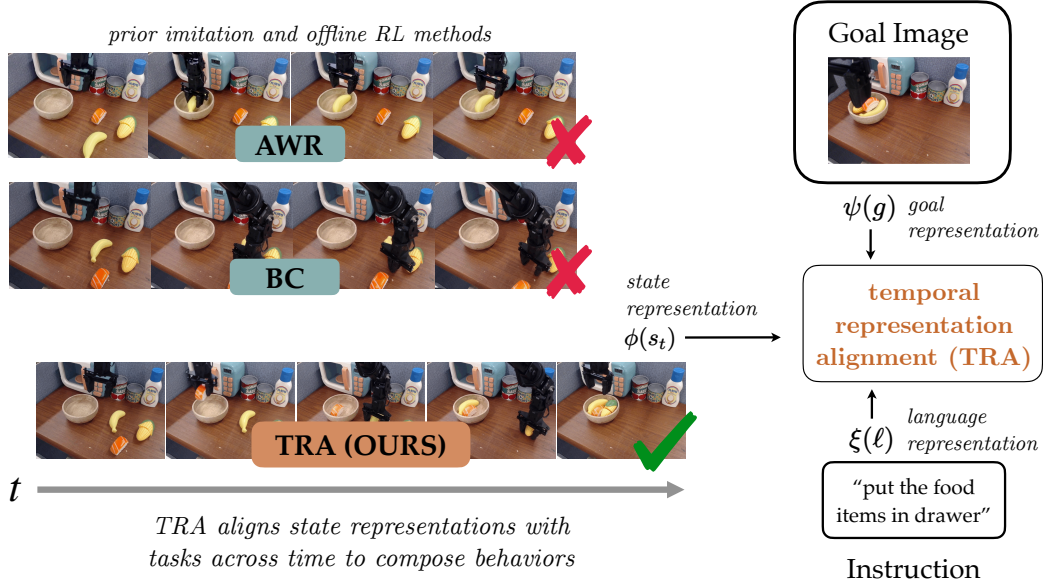


Figure 5.1: Example rollouts of a task with TRA and GCBC to put all food items in the bowl. While TRA can implicitly decompose the task into steps and execute them one by one, GCBC is unable to do that and fails to ground to any relevant objects. GCBC+AWR on the other hand only grounds one object, failing to display any compositionality

## 5.1 TEMPORAL REPRESENTATION ALIGNMENT

Given training on a series of short-horizon goal-reaching and instruction-following tasks, our goal is to learn a representation space such that our policy can generalize to a new (long-horizon) task that can be viewed as a sequence of known subtasks. We propose to structure this representation space by aligning the representations of states, goals, and language in a way that is more amenable to compositional generalization.

**Notation.** We take the setting of a goal- and language-conditioned MDP  $\mathcal{M}$  with state space  $\mathcal{S}$ , continuous action space  $\mathcal{A} \subseteq (0, 1)^{d_{\mathcal{A}}}$ , initial state distribution  $p_0$ , dynamics  $P(s' | s, a)$ , discount factor  $\gamma$ , and language task distribution  $p_{\ell}$ . A policy  $\pi(a | s)$  maps states to a distribution over actions. We inductively define the  $k$ -step (action-conditioned) policy visitation distribution as:

$$\begin{aligned}
 p_1^{\pi}(s_1 | s_1, a_1) &\triangleq p(s_1 | s_1, a_1), \\
 p_{k+1}^{\pi}(s_{k+1} | s_1, a_1) &\triangleq \int_{\mathcal{A}} \int_{\mathcal{S}} p(s_{k+1} | s, a) dp_k^{\pi}(s | s_1, a_1) d\pi(a | s) \\
 p_{k+t}^{\pi}(s_{k+t} | s_t, a_t) &\triangleq p^{\pi}(s_k | s_1, a_1).
 \end{aligned} \tag{5.1}$$

Then, the discounted state visitation distribution can be defined as the distribution over  $s^+$ , the state reached after  $K \sim \text{Geom}(1 - \gamma)$  steps:

$$p_\gamma^\pi(s^+ | s, a) \triangleq \sum_{k=0}^{\infty} \gamma^k p_k^\pi(s^+ | s, a). \quad (5.2)$$

We assume access to a dataset of expert demonstrations  $\mathcal{D} = \{\tau_i, \ell_i\}_{i=1}^K$ , where each trajectory

$$\tau_i = \{s_{t,i}, a_{t,i}\}_{t=1}^H \in \mathcal{S} \times \mathcal{A} \times \mathcal{S} \quad (5.3)$$

is gathered by an expert policy  $\pi^E$ , and is then annotated with  $p_\ell(\ell_i | s_{1,i}, s_{H,i})$ . Our aim is to learn a policy  $\pi$  that can select actions conditioned on a new language instruction  $\ell$ . As in prior work [62], we handle the continuous action space by both our policy and the expert policy as an isotropic Gaussian with fixed variance; we will equivalently write  $\pi(a | s, \varphi)$  or denote the mode as  $\hat{a} = \pi(s, \varphi)$  for a task  $\varphi$ .

Motivation: Representations for Reaching Distant Goals

We learn a goal-conditioned policy  $\pi(a | s, g)$  that selects actions to reach a goal  $g$  from expert demonstrations with behavioral cloning. Suppose we directly selected actions to imitate the expert on two trajectories in  $\mathcal{D}$ :

$$\left. \begin{array}{l} s_1 \longrightarrow s_2 \longrightarrow \dots \longrightarrow s_H \longrightarrow w \\ w \longrightarrow s'_1 \longrightarrow \dots \longrightarrow s'_H \longrightarrow g \end{array} \right\} \tau_i \in \mathcal{D} \quad (5.4)$$

When conditioned with the composed goal  $g$ , we would be unable to imitate effectively as the composed state-goal  $(s, g)$  is jointly out of the training distribution.

What *would* work for reaching  $g$  is to first condition the policy on the intermediate waypoint  $w$ , then upon reaching  $w$ , condition on the goal  $g$ , as the state-goal pairs  $(s_i, w)$ ,  $(w, g)$ , and  $(s'_i, g)$  are all in the training distribution. If we condition the policy on some intermediate waypoint distribution  $p(w)$  (or sufficient statistics thereof) that captures all of these cases, we can stitch together the expert behaviors to reach the goal  $g$ .

Consider the goal-conditioned behavioral cloning [49] loss  $\mathcal{L}_{\text{BC}}^{\phi, \psi, \xi}$  conditioned with waypoints  $w$ .

$$\mathcal{L}_{\text{BC}}(\{s_i, a_i, s_i^+, g_i\}_{i=1}^K) = \sum_{i=1}^K \log \pi(a_i | s_i, \psi(g_i)). \quad (5.5)$$

Enforcing the invariance needed to stitch Eq. (5.4) then reduces to aligning  $\psi(g) \leftrightarrow \psi(w)$ . The temporal alignment objective  $\phi(s) \leftrightarrow \phi(s^+)$  accomplishes this indirectly by aligning both  $\psi(w)$  and  $\psi(g)$  to the shared waypoint representation  $\phi(w)$ :

$$\mathcal{L}_{\text{NCE}}(\{s_i, s_i^+\}_{i=1}^K; \phi, \psi) = \log\left(\frac{e^{\phi(s_i^+)^T \psi(s_i)}}{\sum_{j=1}^K e^{\phi(s_i^+)^T \psi(s_j)}}\right) + \sum_{j=1}^K \log\left(\frac{e^{\phi(s_i^+)^T \psi(s_i)}}{\sum_{i=1}^K e^{\phi(s_i^+)^T \psi(s_j)}}\right) \quad (5.6)$$

### Interfacing with Language Instructions

To extend the representations from Section 5.1 to compositional instruction following with language tasks, we need some way to ground language into the  $\psi$  representation space. We use a similar approach to GRIF [7], which uses an additional CLIP-style [17] contrastive alignment loss with an additional pretrained language encoder  $\zeta$ :

$$\mathcal{L}_{\text{NCE}}(\{g_i, \ell_i\}_{i=1}^K; \psi, \zeta) = \sum_{i=1}^K \log\left(\frac{e^{\psi(g_i)^T \zeta(\ell_i)}}{\sum_{j=1}^K e^{\psi(g_i)^T \zeta(\ell_j)}}\right) + \sum_{j=1}^K \log\left(\frac{e^{\psi(g_i)^T \zeta(\ell_i)}}{\sum_{i=1}^K e^{\psi(g_i)^T \zeta(\ell_j)}}\right) \quad (5.7)$$

### Temporal Alignment

The Temporal Representation Alignment (TRA) approach structures the representation space of goals and language instructions to better enable compositional generalization. We learn encoders  $\phi$ ,  $\psi$ , and  $\zeta$  to map states, goals, and language instructions to a shared representation space.

$$\mathcal{L}_{\text{NCE}}(\{x_i, y_i\}_{i=1}^K; f, h) = \sum_{i=1}^K \log\left(\frac{e^{f(y_i)^T h(x_i)}}{\sum_{j=1}^K e^{f(y_i)^T h(x_j)}}\right) + \sum_{j=1}^K \log\left(\frac{e^{f(y_i)^T h(x_i)}}{\sum_{i=1}^K e^{f(y_i)^T h(x_j)}}\right) \quad (5.8)$$

$$\mathcal{L}_{\text{BC}}(\{s_i, a_i, s_i^+, \ell_i\}_{i=1}^K; \pi) = \sum_{i=1}^K \log \pi(a_i | s_i, \zeta(\ell_i)) + \log \pi(a_i | s_i, \psi(s_i^+)) \quad (5.9)$$

$$\begin{aligned} \mathcal{L}_{\text{TRA}}(\{s_i, a_i, s_i^+, g_i, \ell_i\}_{i=1}^K; \pi, \phi, \psi, \zeta) \\ = \underbrace{\mathcal{L}_{\text{BC}}(\{s_i, a_i, s_i^+, \ell_i\}_{i=1}^K; \pi, \psi, \zeta)}_{\text{behavioral cloning}} + \underbrace{\mathcal{L}_{\text{NCE}}(\{s_i, s_i^+\}_{i=1}^K; \phi, \psi)}_{\text{temporal alignment}} + \underbrace{\mathcal{L}_{\text{NCE}}(\{g_i, \ell_i\}_{i=1}^K; \psi, \zeta)}_{\text{task alignment}} \end{aligned} \quad (5.10)$$

Note that the NCE alignment loss uses a CLIP-style symmetric contrastive objective [17, 11] — we highlight the indices in the NCE alignment loss (5.8) for clarity.

Our overall objective is to minimize Eq. (5.10) across states, actions, future states, goals, and language tasks within the training data:

$$\min_{\pi, \phi, \psi, \zeta} \mathbb{E}_{\substack{(s_{1,i}, a_{1,i}, \dots, s_{H,i}, a_{H,i}, \ell) \sim \mathcal{D} \\ i \sim \text{Unif}(1 \dots H) \\ k \sim \text{Geom}(1 - \gamma)}}} \left[ \mathcal{L}_{\text{TRA}} \left( \{s_{t,i}, a_{t,i}, s_{\min(t+k, H), i}, s_{H,i}, \ell\}_{i=1}^K; \pi, \phi, \psi, \zeta \right) \right]. \quad (5.11)$$

---

**Algorithm 4:** Temporal Representation Alignment (TRA)
 

---

- 1: **input:** dataset  $\mathcal{D} = (\{s_{t,i}, a_{t,i}\}_{t=1}^H, \ell_i)_{i=1}^N$
  - 2: initialize networks  $\Theta \triangleq (\pi, \phi, \psi, \zeta)$
  - 3: **while** training **do**
  - 4:     sample a batch of transitions  $\{(s_{t,i}, a_{t,i}, s_{t+k,i}, \ell_i)\}_{i=1}^K \sim \mathcal{D}$  for  $k \sim \text{Geom}(1 - \gamma)$
  - 5:      $\Theta \leftarrow (\pi, \phi, \psi, \zeta) - \alpha \nabla_{\Theta} \mathcal{L}_{\text{TRA}}(\{s_{t,i}, a_{t,i}, s_{t+k,i}, \ell_i\}_{i=1}^K; \Theta)$
  - 6: **output:** language  $\ell$ -conditioned policy  $\pi(a_t | s_t, \zeta(\ell))$
  - 7:     goal  $g$ -conditioned policy  $\pi(a_t | s_t, \psi(g))$
- 

A summary of our approach is shown in Algorithm 4.

### Temporal Alignment and Compositionality

We will formalize the intuition from Section 5.1 that TRA enables compositional generalization by considering the error on a “compositional” version of  $\mathcal{D}$ , denoted  $\mathcal{D}^*$ . Using the notation from Eq. (5.3), we can say  $\mathcal{D}$  is distributed according to:

$$\mathcal{D} \triangleq \mathcal{D}^H \sim \prod_{i=1}^K p_0(s_{1,i}) p_{\ell}(\ell_i | s_{1,i}, s_{H,i}) \prod_{t=1}^H \pi^E(a_{t,i} | s_{t,i}) P(s_{t+1,i} | s_{t,i}, a_{t,i}), \quad (5.12)$$

or equivalently

$$\mathcal{D}^H \sim \prod_{i=1}^K p_0(s_{1,i}) p_{\ell}(\ell_i | s_{1,i}, s_{H,i}) \prod_{t=1}^H e^{\sigma^2 \|\pi^E(s_{t,i}) - a_{t,i}\|^2} P(s_{t+1,i} | s_{t,i}, a_{t,i}), \quad (5.13)$$

by the isotropic Gaussian assumption. We will define  $\mathcal{D}^* \triangleq \mathcal{D}^{H'}$  to be a longer-horizon version of  $\mathcal{D}$  extending the behaviors gathered under  $\pi^E$  across a horizon  $\alpha H \geq H' \geq H$  that additionally satisfies a “time-isotropy” property: the marginal distribution of the states is uniform across the horizon, i.e.,  $p_0(s_{1,i}) = p_0(s_{t,i})$  for all  $t \in \{1 \dots H'\}$ .

We will relate the in-distribution imitation error  $\text{ERR}(\bullet; \mathcal{D})$  to the compositional out-of-distribution imitation error  $\text{ERR}(\bullet; \mathcal{D}^*)$ . We define

$$\text{ERR}(\hat{\pi}; \tilde{\mathcal{D}}) = \mathbb{E}_{\tilde{\mathcal{D}}} \left[ \frac{1}{H} \sum_{t=1}^H \mathbb{E}_{\hat{\pi}} \left[ \|\tilde{a}_{t,i} - \hat{\pi}(\tilde{s}_{t,i}, \tilde{s}_{H,i})\|^2 / d_{\mathcal{A}} \right] \right] \quad (5.14)$$



$$\text{for } \{\tilde{s}_{t,i}, \tilde{a}_{t,i}, \tilde{\ell}_i\}_{t=1}^H \sim \tilde{\mathcal{D}}. \quad (5.15)$$

On the training dataset this is equivalent to the expected behavioral cloning loss from Eq. (5.9).

**Assumption 5.1.** *The policy factorizes through inferred waypoints as:*

$$\text{goals: } \pi(a | s, g) = \int \pi(a | s, w) \mathbb{P}(s_t = w | s_{t+k} = g) \, dw \quad (5.16)$$

$$\text{language: } \pi(a | s, \ell) = \int \pi(a | s, w) \mathbb{P}(s_t = w | s_{t+k} = g) \mathbb{P}(s_{t+k} = g | \ell) \, dw \, dg, \quad (5.17)$$

where denote by  $\pi(s, g)$  the MLE estimate of the action  $a$ .

**Theorem 5.1.** *Suppose  $\mathcal{D}$  is distributed according to Eq. (5.12) and  $\mathcal{D}^*$  is distributed according to Eq. (5.12). When  $\gamma > 1 - 1/H$  and  $\alpha > 1$ , for optimal features  $\phi$  and  $\psi$  under Eq. (5.11), we have*

$$\text{ERR}(\pi; \mathcal{D}^*) \leq \text{ERR}(\pi; \mathcal{D}) + \frac{\alpha - 1}{2\alpha} + \left(\frac{\alpha - 2}{2\alpha}\right) \mathbf{1}\{\alpha > 2\}. \quad (5.18)$$

We can also define a notion of the language-conditioned compositional generalization error:

$$\text{ERR}^\ell(\pi; \mathcal{D}^*) \triangleq \mathbb{E}_{\mathcal{D}^*} \left[ \frac{1}{H} \sum_{t=1}^H \mathbb{E}_\pi [\|\tilde{a}_{t,i} - \pi(\tilde{s}_{t,i}, \tilde{\ell}_i)\|^2] \right]. \quad (5.19)$$

**Corollary 5.1.1.** *Under the same conditions as Theorem 5.1,*

$$\text{ERR}^\ell(\pi; \mathcal{D}^*) \leq \text{ERR}^\ell(\pi; \mathcal{D}) + \frac{\alpha - 1}{2\alpha} + \left(\frac{\alpha - 2}{2\alpha}\right) \mathbf{1}\{\alpha > 2\}. \quad (5.20)$$

The proofs as well as a visualization of the bound are in Section 5.4.

## 5.2 EXPERIMENTS

Our experimental evaluation aims to answer the following research questions for TRA:

1. Can TRA enable zero-shot composition of multiple sequential tasks without additional prompting or planning methods?
2. How well does TRA perform compared to conventional offline RL algorithms in terms of task generalization and composition?
3. How well does TRA capture skills that are seen at a lower percentage within the dataset, compared to the numerous entries of object manipulation?
4. Is time alignment by itself sufficient for effective compositional generalization?

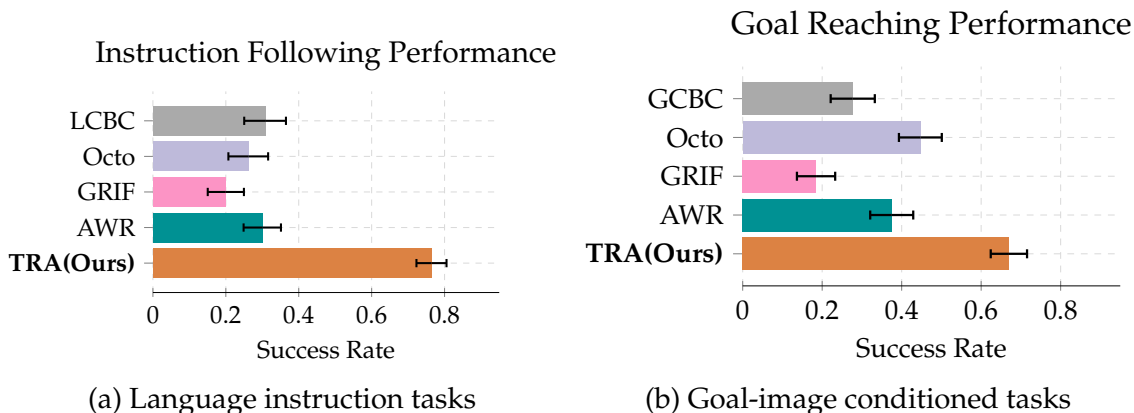


Figure 5.2: Aggregated performance on compositional generalization tasks, consisting of instruction-following and goal-reaching tasks.

Table 5.1: Compositional Generalization Error of Methods

Modality	TRA	GRIF	LCBC	GCBC	Octo
image	$4.25 \pm 0.37$	$5.24 \pm 0.34$	—	$4.84 \pm 0.11$	$5.15 \pm 0.41$
language	$3.82 \pm 0.25$	$4.95 \pm 0.32$	$4.84 \pm 0.11$	—	$4.56 \pm 0.32$

### Experimental Details

We evaluate TRA on a collection of held-out *compositionally-OOD* tasks – tasks for which the individual substeps are represented in the dataset, but the combination of those steps is unseen. For example, in a task such as “removing a bell pepper from a towel, and then sweep the towel”, both the tasks “remove the bell pepper from the towel” and “sweep the towel” have similar entries within BridgeData, but such combined trajectory and language description does not exist. We utilize a real-world robot manipulation interface with a 7 DoF WidowX250 manipulator arm with 5Hz execution frequency. We train on an augmented version of the BridgeDataV2 dataset [62], which contains over 50k trajectories with 72k language annotations. We augment the dataset by rephrasing the language annotations, as described by [7], with 5 additional rephrased language instruction for each language instruction present in the dataset, and randomly sample them during training.

In order to specifically test the ability of TRA to perform compositional generalization, we organize our evaluation tasks into 4 scenes that are unseen in BridgeData, each with increasing difficulty:

**Scene A – One-Step Drawer:** this is the only scene that are not compositionally-*OOD*, as all the tasks are one-step tasks. This scene involves opening, putting an item in, and closing a drawer. These tasks have been seen in BridgeData, although at a lower frequency than object manipulation, but the position in which they are

initialized are unseen. They will be used to compare TRA’s ability to baselines when solving single-step tasks.

**Scene B – Task Concatenation:** this scene involves concatenating multiple tasks of the same nature in sequence, where a robot must be able to perform all tasks within the same trajectory. During evaluation, we instruct the policy with instructions such as sweeping multiple objects in the scene that require composition (though are not sensitive to the *order* of the composition).

**Scene C – Semantic Generalization:** Unlike scene B, these tasks require manipulation with different objects of the same class. We test this using various food items seen within BridgeData and instruct the policy to put various food items within a container. An example of such task would be to have a table containing a banana, a sushi, a bowl, and various distractor objects, and instead of using specific language commands such as “put the banana and the sushi in the bowl”, a more general statement such as “put the food items in a container” will be used.

**Scene D – Tasks with Dependency:** This is the most challenging of the set of tasks: these tasks have subtasks that require previous subtasks being completed for them to succeed. An example of this would be to open a drawer, and to take out an item in the drawer, as one cannot take out an item from the drawer if the drawer is not open.

The complete list of tasks is noted in Appendix C.2.

## Baselines

We compare against the following baselines:

**GRIF [7]** learns a goal- and language- conditioned policy using aligned goal image and language representations. In our experiments, this becomes equivalent to TRA when the temporal alignment objective is removed.

**GCBC [62]** learns a goal-conditioned behavioral cloning policy that concatenates the goal image with the image observation.

**LCBC [62]** learns a language-conditioned policy that concatenates the language with the image observation.

**OCTO [67]** uses a multimodal transformer to learn a goal- and language-conditioned policy. The policy is trained on Open-X dataset [98], which incorporates Bridge-Data in its entirety.

**AWR [182]** uses advantages produced by a value function to effectively extract a policy from an offline dataset. In this experiment, we use the difference between the contrastive loss between the current observation and the goal representation and the contrastive loss between the next observation and the goal representation as a surrogate for value function.

We train GRIF, GCBC, LCBC, and AWR using the same augmented Bridge Dataset as TRA, and we use an Octo-Base 1.5 model for our evaluation. A more

Table 5.2: Real-world Language Conditioned Evaluation

Task	TRA	GRIF	LCBC	Octo	AWR
open the drawer	<b>0.80±0.1</b> <sup>†</sup>	0.20±0.2	<b>0.60±0.2</b>	<b>0.60±0.2</b>	0.40±0.2
mushroom in drawer	<b>0.80±0.1</b>	<b>0.80±0.2</b>	0.40±0.2	0.00±0.0	<b>0.60±0.2</b>
close drawer	<b>0.60±0.2</b>	<b>0.60±0.2</b>	<b>0.40±0.2</b>	<b>0.60±0.2</b>	<b>0.40±0.2</b>
(*) put the spoons on towels	<b>1.00±0.0</b>	0.40±0.2	0.20±0.2	0.00±0.0	0.20±0.2
(*) put the spoons on the plates	<b>0.80±0.2</b>	0.20±0.2	0.20±0.2	0.20±0.2	0.00±0.0
(*) fold cloth into the center	<b>1.00±0.0</b>	0.20±0.2	0.40±0.2	0.40±0.2	0.40±0.2
(*) sweep to the right	<b>0.80±0.1</b>	0.20±0.2	0.40±0.2	0.40±0.2	0.00±0.0
(*) put the corn and sushi on plate	<b>0.90±0.1</b>	0.00±0.0	0.40±0.2	0.00±0.0	0.50±0.2
(*) sushi and mushroom in bowl	<b>0.80±0.2</b>	0.00±0.0	<b>0.60±0.2</b>	0.20±0.2	<b>0.60±0.2</b>
(*) corn, banana, and sushi in bowl	<b>0.80±0.1</b>	0.00±0.0	0.00±0.0	0.00±0.0	0.20±0.1
(*) take the item out of the drawer	<b>0.60±0.2</b>	0.00±0.0	0.00±0.0	0.20±0.2	0.00±0.0
(*) move bell pepper and sweep towel	<b>0.50±0.2</b>	0.00±0.0	0.00±0.0	0.20±0.2	0.00±0.0
(*) corn on plate then sushi in pot	<b>0.70±0.1</b>	0.00±0.0	0.40±0.2	<b>0.60±0.2</b>	0.20±0.2

\* indicates task is compositionally-OOD (has multiple steps never seen together in training)

<sup>†</sup>The best-performing method(s) up to statistical significance are **highlighted**

detail approach is detailed in Appendix C.2. During evaluation, we give all policies the same goal state and language instruction regardless of the architecture, as they are trained on the same language instruction with the exception of Octo, which doesn't benefit from paraphrased language data, but does benefit from a more diverse language annotation set across a larger dataset of varying length and complexity.

## Experimental Evaluation

**Does TRA enable compositionality?** In Table 5.1, we compare the normalized mean squared error (MSE) of the TRA method with other methods on held-out compositionally-OOD image- and goal-specified tasks. These values are derived from passing the inputs through the policy network and sampling the mode of the distribution without unnormalizing the outputs based on the dataset. The validation MSE for these tasks are lower with a statistically significant margin, demonstrating that in a compositionally-OOD setting, TRA provides a trajectory closer to expert demonstrations.

Section 5.2 and Section 5.2 show the success rates of the TRA method compared to other methods on real-world robot evaluation tasks. We marked all policies within the task orange if they achieve the best statistically significant per-

Table 5.3: Real-world Goal-Conditioned Evaluation

Task	TRA	GRIF	GCBC	Octo	AWR
open the drawer	<b>0.60±0.2<sup>†</sup></b>	0.60±0.2	<b>0.40±0.2</b>	<b>0.50±0.2</b>	<b>0.80±0.2</b>
mushroom in drawer	<b>0.90±0.1</b>	0.40±0.2	<b>0.80±0.2</b>	<b>0.90±0.1</b>	<b>0.60±0.2</b>
close drawer	<b>1.00±0.0</b>	0.40±0.2	0.80±0.2	0.60±0.2	0.40±0.2
(*) put the spoons on towels	<b>1.00±0.0</b>	0.20±0.2	0.60±0.2	0.40±0.2	0.60±0.2
(*) put the spoons on the plates	<b>1.00±0.0</b>	0.00±0.0	0.40±0.2	0.00±0.0	0.80±0.2
(*) fold cloth into the center	<b>1.00±0.0</b>	0.00±0.0	0.00±0.0	0.60±0.2	0.00±0.0
(*) sweep to the right	<b>0.70±0.1</b>	0.40±0.2	0.00±0.0	<b>0.80±0.2</b>	0.00±0.0
(*) put the corn and sushi on plate	<b>0.70±0.1</b>	0.00±0.0	0.20±0.2	0.00±0.0	0.30±0.1
(*) sushi and mushroom in bowl	<b>0.60±0.2</b>	0.00±0.0	0.20±0.2	<b>0.40±0.2</b>	<b>0.60±0.2</b>
(*) corn, banana, and sushi in bowl	<b>0.50±0.2</b>	0.00±0.0	0.00±0.0	<b>0.40±0.2</b>	0.50±0.2
(*) take the item out of the drawer	<b>0.40±0.2</b>	0.00±0.0	0.00±0.0	0.20±0.2	0.00±0.0
(*) move bell pepper and sweep towel	<b>0.60±0.2</b>	0.20±0.2	0.20±0.2	<b>0.40±0.2</b>	0.00±0.0
(*) corn on plate then sushi in pot	<b>0.30±0.1</b>	<b>0.20±0.2</b>	0.00±0.0	0.00±0.0	0.00±0.0

\* indicates task is compositionally-OOD (has multiple steps never seen together in training)

<sup>†</sup>The best-performing method(s) up to statistical significance are **highlighted**

formance. We first compare the performance against methods in Scene A. We observe that while TRA performs well with drawer tasks, its performance against baseline methods are not statistically significant. However, when being evaluated on compositionally-OOD **instruction following** tasks, TRA performs considerably better than that of any baseline methods.

While TRA completed 88.9% of tasks seen in Scene B, 83.3% of evaluations in Scene C, and 60% of tasks in Scene D with instruction following, the best-performing baseline for Scene B was 30% with LCBC, 43.3% for Scene C with AWR, and 33.3% on Scene D with Octo. The same improvement was also present in goal reaching tasks, although at a lower level, in which Scene C produced 60% success rate and scene D produced a 43.3% success rate, as compared to 46.7% and 20% for the best-performing baselines.

Qualitatively, we see that policies trained under TRA provides a much smoother trajectory between different subtasks while following instructions, while other cannot replicate the same performance. Take removing the bell pepper + sweep task for example, with its visualization shown Fig. 5.3, while TRA was able to remove the bell pepper by grasping it and putting it to the bottom right corner of the table, LCBC cannot replicate the same performance, choosing to nudge the bell pepper instead and failed to execute the task.

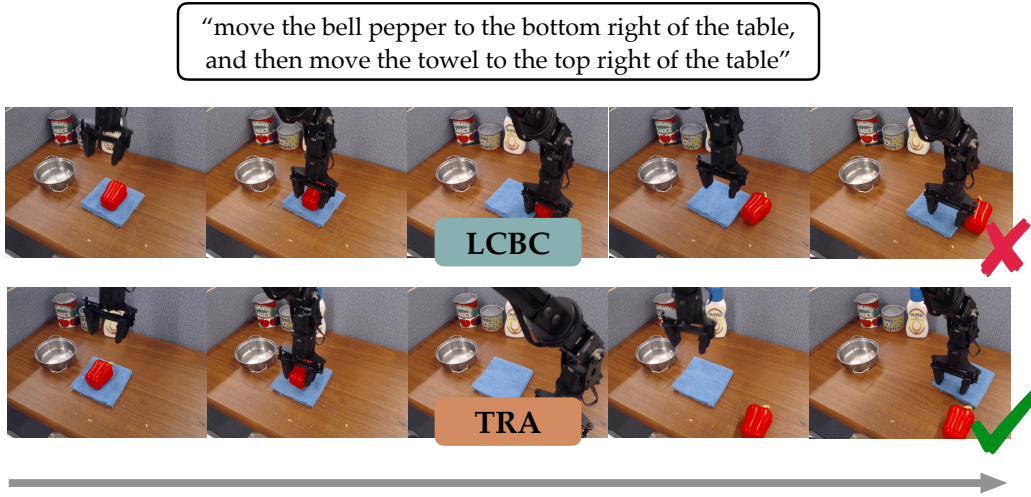


Figure 5.3: Example rollouts of a task with TRA and LCBC. While TRA is able to successfully compose the steps to complete the task, LCBC fails to ground the instruction correctly.

**How well does TRA perform against Conventional Offline RL Algorithms?** While offline reinforcement learning promises good stitching behavior [183], we demonstrate that TRA still outperforms offline reinforcement learning on robotic manipulation. Overall, TRA performs better than AWR for both language and image tasks, outperforming AWR by 45% on instruction following tasks, and by 25% on goal reaching tasks, showing considerable improvement over an offline RL method that promises compositional generalization via stitching.

Qualitatively, it is often seen that a policy trained with AWR would stop after one subtask, even though the goal instruction or image demanded all of the subtasks be completed. We can see this behavior in Fig. 5.1, in which we have the same goal image being fed in to 3 different policies in which all 3 food items must be put in the bowl. While TRA successfully completes all 3 subtasks, AWR chose to only complete one subtask and terminates right after putting the banana in the bowl.

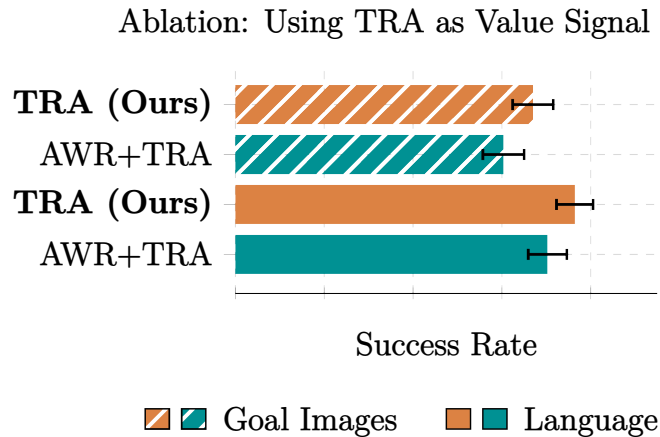


Figure 5.4: Aggregated success rate of using AWR as an additional policy learning metric over all 4 scenes.



This is due to the fact that AWR on an offline dataset has a goal-reaching reward function, in which it does not attempt to align the representations of all trajectories across time unlike TRA.

**Does TRA help capturing rarely-seen skills within the dataset?** We also compare the performance of TRA against AWR across all scenes and compare the performance of the policies with all 3 tasks in Scene D as well as folding the towel, all rarely seen skills within BridgeData, as it mainly focused on object manipulation. When compared by task within language conditioned set, we discover AWR suffered a significant drop off in effectiveness, with its average success rate plummeting from 43.3% in Scene C compared to 6.67% in Scene D, while TRA had a smaller drop off, from 83.3% to 60%, displaying that TRA generates better understanding of tasks that are rarely seen in the dataset. Other agents do not nearly achieve the same performance even as AWR in Scene D, as the lack of such compositional generalization prevented the policies from achieving all of the tasks at a reliable rate.

**Is TRA sufficient in achieving compositional generalization?** We demonstrate in our real-world experiment that only using temporal alignment is sufficient for achieving good compositional generalization. We evaluate this by comparing a policy trained on only temporal alignment loss (our method), and another policy trained on such loss and have these losses weighed by AWR.

Fig. 5.4 shows that across all evaluation tasks, there exists no statistically significant difference between using and not using AWR in addition to temporal alignment, in fact, using AWR marginally decreases the efficacy of TRA, as compared to showing marginal improvement over vanilla GCBC methods and a similar performance with vanilla LCBC methods. While TRA qualitatively improve the smoothness of the execution trajectories, the same cannot be said about using AWR, in which after executing every subtask, the robot chose to return near the starting joint angles before executing the next subtask.

### Failure Cases

While TRA provides an effective mechanism for compositional generalization, it is not immune to failures. Qualitatively, we observe that despite showing better compositional generalization, the policy still fails at a similar rate compared to other multivariate Gaussian policies when multimodal behavior is observed, other cases of early grasping and incorrect reaching are also observed at a similar rate. While TRA did provide marginal improvements as seen in Scene A, it does not provide full coverage of such scenarios. More analysis of failure cases can be seen in Section 5.3.

### 5.3 ADDITIONAL VISUALIZATIONS

In this section, we show additional visualizations of TRA’s execution on compositionally-OOD tasks. We use *folding*, *taking mushroom out of the drawer*, and *corn on plate, then sushi in the pot* as examples, as these tasks require a strong degree of dependency to complete at Section 5.3.

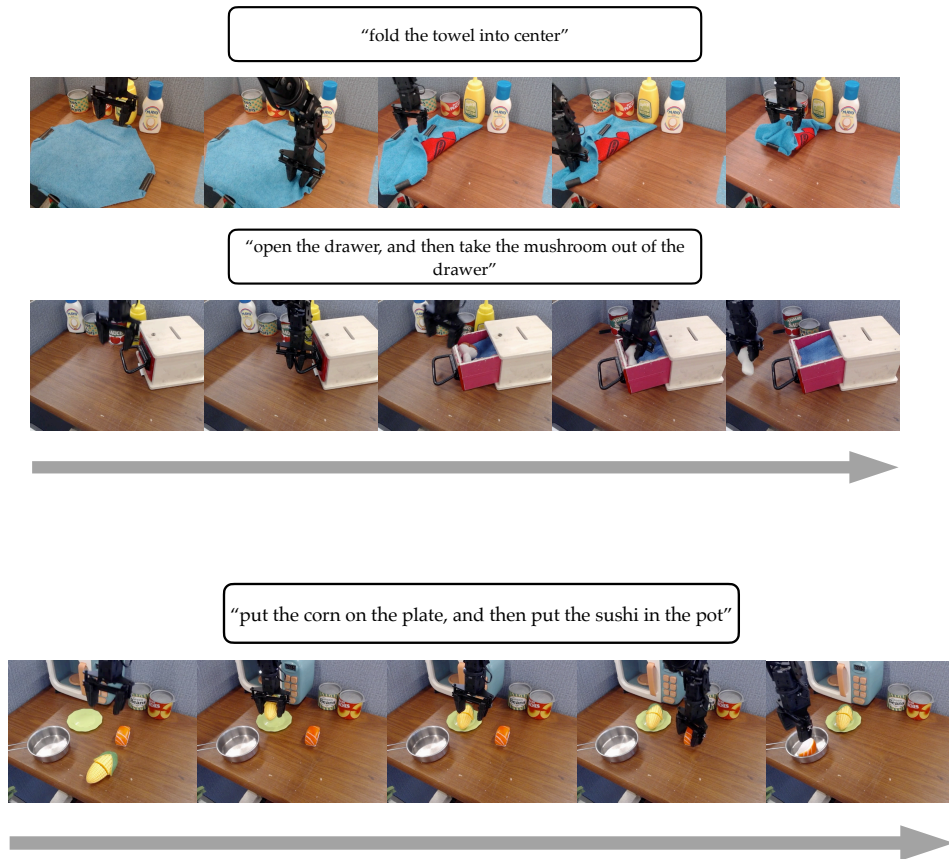


Figure 5.5: In these figures, we see that TRA is able to perform good compositional generalization over a variety of tasks seen within BridgeData

#### Failure Cases

We break down failure cases in this section. While TRA performs well in compositional generalization, it cannot counteract against previous failures seen with behavior cloning with a Gaussian Policy.

### 5.4 ANALYSIS OF COMPOSITIONALITY

We prove the results from Section 5.1.



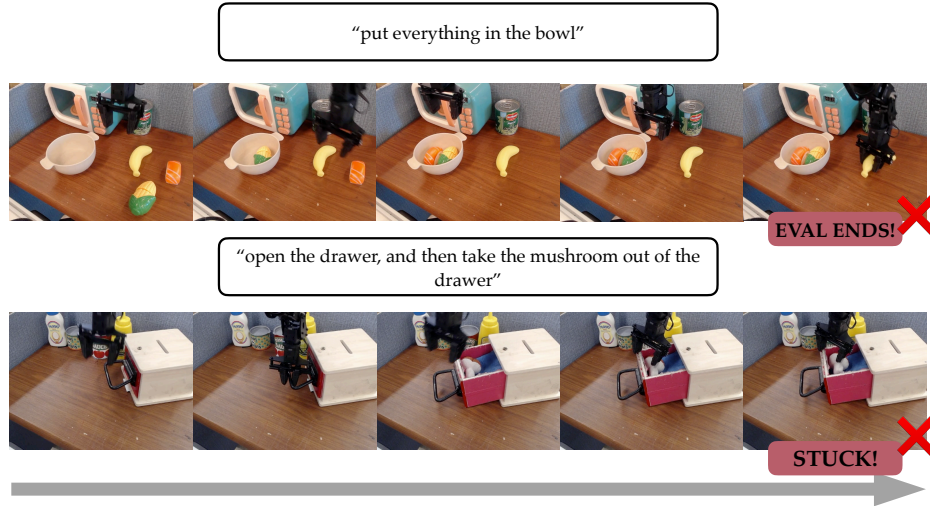


Figure 5.6: Most of the failure cases came from the fact that a policy cannot learn depth reasoning, causing early grasping or late release, and it has trouble reconciling with multimodal behavior

### Goal Conditioned Analysis

**Theorem 5.1.** *Suppose  $\mathcal{D}$  is distributed according to Eq. (5.12) and  $\mathcal{D}^*$  is distributed according to Eq. (5.12). When  $\gamma > 1 - 1/H$  and  $\alpha > 1$ , for optimal features  $\phi$  and  $\psi$  under Eq. (5.11), we have*

$$\text{ERR}(\pi; \mathcal{D}^*) \leq \text{ERR}(\pi; \mathcal{D}) + \frac{\alpha - 1}{2\alpha} + \left(\frac{\alpha - 2}{2\alpha}\right) \mathbf{1}\{\alpha > 2\}. \quad (5.18)$$

*Proof.* We have from Eq. (5.15) for  $K \sim \text{Geom}(1 - \gamma)$ :

$$\begin{aligned} \text{ERR}(\pi; \mathcal{D}^*) &\triangleq \mathbb{E}_{\mathcal{D}^*} \left[ \frac{1}{H'} \sum_{t=1}^{H'} \frac{\|\tilde{a}_{t,i} - \pi(\tilde{s}_{t,i}, \tilde{g}_i)\|^2}{n_{d_A}} \right] \\ &= \mathbb{E}_{\mathcal{D}^*} \left[ \sum_{t=1}^{H'-2H} \frac{\|\tilde{a}_{t,i} - \pi(\tilde{s}_{t,i}, \tilde{g}_i)\|^2}{n_{d_A}} \right] + \mathbb{E}_{\mathcal{D}^*} \left[ \sum_{t=H'-2H+1}^{H'-H} \frac{\|\tilde{a}_{t,i} - \pi(\tilde{s}_{t,i}, \tilde{g}_i)\|^2}{n_{d_A}} \right] \\ &\quad + \mathbb{E}_{\mathcal{D}^*} \left[ \sum_{t=H'-H+1}^{H'} \frac{\|\tilde{a}_{t,i} - \pi(\tilde{s}_{t,i}, \tilde{g}_i)\|^2}{n_{d_A}} \right] \\ &\leq \mathbb{E}_{\mathcal{D}^*} \left[ \sum_{t=H'-H+1}^{H'} \frac{\|\tilde{a}_{t,i} - \pi(\tilde{s}_{t,i}, \tilde{g}_i)\|^2}{n_{d_A}} \right] + \mathbb{E}_{\mathcal{D}^*} \left[ \sum_{t=H'-2H+1}^{H'-H} \frac{\|\tilde{a}_{t,i} - \pi(\tilde{s}_{t,i}, \tilde{g}_i)\|^2}{n_{d_A}} \right] + \left(\frac{\alpha - 2}{2}\right) \mathbf{1}\{\alpha > 2\} \end{aligned}$$

$$\begin{aligned}
&\leq \mathbb{E}_{\mathcal{D}^*} \left[ \sum_{t=H'-H+1}^{H'} \frac{\|\tilde{a}_{t,i} - \pi(\tilde{s}_{t,i}, \tilde{s}_{H',i})\|^2}{n_{d_{\mathcal{A}}}} \right] \\
&\quad + \mathbb{E}_{\mathcal{D}^*} \left[ \sum_{t=H'-2H+1}^{H'-H} \mathbb{E}_K \left[ \frac{\|\tilde{a}_{t,i} - p^\pi(\tilde{s}_{t,i} | \tilde{s}_{H'-K,i})\|^2}{n_{d_{\mathcal{A}}}} \right] \right] + \left( \frac{\alpha - 2}{2} \right) \mathbf{1}\{\alpha > 2\} \\
&\leq \mathbb{E}_{\mathcal{D}^*} \left[ \sum_{t=H'-H+1}^{H'} \frac{\|\tilde{a}_{t,i} - \pi(\tilde{s}_{t,i}, \tilde{s}_{H',i})\|^2}{n_{d_{\mathcal{A}}}} \right] \\
&\quad + \mathbb{E}_{\mathcal{D}^*} \left[ \sum_{t=H'-2H+1}^{H'-H} \mathbb{E}_K \left[ \frac{\|\tilde{a}_{t,i} - p^\pi(\tilde{s}_{t,i} | \tilde{s}_{H'-K,i})\|^2}{n_{d_{\mathcal{A}}}} \right] \right] + \left( \frac{\alpha - 2}{2} \right) \mathbf{1}\{\alpha > 2\} \\
&\leq \mathbb{E}_{\mathcal{D}^*} \left[ \sum_{t=H'-H+1}^{H'} \frac{\|\tilde{a}_{t,i} - \pi(\tilde{s}_{t,i}, \tilde{s}_{H',i})\|^2}{n_{d_{\mathcal{A}}}} \right] \\
&\quad + \mathbb{E}_{\mathcal{D}^*} \left[ \sum_{t=H'-2H+1}^{H'-H} \mathbb{E}_K \left[ \frac{\|\tilde{a}_{t,i} - p^\pi(\tilde{s}_{t,i} | \psi(\tilde{s}_{H'-K,i}))\|^2}{n_{d_{\mathcal{A}}}} \right] \right] + \left( \frac{\alpha - 2}{2} \right) \mathbf{1}\{\alpha > 2\} \\
&\leq \text{ERR}(\pi; \mathcal{D}^*) + \mathbb{E}_{\mathcal{D}^*} \left[ \frac{1 - \gamma^H}{1 - \gamma} \right] + \left( \frac{\alpha - 2}{2} \right) \mathbf{1}\{\alpha > 2\} \\
&\leq \text{ERR}(\pi; \mathcal{D}^*) + \frac{\alpha - 1}{2\alpha} + \left( \frac{\alpha - 2}{2} \right) \mathbf{1}\{\alpha > 2\}. \tag{5.21}
\end{aligned}$$

□

## Language Conditioned Analysis

**Corollary 5.1.1.** *Under the same conditions as Theorem 5.1,*

$$\text{ERR}^\ell(\pi; \mathcal{D}^*) \leq \text{ERR}^\ell(\pi; \mathcal{D}) + \frac{\alpha - 1}{2\alpha} + \left( \frac{\alpha - 2}{2\alpha} \right) \mathbf{1}\{\alpha > 2\}. \tag{5.20}$$

The proof is similar to Section 5.4, but over the predictions of  $\zeta$  instead of  $\psi$ .

## Visualizing the Bound

We compare the bound from Theorem 5.1 with the “worst-case” bound of  $\text{ERR}(\pi; \mathcal{D}^*) - \text{ERR}(\pi; \mathcal{D})$  in Fig. 5.7. The bound from Theorem 5.1 is tighter than the worst-case bound, and it shows that the compositional generalization error decreases as  $\alpha$  increases.

## 5.5 RELATED WORK

Our approach builds upon prior work on goal- and language-conditioned control, focusing particularly on the problem of compositional generalization.

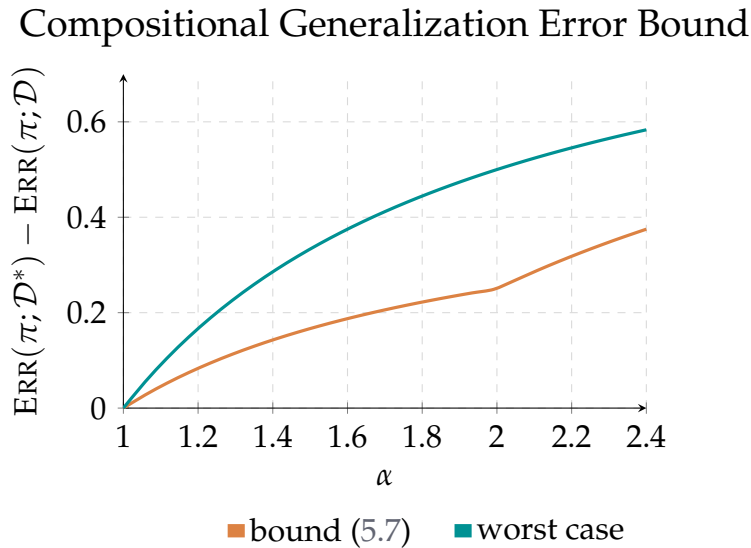


Figure 5.7: Visualizing the bound (Fig. 5.7 from Theorem 5.1) on the compositional generalization error.

**Robot manipulation with language and goals.** Recent improvements in robot learning datasets have enabled the development of robot policies that can be commanded with image goals and language instructions [90, 62, 29]. These policies can be trained with goal- and language-conditioned imitation learning from human demonstrations [184, 26, 51, 86, 85], reinforcement learning [185, 141], or other forms of supervision [186, 187]. When being trained to reach goals, methods can additionally use hindsight relabeling [14, 49] to improve performance [62, 7, 188, 189]. Our work shows how the benefits of goal-conditioned and language-conditioned supervised learning can be combined with temporal representation alignment to enable compositionality that would otherwise require planning or reinforcement learning.

**Compositional generalization in sequential decision making.** In the context of decision making, compositional generalization refers to the ability to generalize to new behaviors that are composed of known sub-behaviors [190, 191]. Biological learning systems show strong compositional generalization abilities [179, 188, 192, 63], and recent work has explored how similar capabilities can be achieved in artificial systems [193, 194, 56]. In the context of policy learning, exploiting the compositionality of the behaviors can lead to generalization to unseen and temporarily extended tasks [1, 195, 196, 170, 197, 198]. Hierarchical and planning-based approaches also aim to enable compositional behavior by explicitly partitioning a task into its components [199, 8, 200, 165]. With improvements in vision-language models (VLMs), many recent works have explored using a pre-trained VLM to

decompose a task into subtasks that are more attainable for the low-level manipulation policy [90, 35, 91, 107, 8, 38, 201]. Our contribution is to show compositional properties can be achieved *without* any explicit hierarchical structure or planning, by learning a structured representation through time-contrastive representation alignment.

**Representation learning for states and tasks.** State and task representations for decision making aim to improve generalization and exploit additional sources of data. Recent work in the robotics domain have explored the use of pre-trained representations across multimodal data, including images and language, for downstream tasks [202, 203, 46, 7, 16, 115, 204, 47, 23]. In reinforcement learning problems, representations are often trained to predict future states, rewards, goals, or actions [205, 119, 206, 44], and can improve generalization and sample efficiency when used as value functions [207–211]. Some recent works have explored the use of additional structural constraints on representations to enable planning [199, 200, 11, 212], or enforced metric properties to improve compositional generalization [127, 9, 126].

The key distinction between our approach and past contrastive representation methods for robotics like VIP [119], GRIF [7], and R3M [16] is that we focus on the real-world compositional generalization capabilities enabled by simply aligning representations across time in addition to the task modalities, without using the learned representations for policy extraction or defining a value function.

## 5.6 CONCLUSIONS AND LIMITATIONS

In this paper, we studied the effects of adding a temporal representation alignment objective in behavior cloning, and we have discovered that by adding this metric, it allows a robot policy to perform robust compositional generalization even when the composition of such tasks are OOD.

Although TRA demonstrates strong performance, there are few limitations remain. First, due to restrictions placed by dataloaders, TRA cannot handle extremely long sequence of language, even though the difficulty of subtasks contained within the instructions still remain easy. It also needs to be shown that such method will be helpful for executing long-horizon tasks with bimanual manipulators or enable cross-embodiment generalization. An interesting future development for this method would look into these directions and also create such compositional generalization across multiple embodiments.

# 6 PLANNING WITH CONTRASTIVE REPRESENTATIONS

Probabilistic modeling of time-series data has applications ranging from robotic control [213] to material science [214], from cell biology [215] to astrophysics [216]. These applications are often concerned with two questions: *predicting* future states (e.g., what will this cell look like in an hour), and *inferring* trajectories between two given states. However, answering these questions often requires reasoning over high-dimensional data, which can be challenging as most tools in the standard probabilistic toolkit require generation. Might it be possible to use discriminative methods (e.g., contrastive learning) to perform such inferences?

Many prior works aim to learn representations that are easy to predict while retaining salient bits of information. For time-series data, we want the representation to remain a sufficient statistic for distributions related to time – for example, they should retain bits required to predict future states (or representations thereof). While generative methods [217–220] have this property, they tend to be computationally expensive (see, e.g., [221]) and can be challenging to scale to high-dimensional observations.

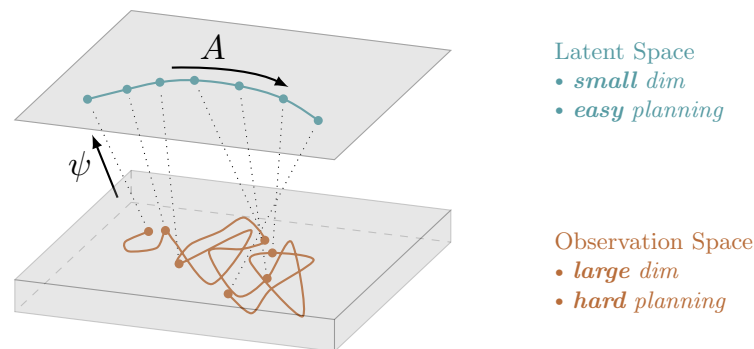


Figure 6.1: We apply temporal contrastive learning to observation pairs to obtain representations  $(\psi(x_0), \psi(x_{t+k}))$  such that  $A\psi(x_0)$  is close to  $\psi(x_{t+k})$ . While inferring waypoints in the high-dimensional observation space is challenging, we show that the distribution over intermediate latent representations has a closed form solution corresponding to linear interpolation between the initial and final representations.

We will study how contrastive methods (which are discriminative, rather than generative) can perform inference over time series. Ideally, we want representations of observations  $x$  to be a sufficient statistic for temporal relationships (e.g., does  $x'$  occur after  $x$ ?) but need not retain other information about  $x$  (e.g. the location of static objects). This intuition motivates us to study how contrastive representation learning methods [152, 128, 222–224] might be used to solve prediction and planning problems on time series data. While prior works in computer vision [159, 152] and natural language processing (NLP) [225] often study the geometry of learned representations, our results show how geometric operations such as interpolation are related to inference. Our analysis will focus on a regularized version of the symmetrized infoNCE objective [17], generating positive examples by sampling pairs of observations from the same time series data. We will study how representations learned in this way can facilitate two inference questions: prediction and planning.<sup>1</sup> As a stepping stone, we will build upon prior work [161] to show that regularized contrastive learning should produce representations whose marginal distribution is an isotropic Gaussian distribution.

The main contribution of this chapter is to demonstrate how intermediate and future time steps in a time series can be inferred easily using contrastive representations. This inference problem captures a number of practical tasks: interpolation, in-filling, and even planning and control, where the intermediate steps represent states between a start and goal. While ordinarily these problems require an iterative inference or optimization procedure, with contrastive representations this can be done simply by inverting a low-dimensional matrix. In one special case, inference will correspond to linear interpolation. Our first step is to prove that, under certain assumptions, the distribution over future representations has a Gaussian distribution, with a mean that is a linear function of the initial state representation (Lemma 6.1). This paves the way to our main result (Theorem 6.2): *given an initial and final state, we show that the posterior distribution over an intermediate state representations also follows a Gaussian distribution*. Said in other words, the representations follow a Gauss-Markov chain,<sup>2</sup> wherein any joint or conditional distribution can be computed by inverting a low-dimensional matrix [230, 231] (See Fig. 6.1). In one special case, inference will correspond to linearly interpolating between the representations of an initial state and final state. Section 6.4 provides numerical experiments.

---

<sup>1</sup>Following prior work [226, 227], we will use *planning* to refer to the problem of inferring intermediate states, not to refer to an optimal control problem.

<sup>2</sup>This probabilistic model is equivalent to a discretized Ornstein-Uhlenbeck process [228] and is also known as an AR(1) model [229, Eq. 3.1.16].

## 6.1 PRELIMINARIES

Our aim is to learn representations of time series data such that the spatial arrangement of representations corresponds to the temporal arrangement of the underlying data: if one example occurs shortly after another, then they should be mapped to similar representations. This problem setting arises in many areas, including video understanding and reinforcement learning. To define this problem formally, we will define a Markov process with states  $x_t$  indexed by time  $t$ :<sup>3</sup>  $p(x_{1:T} | x_0) = \prod_{t=0}^{T-1} p(x_{t+1} | x_t)$ . The dynamics  $p(x_{t+1} | x_t)$  tell us the immediate next state, and we can define the distribution over states  $t$  steps in the future by marginalizing over the intermediate states,  $p_t(x_t | x_0) = \int p(x_{1:t} | x_0) dx_{1:t-1}$ . A key quantity of interest will be the  $\gamma$ -discounted state occupancy measure, which corresponds to a time-averaged distribution over future states:

$$p_{t+}(x_{t+} = x) = (1 - \gamma) \sum_{t=0}^{\infty} \gamma^t p_t(x_t = x). \quad (6.1)$$

**Contrastive learning.** Our analysis will focus on applying contrastive learning to a particular data distribution. Contrastive learning [156, 152, 232] acquires representations using “positive” pairs  $(x, x^+)$  and “negative” pairs  $(x, x^-)$ . While contrastive learning typically learns just one representation, we will use two different representations for the two elements of the pair; that is, our analysis will use terms like  $\phi(x)$ ,  $\psi(x^+)$  and  $\psi(x^-)$ . We assume all representations lie in  $\mathbb{R}^k$ .

The aim of contrastive learning is to learn representations such that positive pairs have similar representations ( $\phi(x) \approx \psi(x^+)$ ) while negative pairs have dissimilar representations ( $\phi(x) \neq \psi(x^-)$ ). Let  $p(x, x^+)$  be the joint distribution over positive pairs (i.e.,  $(x, x^+) \sim p(x, x^+)$ ). We will use the product of the marginal distributions to sample negative pairs ( $(x, x^-) \sim p(x)p(x^-)$ ). Let  $B$  be the batch size, and note that the positive samples  $x_j^+$  at index  $j$  in the batch serve as *negatives* for  $x_i$  for any  $i \neq j$ . Our analysis is based on the infoNCE objective without resubstitution [128, 152]:

$$\max_{\phi(\cdot), \psi(\cdot)} \mathbb{E}_{\{(x_i, x_i^+)\}_{i=1}^B \sim p(x, x^+)} \left[ \sum_{i=1}^B \log \frac{e^{-\frac{1}{2} \|\phi(x_i) - \psi(x_i^+)\|_2^2}}{\sum_{j \neq i} e^{-\frac{1}{2} \|\phi(x_i) - \psi(x_j^+)\|_2^2}} + \log \frac{e^{-\frac{1}{2} \|\phi(x_i) - \psi(x_i^+)\|_2^2}}{\sum_{j \neq i} e^{-\frac{1}{2} \|\phi(x_j) - \psi(x_i^+)\|_2^2}} \right] \quad (6.2)$$

We will use the symmetrized version of this objective [17], where the denominator is the sum across rows of a logits matrix and once where it is a sum across the.

While contrastive learning is typically applied to an example  $x$  and an augmentation  $x^+ \sim p(x | x)$  of that same example (e.g., a random crop), we will follow prior work [157, 152] in using the time series *dynamics* to generate the positive

<sup>3</sup>This can be extended to *controlled* Markov processes appending the previous action to the observations.



pairs, so  $x^+$  will be an observation that occurs temporally after  $x$ . While our experiments will sample positive examples from the discounted state occupancy measure ( $x^+ \sim p_{t+}(x_{t+} | x)$ ) in line with prior work [121], our analysis will also apply to different distributions (e.g., always sampling a state  $k$  steps ahead).

While prior work typically constrains the representations to have a constant norm (i.e., to lie on the unit hypersphere) [152], we will instead constrain the *expected* norm of the representations is bounded, a difference that will be important for our analysis:

$$\frac{1}{k} \mathbb{E}_{p(x)} [\|\psi(x)\|_2^2] \leq c. \quad (6.3)$$

Because the norm scales with the dimension of the representation, we have scaled down the left side by the representation dimension,  $k$ . In practice, we will impose this constraint by adding a regularization term  $\lambda \mathbb{E}_{p(x)} [\|\psi(x)\|_2^2]$  to the infoNCE objective (Eq. 6.2) and dynamically tuning the weight  $\lambda$  via dual gradient descent.

### Key assumptions

This section outlines the two key assumptions behind our analysis, both of which have some theoretical justification. Our main assumption examines the distribution over representations:

**Assumption 6.1.** *Regularized, temporal contrastive learning acquires representations whose marginal distribution representations  $p(\psi) \triangleq \int p(x) \mathbf{1}(\psi(x) = \psi) dx$  is an isotropic Gaussian distribution:*

$$p(\psi) = \mathcal{N}(\psi; \mu = 0, \sigma = c \cdot I). \quad (6.4)$$

In Section 6.3 we extend prior work [161] provide some theoretical intuition for why this assumption should hold: namely, that the isotropic Gaussian is the distribution that maximizes entropy subject to an expected L2 norm constraint (Eq. 6.3) [233–235]. Our analysis also assumes that the learned representations converge to the theoretical minimizer of the infoNCE objective:

**Assumption 6.2.** *Applying contrastive learning to the symmetrized infoNCE objective results in representations that encode a probability ratio:*

$$e^{-\frac{1}{2} \|\phi(x_0) - \psi(x)\|_2^2} = \frac{p_{t+}(x_{t+} = x | x_0)}{p(x)C}. \quad (6.5)$$

This assumption holds under ideal conditions [52, 236] (see Section 6.3),<sup>4</sup> but we nonetheless call this an “assumption” because it may not hold in practice due to sampling and function approximation error. This assumption means the learned

<sup>4</sup>While the result of Ma and Collins [52] has  $C(x)$  depending on  $x$ , the symmetrized version [17] removes the dependence on  $x$ .



representations are sufficient statistics for predicting the probability (ratio) of future states: these representations must retain all the information pertinent to reasoning about *temporal* relationships, but need not retain information about the precise contents of the observations. As such, they may be much more compressed than representations learned via reconstruction.

Combined, these assumptions will allow us to express the distribution over sequences of representations as a Gauss-Markov chain. The denominator in Assumption 6.2,  $p(x)$ , may have a complex distribution, but Assumption 6.1 tells us that the distribution over *representations* has a simpler form. This will allow us to rearrange Assumption 6.2 to express the conditional distribution over representations as the product of two Gaussian likelihoods. Note that the left hand side of Assumption 6.2 already looks like a Gaussian likelihood.

## 6.2 CONTRASTIVE REPRESENTATIONS MAKE INFERENCE EASY

In this section, our main result will be to show how representations learned by (regularized) contrastive learning are distributed according to a Gauss-Markov chain, making it straightforward to perform inference (e.g., planning, prediction) over these representations. Our proof technique will combine (known) results about Gaussian distributions with (known) results about contrastive learning. We start by discussing an important choice of parametrization (Section 6.2) that facilitates prediction (Section 6.2) before presenting the main result in Section 6.2.

### A Parametrization for Shared Encoders

This section describes the two encoders ( $\psi(\cdot), \phi(\cdot)$ ) to compute representations of  $x$  and  $x^+$ . While prior work in computer vision and NLP literature use the same encoder for both  $x$  and  $x^+$ , this decision does not make sense for many time-series data as it would imply that our prediction for  $p(x_t | x_0)$  is the same as our prediction for  $p(x_0 | x_t)$ . However, the difficulty of transiting from  $x_0$  to  $x_t$  (e.g., climbing to the peak of a mountain) might be more difficult than the reverse (e.g., sledding down a mountain). Our proposed parametrization will handle this asymmetry.

We will treat the encoder  $\psi(\cdot)$  as encoding the contents of the state. We will additionally learn a matrix  $A$  so that the function  $\psi \mapsto A\psi$  corresponds to a (multi-step) prediction of the future representation. To map this onto contrastive learning, we will use  $\phi(x) \triangleq A\psi(x)$  as the encoder for the initial state. One way of interpreting this encoder is as an additional linear projection applied on top of  $\psi(\cdot)$ , a design similar to those used in other areas of contrastive learning [142]. Once learned,

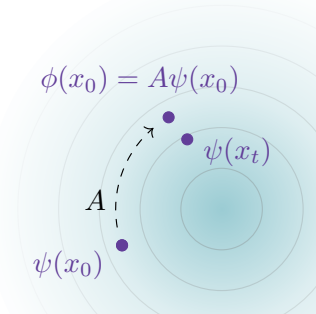


Figure 6.2: A parametrization for temporal contrastive learning.

we can use these encoders to answer questions about prediction (Section 6.2) and planning (Section 6.2).

### Representations Encode a Predictive Model

Given an initial state  $x_0$ , what states are likely to occur in the future? Answering this question directly in terms of high-dimensional states is challenging, but our learned representations provide a straightforward answer.

Let  $\psi_0 = \psi(x_0)$  and  $\psi_{t+} = \psi(x_{t+})$  be random variables representing the representations of the initial state and a future state. Our aim is to estimate the distribution over these future representations,  $p(\psi_{t+} | \psi_0)$ . We will show that the learned representations encode this distribution.

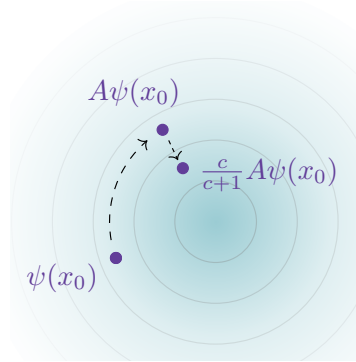


Figure 6.3: Predicting representations of future states.

**Lemma 6.1.** *Under the assumptions from Section 6.1, the distribution over representations of future states follows a Gaussian distribution with mean parameter given by the initial state representation:*

$$p(\psi_{t+} = \psi | \psi_0) = \mathcal{N}\left(\mu = \frac{c}{c+1}A\psi_0, \Sigma = \frac{c}{c+1}I\right). \quad (6.6)$$

The main takeaway here is that the distribution over future representations has a convenient, closed form solution. The representation norm constraint,  $c$ , determines the shrinkage factor  $\frac{c}{c+1} \in [0, 1)$ ; highly regularized settings (small  $c$ ) move the mean closer towards the origin and decrease the variance, as visualized in Fig. 6.3. Regardless of the constraint  $c$ , the predicted mean is a linear function  $\psi \mapsto \frac{c}{c+1}A\psi$ . The proof is in Section 6.3. The proof technique is similar to that of the law of the unconscious statistician.

### Planning over One Intermediate State

We now show how these representations can be used for a specific type of planning: given an initial state  $x_0$  and a future state  $x_{t+}$ , infer the representation of an intermediate “waypoint” state  $x_w$ . The next section will extend this analysis to inferring the entire sequence of intermediate states. We assume  $x_0 \rightarrow x_w \rightarrow x_{t+}$  form a Markov chain where  $x_w \sim p(x_{t+} | x_0 = x_0)$  and  $x_{t+} \sim p(x_{t+} | x_0 = x_w)$  are both drawn from the discounted state occupancy measure (Eq. 6.1). Let random variable  $\psi_w = \psi(x_w)$  be the representation of this intermediate state. Our main result is that the posterior distribution over waypoint representations has a closed form solution in terms of the initial state representation and future state representation:

**Theorem 6.2.** *Under Assumptions 1 and 2, the posterior distribution over waypoint representations is a Gaussian whose mean and covariance are linear functions of the initial and final state representations:*

$$p(\psi_w | \psi_0, \psi_{t+}) = \mathcal{N}\left(\psi_w; \mu = \Sigma(A^T \psi_{t+} + A\psi_0), \Sigma^{-1} = \frac{c}{c+1}A^T A + \frac{c+1}{c}I\right).$$

The proof (Section 6.3) uses the Markov property together with Lemma 6.1. The main takeaway from this lemma is that the posterior distribution takes the form of a simple probability distribution (a Gaussian) with parameters that are linear functions of the initial and final representations.

We give three examples to build intuition:

**Example 1:**  $A = I$  and the  $c$  is very large (little regularization). Then, the covariance is  $\Sigma^{-1} \approx 2I$  and the mean is the simple average of the initial and final representations  $\mu \approx \frac{1}{2}(\psi_0 + \psi_{t+})$ . In other words, the waypoint representation is the midpoint of the line  $\psi_0 \rightarrow \psi_{t+}$ .

**Example 2:**  $A$  is a rotation matrix and  $c$  is very large. Rotation matrices satisfy  $A^T = A^{-1}$  so the covariance is again  $\Sigma^{-1} \approx 2I$ . As noted in Section 6.2, we can interpret  $A\psi_0$  as a *prediction* of which representations will occur after  $\psi_0$ . Similarly,  $A^{-1}\psi_{t+} = A^T \psi_{t+}$  is a prediction of which representations will occur before  $\psi_{t+}$ . Theorem 6.2 tells us that the mean of the waypoint distribution is the simple average of these two predictions,  $\mu \approx \frac{1}{2}(A^T \psi_{t+} + A\psi_0)$ .

**Example 3:**  $A$  is a rotation matrix and  $c = 0.01$  (very strong regularization). In this case  $\Sigma^{-1} = \frac{0.01}{0.01+1}A^T A + \frac{0.01+1}{0.01}I \approx 100I$ , so  $\mu \approx \frac{1}{100}(\psi_0 + \psi_{t+}) \approx 0$ . Thus, in the case of strong regularization, the posterior concentrates around the origin.

### Planning over Many Intermediate States

This section extends the analysis to multiple intermediate states. Again, we will infer the posterior distribution of the representations of these intermediate states,  $\psi_{w_1}, \psi_{w_2}, \dots$ . We assume that these states form a Markov chain.

**Theorem 6.3.** *Given observations from a Markov chain  $x_0 \rightarrow x_1 \dots x_{t+}$ , the joint distribution over representations is a Gaussian distribution. Using  $\psi_{1:n} = (\psi_{w_1}, \dots, \psi_{w_n})$  to denote the concatenated representations of each observation, we can write this distribution as*

$$p(\psi_{1:n}) \propto \exp\left(-\frac{1}{2}\psi_{1:n}^T \Sigma^{-1} \psi_{1:n} + \eta^T \psi_{1:n}\right),$$

where  $\Sigma^{-1}$  is a tridiagonal matrix

$$\Sigma^{-1} = \begin{pmatrix} \frac{c}{c+1}A^T A + \frac{c+1}{c}I & & & & \\ & -A & & & \\ & & \frac{c}{c+1}A^T A + \frac{c+1}{c}I & -A^T & \\ & & & \ddots & \\ & & & & \ddots \end{pmatrix} \quad \text{and } \eta = \begin{pmatrix} A\psi_0 \\ 0 \\ \vdots \\ A^T \psi_{t+} \end{pmatrix}.$$

This distribution can be written in the canonical parametrization as  $\Sigma = \Lambda^{-1}$  and  $\mu = \Sigma\eta$ . Recall that Gaussian distributions are closed under marginalization. Thus, once in this canonical parametrization, the marginal distributions can be obtained by reading off individual entries of these parameters:

$$p(\psi_i | \psi_0, \psi_{t+}) = \mathcal{N}\left(\psi_i; \mu_i = (\Sigma\eta)^{(i)}, \Sigma_i = (\Lambda^{-1})^{(i,i)}\right).$$

The key takeaway here is that this posterior distribution over waypoints is Gaussian, and it has a closed form expression in terms of the initial and final representations (as well as regularization parameter  $c$  and the learned matrix  $A$ ).

In the general case of  $n$  intermediate states, the posterior distribution is

$$p(\psi_{w_1} \cdots \psi_{w_n} | \psi_0, \psi_{t+}) \propto e^{-\frac{1+\frac{1}{c}}{2} \sum_{i=1}^n \|\frac{c}{c+1} A\psi_{w_i} - \psi_{w_{i+1}}\|_2^2},$$

where  $\psi_{w_0} = \psi_0$  and  $\psi_{w_{n+1}} = \psi_{t+}$ . This corresponds to a chain graphical model with edge potentials  $f(\psi, \psi') = e^{-\frac{1+\frac{1}{c}}{2} \|\frac{c}{c+1} A\psi - \psi'\|_2^2}$ .

**Special case.** To build intuition, consider the special case where  $A$  is a rotation matrix and  $c$  is very large, so  $\frac{c}{c+1} A^T A + \frac{c+1}{c} \approx 2I$ . In this case,  $\Sigma^{-1}$  is a (block) second difference matrix [237]:

$$\Sigma^{-1} = \begin{pmatrix} 2I & -I & & \\ -I & 2I & -I & \\ & & \ddots & \ddots \end{pmatrix}.$$

The inverse of this matrix has a closed form solution [238, Pg. 471], allowing us to obtain the mean of each waypoint in closed form:

$$\mu_i = (1 - \lambda(i))A\psi_0 + \lambda(i)A^T\psi_{t+}, \quad (6.7)$$

where  $\lambda(i) = \frac{i}{n+1}$ . Thus, each posterior mean is a convex combination of the (forward prediction from the) initial representation and the (backwards prediction from the) final representation. When  $A$  is the identity matrix, the posterior mean is simple linear interpolation between the initial and final representations!

### 6.3 PROOFS

Marginal Distribution over Representations is Gaussian

The infoNCE objective (Eq. (6.2)) can be decomposed into an alignment term and a uniformity term [161], where the uniformity term can be simplified as follows:

$$\mathbb{E}_{x \sim p(x)} \left[ \log \mathbb{E}_{x^- \sim p(x)} \left[ e^{-\frac{1}{2} \|A\psi(x^-) - \psi(x)\|_2^2} \right] \right]$$

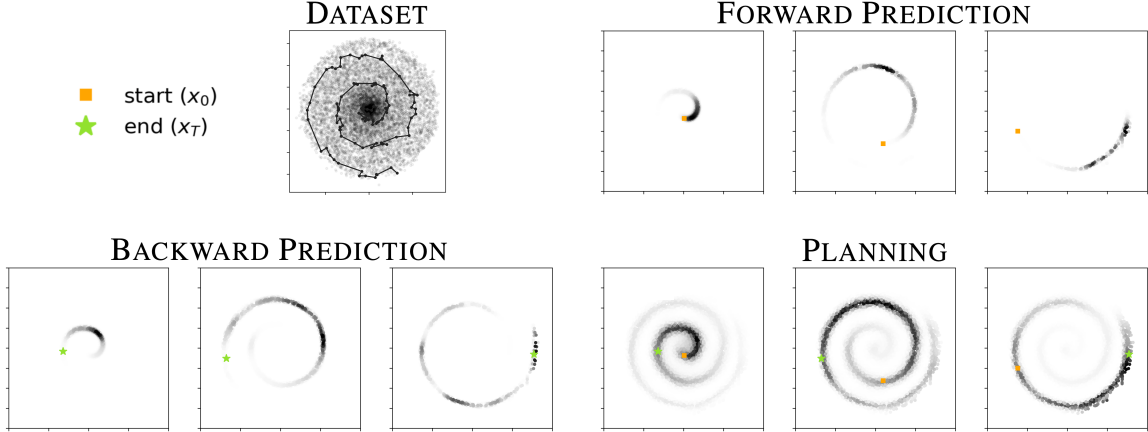


Figure 6.4: **Numerical simulation of our analysis.** (Top Left) Toy dataset of time-series data consisting of many outwardly-spiraling trajectories. We apply temporal contrastive learning to these data. (Top Right) For three initial observations (■), we use the learned representations to predict the distribution over future observations. Note that these distributions correctly capture the spiral structure. (Bottom Left) For three observations (★), we use the learned representations to predict the distribution over preceding observations. (Bottom Right) Given an initial and final observation, we plot the inferred posterior distribution over the waypoint (Section 6.2). The representations capture the shape of the distribution.

$$\begin{aligned}
&= \frac{1}{N} \sum_{i=1}^N \log \left( \frac{1}{N-1} \sum_{j=1 \dots N, j \neq i} e^{-\frac{1}{2} \|A\psi(x_i) - \psi(x_j)\|_2^2} \right) \\
&= \frac{1}{N} \sum_{i=1}^N \log \left( \frac{1}{N-1} \sum_{j=1 \dots N, j \neq i} \underbrace{\frac{1}{(2\pi)^{k/2}} e^{-\frac{1}{2} \|A\psi(x_i) - \psi(x_j)\|_2^2}}_{\mathcal{N}(\mu=\psi(x_j); \Sigma=I)} \right) + \frac{k}{2} \log(2\pi) \\
&= \frac{1}{N} \sum_{i=1}^N \log \hat{p}_{\text{GMM}}(\psi(x_i)) + \frac{k}{2} \log(2\pi) \\
&= -\hat{\mathcal{H}}[\psi(x)] + \frac{k}{2} \log(2\pi).
\end{aligned}$$

The derivation above extends that in Wang and Isola [161] by considering a Gaussian distribution rather than a von Mises Fisher distribution. We are implicitly making the assumption that the marginal distributions satisfy  $p(x) = p(x^-)$ . This difference corresponds to our choice of using a negative squared L2 distance in the infoNCE loss rather than an inner product, a difference that will be important later in our analysis. A second difference is that we do not use the resubstitution estimator (i.e., we exclude data point  $x_i$  from our estimate of  $\hat{p}_{\text{GMM}}$  when evaluating the likelihood of  $x_i$ ), which we found hurt performance empirically. The takeaway from this identity is that maximizing the uniformity term corresponds to maximizing (an estimate of) the entropy of the representations.

We next prove that the maximum entropy distribution with an expected L2 norm constraint is a Gaussian distribution. Variants of this result are well known [233–235], but we include a full proof here for transparency.

**Lemma 6.4.** *The maximum entropy distribution satisfying the expected L2 norm constraint in Eq. (6.3) is a multivariate Gaussian distribution with mean  $\mu = 0$  and covariance  $\Sigma = c \cdot I$*

*Proof.* We start by defining the corresponding Lagrangian, with the second constraint saying that  $p(x)$  must be a valid probability distribution.

$$\mathcal{L}(p) = \mathcal{H}_p[x] + \lambda_1 \left( \mathbb{E}_{p(x)} \left[ \|x\|_2^2 \right] - c \cdot k \right) + \lambda_2 \left( \int p(x) dx - 1 \right)$$

We next take the derivative w.r.t.  $p(x)$ :

$$\frac{\partial \mathcal{L}}{\partial p(x)} = -p(x)/p(x) - \log p(x) + \lambda_1 \|x\|_2^2 + \lambda_2$$

Setting this derivative equal to 0 and solving for  $p(x)$ , we get

$$p(x) = e^{-1+\lambda_2+\lambda_1\|x\|_2^2}.$$

We next solve for  $\lambda_1$  and  $\lambda_2$  to satisfy the constraints in the Lagrangian. Note that  $x \sim \mathcal{N}(\mu = 0, \Sigma = c \cdot I)$  has an expected norm  $\mathbb{E}[\|x\|_2^2] = c \cdot k$ , so we must have  $\lambda_1 = -\frac{1}{2c}$ . We determine  $\lambda_1$  as the normalizing constant for a Gaussian, finally giving us:

$$p(x) = \frac{1}{(2c\pi)^{k/2}} e^{-\frac{1}{2c}\|x\|_2^2}$$

corresponding to an isotropic Gaussian distribution with mean  $\mu = 0$  and covariance  $\Sigma = c \cdot I$ .  $\square$

Proof of Lemma 6.1

Below we present the proof of Lemma 6.1

*Proof.* Our proof technique will be similar to that of the law of the unconscious statistician:

$$\begin{aligned} p(\psi_{t+} | \psi_0) &\stackrel{(a)}{=} \frac{p(\psi_{t+}, \psi_0)}{p(\psi_0)} \propto \iint p(\psi_{t+}, x_{t+}, \psi_0, x_0) dx_{t+} dx_0 \\ &\stackrel{(b)}{=} \iint p(\psi_{t+} | x_{t+}) p(\psi_0 | x_0) p(x_{t+} | x_0) p(x_0) dx_{t+} dx_0 \\ &\stackrel{(c)}{\propto} \iint \mathbf{1}(\psi(x_{t+}) = \psi_{t+}) \mathbf{1}(\psi(x_0) = \psi_0) p(x_{t+}) e^{-\frac{1}{2}\|A\psi(x_0) - \psi(x_{t+})\|_2^2} p(x_0) dx_{t+} dx_0 \end{aligned}$$

$$\begin{aligned}
&\stackrel{(d)}{=} e^{-\frac{1}{2}\|A\psi_0 - \psi_{t+}\|_2^2} \iint \mathbf{1}(\psi(x_{t+}) = \psi_{t+}) \mathbf{1}(\psi(x_0) = \psi_0) p(x_{t+}) p(x_0) dx_{t+} dx_0 \\
&\stackrel{(e)}{=} e^{-\frac{1}{2}\|A\psi_0 - \psi_{t+}\|_2^2} \underbrace{\left( \int p(x_{t+}) \mathbf{1}(\psi(x_{t+})) dx_{t+} \right)}_{p(\psi_{t+})} \underbrace{\left( \int p(x_0) \mathbf{1}(\psi(x_0)) dx_0 \right)}_{p(\psi_0)} \\
&\stackrel{(f)}{\propto} e^{-\frac{1}{2}\|A\psi_0 - \psi_{t+}\|_2^2} e^{-\frac{1}{2c}\|\psi_{t+}\|_2^2} e^{-\frac{1}{2c}\|\psi_0\|_2^2} \\
&\stackrel{(g)}{\propto} e^{-\frac{1+\frac{1}{c}}{2} \left\| \frac{1}{1+\frac{1}{c}} A\psi_0 - \psi_{t+} \right\|_2^2} \\
&\propto \mathcal{N}(\psi_{t+}; \mu = \frac{c}{c+1} A\psi_0, \Sigma = \frac{c}{c+1} I).
\end{aligned}$$

In (a) we applied Bayes' Rule and removed the denominator, which is a constant w.r.t.  $\psi_{t+}$ . In (b) we factored the joint distribution, noting that  $\psi_{t+}$  and  $\psi_0$  are deterministic functions of  $x_{t+}$  and  $x_0$  respectively, so they are conditionally independent from the other random variables. In (c) we used Assumption 6.2 after solving for  $p(x_{t+} | x_0) = p(x_{t+}) e^{-\frac{1}{2}\|A\psi(x_0) - \psi(x)\|_2^2}$ . In (d) we noted that when the integrand is nonzero, it takes on a constant value of  $e^{-\frac{1}{2}\|A\psi_0 - \psi_{t+}\|_2^2}$ , so we can move that constant outside the integral. In (e) we used the definition of the marginal representation distribution (Eq. 6.6). In (f) we used Assumption 6.1 to write the marginal distributions  $p(\psi_{t+})$  and  $p(\psi_0)$  as Gaussian distributions. We removed the normalizing constants, which are independent of  $\psi_{t+}$ . In (g) we completed the square and then recognized the expression as the density of a multivariate Gaussian distribution.  $\square$

### Proof of Theorem 6.2: Waypoint Distribution

*Proof.*

$$\begin{aligned}
p(\psi_w | \psi_0, \psi_{t+}) &\stackrel{(a)}{=} \frac{p(\psi_{t+} | \psi_w) p(\psi_w | \psi_0)}{p(\psi_{t+} | \psi_0)} \\
&\stackrel{(b)}{\propto} e^{-\frac{1+\frac{1}{c}}{2} \left\| \frac{c}{c+1} A\psi_w - \psi_{t+} \right\|_2^2} e^{-\frac{1+\frac{1}{c}}{2} \left\| \frac{c}{c+1} A\psi_0 - \psi_w \right\|_2^2} \\
&\stackrel{(c)}{\propto} e^{-\frac{1}{2}(\psi_w - \mu)^T \Sigma^{-1} (\psi_w - \mu)} = \mathcal{N}(\psi_w; \mu, \Sigma)
\end{aligned}$$

where  $\Sigma^{-1} = \frac{c}{c+1} A^T A + \frac{c+1}{c} I$  and  $\mu = \Sigma(A^T \psi_{t+} + A\psi_0)$ .  $\square$

In line (a) we used the definition of the conditional distribution and then simplified the numerator using the Markov property. Line (b) uses the Lemma 6.1. Line (c) completes the square, the details of which are below:

$$\frac{1}{2} \cdot \frac{c+1}{c} \left( \left\| \frac{c}{c+1} A\psi_w - \psi_{t+} \right\|_2^2 + \left\| \frac{c}{c+1} A\psi_0 - \psi_w \right\|_2^2 \right)$$



$$\begin{aligned}
&= \frac{1}{2} \cdot \frac{c+1}{c} \left( \psi_w^T \left( \frac{c}{c+1} A \right)^T \left( \frac{c}{c+1} A \right) \psi_w - 2\psi_{t+}^T \left( \frac{c}{c+1} A \right) \psi_w + \cancel{\psi_{t+}^T \psi_{t+}} \right. \\
&\quad \left. + \cancel{\psi_0^T \left( \frac{c}{c+1} A \right)^T \left( \frac{c}{c+1} A \right) \psi_0} - 2\psi_0^T \left( \frac{c}{c+1} A \right)^T \psi_w + \psi_w^T \psi_w \right) \\
&\stackrel{\text{const.}}{=} \frac{1}{2} \frac{c+1}{c} \left( \psi_w^T \left( \left( \frac{c}{c+1} \right)^2 A^T A + I \right) \psi_w - 2 \frac{c}{c+1} (A^T \psi_{t+} + A \psi_0)^T \psi_w \right) \\
&= \frac{1}{2} \psi_w^T \underbrace{\left( \frac{c}{c+1} A^T A + \frac{c+1}{c} I \right)}_{\Sigma^{-1}} \psi_w - (A^T \psi_{t+} + A \psi_0)^T \psi_w \\
&\stackrel{\text{const.}}{=} (\psi_w - \mu)^T \Sigma^{-1} (\psi_w - \mu),
\end{aligned}$$

where  $\Sigma^{-1} = \frac{c}{c+1} A^T A + \frac{c+1}{c} I$  and  $\mu = \Sigma(A^T \psi_{t+} + A \psi_0)$ . Above, we have used  $\stackrel{\text{const.}}{=}$  to denote equality up to an additive constant that is independent of  $\psi_w$ .

### Proof of Theorem 6.3: Planning over Many Intermediate States

*Proof.* We start by recalling that the waypoints form a Markov chain, so we can express their joint density as a product of conditional densities:

$$p(\psi_{1:n}) = p(\psi_0) p(\psi_1 | \psi_0) p(\psi_2 | \psi_1) \cdots p(\psi_n | \psi_{n-1}).$$

The aim of this lemma is to express the joint distribution over multiple waypoints, given an initial and final state representation:

$$\begin{aligned}
p(\psi_{1:n} | \psi_0, \psi_{t+}) &\stackrel{(a)}{=} \frac{p(\psi_{1:n} | \psi_0) p(\psi_{t+} | \psi_n)}{\cancel{p(\psi_{t+} | \psi_0)}} \\
&\stackrel{(b)}{\propto} p(\psi_1 | \psi_0) p(\psi_2 | \psi_1) \cdots p(\psi_{t+} | \psi_n) \\
&\stackrel{(c)}{\propto} \exp \left( -\frac{1}{2} \frac{c+1}{c} \left\| \frac{c}{c+1} A \psi_0 - \psi_1 \right\|_2^2 - \frac{1}{2} \frac{c+1}{c} \left\| \frac{c}{c+1} A \psi_1 \right. \right. \\
&\quad \left. \left. - \psi_2 \right\|_2^2 - \cdots - \frac{1}{2} \frac{c+1}{c} \left\| \frac{c}{c+1} A \psi_n - \psi_{t+} \right\|_2^2 \right) \\
&\stackrel{(d)}{=} \exp \left( -\frac{1}{2} \frac{c}{c+1} \cancel{\psi_0^T A^T A \psi_0} + \psi_0^T A^T \psi_1 - \frac{1}{2} \frac{c+1}{c} \psi_1^T \psi_1 \right. \\
&\quad - \frac{1}{2} \frac{c}{c+1} \psi_1^T A^T A \psi_1 + \psi_1^T A^T \psi_2 - \frac{1}{2} \frac{c+1}{c} \psi_2^T \psi_2 \\
&\quad - \frac{1}{2} \frac{c}{c+1} \psi_2^T A^T A \psi_2 + \psi_2^T A^T \psi_3 - \frac{1}{2} \frac{c+1}{c} \psi_3^T \psi_3 \\
&\quad \vdots \\
&\quad \left. - \frac{1}{2} \frac{c}{c+1} \psi_n^T A^T A \psi_n + \psi_n^T A^T \psi_{t+} - \frac{1}{2} \frac{c+1}{c} \cancel{\psi_{t+}^T \psi_{t+}} \right)
\end{aligned}$$



$$= \exp\left(-\frac{1}{2}\psi_{1:n}^T \Sigma^{-1} \psi_{1:n} + \eta^T \psi_{1:n}\right),$$

where  $\Sigma^{-1}$  is a tridiagonal matrix

$$\Sigma^{-1} = \begin{pmatrix} \frac{c}{c+1}A^T A + \frac{c+1}{c}I & -A^T & & & \\ -A & \frac{c}{c+1}A^T A + \frac{c+1}{c}I & -A^T & & \\ & -A & \frac{c}{c+1}A^T A + \frac{c+1}{c}I & & \\ & & & \ddots & \\ & & & & \ddots \end{pmatrix} \quad \text{and } \eta = \begin{pmatrix} A\psi_0 \\ 0 \\ \vdots \\ 0 \\ A^T \psi_{t+} \end{pmatrix}.$$

In (a) we applied Bayes' rule and removed the denominator because it is a constant with respect to  $\psi_{1:n}$ . In (b) we applied the Markov assumption. In (c) we used Lemma 6.1 to express the conditional probabilities as Gaussians, ignoring the proportionality constants (which are independent of  $\psi$ ). In (d) we simplified the exponents, removing terms that do not depend on  $\psi_{1:n}$ .  $\square$

Formalizing Assumption 6.2

Assumption 6.2 states

$$e^{-\frac{1}{2}\|\phi(x_0) - \psi(x)\|_2^2} = \frac{p(x | x_0)}{p(x^+)C} \quad (6.5)$$

when Eq. (6.2) is optimized.

We can justify this assumption by analyzing the general solution to the symmetrized version of the Oord et al. [152] infoNCE objective, which we do in Lemma 6.5. Applying this lemma to our representation learning objective (6.2) for sufficiently large batch size  $B$  then yields Eq. (6.5), with the function approximator  $\|\phi(x) - \psi(x^+)\|^2 \approx f(x, x^+)$ .

**Lemma 6.5.** *The solution to the optimization problem*

$$\max_{f(x, x^+)} \lim_{B \rightarrow \infty} \mathbb{E}_{\{(x_i, x_i^+)\}_{i=1}^B \sim p(x, x^+)} \left[ \frac{1}{B} \sum_{i=1}^B \log \frac{e^{f(x_i, x_i^+)}}{\sum_{j \neq i} e^{f(x_i, x_j^+)}} + \log \frac{e^{f(x_i, x_i^+)}}{\sum_{j \neq i} e^{f(x_j, x_i^+)}} \right] \quad (6.8)$$

satisfies

$$f(x, x^+) = \log \left( \frac{p(x^+ | x)}{p(x^+)C} \right) \quad (6.9)$$

for some  $C$ .

*Proof of Lemma 6.5.* We first break down the LHS and RHS of Eq. (6.2):

$$\max_f \lim_{B \rightarrow \infty} \mathbb{E}_{\{(x_i, x_i^+)\}_{i=1}^B \sim p(x, x^+)} \left[ \underbrace{\frac{1}{B} \sum_{i=1}^B \log \frac{e^{f(x_i, x_i^+)}}{\sum_{j \neq i} e^{f(x_i, x_j^+)}}}_{\mathcal{J}_1} + \underbrace{\log \frac{e^{f(x_i, x_i^+)}}{\sum_{j \neq i} e^{f(x_j, x_i^+)}}}_{\mathcal{J}_2} \right]$$

$$\mathcal{J}_1(f) = \lim_{B \rightarrow \infty} \mathbb{E}_{\{(x_i, x_i^+)\}_{i=1}^B \sim p(x, x^+)} \left[ \frac{1}{B} \sum_{i=1}^B \log \frac{e^{f(x_i, x_i^+)}}{\sum_{j \neq i} e^{f(x_i, x_j^+)}} \right]$$

$$\mathcal{J}_2(f) = \lim_{B \rightarrow \infty} \mathbb{E}_{\{(x_i, x_i^+)\}_{i=1}^B \sim p(x, x^+)} \left[ \frac{1}{B} \sum_{i=1}^B \log \frac{e^{f(x_i, x_i^+)}}{\sum_{j \neq i} e^{f(x_j, x_i^+)}} \right]$$

We now use the following result from Ma and Collins [52]:

**Lemma 6.6.** *The optimal solutions  $f_1$  and  $f_2$  for  $\mathcal{J}_1$  and  $\mathcal{J}_2$  satisfy*

$$f_1(x, x^+) = \log p(x | x^+) - \log c_1(x) \quad (6.10)$$

$$f_2(x, x^+) = \log p(x^+ | x) - \log c_2(x^+) \quad (6.11)$$

for arbitrary  $c_1(x), c_2(x^+)$ .

For any  $C$ , when  $c_1(x) = Cp(x)$  and  $c_2(x^+) = Cp(x^+)$ ,

$$f_1(x, x^+) = \log \left( \frac{p(x | x^+)}{p(x)C} \right) = \log \left( \frac{p(x^+ | x)}{p(x^+)C} \right) = f_2(x, x^+). \quad (6.12)$$

It follows that Eq. (6.12) maximizes both  $\mathcal{J}_1$  and  $\mathcal{J}_2$ , and is precisely the optimal solution Eq. (6.9) for Eq. (6.8).  $\square$

**What does  $C$  represent?** From Eq. (6.9), we can connect  $C$  to the mutual information  $I(x, x^+)$ :

$$C = \frac{\mathbb{E}_{(x, x^+) \sim p(x, x^+)} [f(x, x^+)]}{I(x, x^+)}. \quad (6.13)$$

*Proof of Lemma 6.6.* We can first consider  $\mathcal{J}_1$  without loss of generality. Denoting  $g(x, x^+) = e^{f(x, x^+)}$ , we take the functional derivative:

$$\begin{aligned} \delta \mathcal{J}_1(\log g) &= \lim_{B \rightarrow \infty} \delta \mathbb{E}_{\{(x_i, x_i^+)\}_{i=1}^B \sim p(x, x^+)} \left[ \frac{1}{B} \sum_{i=1}^B \log \frac{g(x_i, x_i^+)}{\sum_{j \neq i} g(x_i, x_j^+)} \right] \\ &= \lim_{B \rightarrow \infty} \mathbb{E}_{\{(x_i, x_i^+)\}_{i=1}^B \sim p(x, x^+)} \left[ \frac{1}{B} \sum_{i=1}^B \frac{(\sum_{j \neq i} g(x_i, x_j^+)) \delta g(x_i, x_i^+) - g(x_i, x_i^+) \delta (\sum_{j \neq i} g(x_i, x_j^+))}{g(x_i, x_i^+) (\sum_{j \neq i} g(x_i, x_j^+))} \right] \\ &= \lim_{B \rightarrow \infty} \mathbb{E}_{\{(x_i, x_i^+)\}_{i=1}^B \sim p(x, x^+)} \left[ \frac{1}{B} \sum_{i=1}^B \frac{\delta g(x_i, x_i^+)}{g(x_i, x_i^+)} - \frac{\delta (\sum_{j \neq i} g(x_i, x_j^+))}{\sum_{j \neq i} g(x_i, x_j^+)} \right] \\ &= \lim_{B \rightarrow \infty} \mathbb{E}_{\{(x_i, x_i^+)\}_{i=1}^B \sim p(x, x^+)} \left[ \frac{1}{B} \sum_{i=1}^B \int \left( \left( \frac{\delta g(x_i, x_i^+)}{g(x_i, x_i^+)} \right) p(x^+ | x_i) \right. \right. \\ &\quad \left. \left. - \sum_{k \neq i} \left( \frac{\delta g(x_i, x_i^+)}{g(x_i, x_i^+) - g(x_i, x_k^+) + \sum_{j \neq i} g(x_i, x_j^+)} \right) p(x^+) \right) dx^+ \right] \end{aligned}$$

$$\begin{aligned}
&= \lim_{B \rightarrow \infty} \mathbb{E}_{\{(x_i, x_i^+)\}_{i=1}^B \sim p(x, x^+)} \left[ \frac{1}{B} \sum_{i=1}^B \int \delta g(x_i, x^+) \left( \frac{p(x^+ | x_i)}{g(x_i, x^+)} \right. \right. \\
&\quad \left. \left. - \underbrace{\mathbb{E}_{\{(x_i, x_i^+)\}_{i=1}^B} \left[ \frac{1}{\sum_{j \neq i} g(x_i, x_j^+)} \right]}_{\text{as } B \rightarrow \infty} p(x^+) \right) dx^+ \right] \\
&= \lim_{B \rightarrow \infty} \mathbb{E}_{\{(x_i, x_i^+)\}_{i=1}^B \sim p(x, x^+)} \left[ \frac{1}{B} \sum_{i=1}^B \int \delta g(x_i, x^+) \left( \frac{p(x^+ | x_i)}{g(x_i, x^+)} \right. \right. \\
&\quad \left. \left. - \underbrace{\mathbb{E}_{\{(x_i, x_i^+)\}_{i=1}^B} \left[ \frac{1}{\sum_{j \neq i} g(x_i, x_j^+)} \right]}_{\triangleq k(x_i) \text{ indep. of } x^+} p(x^+) \right) dx^+ \right] \\
&= \lim_{B \rightarrow \infty} \mathbb{E}_{\{(x_i, x_i^+)\}_{i=1}^B \sim p(x, x^+)} \left[ \frac{1}{B} \sum_{i=1}^B \int \delta g(x_i, x^+) \left( \frac{p(x^+ | x_i)}{g(x_i, x^+)} - k(x_i) p(x^+) \right) dx^+ \right] \\
&= \int \delta g(x, x^+) \left( \frac{p(x^+ | x)}{g(x, x^+)} - k(x) p(x^+) \right) dx^+.
\end{aligned}$$

This is zero when

$$g(x, x^+) = \frac{p(x | x^+)}{k(x)p(x)},$$

i.e.,

$$f(x, x^+) = \log p(x | x^+) - \log \underbrace{c_1(x)}_{k(x)p(x)}.$$

as in Eq. (6.10), and Eq. (6.11) follows similarly, exchanging  $x$  and  $x^+$ .  $\square$

## 6.4 NUMERICAL SIMULATION

We include several didactic experiments to illustrate our results. All results and figures can be reproduced by running make in the source code: [https://github.com/vivekmyers/contrastive\\_planning](https://github.com/vivekmyers/contrastive_planning). The expected compute time is a few hours on a A6000 GPU. Figures in this section show error across different training and dataset split seeds.

### Synthetic Dataset

To validate our analysis, we design a time series task with 2D points where inference over intermediate points (i.e., in-filling) requires nonlinear interpolation. Fig. 6.4 (*Top Left*) shows the dataset of time series data, starting at the origin and spiraling outwards, with each trajectory using a randomly-chosen initial angle. We applied contrastive learning with the parametrization in Section 6.2 to these data and used the learned representations to solve prediction and planning problems (see Fig. 6.4 for details). Note that these predictions correctly handle the nonlinear structure of

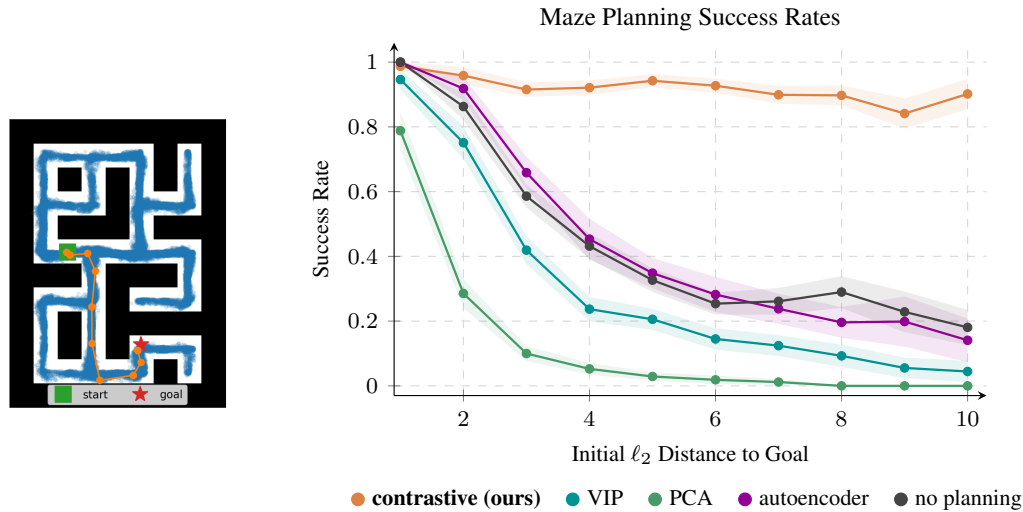


Figure 6.5: Using inferred paths over our contrastive representations for control boosts success rates by  $4.5\times$  on the most difficult goals (18%  $\rightarrow$  84%). Alternative representation learning techniques fail to improve performance when used for planning.

these data – states nearby the initial state in Euclidean space that are not temporally adjacent are assigned low likelihood.

### Solving Mazes with Inferred Representations

Our next experiment studies whether the inferred representations are useful for solving a control task. We took a 2d maze environment and dataset from prior work (Fig. 6.5, *Left*) [2] and learned encoders from this dataset. To solve the maze, we take the observation of the starting state and goal state, compute the representations of these states, and use the analysis in Section 6.2 to infer the sequence of intermediate representations. We visualize the results using a nearest neighbor retrieval (Fig. 6.5, *Left*). Appendix Fig. 6.7 contains additional examples.

Finally, we studied whether these representations are useful for control. We implemented a simple proportional controller for this maze. As expected, this proportional controller can successfully navigate to close goals, but fails to reach distant goals (Fig. 6.5, *Right*). However, if we use the proportional controller to track a series of waypoints planned using our representations (i.e., the orange dots shown in Fig. 6.5 (*Left*)), the success rate increases by up to  $4.5\times$ . To test the importance of *nonlinear* representations, we compare with a “PCA” baseline that predicts waypoints by interpolating between the principal components of the initial state and goal state. The better performance of our method indicates the importance of doing the interpolation using representations that are *nonlinear* functions of the input observations. While prior methods learn representations to encode temporal distances, it is unclear whether these methods support inference via interpolation.

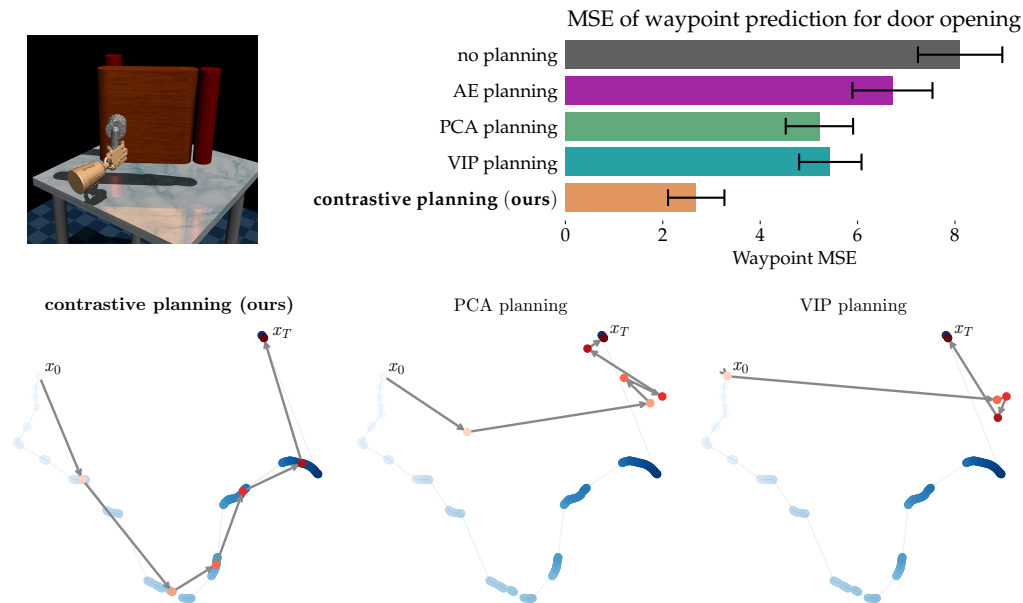


Figure 6.6: Planning for 39-dimensional robotic door opening. (*Top Left*) We use a dataset of trajectories demonstrating door opening from prior work [2] to learn representations. (*Top Right*) We use our method and three baselines to infer one intermediate waypoint between the first and last observation in a trajectory from a held-out validation set. Errors are measured using the mean squared error with the true waypoint observation; predicted representations are converted to observations using nearest neighbors on a validation set. (*Bottom*) We visualize a TSNE [3] of the states along the sampled trajectory as blue circles, with the transparency indicating the index along the trajectory. The inferred plan is shown as red circles connected by arrows. Our method generates better plans than alternative representation learning methods (PCA, VIP).

To test this hypothesis, we use one of these methods (“VIP” [119]) as a baseline. While the VIP representations likely encode similar bits as our representations, the better performance of the contrastive representations indicates that the VIP representations do not expose those bits in a way that makes planning easy.

### Higher dimensional tasks

In this section we provide preliminary experiments showing the planning approach in Section 6.2 scales to higher dimensional tasks. We used two datasets from prior work [2]: `door-human-v0` (39-dimensional observations) and `hammer-human-v0` (46-dimensional observations). After learning encoders on these tasks, we evaluated the inference capabilities of the learned representations. Given the first and last observation from a trajectory in a validation set, we use linear interpolation (see Eq. 6.7) to infer the representation of five intermediate waypoint representations.

We evaluate performance in two ways. **Quantitatively**, we measure the mean squared error between each of the true waypoint observations and those inferred by our method. Since our method infers representations, rather than observations, we

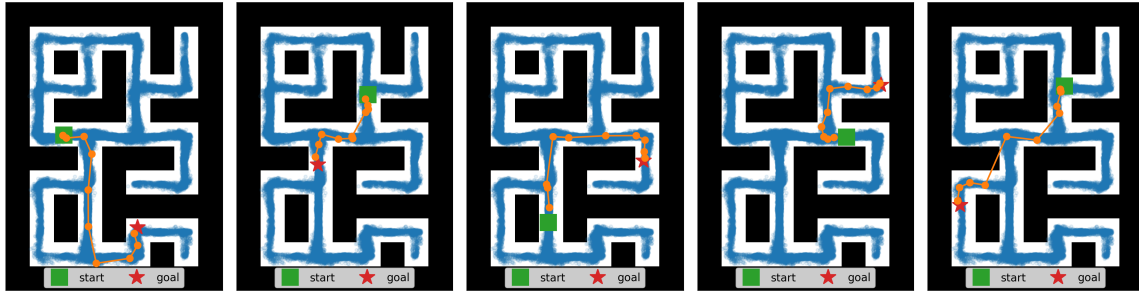


Figure 6.7: Our approach enables a goal-conditioned policy to reach farther targets (red) from the start (green) by planning over intermediate waypoints (orange).

use a nearest-neighbor retrieval on a validation set so that we can measure errors in the space of observations. **Qualitatively**, we visualize the high-dimensional observations from the validation trajectory using a 2-dimensional TSNE [3] embedding, overlying the inferred waypoints from our method; as before, we convert the representations inferred by our method to observations using nearest neighbors.

We compare with three alternative methods in Fig. 6.6. To test the importance of representation learning, we first naively interpolate between the initial and final observations (“no planning”). The poor performance of this baseline indicates that the input time series are highly nonlinear. Similarly, interpolating the principle components of the initial and final observations (“PCA”) performs poorly, again highlighting that the input time series is highly nonlinear and that our representations are doing more than denoising (i.e., discarding directions of small variation). The third baseline, “VIP” [119], learns representations to encode temporal distances using approximate dynamic programming. Like our method, VIP avoids reconstruction and learns nonlinear representations of the observations. However, the results in Fig. 6.6 highlight that VIP’s representations do not allow users to plan by interpolation. The error bars shown in Fig. 6.6 (*Top Right*) show the standard deviation over 500 trajectories sampled from the validation set. For reproducibility, we repeated this entire experiment on another task, the 46-dimensional hammer-human-v0 from D4RL. The results, shown in Appendix Fig. 6.8, support the conclusions above. Taken together, these results show that our procedure for interpolating contrastive representations continues to be effective on tasks where observations have dozens of dimensions.

## 6.5 ADDITIONAL EXPERIMENTS

Fig. 6.7 visualizes the inferred waypoints from the task in Fig. 6.5.

Fig. 6.8 visualizes the representations learned on a 46-dimensional robotic hammering task (see Section 6.4).

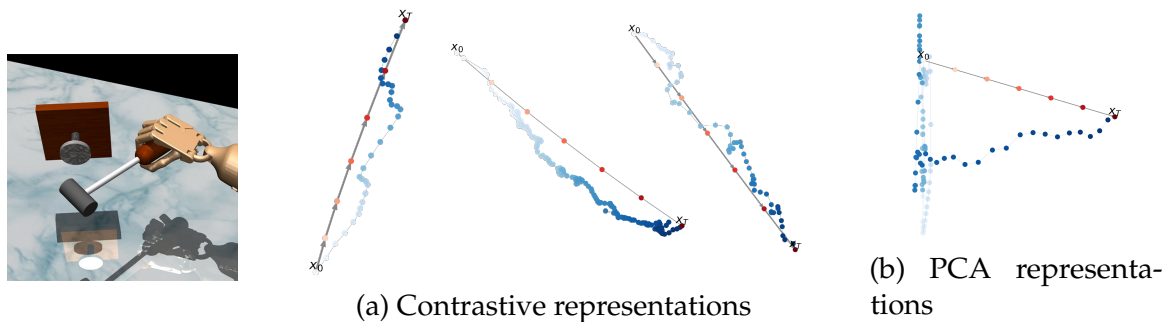


Figure 6.8: Planning for 46-dimensional robotic hammering. (Left) A dataset of trajectories demonstrating a hammer knocking a nail into a board [2]. (Center) We visualize the learned representations as blue circles, with the transparency indicating the index of that observation along the trajectory. We also visualize the inferred plan (Section 6.2) as red circles connected by arrows. (Right) Representations learned by PCA on the same trajectory as (a, left).

## Stock Prediction

We show results on a stock opening price task in Fig. 6.9.

## 6.6 RELATED WORK

**Representations for time-series data.** In applications ranging from robotics to vision to NLP, users often want to learn representations of observations from time series data such that the *spatial* arrangement of representations reflects the *temporal* arrangement of the observations [152, 225, 158, 168]. Ideally, these representations should retain information required to predict future observations and infer likely paths between pairs of observations. Many approaches use an autoencoder, learning representations that retain the bits necessary to reconstruct the input observation, while also regularizing the representations to compressed or predictable [202, 239–241, 218, 242]. A prototypical method is the sequential VAE [217], which is computationally expensive to train because of the reconstruction loss, but is easy to use for inference. Our work shares the aims of prior methods that attempt to linearize the dynamics of nonlinear systems [243–248], including videos [249, 250]. Our work aims to retain uncertainty estimates over predictions (like the sequential VAE) without requiring reconstruction. Avoiding reconstruction is appealing *practically* because it decreases the computational requirements and number of hyperparameters; and *theoretically* because it means that representations only need to retain bits about temporal relationships and not about the bits required to reconstruct the original observation.

**Contrastive Learning.** Contrastive learning methods circumvent reconstruction by learning representations that merely classify if two events were sampled from the



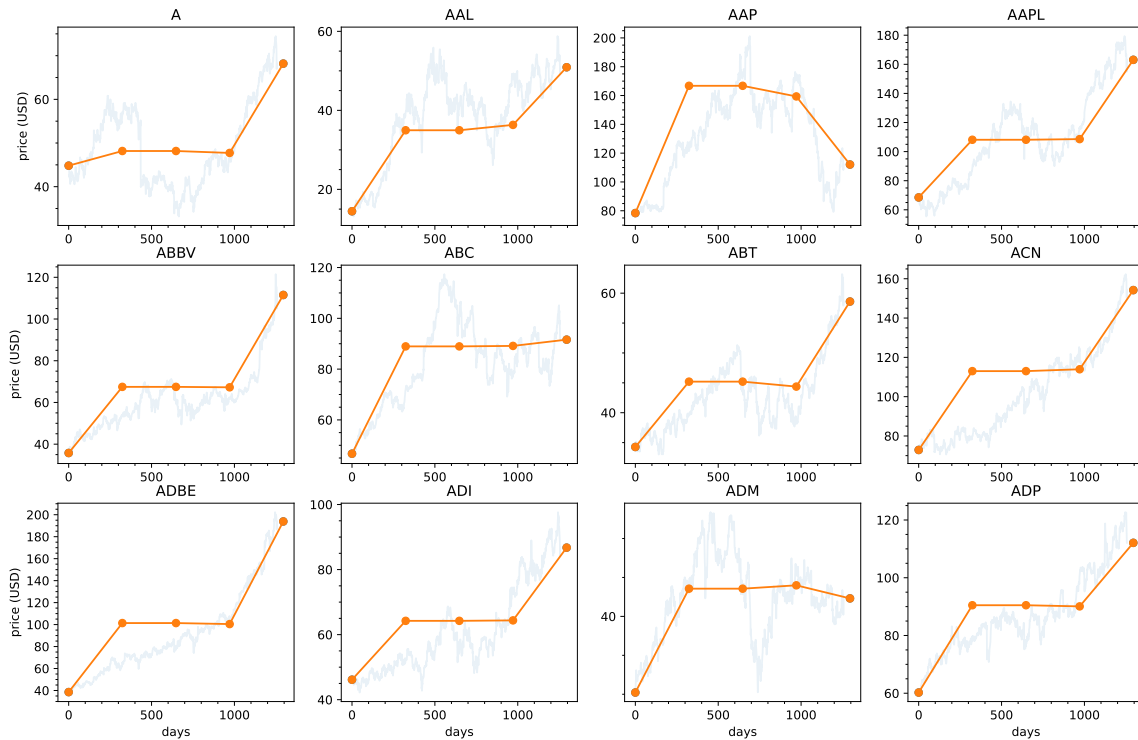


Figure 6.9: **Stock Prediction.** We apply temporal contrastive learning to time series data of the stock market. Data are the opening prices for the 500 stocks in the S&P 500, over a four year window. We remove 30 stocks that are missing data. For evaluation, we choose a 100 day window from a validation set, and use Theorem 6.2 to perform “inpainting”, predicting the intermediate stock prices *jointly* for all stocks (orange), given the first and last stock price. The true stock prices are shown in blue. While we do not claim that this is a state-of-the-art model for stock prediction, this experiment demonstrates another potential application of our theoretical results.

same joint distribution [156, 142, 17]. When applied to representing states along trajectories, contrastive representations learn to classify whether two points lie on the same trajectory or not [152, 157, 121, 158, 251]. Empirically, prior work in computer vision and NLP has observed that contrastive learning acquires representations where interpolation between representations corresponds to changing the images in semantically meaningful ways [252–256, 225].

Our analysis will be structurally similar to prior theoretical analysis on explaining why word embeddings can solve analogies [257–259]. Our work will make a Gaussianity assumption similar to Arora et al. [258] and our Markov assumption is similar to the random walks analyzed in Arora et al. [258], Hashimoto et al. [260]. We build upon and extends these results to answer questions such as: “what is the distribution over future observations representations?” and “what is the



distribution over state (representations) that would occur on the path between one observation and another?” While prior work is primarily aimed at explaining the good performance of contrastive word embeddings (see, e.g., [258]), we are primarily interested in showing how similar contrastive methods are an effective tool for inference over high-dimensional time series data. Our analysis will show how representations learned via temporal contrastive learning (i.e., without reconstruction) are sufficient statistics for inferring future outcomes and can be used for performing inference on a graphical model (a problem typically associated with generative methods).

**Goal-oriented decision making.** Much work on time series representations is done in service of learning goal-reaching behavior, an old problem [261, 164] that has received renewed attention in recent years [141, 171, 262, 167, 263–265, 151]. Some of the excitement in goal-conditioned RL is a reflection of the recent success of self-supervised methods in computer vision [266] and NLP [94]. Our analysis will study a variant of contrastive representation learning proposed in prior work for goal-conditioned RL [121, 157]. These methods are widespread, appearing as learning objectives for learning value functions [168, 121, 267, 268, 119, 46, 16, 127, 126], as auxiliary objectives [125, 269, 16, 270, 271, 205, 272], in objectives for model-based RL [243, 273–275], and in exploration methods [276, 277]. Our analysis will highlight connections between these prior methods, the classic successor representation [124, 207], and probabilistic inference.

**Planning.** Planning lies at the core of many RL and control methods, allowing methods to infer the sequence of states and actions that would occur if the agent navigated from one state to a goal state. While common methods such as PRM [278] and RRT [279] focus on building random graphs, there is a strong community focusing on planning methods based on probabilistic inference [227, 280, 281]. The key challenge is scaling to high-dimensional settings. While semi-parametric methods make progress on this problem this limitation through semi-parametric planning [170, 282, 200], it remains unclear how to scale any of these methods to high-dimensional settings when states do not lie on a low-dimensional manifold. Our analysis will show how contrastive representations may lift this limitation, with experiments validating this theory on 39-dimensional and 46-dimensional tasks.

## 6.7 DISCUSSION

Representation learning is at the core of many high-dimensional time-series modeling questions, yet how those representations are learned is often disconnected with the inferential task. The precise objective and parametrization we studied is not much different from that used in practice, suggesting that either our theoretical results might be adapted to the existing methods, or that practitioners might adopt

these details so they can use the closed-form solutions to inference questions. Our work may also have implications for studying the structure of learned representations. While prior work often studies the geometry of representations as a post-hoc check, our analysis provides tools for studying *when* interpolation properties are guaranteed to emerge, as well as *how* to learn representations with certain desired geometric properties.

## 7

## INVARIANCE TO PLANNING

Reinforcement learning (RL) remains alluring for its capacity to use data to determine optimal solutions to long-horizon reasoning problems. However, it is precisely this horizon that makes solving the RL problem difficult—the number of possible solutions to a control problem often grows exponentially in the horizon [283]. Indeed, the requirement of collecting long horizon data precludes several potential applications of RL (e.g., health care, robotic manipulation). As a result, RL systems tend to only solve short horizon tasks, or long horizon tasks characterized by repetitive motion.

The classical solution to the “curse of horizon” is dynamic programming [284, 285] (i.e., temporal difference learning [286]): stitching together data to find new solutions. However, TD methods can be complex to implement and challenging to stabilize in high-dimensional settings. There is also a more subtle challenge with these methods: adopting TD methods typically means forging mental models associated with “standard” ML problems, such as generalization and invariance. We will discuss how these tools provide new ways of thinking about long-horizon problems.

While there is prior work studying generalization in RL, it almost exclusively focuses on either (i) *perceptual* changes (e.g., changes in lighting conditions) or (ii) simple randomizations of simulator parameters. We will discuss a different sort of generalization: generalization with respect to horizon. We will study this notion of *horizon generalization* within the setting of goal-conditioned RL: after training the RL agent on nearby goals, can the agent succeed at reaching more distant goals (see Fig. 7.1). While these goals may have been seen in different contexts before (e.g., reaching this goal from a different state), they have never been used in

$\pi$  optimal for all  
( $s, s'$ ) given:

$$d(s, s') < c$$

$$\downarrow$$

$$d(s, s'') < 2c$$

$$\downarrow$$

$$d(s, s''') < 4c$$

...

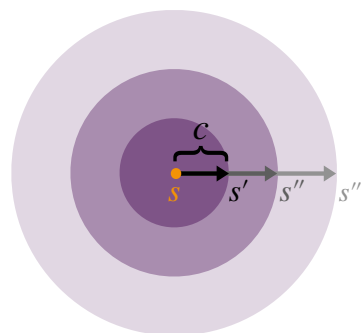


Figure 7.1: **Horizon generalization.** A policy generalizes over the horizon if optimality over all start-goal pairs ( $s, s'$ ) a small temporal distance  $d(s, s') < c$  apart (say, in the training set) leads to optimality over all possible start-goal pairs.

learning long-horizon tasks. Horizon generalization is a type of extrapolation [287]; however, while extrapolation is sometimes seen as alchemy, in some settings horizon generalization is guaranteed (proof: Dijkstra’s algorithm does this).

Our key mathematical tool for understanding horizon generalization is a notion of *planning invariance* (Fig. 7.2): that a RL agent selects similar actions when headed towards a goal, as when headed towards a subgoal (i.e., a waypoint) along the route to that goal. In the same way that (say) an image classification model that is invariant to image brightness will generalize to images of varying brightness, we will show how RL agents that are invariant to planning will generalize to goal-reaching tasks of varying horizons. When a policy is invariant to planning, tasks of length  $n$  and length  $2n$  will be mapped to similar internal representations, as will tasks of length  $4n$ , and  $8n$ , and so on (see Fig. 7.3). This reasoning also explains how a policy exhibiting horizon generalization must solve problems: by recursion, mapping a task of length  $n$  to an (isomorphic) task of length  $n/2$  to a task of length  $n/4$  and so on until the task is simple and similar to one seen during training.

## 7.1 PRELIMINARIES

We consider a controlled Markov process  $\mathcal{M}$  with state space  $\mathcal{S}$ , action space  $\mathcal{A}$ , and dynamics  $p(s' | s, a)$ . The agent interacts with the environment by selecting actions according to a policy  $\pi(a | s)$ , i.e., a mapping from  $\mathcal{S}$  to distributions over  $\mathcal{A}$ . We further assume the state and action spaces are compact.

We equip  $\mathcal{M}$  with an additional notion of *distances* between states. At the most basic level, a distance  $\mathcal{S} \times \mathcal{S} \rightarrow \mathbb{R}$  must be positive everywhere except for a zero diagonal (positive definiteness). We will denote the set of all distances as  $\mathcal{D}$ :

$$\mathcal{D} \triangleq \{d : \mathcal{S} \times \mathcal{S} \rightarrow \mathbb{R} : d(s, s) = 0, d(s, s') > 0 \text{ for each } s, s' \in \mathcal{S} \text{ where } s \neq s'\}. \quad (7.1)$$

A desirable property for distances to satisfy is the triangle inequality. A distance satisfying this property is known as a *quasimetric*, and we define the set of all quasimetric functions as

$$\mathcal{Q} \triangleq \{d \in \mathcal{D} : d(s, g) \leq d(s, w) + d(w, g) \text{ for all } s, g, w \in \mathcal{S}\}. \quad (7.2)$$

If we further restrict distances to be symmetric ( $d(x, y) = d(y, x)$ ), we obtain the set of traditional metrics over  $\mathcal{S}$ . However, we wish to preserve this asymmetry over interchange of start and end states with a quasimetric: navigating  $s \rightarrow g$  may be a completely different task from navigating  $g \rightarrow s$ .

A particular quasimetric of note here is the *successor state distance* [9],  $d_{\text{SD}}^\gamma$ , defined as

$$d_{\text{SD}}^\gamma(s, g) \triangleq \min_{\pi} \left[ \log \frac{p_{\gamma}^{\pi}(\mathfrak{s}_K = g | \mathfrak{s}_0 = g)}{p_{\gamma}^{\pi}(\mathfrak{s}_K = g | \mathfrak{s}_0 = s)} \right], \text{ where } K \sim \text{Geom}(1 - \gamma). \quad (7.3)$$

where the *discounted state occupancy measure*  $p_\gamma^\pi(\mathfrak{s}_K = g \mid \mathfrak{s}_0 = s)$  is defined as

$$p_\gamma^\pi(\mathfrak{s}_K = g \mid \mathfrak{s}_0 = s) \triangleq \sum_{t=0}^{\infty} \gamma^t p^\pi(\mathfrak{s}_t = g \mid \mathfrak{s}_0 = s). \quad (7.4)$$

The distance  $d_{\text{SD}}^\gamma$  is interesting because minimizing the distance to the goal  $d_{\text{SD}}^\gamma(s, g)$  with respect to  $s$  corresponds to optimal goal reaching with a discount factor  $\gamma$ . Formally, if we augment  $\mathcal{M}$  with the goal-conditioned reward function  $r_g(s) = \delta_{(s,g)}$ , a Kronecker delta function which evaluates to 1 if  $s = g$  and 0 otherwise, we obtain an MDP under which the  $d_{\text{SD}}^\gamma$ -minimizing policy is the optimal policy. The related *successor distance with actions*  $d_{\text{SD}}^\gamma(s, a, g)$  [9] allows us to optimize this distance over actions, where the  $d_{\text{SD}}^\gamma(s, a, g)$ -minimizing action is the optimal action over the same MDP:

$$d_{\text{SD}}^\gamma(s, a, g) \triangleq \min_{\pi} \left[ \log \frac{p_\gamma^\pi(\mathfrak{s}_K = g \mid \mathfrak{s}_0 = g)}{p_\gamma^\pi(\mathfrak{s}_K = g \mid \mathfrak{s}_0 = s, a)} \right], \text{ where } K \sim \text{Geom}(1 - \gamma) \quad (7.5)$$

where the *discounted state occupancy measure with actions* is defined as

$$p_\gamma^\pi(\mathfrak{s}_K = g \mid \mathfrak{s}_0 = s, a) \triangleq \sum_{t=0}^{\infty} \gamma^t p^\pi(\mathfrak{s}_t = g \mid \mathfrak{s}_0 = s, a). \quad (7.6)$$

## 7.2 PLANNING INVARIANCE AND HORIZON GENERALIZATION

Our analysis will focus on the goal-conditioned setting. We will start by providing intuition for our key definitions (planning invariance and horizon generalization) and then prove that these properties can exist.

### Intuition for Planning Invariance and Horizon Generalization

Many prior works have found that augmenting goal-conditioned policies with planning can significantly boost performance [117, 165]: instead of aiming for the final goal, these methods use planning to find a waypoint en route to that goal and aim for that waypoint instead. In effect, the policy chooses a closer, easier waypoint that will naturally bring the agent closer to the final goal. We say that a policy is *invariant to planning* if it takes similar actions when directed towards this waypoint as when directed towards the final goal (see Fig. 7.2).

Invariance to planning is an appealing property for several reasons. First, it implies that the policy realizes the benefits of planning without the complex machinery typically associated with hierarchical and model-based methods. Second, it implies that the policy will exhibit *horizon generalization*: given a training dataset of short trajectories covering some state space  $\mathcal{S}$ , it will succeed at performing long-horizon tasks over the same state space  $\mathcal{S}$  (see Fig. 7.1). Say, for example, a

given policy exhibits horizon generalization, and the policy succeeds at reaching a goal that is  $n$  steps (“temporal distance”) away from any initial state in  $\mathcal{S}$ . Then, the horizon generalization property means that this same policy should be able to reach any new goal in  $\mathcal{S}$  for which that original goal is a waypoint, capturing the set of goal states  $2n$  steps away from the initial state. Importantly, we can apply this argument again, reasoning that the policy must also be able to reach goals  $4n$  steps away. This simple recursive argument suggests that a policy with horizon generalization, assuming it can reach very close goals that span a desired state space, must also be able to reach the most distant goals available in this space. Taking a “forward” looking perspective, a policy will generalize from an initial narrow set of seen tasks to vastly more distant goals with trajectories *composed* of these seen tasks.

A similar argument can also be applied in reverse, providing intuition on how a planning invariant policy selects actions. In the broad context of machine learning, a model that is invariant to some transformation (i.e. brightness) assigns similar internal representations to inputs that differ by this transformation (i.e. darkened and brightened version of the same image). The same applies for planning invariant policies: a start-goal pair  $n$  steps apart and a start-waypoint pair  $n/2$  steps apart have the same representation when the waypoint is along the shortest path to that goal (Fig. 7.3). We can repeatedly

apply this argument until mapping the original start-goal pair to a start-waypoint pair that is just one action (in deterministic settings) apart from each other. In short, the “forward” argument predicts *what* tasks a policy with horizon generalization can solve, while the “reverse” argument explains *how* the policy solves tasks that appear to be out of distribution.

With this motivation in hand, how do we actually construct methods that are planning invariant and lead to horizon generalization? To answer this question, we build upon prior work on quasimetric neural network architectures [127, 143, 130] and show that quasimetric policies, where latents obey the triangle inequality, are invariant to planning.

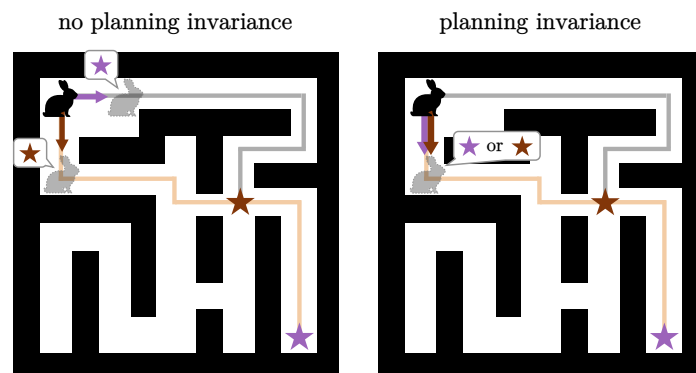


Figure 7.2: **Visualizing planning invariance.** Planning invariance (Definition 7.1) means that a policy should take similar actions when directed towards a goal (purple arrow and purple star) as when directed towards an intermediate waypoint (brown arrow and brown star). We visualize a policy with (Right) and without (Left) this property via the misalignment and alignment of actions towards the waypoint and the goal, where the optimal path is tan and the suboptimal path is gray.

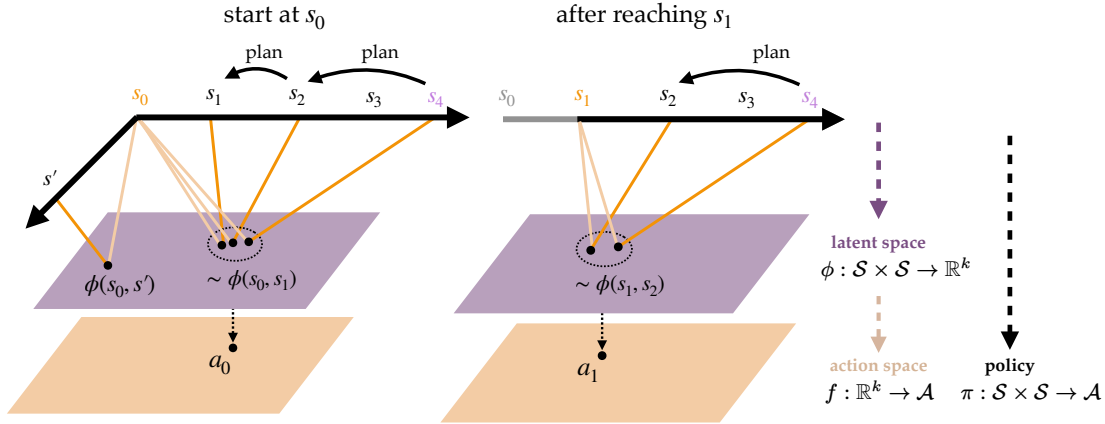


Figure 7.3: **Invariance to planning leads to horizon generalization.** (Left) Invariance to planning maps  $(s_0, \{s_1, s_2, s_4\})$  together in latent space, which results in a shared optimal action. (Right) After successfully reaching the closest waypoint  $s_1$  in 1 step, the next optimal action is also shared, meaning any trajectory of length 2 is optimal. We can repeat this argument for trajectories of length 4, 8,  $\dots$  until the entire reachable state space is covered.

### Definitions of planning invariance and horizon generalization

To construct general definitions of planning invariance and horizon generalization, we will need to define a general notion of a planning operator which proposes waypoints at a given state to reach a target distribution of goals.

We denote by

$$\mathbf{plan} \triangleq \{\text{PLAN} : \mathcal{S} \times \mathcal{A} \times \mathcal{P}(\mathcal{S}) \mapsto \mathcal{P}(\mathcal{S})\} \quad (7.7)$$

the class of “planning functions” that given a state, action, and goal distribution, produce a distribution of possible waypoints. In the special case of a fixed waypoint and goal we write

$$\mathbf{plan}^{\text{FIX}} \triangleq \{\text{PLAN}^{\text{FIX}} : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \mapsto \mathcal{S}\} \subset \mathbf{plan}. \quad (7.8)$$

Our analysis in the rest of this section will focus on the simpler “fixed” setting of  $\text{PLAN}^{\text{FIX}} \in \mathbf{plan}^{\text{FIX}}$ .

We will use  $w$  or  $w_{\text{PLAN}}$  to denote the waypoint produced by  $\text{PLAN}^{\text{FIX}}(s, g)$ . The proofs and quasimetric objects in the stochastic settings are slightly more complicated, but carry the same structure and takeaways as this simpler case; the general stochastic proofs and definitions are presented in Section 7.6.

There are several different types of planning algorithms one might consider (e.g., Dijkstra’s algorithm [285],  $A^*$  [288], RRT [289]). Importantly, the constraints of a quasimetric (see Section 7.1) and the related idea of *path relaxations* from Dijkstra’s



algorithm provide clues for specifying our planning operator later in our analysis. We use this planning operator in one of our key definitions (visualized in Fig. 7.2):

**Definition 7.1** (Planning invariance). *Let an MDP with states  $\mathcal{S}$ , actions  $\mathcal{A}$ , and goal-conditioned Kronecker delta reward function  $r_g(s) = \delta_{(s,g)}$  be given. For any given goal-conditioned policy  $\pi(a | s, G)$  where  $G \in \mathcal{P}(\mathcal{S})$ , we say that  $\pi(a | s, G)$  is invariant under planning operator  $\text{PLAN} \in \mathbf{plan}$  if and only if*

$$\pi(a | s, G) = \pi(a | s, W), \text{ where } W \sim \text{PLAN}(s, a, G). \quad (7.9)$$

*In the single-goal, controlled (“fixed”) case,*

$$\pi(a | s, g) = \pi(a | s, w), \text{ where } w = \text{PLAN}^{\text{FIX}}(s, g). \quad (7.10)$$

Our second key definition is horizon generalization (see Fig. 7.1):

**Definition 7.2** (Horizon generalization). *In the single-goal, controlled (“fixed”) case, a policy  $\pi(a | s, g)$  **generalizes over the horizon** if optimality over  $\mathcal{B}_c = \{(s, g) \in \mathcal{S} \times \mathcal{S} \mid d(s, g) < c\}$  for some finite  $c > 0$  implies optimality over the entire state space  $\mathcal{S}$ , where  $d(s, g)$  is any arbitrary quasimetric over the state and goal distribution space  $\mathcal{S} \times \mathcal{S}$ .*

We highlight the key base case assumption: optimality over shorter trajectories where start-goal pairs *cover the entire desired state space*  $\mathcal{S}$  can generalize over the horizon across the same space  $\mathcal{S}$  (optimal trajectories are contained within  $\mathcal{S}$ )—without additional assumptions about the symmetries of the MDP, it is beyond the scope of this work to consider horizon generalization to completely unseen states. Rather, we analyze generalization to unseen, long-horizon  $(s, g)$  state pairs.

Existence of planning invariance

With these notions of planning invariance and horizon generalization in hand, we will consider planning algorithms  $\text{PLAN}_d^{\text{FIX}} \in \mathbf{plan}^{\text{FIX}}$  that acquire a quasimetric  $d(s, g)$  and output a single waypoint  $w \in \mathcal{S}$ :

$$\text{PLAN}_d^{\text{FIX}}(s, g) = w_{\text{PLAN}} \in \underset{w \in \mathcal{S}}{\text{arg min}} d(s, w) + d(w, g). \quad (7.11)$$

where, by the triangle inequality, we have  $d(s, w_{\text{PLAN}}) + d(w_{\text{PLAN}}, g) = d(s, g)$ .

**Theorem 7.1** (Planning invariance exists). *Assume a controlled, fixed goal setting. For every quasimetric  $d(s, g)$  over state space  $\mathcal{S}$ , there exists a policy  $\pi_d^{\text{FIX}}(a | s, g)$  and planning operator  $\text{PLAN}_d^{\text{FIX}} \in \mathbf{plan}^{\text{FIX}}$  such that  $\pi_d^{\text{FIX}}(a | s, g) = \pi_d^{\text{FIX}}(a | s, w)$  for  $w = \text{PLAN}_d^{\text{FIX}}(s, g)$ .*



The proof is in Section 7.6. In practice, we measure planning invariance by comparing the relative *performance* of algorithms with and without planning. For this condition, we do not necessarily need  $\pi_d(a | s, g) = \pi_d(a | s, w_{\text{PLAN}})$ ; rather, the weaker condition  $d(s, \pi_d(a | s, g), g) = d(s, \pi_d(a | s, w_{\text{PLAN}}), w_{\text{PLAN}})$  is sufficient and necessary for planning invariance when there are no errors from function approximation, noise, etc. We extend this result to stochastic settings in Section 7.6.

### Horizon Generalization Exists

Finally, we prove the existence of horizon generalization using induction, where the inductive step invokes planning invariance. We begin by defining a quasimetric policy.

**Definition 7.3** (Quasimetric policy). *We define the quasimetric policy as some policy  $\pi_d^{\text{FIX}}(a | s, g)$  where*

$$\pi_d^{\text{FIX}}(a | s, g) \in \text{OPT}_d(s, g) \triangleq \arg \min_{a \in \mathcal{A}} d(s, a, g)$$

and  $d(s, a, g)$  is the successor distance with actions (Eq. 7.5). We can extend this definition to stochastic settings (see Definition 7.8) where  $\pi_d(a | s, G)$  is defined over state-goal distribution pairs.

**Theorem 7.2** (Horizon generalization exists). *A quasimetric policy  $\pi_d^{\text{FIX}}(a | s, g)$  that is optimal over  $\mathcal{B}_c = \{(s, g) \in \mathcal{S} \times \mathcal{S} \mid d(s, g) < c\}$  for some finite  $c > 0$  implies optimality over the entire state and goal space  $\mathcal{S} \times \mathcal{S}$ .*

The idea of the proof is to begin with a ball of states  $\mathcal{D}_c = \{s' \in \mathcal{S} \mid d(s, s') < c\}$  for some arbitrary  $s \in \mathcal{S}$ ; we assume policy  $\pi_d^{\text{FIX}}(a | s, \cdot)$  is optimal over this ball by the base case. Then, we use planning invariance and the triangle inequality to show that policy optimality over  $\mathcal{D}_n = \{s' \in \mathcal{S} \mid d(s, s') < 2^n c\}$  implies optimality over  $\mathcal{D}_{n+1}$ , a ball with double the radius. This proof shows that a goal-conditioned, planning invariant policy with optimality over pairs of close states (with respect to the quasimetric) covering state space  $\mathcal{S}$  can be optimal over pairs drawn arbitrarily from the *entire* state space  $\mathcal{S}$ ; the complete proof, extended to stochastic settings and thus applicable to the fixed setting, is in Section 7.6.

Importantly, this property is *not* guaranteed for any arbitrary optimal goal-reaching policy on some restricted horizon:

**Remark 7.3** (Horizon generalization is nontrivial). *For an arbitrary policy, optimality over  $\mathcal{B}_c = \{(s, g) \in \mathcal{S} \times \mathcal{S} \mid d(s, g) < c\}$  for some finite  $c > 0$  is not a sufficient condition for optimality over the entire state space  $\mathcal{S}$ .*

To prove this remark, we construct policies that are optimal over horizon  $H$  but suboptimal over horizon  $H + 1$ . The complete proof is in Section 7.6.

Combined, these results show that (1) planning invariance and horizon generalization, as defined in Section 7.2, exist, (2) planning invariance *and* local policy optimality and coverage are sufficient conditions to achieve horizon generalization, and (3) horizon generalization is not a trivially achievable property.

### Limitations and Assumptions

Despite our theoretical results proving that both horizon generalization and planning invariance do exist, we expect that practical algorithms will not *perfectly* achieve these properties. This section highlights the assumptions that belie our key results, and our experiments in Section 7.4 will empirically study the degree to which current methods achieve these properties.

The main assumption behind our inductive proof is that horizon generalization is unlikely to be a binary category, but rather exists on a spectrum. As such, each application of the inductive argument is likely to incur some error, such that the argument (and, hence, the degree of generalization) will not extend infinitely. To make this a bit more concrete, define  $\text{SUCCESS}(c)$  as the success rate for reaching goals in radius  $c$ , and assume that we choose constant  $c_0$  small enough that  $\text{SUCCESS}(c_0) = 1$ . Then, let us assume that each time the horizon is doubled ( $c_0 \rightarrow 2c_0 \rightarrow 4c_0 \rightarrow \dots$ ), the success rate decreases by a factor of  $\eta$ . We will refer to  $\eta$  as the degree of planning invariance. In addition, we assume that  $\text{SUCCESS}(c)$  is monotonically decreasing; goals further in time should be harder. We can now define the REACH as the sum of  $\text{SUCCESS}(c)$  over  $c \geq c_0$ . With the above constraints on  $\text{SUCCESS}(c)$ , in the worst case,

$$\text{REACH}_{wc} = 1 + \eta(2 - 1) + \eta^2(4 - 2) + \eta^3(8 - 4) + \dots = \begin{cases} 1 + \eta \frac{1}{1-2\eta} & \text{if } 0 < \eta < 1/2 \\ \infty & \text{if } \eta \geq 1/2 \end{cases} \quad (7.12)$$

When the degree of horizon generalization has a low value of (say)  $\eta = 0.1$  (i.e., it generalizes for only 1 out of every 10 goals), the Reach is 1.125, not much bigger than that of a policy without horizon generalization. Once the degree of horizon generalization reaches  $\eta = 1/2$  (i.e., generalizes for 1 out of every two goals), the Reach is infinite. In short, the potential reach of horizon generalization is infinite, even when each step of the recursive argument incurs a non-negligible degree of error.

A second important assumption behind our analysis is that very easy goals that *cover the desired space of possible hard goals* (and waypoints to these hard goals) can be reached 100% of the time. In terms of our induction proof, we need the base case to hold. If the base case does not hold (poor performance on easy goals, or easy goals do not have sufficient state coverage to capture harder goals or their

waypoints) but planning invariance holds, then we should not expect to see optimality over arbitrary harder goals. We will observe this empirically with a random policy in our experiments (Fig. 7.4): a random policy is invariant to planning (it always selects random actions, regardless of the goal) yet its performance on nearby goals is mediocre, so it is not surprising that this policy fails to exhibit horizon generalization.

### 7.3 METHODS FOR PLANNING INVARIANCE: OLD AND NEW

In this section we discuss how planning invariance relates to several classes of RL algorithms. Section 7.7 discusses several new directions for designing RL algorithms that are invariant to planning. Section 7.9 recalls figures from prior works in search of evidence for horizon generalization.

**Dynamic programming and temporal difference (TD) learning.** The capacity for TD methods to “stitch” [290] together trajectories offers one route for obtaining policies with horizon generalization. Indeed, our definition of planning invariance is very closely tied with the optimal substructure property [291, pp. 382-387] of dynamic programming *problems*, and likely could be redefined entirely in terms of optimal substructure. Viewing horizon generalization and planning invariance through the lens of machine learning allows us to consider a broader set of tools for achieving invariance and generalization (e.g., special neural network layers, data augmentation).

**Quasimetric Architectures (implicit planning).** Prior methods that employ special neural networks may have some degree of horizon generalization. For example, some prior methods [126, 296, 9] use quasimetric networks to represent a distance function. As the correct distance function satisfies the triangle inequality, it makes sense to use special architectures that are guaranteed to satisfy the triangle inequality. However, prior work rarely examines the generalization or invariance properties of these quasimetric architectures. One way of thinking about quasimetric architectures is that they are invariant to path relaxation ( $d(s, g) \leftarrow \min_w d(s, w) + d(w, g)$ ) [291, p. 609]. This path relaxation is exactly the notion of planning used in our theoretical construction (Theorem 7.1). Thus, these architectures are, by construction, invariant to planning! We use these architectures in our experiments in Section 7.4.

While quasimetric architectures are invariant to path relaxation, other prior methods [297, 298] have proposed architectures that perform value iteration internally and (hence) may be invariant to the Bellman operator. Because Bellman optimality implies planning invariance (c.f. optimal substructure), we expect that these value iteration networks may exhibit some degree of horizon generalization as well.

Table 7.1: Summary of methods and modifications tested

Method	Description	Losses	Critics
CRL	Contrastive RL [121]	$\{\mathcal{L}_{\text{fwd}}, \mathcal{L}_{\text{bwd}}, \mathcal{L}_{\text{sym}}\}$	$\{d_{\ell_2}, d_{\text{MLP}}\}$
SAC	Soft Actor-Critic [292]	$\{\mathcal{L}_{\text{sac}}\}$	$\{Q_{\text{MLP}}\}$
CMD-1	Contrastive metric distillation [9]	$\{\mathcal{L}_{\text{bwd}}\}$	$\{d_{\text{MRN}}\}$

(a) Losses		(b) Architectures	
$\mathcal{L}_{\text{fwd}}$	InfoNCE loss: predict goal $g$ from current state-action $(s, a)$ pair [128]	$d_{\ell_2}$	L2-distance parameterized architecture, uses $\ \phi(s) - \psi(g)\ $ as a distance/critic [11]
$\mathcal{L}_{\text{bwd}}$	Backward InfoNCE loss: predict current state and action $(s, a)$ from future state $g$ [293]	$d_{\text{MLP}}$	Uses multi-layer perceptron (MLP) to parameterize the distance/critic [294, 295]
$\mathcal{L}_{\text{sym}}$	Symmetric contrastive loss: combine the forward and backward contrastive losses [17]	$d_{\text{MRN}}$	Metric residual network, uses a quasimetric architecture to parameterize the distance/critic [127]
$\mathcal{L}_{\text{sac}}$	Temporal difference loss [292]	$Q_{\text{MLP}}$	MLP-parameterized Q-function [292]

**Explicit planning methods.** While our proof of planning used a specific notion of planning, prior work has proposed RL methods that employ many different styles of planning: graph search methods [117, 200, 299, 171], model-based methods [300–304], collocation methods [305], and hierarchical methods [176, 306, 198, 307]. Insofar as these methods approximate the method used in our proof, it is reasonable to expect that they may achieve some degree of planning invariance and horizon generalization (see Fig. 7.9). Prior methods in this space are typically evaluated on the *training* distribution, so their horizon generalization capabilities are typically not evaluated. However, the improved generalization properties might have still contributed to the faster learning on the *training* tasks: after just learning the easier tasks, these methods would have already solved the complex tasks, leading to higher average success rates.

**Data augmentation.** Finally, prior work [1, 171] has argued that data augmentation provides another avenue for achieving the benefits typically associated with planning or dynamic programming.

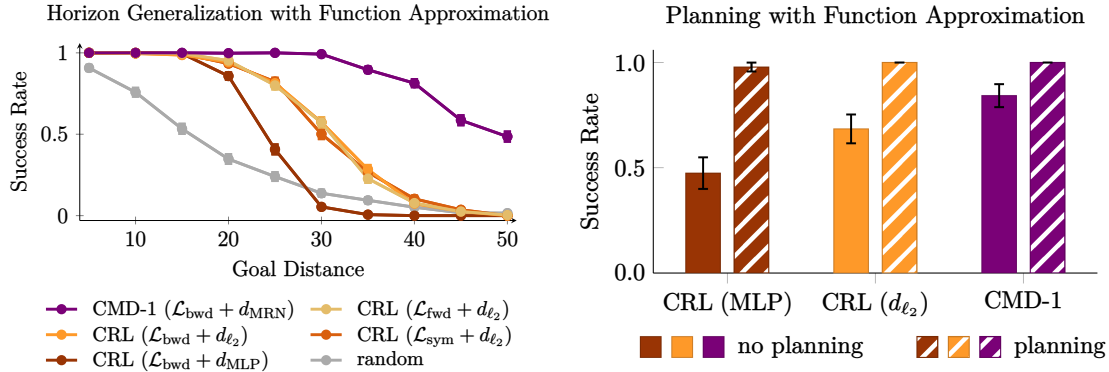


Figure 7.4: **Quantifying horizon generalization and invariance to planning.** On a simple navigation task, we collect short trajectories and train two goal-conditioned policies, comparing both to a random policy. (*Left*) We evaluate on  $(s, g)$  pairs of varying distances, observing that metric regression with a quasimetric exhibits strong horizon generalization. (*Right*) In line with our analysis, the policy that has strong horizon generalization is also more invariant to planning: combining that policy with planning does not increase performance. Figure 7.7 shows a version of this plot that also includes the tabular setting.

## 7.4 EXPERIMENTS

The aim of our experiments is to provide intuition into what horizon generalization and planning invariance are, why it should be possible to achieve these properties, and to study the extent to which existing methods already achieve these properties. We also present an experiment highlighting why horizon generalization is a useful notion even when considering temporal difference methods (Section 7.4).

We start with a didactic, tabular navigation task (Fig. C.2), connecting short horizon trajectories and evaluating performance on long-horizon tasks. In our first experiment, we measure the empirical average hitting time distance between all pairs of states. We define a policy that acts greedily with respect to these distances, measuring performance of this “metric regression” policy in Fig. 7.7 (*Top Left*). The degree of horizon generalization can be quantified by comparing its success rate on nearby  $(s, g)$  pairs to more distant pairs. We compare to a “metric regression with quasimetric” method that projects the empirical hitting times *into a quasimetric* by performing path relaxation updates until convergence ( $d(s, g) \leftarrow \min_w d(s, w) + d(w, g)$ ). Fig. 7.7 (*Top Left*) shows that this policy achieves near perfect horizon generalization. While this result makes intuitive sense (this algorithm is very similar to Dijkstra’s algorithm), it nonetheless highlights one way in which a method trained on nearby start-goal pairs can generalize to more distant pairs.

We study planning invariance of these policies by comparing the success rate

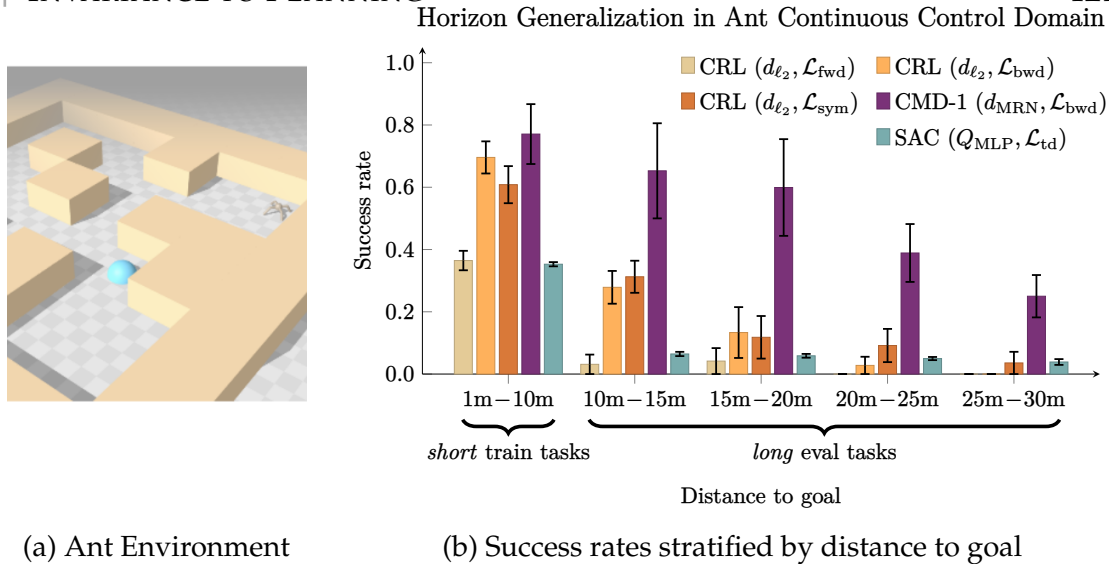


Figure 7.5: **Measuring horizon generalization in a high-dimensional (27D observation, 8DoF control) task.** (Left) We use an enlarged version of the quadruped “ant” environment, training all goal-conditioned RL methods on (start, goal) pairs that are at most 10 meters apart. (Right) We evaluate several RL methods, measuring the horizon generalization of each. These results reveal that (i) some degree of horizon generalization is possible; (ii) the learning algorithm influences the degree of generalization; (iii) the value function architecture influences the degree of generalization; and (iv) no method achieves perfect generalization, suggesting room for improvement in future work. The ratio of success at 10m vs 5m and 20m vs 10m corresponds to  $\eta$  from Section 7.2. Results are plotted with standard errors across random seeds.

of each policy (on distant start-goal pairs) when the policy is conditioned on the goal versus on a waypoint. See Appendix C.5 for details. As shown in Fig. 7.7 (Top Right), the “metric regression with quasimetric” policy exhibits stronger planning invariance, supporting our theoretical claim that (Theorem 7.1) planning invariance is possible.

We next study whether these properties exist when using function approximation. For this experiment, we adopt the contrastive RL method [121] for estimating the distances, comparing different architectures and loss functions. The results in Fig. 7.4 (Left) show that both the architecture and the loss function can influence horizon generalization, with the strongest generalization being achieved by a CMD-1 [9]. Intuitively this makes sense, as this method was explicitly designed to exploit the triangle inequality, which is closely linked to planning invariance. Fig. 7.4 (Right) shows the degree of planning invariance for these policies. Supporting our analysis, the policy most invariant to planning trained over short horizon tasks shows the strongest horizon generalization.

To better understand the relationship between planning invariance and horizon



generalization, we used the data from Fig. 7.4 (*Left*) to estimate the horizon generalization parameter  $\eta$ , and used the data from the (*Right*) to compute the ratio of performance with and without planning. Fig. 7.8 shows these data as a scatter plot. These two quantities are well correlated, supporting Theorem 7.2’s claim that horizon generalization is closely linked to planning invariance. Note that methods that use an L2-distance parameterized architecture demonstrate stronger horizon generalization and planning invariance than that which uses an MLP, suggesting that some degree of planning invariance might be had even without a quasimetric architecture. Intriguingly, these methods using the L2 architecture have a value of  $\eta \approx 0.5$ , right at the critical point between bounded and unbounded reach (see Section 7.2). The CMD-1 method, which is explicitly designed to incorporate the triangle inequality, exhibits much stronger planning invariance and horizon generalization ( $\eta \approx 0.8 \gg 0.5$ ), well above the critical point. Finally, note that the random policy is an outlier: it achieves perfect planning invariance (it always takes random actions, regardless of the goal) yet poor horizon generalization. This random policy highlights a key assumption in our analysis: that the policy *always* succeeds at reaching nearby goals (in Fig. 7.4, note that the success rate on the easiest goals is strictly less than 1).

### Empirically Studying Horizon Generalization in a High-dimensional Setting

Our next set of experiments study horizon generalization and planning invariance in the context of a high-dimensional quadrupedal locomotion task (see Fig. 7.5). We start by running a series of experiments to compare the horizon generalization of different learning algorithms (CRL [121] and SAC [292]) and distance metric architectures (details in Appendix C.5). The results in Fig. 7.5 highlight that both the learning algorithm and the architecture can play an important role in horizon generalization, while also underscoring that achieving high horizon generalization in high-dimensional settings remains an open problem. See Section 7.3 for a summary of the methods used in these experiments.

### Impact of Horizon Generalization on Bellman Errors

Why should someone using a temporal difference method care about horizon generalization, if TD methods are supposed to provide this property for free? One hypothesis is that methods for achieving horizon generalization will also help decrease the Bellman error, especially for unseen start-goal pairs. We test this hypothesis by measuring the Bellman error throughout training of the contrastive RL method (same method as Fig. 7.4), with two different architectures. The results in Fig. 7.6 show that the architecture that exhibits stronger horizon generalization ( $d_{\ell_2}$ ) also has a lower Bellman error throughout training. Thus, while TD methods may achieve horizon generalization at convergence (at least in the tabular setting with infinite data), a stronger understanding of horizon generalization may nonethe-

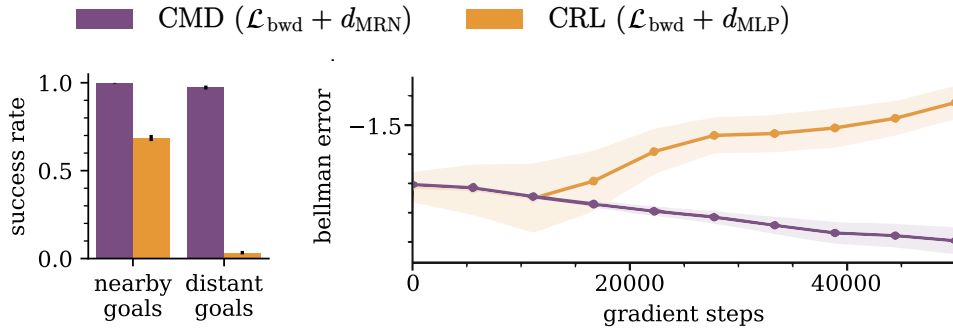


Figure 7.6: **Impact of horizon generalization on Bellman errors.** (Left) Two goal-reaching methods exhibit different horizon generalization. (Right) Despite neither method being trained with the Bellman loss, we observe that the method with stronger horizon generalization has a lower Bellman loss. Thus, understanding horizon generalization may be important even when using TD methods (which guarantee horizon generalization at convergence).

less prove useful for designing architectures that enable faster convergence of TD methods.

## 7.5 DEFINITION OF PATH RELAXATION

**Definition 7.4** (Path relaxation operator). Let  $\text{PATH}_d(s, G)$  be the path relaxation operator over quasimetric  $d(s, G)$ . For any triplet of state and state distributions  $(s, W, G) \in \mathcal{S} \times \mathcal{P}(\mathcal{S}) \times \mathcal{P}(\mathcal{S})$ ,

$$\text{PATH}_d(s, G) \triangleq \min_W d(s, W) + d(W, G). \quad (7.13)$$

In the controlled, fixed goal setting, define

$$\text{PATH}_d^{\text{FIX}}(s, g) \triangleq \min_w d(s, w) + d(w, g). \quad (7.14)$$

Thus, invariance to the path relaxation operator is a form of self-consistency; any triplet of predictions should satisfy the following identity:

$$d(s, G) \leq d(s, W) + d(W, G)$$

or in the controlled, fixed goal setting

$$d(s, g) \leq d(s, w) + d(w, g).$$

which is the familiar triangle inequality. Conveniently, the quasimetric neural network architecture [127, 143, 130] innately satisfies the triangle inequality before seeing any training data.



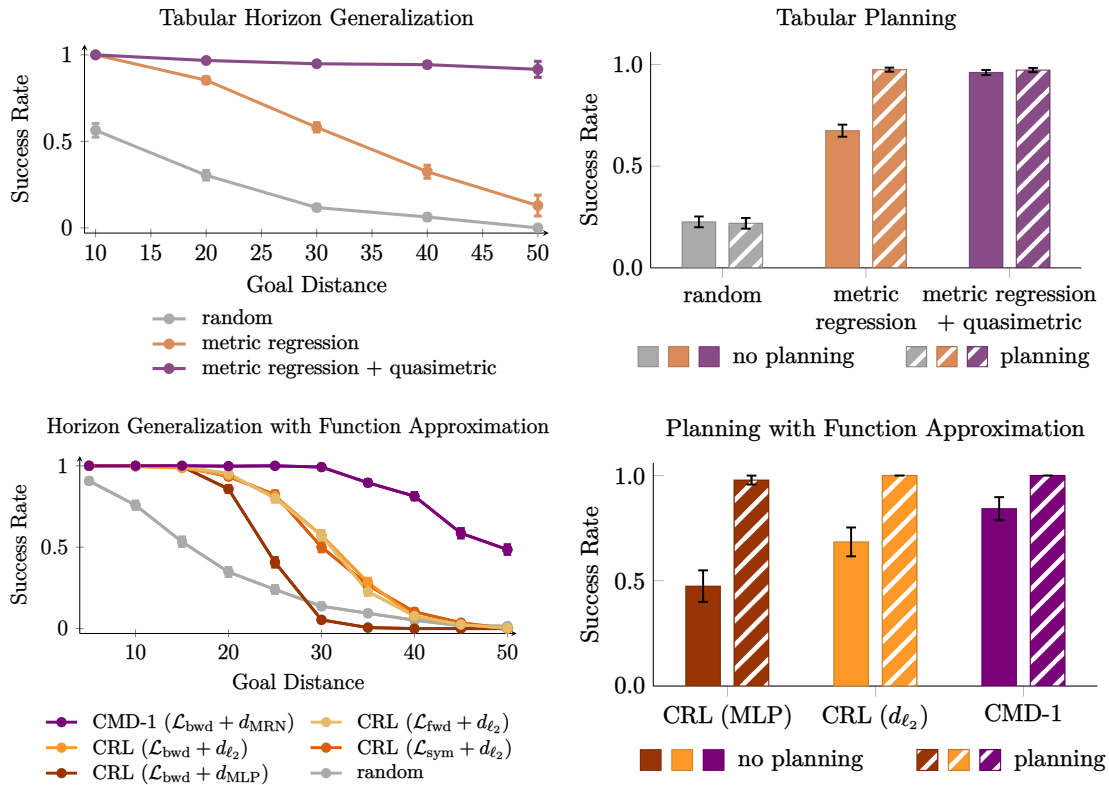


Figure 7.7: **Quantifying horizon generalization and invariance to planning.** On a simple navigation task, we collect short trajectories and train two goal-conditioned policies, comparing both to a random policy. (*Top Left*) We evaluate on  $(s, g)$  pairs of varying distances, observing that metric regression with a quasimetric exhibits strong horizon generalization. (*Top Right*) In line with our analysis, the policy that has strong horizon generalization is also more invariant to planning: combining that policy with planning does not increase performance. (*Bottom Row*) We repeat these experiments using function approximation (instead of a tabular model), observing similar trends.

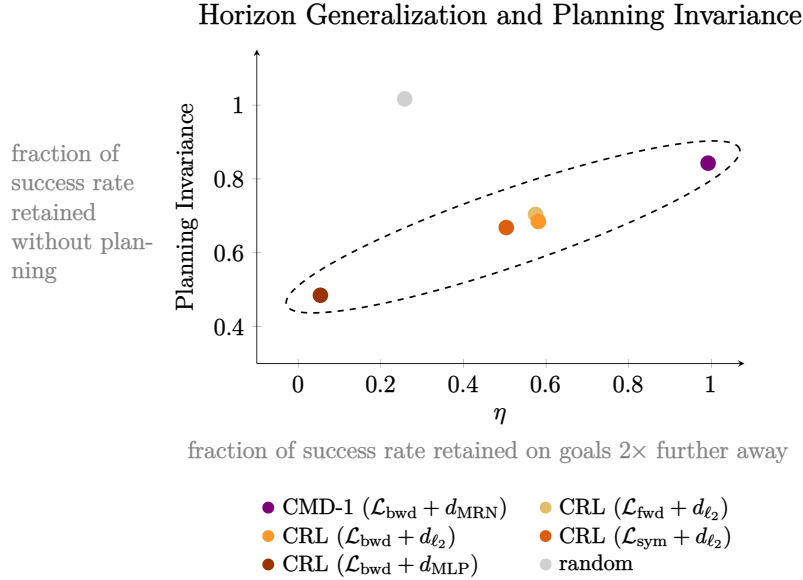


Figure 7.8: Quantifying horizon generalization ( $x$ -axis) and planning invariance ( $y$ -axis). See text Section 7.4 for more details.

**Definition 7.5** (Path relaxation operator with actions). Let  $\text{PATH}_d(s, a, G)$  be the path relaxation operator over quasimetric  $d(s, a, G)$ . For any triplet of state and state distributions  $(s, W, G) \in \mathcal{S} \times \mathcal{S} \times \mathcal{S}$ ,

$$\text{PATH}_d(s, G) \triangleq \min_w d(s, W) + d(W, G). \quad (7.15)$$

In the controlled, fixed goal setting, define

$$\text{PATH}_d^{\text{FIX}}(s, g) \triangleq \min_w d(s, w) + d(w, g). \quad (7.16)$$

## 7.6 FORMALIZING PLANNING INVARIANCE

In this section, we prove results discussed in Section 7.2 and versions of results in Section 7.2 for the general stochastic, distributional setting.

Planning invariance exists

**Theorem 7.1** (Planning invariance exists). Assume a controlled, fixed goal setting. For every quasimetric  $d(s, g)$  over state space  $\mathcal{S}$ , there exists a policy  $\pi_d^{\text{FIX}}(a | s, g)$  and planning operator  $\text{PLAN}_d^{\text{FIX}} \in \mathbf{plan}^{\text{FIX}}$  such that  $\pi_d^{\text{FIX}}(a | s, g) = \pi_d^{\text{FIX}}(a | s, w)$  for  $w = \text{PLAN}_d^{\text{FIX}}(s, g)$ .

*Proof.* Let  $s, g \in \mathcal{S}$  and the action-free distance function be  $d(s, g) = \min_a d(s, a, g)$ ; this statement is true for the contrastive successor distances (Eq. 7.3). Define the

(deterministic) planned waypoint as

$$w_{\text{PLAN}} \leftarrow \text{PLAN}_d^{\text{FIX}}(s, g) \in \arg \min_{w \in \mathcal{S}} d(s, w) + d(w, g). \quad (7.17)$$

We can then construct the following policy:

$$\pi_d^{\text{FIX}}(a \mid s, g) \in \text{OPT}_d(s, g) \triangleq \arg \min_{a \in \mathcal{A}} d(s, a, g). \quad (7.18)$$

and later restrict the selection of the action to reach waypoint  $w_{\text{PLAN}}$  to get planning invariance, where  $w_{\text{PLAN}} \in \arg \min_{w \in \mathcal{S}} d(s, w) + d(w, g)$ . Applying this policy to  $(s, w_{\text{PLAN}})$ ,

$$\begin{aligned} \pi_d^{\text{FIX}}(a \mid s, w_{\text{PLAN}}) \in \text{OPT}_d(s, w_{\text{PLAN}}) &\triangleq \arg \min_{a \in \mathcal{A}} d(s, a, w_{\text{PLAN}}) \\ &= \arg \min_{a \in \mathcal{A}} d(s, a, w_{\text{PLAN}}) + d(w_{\text{PLAN}}, g) \\ &= d(s, w_{\text{PLAN}}) + d(w_{\text{PLAN}}, g) \\ &\subseteq \arg \min_{a \in \mathcal{A}} d(s, a, g) \\ &= \text{OPT}_d(s, g). \end{aligned} \quad (7.19)$$

Thus, for a given deterministic planning algorithm defined as in Eq. (7.17), there exists some deterministic policy  $\pi_d^{\text{FIX}}(a \mid s, g) = \pi_d^{\text{FIX}}(a \mid s, w_{\text{PLAN}}) \in \text{OPT}_d(s, w_{\text{PLAN}}) \subseteq \text{OPT}_d(s, g)$  which is planning invariance.  $\square$

### Quasimetric Over Distributions

**Definition 7.6** (Quasimetric over distributions). *For a given quasimetric  $d_{\text{QM}} \in \mathcal{Q}$ , we define the quasimetric over distributions as*

$$d_{\text{QMD}}(L, M) = \left( \int_{\mathcal{S} \times \mathcal{S}} p_L(l) p_M(m) d_{\text{QM}}(L, M) dl dm \right) \times \left( 1 - \int_{\mathcal{S}} \sqrt{p_L(s) p_M(s)} ds \right). \quad (7.20)$$

We show Definition 7.6 is a valid quasimetric.

This is a known result given the definition of  $d_{\text{QMD}}$  as a Wasserstein distance and cost function  $d_{\text{QM}}(a, b)$  that is a quasimetric, but we reproduce the proof here for completeness.

*Proof.* We check the conditions of a quasimetric for  $d_{\text{QMD}}(A, B)$  with quasimetric cost function  $d_{\text{QM}}(a, b)$ .

**Positive semidefiniteness:** By definition of  $\gamma(a, b)$  and  $d_{\text{QM}}(a, b)$ , we have  $d_{\text{QMD}}(A, B) \geq 0$  for all  $A, B$ . To show that  $d_{\text{QMD}}(A, B) = 0$  if and only if  $A = B$ :

$$\begin{aligned} d_{\text{QMD}}(A, A) &= \inf_{\gamma \in \Pi(A, A)} \int_{\mathcal{S} \times \mathcal{S}} d_{\text{QM}}(a, b) \gamma(a, b) \, da \, db \\ &\leq \int_{\mathcal{S} \times \mathcal{S}} d_{\text{QM}}(a, b) \gamma_D(a, b) \, da \, db \quad (\text{set } \gamma \text{ as diagonal matrix } \gamma_D) \\ &= \int_{\mathcal{S}} d_{\text{QM}}(a, a) \mu(a) \, da = 0 \end{aligned}$$

For the other direction, we have that  $d_{\text{QMD}}(A, B) = 0$  implies that  $\gamma(a, b) = 0$  for all  $a \neq b$ . Thus,  $A = B$ .

**Asymmetry:** We have that  $d_{\text{QMD}}(A, B) \neq d_{\text{QMD}}(B, A)$  in general, as the quasimetric  $d_{\text{QM}}(a, b)$  is not necessarily symmetric.

**Triangle inequality:** Let  $A, B, C$  be three probability measures. Let  $\gamma_{1,2}^*$  and  $\gamma_{2,3}^*$  be minimizers of  $d_{\text{QMD}}(A, B)$  and  $d_{\text{QMD}}(B, C)$  respectively. We can construct some  $\gamma_{1,2,3}(a, b, c)$  such that

$$\begin{aligned} \int_{\mathcal{X}} \gamma_{1,2,3}(a, b, c) \, da &= \gamma_{1,2}^* \\ \int_{\mathcal{X}} \gamma_{1,2,3}(a, b, c) \, db &= \gamma_{2,3}^* \\ \int_{\mathcal{X}} \gamma_{1,2,3}(a, b, c) \, dc &= \gamma_{1,3} \end{aligned}$$

where  $\gamma_{1,3}$  is **not necessarily** the optimal joint distribution to minimize  $d_{\text{QMD}}(A, C)$ . Then, we have:

$$\begin{aligned} d_{\text{QMD}}(A, C) &\leq \int_{\mathcal{S} \times \mathcal{S}} d_{\text{QM}}(a, c) \gamma_{1,3}(a, c) \, da \, dc \\ &= \int_{\mathcal{S} \times \mathcal{S}} d_{\text{QM}}(a, c) \gamma_{1,2,3}(x, y, z) \, da \, db \, dc \\ &\leq \int_{\mathcal{S} \times \mathcal{S} \times \mathcal{S}} (d_{\text{QM}}(a, b) + d_{\text{QM}}(b, c)) \gamma_{1,2,3}(a, b, c) \, da \, db \, dc \\ &\quad (d_{\text{QM}}(a, b) \text{ satisfies } \Delta\text{-ineq}) \\ &= \int_{\mathcal{S} \times \mathcal{S} \times \mathcal{S}} d_{\text{QM}}(a, b) \gamma_{1,2,3}(a, b, c) \, da \, db \, dc \\ &\quad + \int_{\mathcal{S} \times \mathcal{S} \times \mathcal{S}} d_{\text{QM}}(b, c) \gamma_{1,2,3}(a, b, c) \, da \, db \, dc \\ &= d_{\text{QMD}}(A, B) + d_{\text{QMD}}(B, C) \end{aligned}$$

as desired. Therefore,  $d_{\text{QMD}}$  is a quasimetric and we are done.  $\square$

## Quasimetrics, Policies, and Planning Invariance (Stochastic Setting)

**Definition 7.7** (Quasimetric over actions in general stochastic setting). *Assume  $d(s, g)$  is the Contrastive Successor Distance [9]. Define the stochastic-setting quasimetric over actions as*

$$d(s, a, G) \triangleq d(s, S'_{(s,a)}) + d(S'_{(s,a)}, G)$$

where  $S'_{(s,a)} = p(s' | s, a)$  is the distribution over next-step states after taking action  $a$  from state  $s$ .

**Definition 7.8** (Quasimetric policy in general stochastic setting). *Extending the deterministic quasimetric policy to stochastic settings,*

$$\pi_d(a | s, G) \in \text{OPT}_d(s, G) \triangleq \arg \min_a d(s, a, G).$$

The existence of planning invariance in stochastic settings follows from these quasimetric definitions.

**Lemma 7.4** (Planning invariance exists in general stochastic setting). *For every quasimetric  $d(s, G)$  where  $G \in \mathcal{P}(\mathcal{S})$ , there exists a policy*

$$\pi_d(a | s, G) \in \arg \min_{a \in \mathcal{A}} d(s, a, G)$$

where  $\pi_d(a | s, G) = \pi_d(a | s, W)$ , and planning operator

$$\text{PLAN}_d(s, a, G) = W_{\text{PLAN}} \in \arg \min_{W \in \mathcal{P}(\mathcal{S})} (d(s, a, W) + d(W, G)).$$

*Proof.* For any  $s, G$  pairs,

$$\begin{aligned} \min_a d(s, a, G) &= \min_a d(s, S'_{(s,a)}) + d(S'_{(s,a)}, G) \\ &= \min_a \min_W d(s, S'_{(s,a)}) + d(S'_{(s,a)}, W) + d(W, G) && (\Delta\text{-ineq}) \\ &= \min_a \min_W d(s, a, W) + d(W, G) \end{aligned}$$

From Definition 7.8, let quasimetric policy  $\pi$  be

$$\pi_d(a | s, G) \in \text{OPT}_d(s, G) \triangleq \arg \min_{a \in \mathcal{A}} d(s, a, G).$$

Now, applying this policy to state-waypoint pair  $(s, W_{\text{PLAN}})$ ,

$$\begin{aligned} \pi(a|s, W_{\text{PLAN}}) &\in \text{OPT}_d(s, W_{\text{PLAN}}) \\ &\triangleq \arg \min_{a \in \mathcal{A}} d(s, a, W_{\text{PLAN}}) \\ &= \arg \min_{a \in \mathcal{A}} d(s, a, W_{\text{PLAN}}) + d(W_{\text{PLAN}}, G) \\ &\subseteq \arg \min_{a \in \mathcal{A}} d(s, a, G) \end{aligned}$$

as desired. Thus, for the given stochastic planning algorithm, there exists some policy  $\pi_d(a | s, G) = \pi_d(a | s, W_{\text{PLAN}}) \in \text{OPT}_d(s, W_{\text{PLAN}}) \in \text{OPT}_d(s, G)$ .  $\square$

Horizon generalization exists

**Theorem 7.2** (Horizon generalization exists). *A quasimetric policy  $\pi_d^{\text{FIX}}(a | s, g)$  that is optimal over  $\mathcal{B}_c = \{(s, g) \in \mathcal{S} \times \mathcal{S} \mid d(s, g) < c\}$  for some finite  $c > 0$  implies optimality over the entire state and goal space  $\mathcal{S} \times \mathcal{S}$ .*

*Proof.* We use induction and prove the following more general result for policies  $\pi_d(a | s, G)$  defined over state-goal distribution pairs  $(s, G)$ . See earlier sections in Section 7.6 for quasimetric, policy, and planning definitions over distributions:

**Lemma 7.5** (Horizon generalization exists, stochastic settings). *A quasimetric policy  $\pi_d(a | s, G)$  that is optimal over  $\mathcal{B}_c = \{(s, G) \in \mathcal{S} \times \mathcal{P}(\mathcal{S}) \mid d(s, G) < c\}$  for some finite  $c > 0$  implies optimality over the entire state and goal distribution space  $\mathcal{S} \times \mathcal{P}(\mathcal{S})$ .*

Note that we can set  $G$  to a Delta function at a single goal  $g$  to recover the fixed policy  $\pi_d(a | s, G)$ .

Assume optimality over  $\mathcal{B}_n = \{(s, G) \in \mathcal{S} \times \mathcal{P}(\mathcal{S}) \mid d(s, G) < 2^n c\}$  for arbitrary  $n \in \mathbb{Z}^+$ . Without loss of generality, consider arbitrary state  $s \in \mathcal{S}$  and all goal distributions  $\mathcal{D}_n = \{G \in \mathcal{P}(\mathcal{S}) \mid d(s, G) < 2^n c\}$ .

We can consider the space of distributions  $\mathcal{D}'_n$  that are  $2^n c$  distance away from goal distribution  $G \in \mathcal{D}_n$ :

$$\mathcal{D}'_n = \{S' \in \mathcal{P}(\mathcal{S}) \mid d(G, S') < 2^n c, G \in \mathcal{D}_n\} = \{S' \in \mathcal{P}(\mathcal{S}) \mid d(s, S') < 2^{n+1} c\}.$$

where

$$\mathcal{B}'_n = \{(s, S') \mid S' \in \mathcal{D}'_n\} = \mathcal{B}_{n+1}.$$

Invoking the definition of the quasimetric policy  $\pi_d(a | S, S')$ , for some waypoint distribution  $W_{\text{PLAN}} \in \arg \min_{W \in \mathcal{D}'_n} (d(s, a, W) + d(W, G))$  over distributions  $G \in \mathcal{D}'_n$ :

$$\pi_d(a | s, G) \in \arg \min_{a \in \mathcal{A}} d(s, a, W_{\text{PLAN}}).$$

To show that there always exists some planned waypoint distribution  $W_{\text{PLAN}}$  within the region of assumed optimality  $\mathcal{D}_n$  from the inductive assumption, we consider the case  $W_{\text{PLAN}} \notin \mathcal{D}_n$  and show that there exists some  $W_{\text{PLAN, IN}} \in \mathcal{D}_n$  such that  $d(s, a, W_{\text{PLAN, IN}}) + d(W_{\text{PLAN, IN}}, G) = d(s, a, G)$ . By the triangle inequality,

$$\begin{aligned}
d(s, a, G) &= \min_{W \in \mathcal{D}'_n} (d(s, a, W) + d(W, G)) \\
&= d(s, a, W_{\text{PLAN}}) + d(W_{\text{PLAN}}, G) \\
&= \min_{W_{\text{OUT}} \in \mathcal{D}'_n \setminus \mathcal{D}_n} d(s, a, W_{\text{OUT}}) + d(W_{\text{OUT}}, G) \\
&= \min_{W_{\text{OUT}} \in \mathcal{D}'_n \setminus \mathcal{D}_n} \min_{W_{\text{IN}} \in \mathcal{D}_n} (d(s, a, W_{\text{IN}}) + d(W_{\text{IN}}, W_{\text{OUT}})) + d(W_{\text{OUT}}, G) \\
&= \min_{W_{\text{IN}} \in \mathcal{D}_n} \min_{W_{\text{OUT}} \in \mathcal{D}'_n \setminus \mathcal{D}_n} d(s, a, W_{\text{IN}}) + (d(W_{\text{IN}}, W_{\text{OUT}}) + d(W_{\text{OUT}}, G)) \\
&= \min_{W_{\text{IN}} \in \mathcal{D}_n} d(s, a, W_{\text{IN}}) + d(W_{\text{IN}}, G) \quad (\triangle\text{-ineq}) \\
&= d(s, a, W_{\text{PLAN, IN}}) + d(W_{\text{PLAN, IN}}, G),
\end{aligned}$$

so there always exists an optimal state-waypoint distribution pair within the assumed optimality region  $\mathcal{B}_n$ ; we can then restrict  $(s, W_{\text{PLAN}}) \in \mathcal{B}_n$ . Therefore, with the previously defined quasimetric policy  $\pi_d(a \mid s, G)$ ,

$$\begin{aligned}
\pi_d(a \mid (s, W_{\text{PLAN}}) \in \mathcal{B}_n) &\in \arg \min_{a \in \mathcal{A}} d(s, a, W_{\text{PLAN}}) \quad (\text{inductive assumption}) \\
&\subseteq \arg \min_{a \in \mathcal{A}} d(s, a, G). \quad (\text{Lemma 7.4: planning invariance})
\end{aligned}$$

Therefore, policy  $\pi_d(a \mid s, G)$  is optimal over  $\mathcal{B}_{n+1}$  following the inductive assumption, and, since  $d(s, G)$  is finite for all  $(s, G) \in \mathcal{S} \times \mathcal{S}'_s$  where goal distribution  $G$  is reachable from state  $s$ , Theorem 7.2 follows.  $\square$

Horizon generalization is nontrivial

**Remark 7.3** (Horizon generalization is nontrivial). *For an arbitrary policy, optimality over  $\mathcal{B}_c = \{(s, g) \in \mathcal{S} \times \mathcal{S} \mid d(s, g) < c\}$  for some finite  $c > 0$  is not a sufficient condition for optimality over the entire state space  $\mathcal{S}$ .*

*Proof.* We restrict our proof to the fixed, controlled setting and let quasimetric  $d(s, g)$  be the successor distance  $d_{\text{SD}}(s, g)$ —this assumption lets us directly equate the optimal horizon  $H$  to the distance  $d_{\text{SD}}(s, g)$ , but note that similar arguments can be applied by treating  $d(s, g)$  as a generalized notion of horizon.

Consider goal-conditioned policy  $\pi^{*,H}(a \mid s, g)$  that is optimal for  $(s, g)$  pairs over some horizon  $H$ . Assume there is at least one goal  $g'$  that is optimally  $H + 1$  actions away from  $s$ , and that there exists some optimal waypoint  $s'$  en route to

$g'$  reachable via actions  $\mathcal{A}' \subset \mathcal{A}$  (where  $\mathcal{A} \setminus \mathcal{A}'$ , the set of suboptimal actions, is nonempty).

We can then construct a policy  $\pi^{H+1}$  where  $\pi^{H+1}(a | s, g')$  returns an action in the suboptimal set  $\mathcal{A} \setminus \mathcal{A}'$ , and  $\pi^{H+1}$  restricted to state-goal pairs horizon  $H$  away is equivalent to  $\pi^{*,H}$ . Therefore, an arbitrary optimal goal-reaching policy over some restricted horizon  $H$  does not necessarily exhibit horizon generalization.  $\square$

## 7.7 NEW METHODS FOR PLANNING INVARIANCE

While our aim is not to propose a new method, we will discuss several new directions that may be examined for achieving planning invariance.

**Representation learning.** As shown in Fig. 7.2, planning invariance implies that some internal representation inside a policy must map start-goal inputs and start-waypoint inputs to similar representations. What representation learning objective would result in representations that, when used for a policy, guarantee horizon generalization?<sup>1</sup> The fact that plans over representations sometimes correspond to geodesics [308, 11] hints that this may be possible.

**Flattening hierarchical methods.** While hierarchical methods often achieve higher success rates in practice, it remains unclear why flat methods cannot achieve similar performance given the same data. While prior work has suggested that hierarchies may aid in exploration [309], it may be the case that they (somehow) exploit the metric structure of the problem. Once this inductive bias is identified, it may be possible to imbue it into a “flat” policy so that it can achieve similar performance (without the complexity of hierarchical methods).

**Policies that learn to plan.** While explicit planning methods may be invariant to planning, recent work has suggested that certain policies can *learn* to plan when trained on sufficient data [171, 310]. Insofar as neural networks are universal function approximators, they may learn to approximate a planning operator internally. The best way of learning such networks that implicitly learn to perform planning remains an open question.

**MDP reductions.** Finally, is it possible to map one MDP to another MDP (e.g., with different observations, with different actions) so that any RL algorithm applied to this transformed MDP automatically achieves the planning invariance property?

## 7.8 SELF-CONSISTENT MODELS

In machine learning, we usually strive for *consistent* models: ones that faithfully predict the training data. Sometimes (often), however, a model that is consistent

---

<sup>1</sup>The construction in our proof is a degenerate case of this, where the internal representations are equal to the output actions.

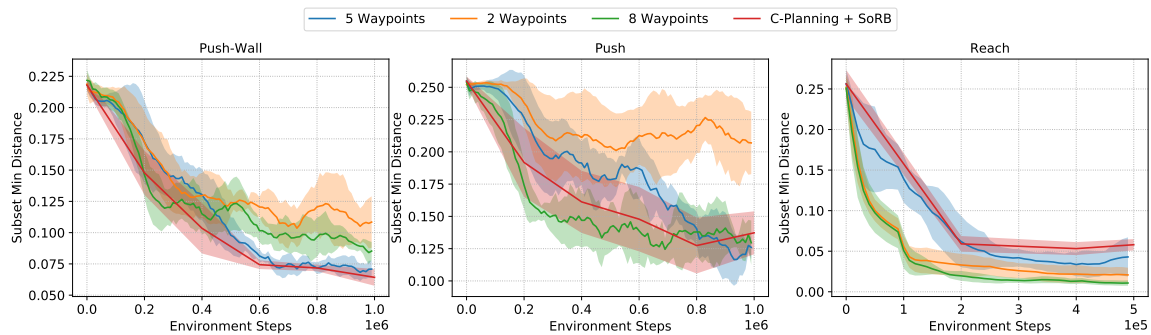


with the training data may be inconsistent with other yet-to-be-seen training examples. In the absence of infinite data, one way of performing model selection is to see whether a model's predictions are self-consistent with one another. This is perhaps most easily seen in the case of metric learning, as discussed in this chapter. If we are trying to learn a metric  $d(x, y)$ , then the properties of metrics tell us something about the predictions that our model should make, both on seen and unseen inputs. For example, even on unseen inputs, our model's predictions should obey the triangle inequality. Given many candidate models that are all consistent with the training data, we may be able to rule out some of those models if their predictions on unseen examples are not "logically" consistent (e.g., if they violate the triangle inequality). *One way of interpreting quasimetric neural networks is that they are architecturally constrained to be self-consistent.* We will discuss a few implications of this observation.

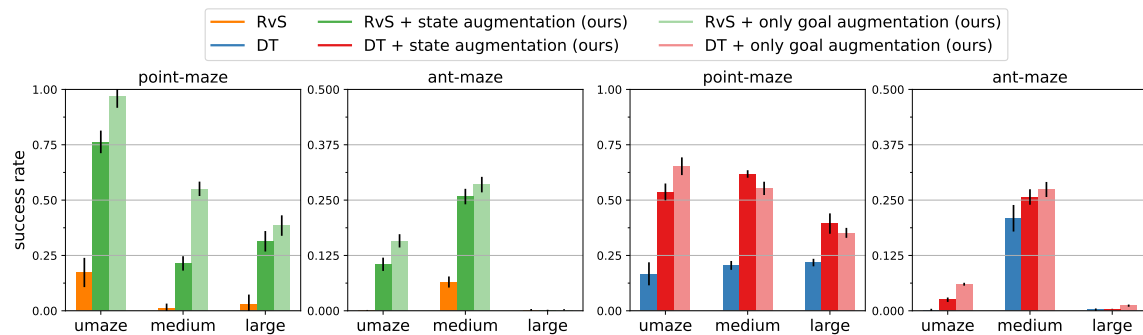
**Do self-consistent models know what they know?** What if we assume that quasimetric networks can generalize? That is, after learning that (say)  $s_1$  and  $s_2$  are 5 steps apart, it will predict that similar states  $s'_1$  and  $s'_2$  are also 5 steps apart. Because the model is architecturally constrained to be a quasimetric, this prediction (or "hallucination") could also result in changing the predictions for other s-g pairs. That is, this new "hallucinated" edge  $s'_1 \rightarrow s'_2$  might result in path relaxation for yet other edges.

**What other sorts of models are self-consistent?** There has been much discussion of self-consistency in the language-modeling literature [311, 312]. Many of these methods are predicated on the same underlying as self-consistency in quasimetric networks: checking whether the model makes logically consistent predictions on unseen inputs. Logical consistency might be used to determine that a prediction is unlikely, and so the model can be updated or revised to make a different prediction instead.

There is an important difference between this example and the quasimetrics. While the axiom used for checking self-consistency in quasimetrics was the triangle inequality, in this language modeling example self-consistency is checked using the predictions from the language model itself. In the example of quasimetrics, our ability to precisely write down a mathematical notion of consistency enabled us to translate that axiom into an architecture that is self-consistent with this property. This raises an intriguing question: *Can we quantify the rules of logic in such a way that they can be translated into a logically self-consistent language model?* What makes this claim seem alluringly tangible is that there is abundant literature from mathematics and philosophy on quantifying logical rules [313].



(a) Zhang et al. [200]



(b) Ghugare et al. [1]

**Figure 7.9: Evidence of Horizon Generalization and Planning Invariance from Prior work.** (a) Prior work has observed that if policies are trained in an online setting and perform planning during exploration, then those policies see little benefit from doing planning during evaluation. This observation suggests that these policies may have learned to be planning invariant. While results are not stratified into training and testing tasks, we speculate that the faster learning of that method (relative to baselines, not shown) may be explained by the policy generalizing from easy tasks (which are learned more quickly) to more difficult tasks. (b) Prior work studies how data augmentation can facilitate combinatorial generalization. While the notion of combinatorial generalization studied there is slightly from horizon generalization, a method that performs combinatorial generalization would also achieve effective horizon generalization.

## 7.9 EVIDENCE OF HORIZON GENERALIZATION AND PLANNING INVARIANCE FROM PRIOR WORK

Not only do the experiments in Section 7.4 provide evidence for horizon generalization and planning invariance, but we also can find evidence of these properties in the experiments run by prior work. This section reviews three such examples, with the corresponding figures from prior work in Fig. 7.9:

1. Zhang et al. [200] propose a method for goal-conditioned RL in the online setting that performs planning during exploration. While not the main focus of the paper, an ablation experiment in that paper hints that their method may have some degree of planning invariance: after training, the policy produced by their method is evaluated both with and without planning, and achieves similar success rates. This experiment hints at another avenue for achieving planning invariance: rather than changing the architecture or learning rule, simply changing how data are collected may be sufficient.
2. Ghugare et al. [1] propose a method for goal-conditioned RL in the offline setting that performs temporal data augmentation. Their key result, reproduced above, is that the resulting method generalizes better to unseen start-goal pairs, as compared with a baseline. While this notion of generalization is not exactly the same as horizon generalization (unseen start-goal pairs may still be close to one another), the high success rates of the proposed method suggest that method does not *just* generalize to nearby start-goal pairs, but also exhibits horizon generalization by succeeding in reaching unseen distant start-goal pairs.

## 7.10 CONCLUSION

Our aim in this chapter is to give a name to a type of generalization that has been observed before, but (to the best of our knowledge) has never been studied in its own right: the capacity to generalize from nearby start-goal pairs to distant goals. Seen from one perspective, this property is trivial — it is an application of the optimal substructure property, and the original Q-learning method [314] already achieves this property. Seen from another perspective, this property may seem magical: how can one *guarantee* that a policy trained over easy tasks can *extrapolate* from easy tasks to hard tasks?

Our contribution in this paper is to provide a theoretical framework for understanding this property as a form of self-consistency over model architecture, and show how we can obtain and measure this property in practice. The experiments in Section 7.4 then connect these insights to concrete advice for structuring the representation for goal-reaching.

1. Policies defined over metric architectures that measure state dissimilarity have *planning invariance*.
2. Planning invariance is a desirable feature that is correlated with the notion of *horizon generalization*.
3. Quasimetric architectures provide a realistic approach to achieve planning invariance and horizon generalization.

In Section 7.8, we discuss further implications of these notions of invariance on self-consistent models for decision-making.

**Limitations and Future Work.** Future work should examine how the properties of planning invariance and horizon generalization are conserved in more complex decision-making environments, such as robotic manipulation and language-based agents. Which versions of the distance parameterizations in Section 7.3 are most effective at scale should be investigated with larger-scale empirical experiments. In this paper, we assume a goal-conditioned setting, but there are many alternative forms of task specification (rewards, language, preferences, etc.) that could similarly benefit from generalization over long horizons. Future work should explore how planning-invariant geometry could be extended or mapped onto these task spaces.

# III

REPRESENTATIONS FOR TRACTABLE  
INTRINSIC MOTIVATION

## 8

EMPOWERMENT VIA SUCCESSOR  
REPRESENTATIONS

AI agents deployed in the real world should be helpful to humans. When we know the utility function of the humans an agent could interact with, we can directly train assistive agents through reinforcement learning with the known human objective as the agent’s reward. In practice, agents rarely have direct access to a scalar reward corresponding to human preferences (if such a consistent model even exists) [315], and must infer them from human behavior [316, 317]. This inference can be challenging, as humans may act suboptimally with respect to their stated goals, not know their goals, or have changing preferences [318]. Optimizing a misspecified reward function can have poor consequences [319].

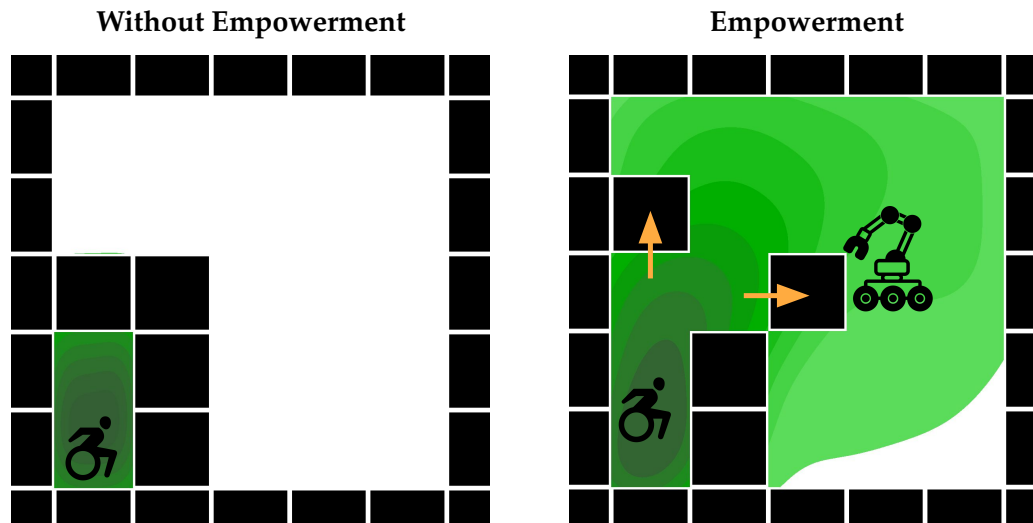


Figure 8.1: We propose an algorithm training assistive agents to empower human users—the assistant should take actions that enable human users to visit a wide range of future states, and the human’s actions should exert a high degree of influence over the future outcomes. Our algorithm scales to high-dimensional settings, opening the door to building assistive agents that need not directly reason about human intentions.

An alternative paradigm for assistance is to train agents that are *intrinsically* motivated to assist humans, rather than directly optimizing a model of their preferences. An analogy can be drawn to a parent raising a child. A good parent will empower the child to make impactful decisions and flourish, rather than proscribing an “optimal” outcome for the child. Likewise, AI agents might seek to *empower* the human agents they interact with, maximizing their capacity to change the environment [4]. In practice, concrete notions of empowerment can be difficult to optimize as an objective, requiring extensive modeling assumptions that don’t scale well to the high-dimensional settings deep reinforcement learning agents are deployed in.

What is a good intrinsic objective for assisting humans that doesn’t require these assumptions? We propose a notion of assistance based on maximizing the influence of the human’s actions on the environment. This approach only requires one structural assumption: the AI agent is interacting with an environment where there is a notion of actions taken by the human agent — a more general setting than the case where we model the human actions as the outcome of some optimization procedure, as in inverse RL [320, 321] or preference-based RL [322].

Prior work has studied many effective objectives for empowerment. For instance, Du et al. [4] approximates human empowerment as the variance in the final states of random rollouts. Despite excellent results in certain settings, this approach can be challenging to scale to higher dimensional settings, and does not necessarily enable human users to achieve the goals they want to achieve. By contrast, our approach exclusively empowers the human with respect to the distribution of (useful) behaviors induced by their current policy, and can be implemented through a simple objective derived from contrastive successor features, which can then be optimized with scalable deep reinforcement learning (Fig. 8.1). We provide a theoretical framework connecting our objective to prior work on empowerment and goal inference, and empirically show that agents trained with this objective can assist humans in the Overcooked environment [6] as well as the obstacle gridworld assistance benchmark proposed by Du et al. [4].

Our core contribution is a novel objective for training agents that are intrinsically motivated to assist humans without requiring a model of the human’s reward function. Our objective, Empowerment via Successor Representations (ESR), maximizes the influence of the human’s actions on the environment, and, unlike past approaches for assistance without reward inference, is based on a scalable model-free objective that can be derived from learned successor features that encode which states the human is likely to want to reach given their current action. Our objective empowers the human to reach the desired states, not all states, without assuming a human model. We analyze this objective in terms of empowerment and goal inference, drawing novel mathematical connections between time-series representations, decision-making, and assistance. We empirically show that agents trained with our objective can assist humans in two benchmarks proposed by past work: the

Overcooked environment [6] and an obstacle-avoidance gridworld [4].

## 8.1 THE INFORMATION GEOMETRY OF EMPOWERMENT

We will first state a general notion of an assistive setting, then show how an empowerment objective based on learned successor representations can be used to assist humans without making assumptions about the human following an underlying reward function. In Section 8.3, we provide empirical evidence supporting these claims.

### Preliminaries

Formally, we adapt the notation of Hadfield-Menell et al. [316], and assume a “robot” (**R**) and “human” (**H**) policy are training together in an MDP  $M = (\mathcal{S}, \mathcal{A}_H, \mathcal{A}_R, R, P, \gamma)$ . The states  $s$  consist of the joint states of the robot and the human; we do not have separate observations for the human and robot. At any state  $s \in \mathcal{S}$ , the robot policy selects actions distributed according to  $\pi_R(a^R | s)$  for  $a^R \in \mathcal{A}_R$  and the human selects actions from  $\pi_H(a^H | s)$  for  $a^H \in \mathcal{A}_H$ . The transition dynamics are defined by a distribution  $P(s' | s, a^H, a^R)$  over the next state  $s' \in \mathcal{S}$  given the current state  $s \in \mathcal{S}$  and actions  $a^H \in \mathcal{A}_H$  and  $a^R \in \mathcal{A}_R$ , as well as an initial state distribution  $P(s_0)$ . For notational convenience, we will additionally define random variables  $s_t$  to represent the state at time  $t$ , and  $a_t^R \sim \pi_R(\cdot | s_t)$  and  $a_t^H \sim \pi_H(\cdot | s_t)$  to represent the human and robot actions at time  $t$ , respectively.

**Empowerment.** Our work builds on a long line of prior methods that use information theoretic objectives for RL. Specifically, we adopt *empowerment* as an objective for training an assistive agent [4, 323, 324]. This section provides the mathematical foundations for empowerment, as developed in prior work. Our work will build on the prior work by (1) providing an information geometric interpretation of what empowerment does (Section 8.1) and (2) providing a scalable algorithm for estimating and optimizing empowerment, going well beyond the gridworlds studied in prior work.

The idea behind empowerment is to think about the changes that an agent can effect on a world; an agent is more empowered if it can effect a larger degree of change over future outcomes. Following prior work [211, 324, 323], we measure empowerment by looking at how much the actions taken *now* affect outcomes *in the future*. An agent with a high degree of empowerment exerts a high degree of control of the future states by simply changing the actions taken now. Like prior work, we measure this degree of control through the mutual information  $I(s^+; a^H)$  between the current action  $a^H$  and the future states  $s^+$ . Note that these future states might occur many time steps into the future.

Empowerment depends on several factors: the environment dynamics, the choice of future actions, the current state, and other agents in the environment.



Different problem settings involve maximizing empowerment using these different factors. In this work, we study the setting where a “human” agent and a “robot” agent collaborate in an environment; the robot will aim to maximize the empowerment of the human. This problem setting was introduced in prior work [4]. Compared with other mathematical frameworks for learning assistive agents [325], framing the problem in terms of empowerment means that the assistive agent need not infer the human’s underlying intention, an inference problem that is typically challenging [326, 327].

We now define our objective. To do this, we introduce random variable  $\mathfrak{s}^+$ , which corresponds to a state sampled  $K \sim \text{Geom}(1 - \gamma)$  steps into the future under the behavior policies  $\pi_{\mathbf{H}}$  and  $\pi_{\mathbf{R}}$ . We will use  $\rho(\mathfrak{s}^+ | s_t)$  to denote the density of this random variable; this density is sometimes referred to as the discounted state occupancy measure. We will use mutual information to measure how much the action  $a_t$  at time  $t$  changes this distribution:

$$I(a_t^{\mathbf{H}}; \mathfrak{s}^+ | s_t) \triangleq \mathbb{E}_{s_t, s_{t+K}, a_t^{\mathbf{H}}, a_t^{\mathbf{R}}} \left[ \log \frac{\mathbb{P}(\mathfrak{s}_{t+K} = s_{t+K} | \mathfrak{s}_t = s_t, a_t^{\mathbf{H}} = a_t)}{\mathbb{P}(\mathfrak{s}_{t+K} = s_{t+K} | \mathfrak{s}_t = s_t)} \right]. \quad (8.1)$$

Our overall objective is *empowerment*,  $\mathcal{E}(\pi_{\mathbf{H}}, \pi_{\mathbf{R}})$ : the mutual information between the human’s actions and the future states  $\mathfrak{s}^+$  while interacting with the robot:

$$\mathcal{E}(\pi_{\mathbf{H}}, \pi_{\mathbf{R}}) = \mathbb{E} \left[ \sum_{t=0}^{\infty} \gamma^t I(a_t^{\mathbf{H}}; \mathfrak{s}^+ | s_t) \right]. \quad (8.2)$$

Note that this objective resembles an RL objective: we do not just want to maximize this objective greedily at each time step, but rather want the assistive agents to take actions now that help the human agent reach states where it will have high empowerment in the future.

### Intuition and Geometry of Empowerment

Intuitively, the assistive agent should aim to maximize the number of future outcomes. We will mathematically quantify this in terms of the discounted state occupancy measure,  $\rho^\pi(\mathfrak{s}^+ | s)$ . Intuitively, an agent has a large empowerment if the future states for one action are very different from the future actions after taking a different action; i.e., when  $\rho(a_t = a_1; \mathfrak{s}^+ | s_t)$  is quite different from  $\rho(a_t = a_2; \mathfrak{s}^+ | s_t)$  for actions  $a_1 \neq a_2$ . The mutual information (Eq. (8.1)) quantifies this degree of control:  $I(a_t; \mathfrak{s}^+ | s_t)$ .

One way of understanding this mutual information is through *information geometry* [328, 329, 329, 330]. For a fixed current state  $s_t$ , assistant policy  $\pi_{\mathbf{R}}$  and human policy  $\pi_{\mathbf{H}}$ , each potential action  $a_t$  that the human takes induces a different distribution over future states:  $\rho^{\pi_{\mathbf{R}}, \pi_{\mathbf{H}}}(\mathfrak{s}^+ | s_t, a_t)$ . We can think about the set of these possible distributions:  $\{\rho^{\pi_{\mathbf{R}}, \pi_{\mathbf{H}}}(\mathfrak{s}^+ | s_t, a_t) \mid a_t \in \mathcal{A}\}$ . Figure 8.2 (Left) visualizes

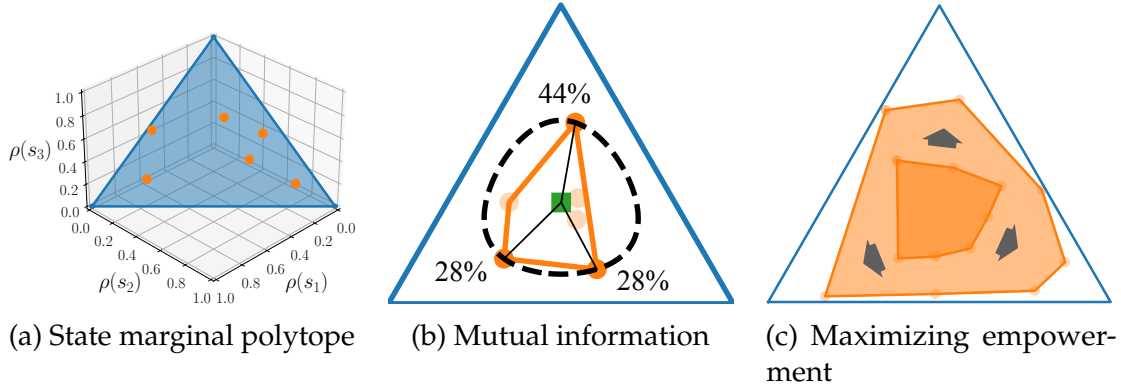


Figure 8.2: **The Information Geometry of Empowerment**, illustrating the analysis in Section 8.1. (Left) For a given state  $s_t$  and assistant policy  $\pi_{\mathbf{R}}$ , we plot the distribution over future states for 6 choices of the human policy  $\pi_{\mathbf{H}}$ . In a 3-state MDP, we can represent each policy as a vector lying on the 2-dimensional probability simplex. We refer to the set of all possible state distributions as the *state marginal polytope*. (Center) Mutual information corresponds to the distance between the center of the polytope and the vertices that are maximally far away. (Right) Empowerment corresponds to maximizing the size of this polytope. For example, when an assistive agent moves an obstacle out of a human user’s way, the human user can spend more time at desired state.

this distribution on a probability simplex for 6 choices of action  $a_t$ . If we look at any possible distribution over actions, then this set of possible future distributions becomes a polytope (see orange polygon in Fig. 8.2 (Center)).

Intuitively, the mutual information  $I(a_t; \mathbf{s}^+ | s_t)$  used to define our empowerment objective corresponds to the *size* or *volume* of this state marginal polytope. This intuition can be formalized by using results from information geometry [331–333]. The human policy  $\pi_{\mathbf{H}}(a_t | s_t)$  places probability mass on the different points in Figure 8.2 (Center). Maximizing the mutual information corresponds to “picking out” the state distributions that are maximally spread apart (see probabilities in Fig. 8.2 (Center)). To make this formal, define

$$\rho(\mathbf{s}^+ | s_t) \triangleq \mathbb{E}_{\pi(a_t | s_t)}[\rho(\mathbf{s}^+ | s_t, a_t)] \quad (8.3)$$

as the *average* state distribution from taking the human’s actions (see green square in Fig. 8.2 (Center)).

**Remark 8.1.** *Mutual information corresponds to the distance between the average state distribution (Eq. 8.3) and the furthest achievable state distributions:*

$$I(a_t; \mathbf{s}^+ | s_t) = \max_{a_t} D_{KL}(\rho(a_t; \mathbf{s}^+ | s_t) \| \rho(\mathbf{s}^+ | s_t)) \triangleq d_{max}. \quad (8.4)$$

This distance is visualized as the black lines in Fig. 8.2. When we talk about the “size” of the state marginal polytope, we are specifically referring to the length of these black lines (as measured with a KL divergence).

This sort of mutual information is a way for measuring the degree of control that an agent exerts on an environment. This measure is well defined for any agent/policy; that agent need not be maximizing mutual information, and could instead be maximizing some arbitrary reward function. This point is important in our setting: this means that the assistive agent can estimate and maximize the empowerment of the human user *without having to infer what reward function the human is trying to maximize*.

Finally, we come back to our empowerment objective, which is a discounted sum of the mutual information terms that we have been analyzing above. This empowerment objective says that the human is more empowered when this set has a larger size — i.e., the human can visit a wider range of future state (distributions). The empowerment objective says that the assistive agent should act to try to maximize the size of this polytope. Importantly, this maximization problem is done sequentially: the assistive agent wants the size of this polytope to be large both at the current state and at future states; the human’s actions should exert a high degree of influence over the future outcomes both now and in the future. Thus, our overall objective looks at a sum of these mutual informations.

Not only does this analysis provides a geometric picture for what empowerment is doing, it also lays the groundwork for formally relating empowerment to reward.

### Relating Empowerment to Reward

In this section we take aim at the question: when humans are well-modeled as optimizing a reward function, when does maximizing empowerment help humans maximize their rewards? Answering this question is important because for empowerment to be a safe and effective assistive objective, it should enable the human to better achieve their goals. We show that under certain assumptions, empowerment yields a provable lower bound on the average-case reward achieved by the human for sufficiently long-horizon empowerment (i.e.,  $\gamma \rightarrow 1$ ).

For constructing the formal bound, we suppose the human is Boltzmann-rational [334, 290] with respect to some reward function  $R \sim \mathcal{R}$ , where  $\mathcal{R}$  is some distribution that could be interpreted as a prior over the human’s objective, a set of skills the human may try and carry out, or a population of humans with different objectives that the agent could be interacting with. Our quantity of interest, the average-case reward achieved by the human with our empowerment objective, is given by

$$\mathcal{J}_{\pi_{\mathbf{R}}}^{\gamma}(\pi_{\mathbf{H}}) = \mathbb{E}_{R \sim \mathcal{R}} [V_{R, \gamma}^{\pi_{\mathbf{H}}, \pi_{\mathbf{R}}}(s_0)] \quad (8.5)$$

where  $V_{R, \gamma}^{\pi_{\mathbf{H}}, \pi_{\mathbf{R}}}(s_0)$  is the value function of the human policy  $\pi_{\mathbf{H}}$  under the reward

function  $R$  when interacting with  $\pi_{\mathbf{R}}$ . Recalling Eq. (8.2), we will express the overall empowerment objective we are trying to relate to Eq. (8.5) as

$$\mathcal{E}_\gamma(\pi_{\mathbf{H}}, \pi_{\mathbf{R}}) = \mathbb{E} \left[ \sum_{t=0}^{\infty} \gamma^t I(\mathfrak{s}^+; \mathfrak{a}_t^{\mathbf{H}} \mid \tilde{\mathfrak{s}}_t) \right]. \quad (8.6)$$

The two key assumptions used in our analysis are Assumption 8.1, which states that the human will optimize for behaviors that uniformly cover the state space, and Assumption 8.2, which simply states that with infinite time, the human will be able to reach any state in the state space.

**Assumption 8.1** (Skill Coverage). *The rewards  $R \sim \mathcal{R}$  are uniformly distributed over the scaled  $|\mathcal{S}|$ -simplex  $\Delta^{|\mathcal{S}|}$  such that:*

$$\left(R + \frac{1}{|\mathcal{S}|}\right) \left(\frac{1}{1-\gamma}\right) \sim \text{Unif}(\Delta^{|\mathcal{S}|}) = \text{Dirichlet}(\underbrace{1, 1, \dots, 1}_{|\mathcal{S}| \text{ times}}). \quad (8.7)$$

**Assumption 8.2** (Ergodicity). *For some  $\pi_{\mathbf{H}}, \pi_{\mathbf{R}}$ , we have*

$$P^{\pi_{\mathbf{H}}, \pi_{\mathbf{R}}}(\mathfrak{s}^+ = s \mid s_0) > 0 \quad \text{for all } s \in \mathcal{S}, \gamma \in (0, 1). \quad (8.8)$$

Our main theoretical result is Theorem 8.2, which shows that under these assumptions, maximizing empowerment yields a lower bound on the (squared) average-case reward achieved by the human for sufficiently large  $\gamma$ . In other words, for a sufficiently long empowerment horizon, the empowerment objective Eq. (8.2) is a meaningful proxy for reward maximization.

**Theorem 8.2.** *Under Assumption 8.1 and Assumption 8.2, for sufficiently large  $\gamma$  and any  $\beta > 0$ ,*

$$\mathcal{E}_\gamma(\pi_{\mathbf{H}}, \pi_{\mathbf{R}})^{1/2} \leq (\beta/e) \mathcal{J}_{\pi_{\mathbf{R}}}^\gamma(\pi_{\mathbf{H}}). \quad (8.9)$$

To the best of our knowledge, this result provides the first formal link between empowerment maximization and reward maximization. This motivates us to develop a scalable algorithm for empowerment maximization, which we introduce in the following section.

## Notation for Analysis

To connect our empowerment objective to reward, we will extend the notation in Section 8.1 to include a distribution over possible tasks the human might be trying to solve,  $\mathcal{R}$ , such that each  $R \sim \mathcal{R}$  defines a distinct reward function  $R : \mathcal{S} \rightarrow \mathbb{R}$ . We assume  $\pi_{\mathbf{R}}$  tries to maximize the  $\gamma$ -discounted empowerment'' of the human, defined as

$$\mathcal{E}_\gamma(\pi_{\mathbf{H}}, \pi_{\mathbf{R}}) = \mathbb{E} \left[ \sum_{t=0}^{\infty} \gamma^t I(\mathfrak{s}_+^\gamma; \mathfrak{a}_t^{\mathbf{H}} \mid \tilde{\mathfrak{s}}_t) \right] \quad (\text{Eq. 8.6})$$

for

$$\mathfrak{s}_+^\gamma \triangleq \{\mathfrak{s}_k \text{ for } k \sim \text{Geom}(1 - \gamma)\}. \quad (8.10)$$

We additionally define  $\bar{\mathfrak{s}}_t$  to be the full history of states up to time  $t$  and  $\bar{\mathfrak{a}}_t^{\mathbf{H}}$  to be the full history of human actions up to time  $t$ ,

$$\begin{aligned} \bar{\mathfrak{s}}_t &= \{\mathfrak{s}_i\}_{i=0}^t, \\ \bar{\mathfrak{a}}_t^{\mathbf{H}} &= \{\mathfrak{a}_i^{\mathbf{H}}\}_{i=0}^t. \end{aligned} \quad (8.11)$$

Then,  $\tilde{\mathfrak{s}}_t$  is the full history of states and past human actions up to time  $t$ ,

$$\tilde{\mathfrak{s}}_t = \bar{\mathfrak{s}}_t \cup \bar{\mathfrak{a}}_{t-1}^{\mathbf{H}}. \quad (8.12)$$

Note that the definition of empowerment in Eq. (8.6) differs slightly from the original construction Eq. (8.2) — we condition on the full history of human actions, not just the most recent one. This distinction becomes irrelevant in practice if our MDP maintains history in the state, in which case we can equivalently use  $\mathfrak{s}_t$  in place of  $\tilde{\mathfrak{s}}_t$ .

Meanwhile, for any fixed  $\pi_{\mathbf{R}}$  and  $\beta > 0$ , the human is Boltzmann-rational with respect to the robot’s policy:

$$\pi_{\mathbf{H}}(a_t^{\mathbf{H}} \mid \tilde{\mathfrak{s}}_t) \propto \exp(\beta Q_{R,\gamma}^{\pi_{\mathbf{H}},\pi_{\mathbf{R}}}(s_t, a_t^{\mathbf{H}})) \quad (8.13)$$

$$\text{where } Q_{R,\gamma}^{\pi_{\mathbf{H}},\pi_{\mathbf{R}}}(s_t, a_t^{\mathbf{H}}) = \mathbb{E} \left[ \sum_{k=0}^{\infty} \gamma^k R(s_{t+k}) \mid s_t, a_t^{\mathbf{H}} \right]. \quad (8.14)$$

Equivalently, we can define the human’s (soft) Q-function and value as

$$\begin{aligned} Q_{R,\gamma}^{\pi_{\mathbf{H}},\pi_{\mathbf{R}}}(s_t, a_t^{\mathbf{H}}) &= R(s_t) + \gamma \mathbb{E} \left[ V_{R,\gamma}^{\pi_{\mathbf{H}},\pi_{\mathbf{R}}}(s_{t+1}) \mid s_t, a_t^{\mathbf{H}} \right] \\ \text{for } V_{R,\gamma}^{\pi_{\mathbf{H}},\pi_{\mathbf{R}}}(s_t) &= \mathbb{E} \left[ R(s_t) + \gamma R(s_{t+1}) + \gamma^2 R(s_{t+2}) + \dots \mid s_t, a_t^{\mathbf{H}} \right]. \end{aligned} \quad (8.15)$$

The overall human objective is to maximize the expected soft value:

$$\mathcal{J}_{\pi_{\mathbf{R}}}^\gamma(\pi_{\mathbf{H}}) = \mathbb{E}_{\substack{R \sim \mathcal{R} \\ s_0 \sim p_0}} \left[ V_{R,\gamma}^{\pi_{\mathbf{H}},\pi_{\mathbf{R}}}(s_0) \right]. \quad (\text{Eq. 8.5})$$

Note that this definition of  $\pi_{\mathbf{H}}$  depends on  $R$  and  $\pi_{\mathbf{R}}$  and is bounded  $0 \leq \mathcal{J}_{\pi_{\mathbf{R}}}^\gamma(\pi_{\mathbf{H}}) \leq 1$ . As in the CIRL setting [316], we assume robot is unable to access the true human reward  $R : \mathcal{S} \rightarrow \mathbb{R}$ . One way to think of the robot’s task is as finding a Nash equilibrium between the objectives Eq. (8.6) and the human best response in Eq. (8.13).

For convenience, we will also define a multistep version of  $Q_{R,\gamma}^{\pi_{\mathbf{H}},\pi_{\mathbf{R}}}$ ,

$$Q_{R,\gamma}^{\pi_{\mathbf{H}},\pi_{\mathbf{R}}}(s_t, a_t^{\mathbf{H}}, \dots, a_{t+K}^{\mathbf{H}}) = \mathbb{E} \left[ \sum_{k=0}^{\infty} \gamma^k R(\mathfrak{s}_{t+k}) \mid s_t, a_t^{\mathbf{H}}, \dots, a_{t+K}^{\mathbf{H}} \right]. \quad (8.16)$$

**Proof of Theorem 8.2**

Our approach will be to first relate the empowerment (influence of  $\mathbf{a}_t^{\mathbf{H}}$  on  $\mathfrak{s}_+^\gamma$ ) to the mutual information between  $\mathbf{a}_t^{\mathbf{H}}$  and the reward  $R$ .

Then, we will connect this quantity to a notion of “advantage” for the human (Eq. 8.23), which in turn can be related to the expected reward under the human’s policy. In its simplest form, this argument will require an assumption over the reward distribution:

**Assumption 8.1** (Skill Coverage). *The rewards  $R \sim \mathcal{R}$  are uniformly distributed over the scaled  $|\mathcal{S}|$ -simplex  $\Delta^{|\mathcal{S}|}$  such that:*

$$\left(R + \frac{1}{|\mathcal{S}|}\right) \left(\frac{1}{1-\gamma}\right) \sim \text{Unif}(\Delta^{|\mathcal{S}|}) = \text{Dirichlet}(\underbrace{1, 1, \dots, 1}_{|\mathcal{S}| \text{ times}}). \quad (8.7)$$

In other words, Assumption 8.1 says our prior over the human’s reward function is uniform with zero mean. This is not the only prior for which this argument works, but for general  $\mathcal{R}$  we will need a correction term to incentivize states that are more likely across the distribution of  $\mathcal{R}$ . Another way to view Assumption 8.1 is that the human is trying to execute diverse “skills”  $z \sim \text{Unif}(\Delta^{|\mathcal{S}|})$ .

We also assume ergodicity (Assumption 8.2). In the special case of an MDP that resets to some distribution with full support over  $\mathcal{S}$ , this assumption is automatically satisfied.

**Assumption 8.2** (Ergodicity). *For some  $\pi_{\mathbf{H}}, \pi_{\mathbf{R}}$ , we have*

$$P^{\pi_{\mathbf{H}}, \pi_{\mathbf{R}}}(\mathfrak{s}_+^\gamma = s \mid s_0) > 0 \quad \text{for all } s \in \mathcal{S}, \gamma \in (0, 1). \quad (8.8)$$

Our main result connects empowerment directly to a (lower bound on) the human’s expected reward.

**Theorem 8.2.** *Under Assumption 8.1 and Assumption 8.2, for sufficiently large  $\gamma$  and any  $\beta > 0$ ,*

$$\mathcal{E}_\gamma(\pi_{\mathbf{H}}, \pi_{\mathbf{R}})^{1/2} \leq (\beta/e) \mathcal{J}_{\pi_{\mathbf{R}}}^\gamma(\pi_{\mathbf{H}}). \quad (8.9)$$

Theorem 8.2 says that for a long enough horizon (i.e.,  $\gamma$  close to 1), the robot’s empowerment objective will lower bound the (squared, MaxEnt) human objective.

We make use of the following lemmas in the proof.

**Lemma 8.3.** *For  $t \sim \text{Geom}(1 - \gamma)$  and any  $K \geq 0$ ,*

$$\liminf_{\gamma \rightarrow 1} I(\mathfrak{s}_+^\gamma; \mathbf{a}_t^{\mathbf{H}}, \dots, \mathbf{a}_{t+K}^{\mathbf{H}} \mid \tilde{\mathfrak{s}}_t) \leq I(R; \mathbf{a}_t^{\mathbf{H}}, \dots, \mathbf{a}_{t+K}^{\mathbf{H}} \mid \tilde{\mathfrak{s}}_t). \quad (8.17)$$

*Proof.* For sufficiently large  $\gamma$ ,  $\mathfrak{s}_+^\gamma$  will approach the stationary distribution of  $P^{\pi_H, \pi_R}$  for a fixed  $\pi_H, \pi_R$ , irrespective of  $\tilde{\mathfrak{s}}_t$  and  $\mathfrak{a}_t^H, \dots, \mathfrak{a}_{t+K}^H$  from Assumption 8.2. So,

$$\liminf_{\gamma \rightarrow 1} I(\mathfrak{s}_+^\gamma; \mathfrak{a}_t^H, \dots, \mathfrak{a}_{t+K}^H \mid \tilde{\mathfrak{s}}_t) \leq I\left(\lim_{\gamma \rightarrow \infty} \mathfrak{s}_+^\gamma; \mathfrak{a}_t^H, \dots, \mathfrak{a}_{t+K}^H \mid \tilde{\mathfrak{s}}_t\right) \quad (8.18)$$

Since each  $R, \pi_R, \gamma$  defines a human policy  $\pi_H$  via Eq. (8.13), we can express the dependencies as the following Markov chain:

$$\hat{\mathfrak{a}}_t \longrightarrow R \longrightarrow \lim_{\gamma \rightarrow 1} \mathfrak{s}_+^\gamma. \quad (8.19)$$

Applying the data processing inequality [328], we get

$$I\left(\lim_{\gamma \rightarrow \infty} \mathfrak{s}_+^\gamma; \mathfrak{a}_t^H, \dots, \mathfrak{a}_{t+K}^H \mid \tilde{\mathfrak{s}}_t\right) \leq I(R; \mathfrak{a}_t^H, \dots, \mathfrak{a}_{t+K}^H \mid \tilde{\mathfrak{s}}_t), \quad (8.20)$$

from which Eq. (8.17) follows.  $\square$

**Lemma 8.4.** *Suppose we have  $k$  logits, denoted by the map  $\alpha : \{1 \dots k\} \rightarrow [0, 1]$ . For any  $\beta > 0$ , we can construct the (softmax) distribution*

$$p_\beta(i) \propto \exp(\beta \alpha(i)).$$

Then,

$$\mathcal{H}(p_\beta) \geq \log k - \left(\frac{\beta}{e}\right)^2. \quad (8.21)$$

*Proof.* We lower bound the “worst-case” of the RHS,  $\alpha = (1, 0, \dots, 0)$ :

$$\begin{aligned} \mathcal{H}(p_\beta) &= \frac{(1-n) \log\left(\frac{1}{k+e^\beta-1}\right)}{k+e^\beta-1} - \frac{e^\beta \log\left(\frac{e^\beta}{k+e^\beta-1}\right)}{k+e^\beta-1} \\ &= \frac{(k+e^\beta-1) \log(k+e^\beta-1) - e^\beta \log(e^\beta)}{k+e^\beta-1} \\ &= \log(k+e^\beta-1) - \frac{e^\beta \log(e^\beta)}{k+e^\beta-1} \\ &\geq \log k - (\beta/e)^2. \end{aligned} \quad (8.22)$$

$\square$

**Lemma 8.5.** *For any  $t$  and  $K \geq 0$ ,*

$$I(R; \mathfrak{a}_t^H, \dots, \mathfrak{a}_{t+K}^H \mid \tilde{\mathfrak{s}}_t) \leq \lim_{\gamma \rightarrow 1} \left( \frac{\beta}{e} \mathbb{E} \left[ Q_{R, \gamma}^{\pi_H, \pi_R}(s_t, \mathfrak{a}_t^H, \dots, \mathfrak{a}_{t+K}^H) \right] \right)^2. \quad (8.23)$$



*Proof.* Denote by  $\hat{\mathbf{a}}_t^{\mathbf{H}} \dots \hat{\mathbf{a}}_{t+K}^{\mathbf{H}} \sim \text{Unif}(\mathcal{A}^{\mathbf{H}})$  a sequence of  $K$  random actions. From Lemma 8.4:

$$\begin{aligned} I(R; \mathbf{a}_t^{\mathbf{H}}, \dots, \mathbf{a}_{t+K}^{\mathbf{H}} \mid \tilde{\mathbf{s}}_t) &= \mathcal{H}(\mathbf{a}_t^{\mathbf{H}}, \dots, \mathbf{a}_{t+K}^{\mathbf{H}} \mid \tilde{\mathbf{s}}_t) - \mathcal{H}(\mathbf{a}_t^{\mathbf{H}}, \dots, \mathbf{a}_{t+K}^{\mathbf{H}} \mid R, \tilde{\mathbf{s}}_t) \\ &\leq \log(K|\mathcal{A}|) - \mathcal{H}(\mathbf{a}_t^{\mathbf{H}}, \dots, \mathbf{a}_{t+K}^{\mathbf{H}} \mid R, \tilde{\mathbf{s}}_t) \\ &\leq \lim_{\gamma \rightarrow 1} \left( \frac{\beta}{e} \mathbb{E} [Q_{R,\gamma}^{\pi_{\mathbf{H}}, \pi_{\mathbf{R}}}(s_t, \mathbf{a}_t^{\mathbf{H}}, \dots, \mathbf{a}_{t+K}^{\mathbf{H}}) - Q_{R,\gamma}^{\pi_{\mathbf{H}}, \pi_{\mathbf{R}}}(s_t, \hat{\mathbf{a}}_t^{\mathbf{H}}, \dots, \hat{\mathbf{a}}_{t+K}^{\mathbf{H}})] \right)^2, \end{aligned} \quad (8.24)$$

where the last inequality follows from Lemma 8.4 and  $Q_{R,\gamma}^{\pi_{\mathbf{H}}, \pi_{\mathbf{R}}}(\dots) \leq 1$ . We also have

$$0 \leq Q_{R,\gamma}^{\pi_{\mathbf{H}}, \pi_{\mathbf{R}}}(s_t, \hat{\mathbf{a}}_t^{\mathbf{H}}, \dots, \hat{\mathbf{a}}_{t+K}^{\mathbf{H}}) \leq Q_{R,\gamma}^{\pi_{\mathbf{H}}, \pi_{\mathbf{R}}}(s_t, \mathbf{a}_t^{\mathbf{H}}, \dots, \mathbf{a}_{t+K}^{\mathbf{H}}) \leq 1, \quad (8.25)$$

which lets us conclude from Eq. (8.24) that

$$I(R; \mathbf{a}_t^{\mathbf{H}}, \dots, \mathbf{a}_{t+K}^{\mathbf{H}} \mid \tilde{\mathbf{s}}_t) \leq \left( \frac{\beta}{e} \mathbb{E} [Q_{R,\gamma}^{\pi_{\mathbf{H}}, \pi_{\mathbf{R}}}(s_t, \mathbf{a}_t^{\mathbf{H}}, \dots, \mathbf{a}_{t+K}^{\mathbf{H}})] \right)^2. \quad (\text{Eq. 8.23})$$

□

We can now prove Theorem 8.2 directly by combining Lemmas 8.3 and 8.5.

*Proof of Theorem 8.2.* Simplifying the limit in Eq. (8.9), we get

$$\begin{aligned} \liminf_{\gamma \rightarrow 1} \mathcal{E}_{\gamma}(\pi_{\mathbf{H}}, \pi_{\mathbf{R}}) &\leq \liminf_{\gamma \rightarrow 1} \left( \sum_{t=0}^{\infty} \gamma^t I(\tilde{\mathbf{s}}_t^{\gamma}; \mathbf{a}_t^{\mathbf{H}} \mid \tilde{\mathbf{s}}_t) \right) \\ &\leq \liminf_{\gamma \rightarrow 1} I(\tilde{\mathbf{s}}_t^{\gamma}; \mathbf{a}_t^{\mathbf{H}}, \dots, \mathbf{a}_{t+K}^{\mathbf{H}} \mid \tilde{\mathbf{s}}_t) \quad (\text{chain rule}) \\ &\leq I(R; \mathbf{a}_t^{\mathbf{H}}, \dots, \mathbf{a}_{t+K}^{\mathbf{H}} \mid \tilde{\mathbf{s}}_t) \quad (\text{Lemma 8.3}) \\ &\leq \lim_{\gamma \rightarrow 1} \left( \frac{\beta}{e} \mathbb{E} [Q_{R,\gamma}^{\pi_{\mathbf{H}}, \pi_{\mathbf{R}}}(s_t, \mathbf{a}_t^{\mathbf{H}}, \dots, \mathbf{a}_{t+K}^{\mathbf{H}})] \right)^2 \quad (\text{Lemma 8.5}) \\ &\leq \lim_{\gamma \rightarrow 1} \left( \frac{\beta \mathcal{J}_{\pi_{\mathbf{R}}}^{\gamma}(\pi_{\mathbf{H}})}{e} \right)^2. \end{aligned} \quad (8.26)$$

It follows that for sufficiently large  $\gamma$ ,

$$\mathcal{E}_{\gamma}(\pi_{\mathbf{H}}, \pi_{\mathbf{R}})^{1/2} \leq (\beta/e) \mathcal{J}_{\pi_{\mathbf{R}}}^{\gamma}(\pi_{\mathbf{H}}). \quad (\text{Eq. 8.9})$$

□

## 8.2 MAXIMIZING EMPOWERMENT WITH CONTRASTIVE REPRESENTATIONS

Directly computing Eq. (8.2) would require access to the human policy, which we don't have. Therefore, we want a tractable estimation that still performs well in



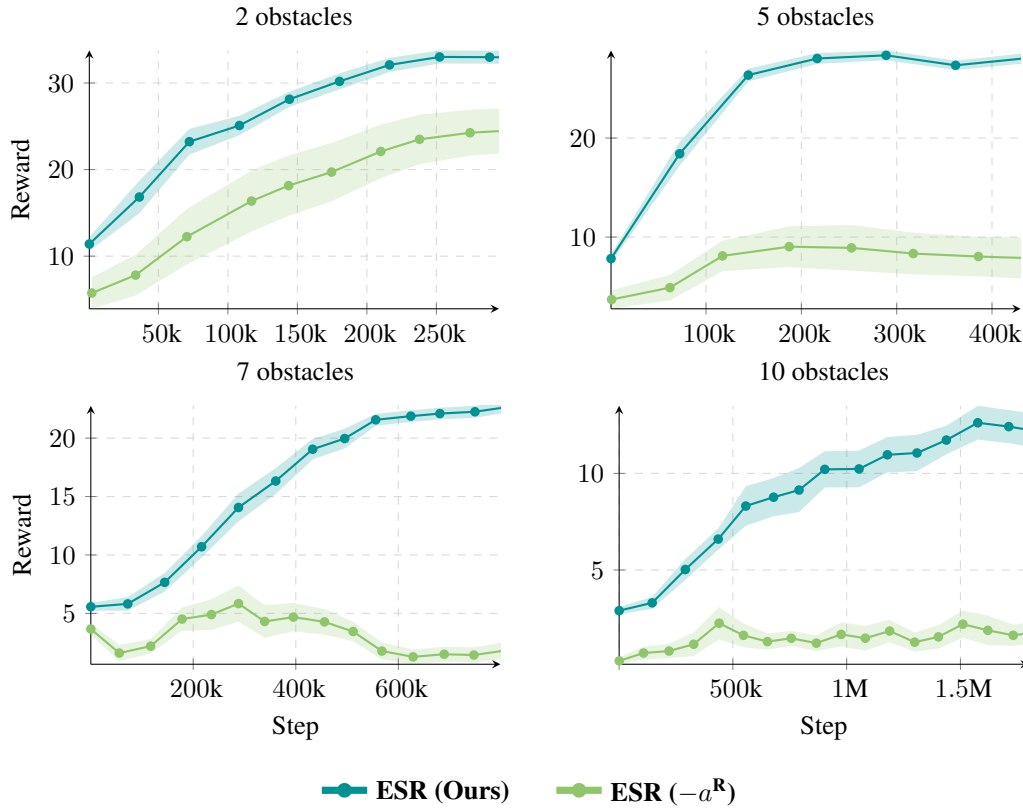


Figure 8.3: We evaluate our method with and without conditioning on the robot action  $a^R$ . Conditioning aids learning significantly, which we theorize is because it removes uncertainty in the classification.

large environments which are more difficult to model due to the exponentially increasing set of possible future states. To better-estimate empowerment, we learn contrastive representations that encode information about which future states are likely to be reached from the current state. These contrastive representations learn to model mutual information between the current state, action, and future state, which we then use to compute the empowerment objective.

### Estimating Empowerment

To estimate this empowerment objective, we need a way of learning the probability ratio inside the expectation. Prior methods such as Du et al. [4] and Salge et al. [323] rollout possible future states and compute a measure of their variance as a proxy for empowerment, however this doesn't scale when the environment becomes complex. Other methods learn a dynamics model, which also doesn't scale when dynamics become challenging to model [335]. Modeling these probabilities directly is challenging in settings with high-dimensional states, so we opt for an indirect

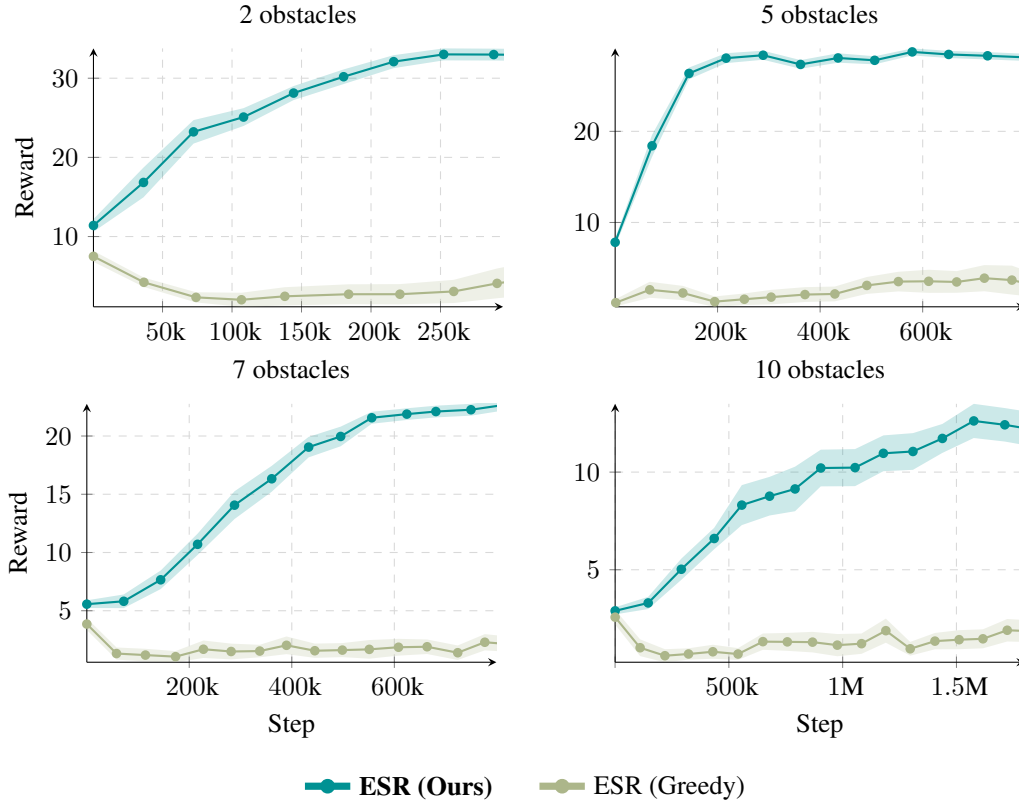


Figure 8.4: We compare a greedy policy ( $\gamma = 0$ ) against our standard policy ( $\gamma = 0.9$ ).

approach. Specifically, we will learn representations that encode two probability ratios. Then, we will be able to compute the desired probability ratio by combining these other probability ratios.

Our method learns three representations:

1.  $\phi(s, a^R, a^H)$  — This representation can be understood as a sort of latent-space model, predicting the future representation given the current state  $s$  and the human’s current action  $a^H$  as well as the robot’s current action  $a^R$ .
2.  $\phi'(s, a^R)$  — This representation can be understood as an uncontrolled model, predicting the representation of a future state without reference to the current human action  $a^H$ . This representation is analogous to a value function.
3.  $\psi(s^+)$  — This is a representation of a future state.

We will learn these three representations with two contrastive losses, one that aligns  $\phi(s, a^R, a^H) \leftrightarrow \psi(s^+)$  and one that aligns  $\phi'(s, a^R) \leftrightarrow \psi(s^+)$

$$\max_{\phi, \phi', \psi} \mathbb{E}_{\{(s_i, a_i, s'_i) \sim P(s_t, a_t^H, s_{t+k})\}_{i=1}^N} [\mathcal{L}_c(\{\phi(s_i, a_i)\}, \{\psi(s'_i)\}) + \mathcal{L}_c(\{\phi'(s_i)\}, \{\psi(s'_i)\})],$$

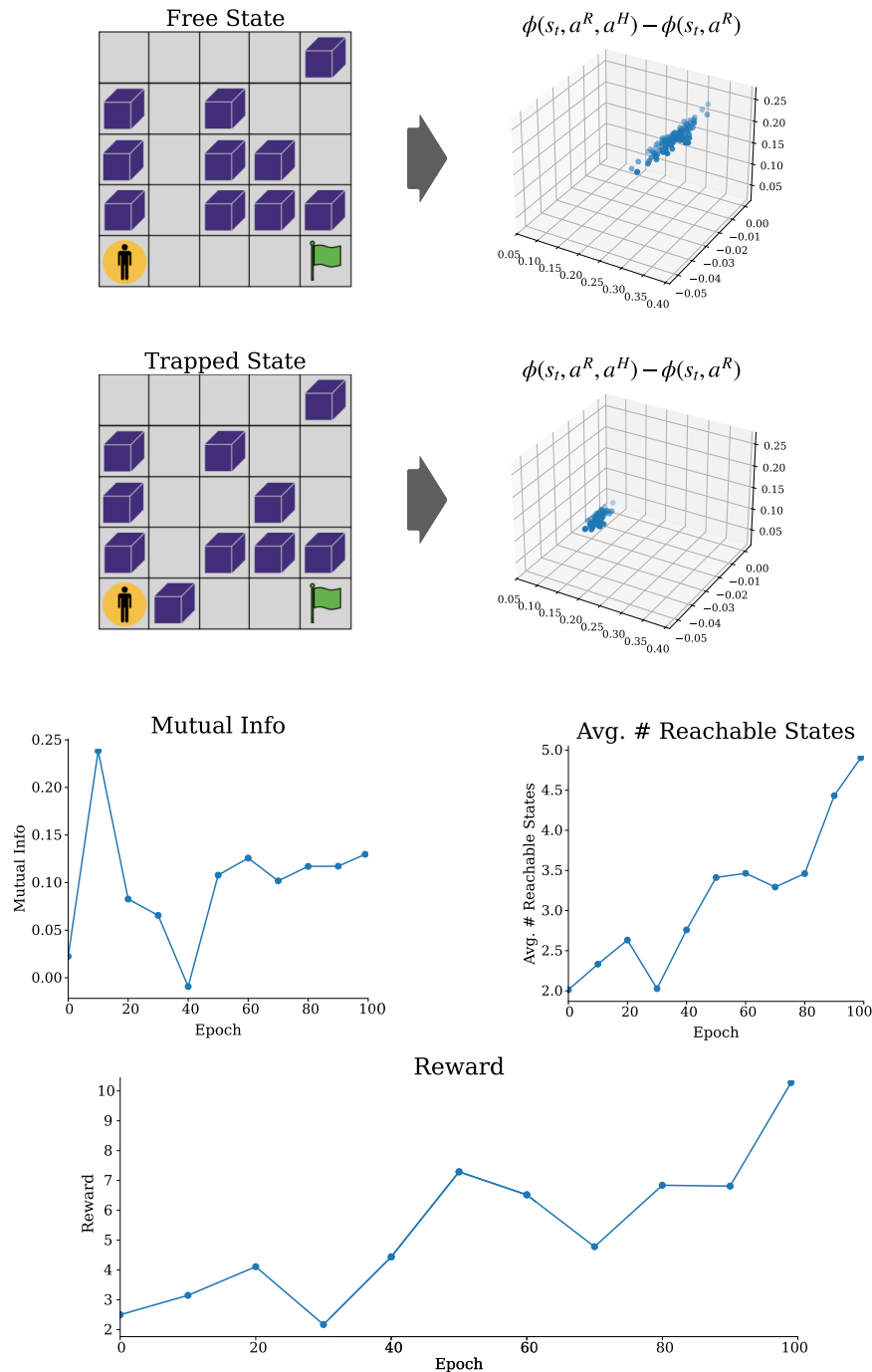


Figure 8.5: Visualizing training empowerment in a 5x5 Gridworld with 10 obstacles. Our empowerment objective maximizes the influence of the human’s actions on the future state, preferring the state where the human can reach the goal to the trapped state. This corresponds to maximizing the volume of the state marginal polytope, which is proportional to the number of states that the human can reach from their current position. To visualize the representations, we set the latent dimension to 3 instead of 100.

where the contrastive loss  $\mathcal{L}_c$  is the symmetrized infoNCE objective [152]:

$$\mathcal{L}_c(\{x_i\}, \{y_j\}) \triangleq \sum_{i=1}^N \left[ \log \left( \frac{e^{x_i^T y_i}}{\sum_{j=1}^N e^{x_i^T y_j}} \right) + \log \left( \frac{e^{x_i^T y_i}}{\sum_{j=1}^N e^{x_j^T y_i}} \right) \right]. \quad (8.27)$$

We have colored the index  $j$  for clarity. At convergence, these representations encode two probability ratios [236], which we will ultimately be able to use to estimate empowerment (Eq. 8.2):

$$\phi(s, a^{\mathbf{R}}, a^{\mathbf{H}})^T \psi(g) = \log \left[ \frac{\mathbb{P}(\mathfrak{s}_{t+K} = g \mid \mathfrak{s}_t = s, \mathfrak{a}_t^{\mathbf{H}} = a^{\mathbf{H}}, \mathfrak{a}_t^{\mathbf{R}} = a^{\mathbf{R}})}{C_1 \mathbb{P}(\mathfrak{s}_{t+K} = g)} \right] \quad (8.28)$$

$$\phi'(s, a^{\mathbf{R}})^T \psi(g) = \log \left[ \frac{\mathbb{P}(\mathfrak{s}_{t+K} = s_{t+k} \mid \mathfrak{s}_t = s_t, \mathfrak{a}_t^{\mathbf{R}} = a^{\mathbf{R}})}{C_2 \mathbb{P}(\mathfrak{s}_{t+K} = g)} \right]. \quad (8.29)$$

Note that our definition of empowerment (Eq. 8.2) is defined in terms of similar probability ratios. The constants  $C_1$  and  $C_2$  will mean that our estimate of empowerment may be off by an additive constant, but that constant will not affect the solution to the empowerment maximization problem.

### Estimating Empowerment with the Learned Representations

To estimate empowerment, we will look at the difference between these two inner products:

$$\phi(s_{t+K}, a^{\mathbf{R}}, a^{\mathbf{H}})^T \psi(g) - \phi(s_{t+K}, a^{\mathbf{R}})^T \psi(g) \quad (8.30)$$

$$= \log \mathbb{P}(s_{t+K} \mid s, a^{\mathbf{H}}) - \log C_1 - \log \mathbb{P}(s_{t+K}) \\ - \log \mathbb{P}(s_{t+K} \mid s) + \log C_2 + \log \mathbb{P}(s_{t+K}) \quad (8.31)$$

$$= \log \frac{\mathbb{P}(s_{t+K} \mid s, a^{\mathbf{H}})}{\mathbb{P}(s_{t+K} \mid s)} + \log \frac{C_2}{C_1}. \quad (8.32)$$

Note that the expected value of the first term is the *conditional* mutual information  $I(s_{t+K}; a^{\mathbf{H}} \mid s)$ . Our empowerment objective corresponds to averaging this mutual information across all the visited states. In other words, our objective corresponds to an RL problem, where empowerment corresponds to the expected discounted sum of these log ratios:

$$\mathcal{E}(\pi_{\mathbf{H}}, \pi_{\mathbf{R}}) = \mathbb{E}_{\pi_{\mathbf{H}}, \pi_{\mathbf{R}}} \left[ \sum_{t=0}^{\infty} \gamma^t I(s_{t+K}; a_t^{\mathbf{H}} \mid s_t) \right] \quad (8.33)$$

$$\approx \mathbb{E}_{\pi_{\mathbf{H}}, \pi_{\mathbf{R}}} \left[ \sum_{t=0}^{\infty} \gamma^t (\phi(s_t, a^{\mathbf{R}}, a^{\mathbf{H}}) - \phi(s_t, a^{\mathbf{R}}))^T \psi(g) - \log \frac{C_2}{C_1} \right]. \quad (8.34)$$

**Algorithm 5:** Empowerment via Successor Representations (ESR)

---

**Input:** Human policy  $\pi_{\mathbf{H}}(a | s)$   
 Randomly initialize assistive agent policy  $\pi_{\mathbf{R}}(a | s)$ , and representations  $\phi(s, a^{\mathbf{R}}, a^{\mathbf{H}})$ ,  $\psi(s, a^T)$ , and  $\psi(g)$ .  
 Initialize replay buffer  $\mathcal{B}$ .  
**while** not converged **do**  
   Collect a trajectory of experience with human policy and assistive agent policy, store in replay buffer  $\mathcal{B}$ .  
   Update representations  $\phi(s, a^{\mathbf{R}}, a^{\mathbf{H}})$ ,  $\psi(s, a^T)$ , and  $\psi(g)$  with the contrastive losses in Eq. (8.27).  
   Update  $\pi_{\mathbf{R}}(a | s)$  with RL using reward function  $r(s, a^{\mathbf{R}}, a^{\mathbf{H}}) = (\phi(s, a^{\mathbf{R}}, a^{\mathbf{H}}) - \phi'(s, a^{\mathbf{R}}))^T \psi(g)$ .  
**Return:** Assistive policy  $\pi_{\mathbf{R}}(a | s)$ .

---

The approximation above comes from function approximation in learning the Bayes optimal representations. Again, note that the constants  $C_1$  and  $C_2$  do not change the optimization problem. Thus, to maximize empowerment we will apply RL to the assistive agent  $\pi_{\mathbf{R}}(a | s)$  using a reward function

$$r(s, a^{\mathbf{R}}) = (\phi(s_t, a^{\mathbf{R}}, a^{\mathbf{H}}) - \phi(s_t, a^{\mathbf{R}}))^T \psi(g). \quad (8.35)$$

### Algorithm Summary

We propose an actor-critic method for learning the assistive agent. Our method will alternate between updating these contrastive representations and using them to estimate a reward function (Eq. (8.35)) that is optimized via RL. We summarize the algorithm in Algorithm 5. In practice, we use SAC [292] as our RL algorithm. In our experiments, we will also study the setting where the human user updates their policy alongside the assistive agent.

## 8.3 EXPERIMENTS

We seek to answer two questions with our experiments. *First*, does our approach enable assistance in standard cooperation benchmarks? *Second*, does our approach scale to harder benchmarks where prior methods fail?

Our experiments will use two benchmarks designed by prior work to study assistance: the obstacle gridworld [4] and Overcooked [6]. Our main **baseline** is AvE [4], a prior empowerment-based method. Our conjecture is that both methods will perform well on the lower-dimensional gridworld task, and that our method will scale more gracefully to the higher dimensional Overcooked environment. We will also compare against a naïve baseline where the assistive agent acts randomly.

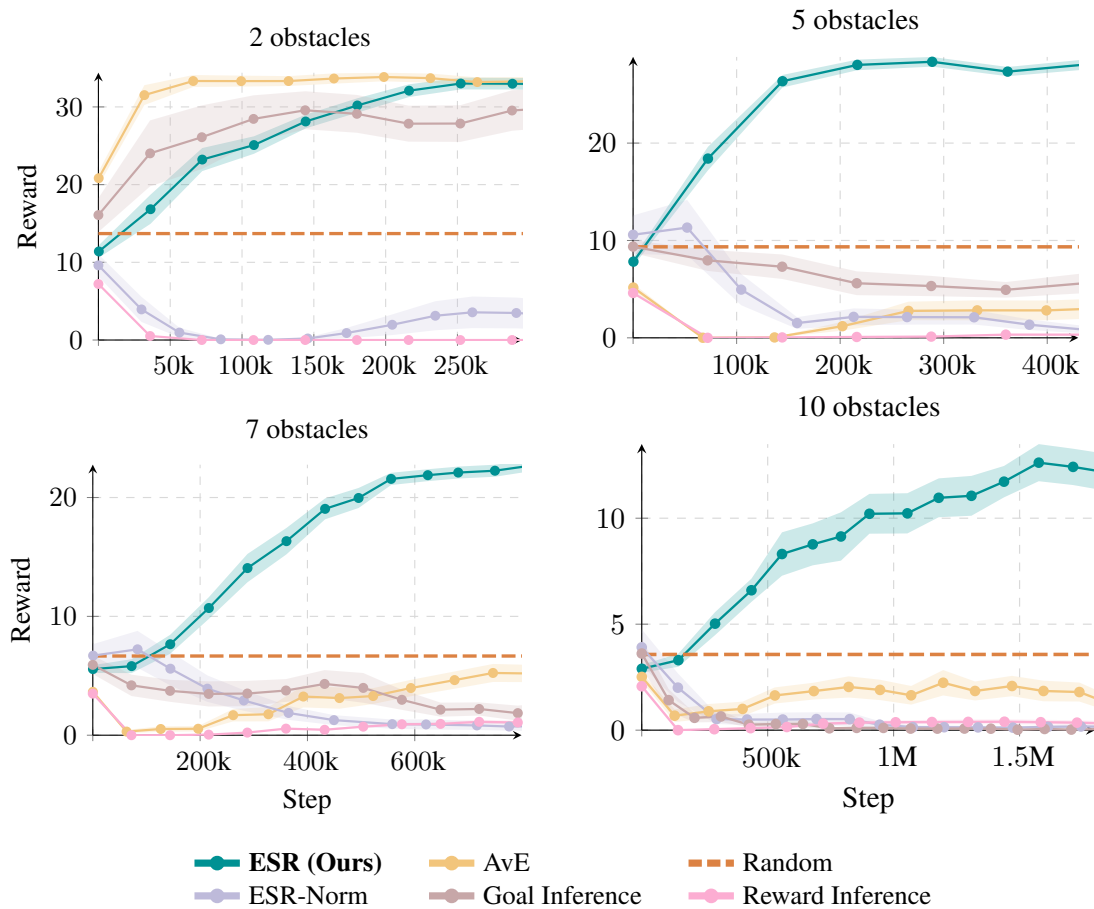


Figure 8.6: We apply our method to the benchmark proposed in prior work [4], visualized in Fig. 8.7a. The four subplots show variant tasks of increasing complexity (more blocks), ( $\pm 1$  SE). We compare against AvE [4], the Goal Inference baseline from [4] which assumes access to a world model, and Reward Inference [5] where we recover the reward from a learned q-value. These prior approaches fail on all except the easiest task, highlighting the importance of scalability.

Do contrastive successor representations effectively estimate empowerment?

We test our approach in the assistance benchmark suggested in Du et al. [4]. The human (orange) is tasked with reaching a goal state (green) while avoiding the obstacles (purple). The AI assistant can move blocks one step at a time in any direction [4]. While the original benchmark used  $N = 2$  obstacles, we will additionally evaluate on harder versions of this task with  $N = 5, 7, 10$  obstacles. We show results in Fig. 8.6. On the easiest task, both our method and AvE achieve similar asymptotic reward, though our method learns more slowly than AvE. However, on the tasks with moderate and high degrees of complexity, our approach (ESR) achieves signifi-

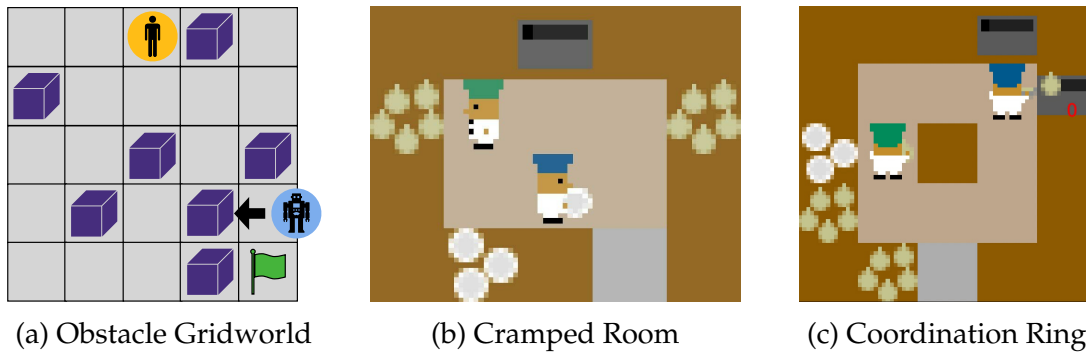


Figure 8.7: (a) The modified environment from Du et al. [4] scaled to  $N = 7$  blocks, and (b, c) the two layouts of the Overcooked environment [6].

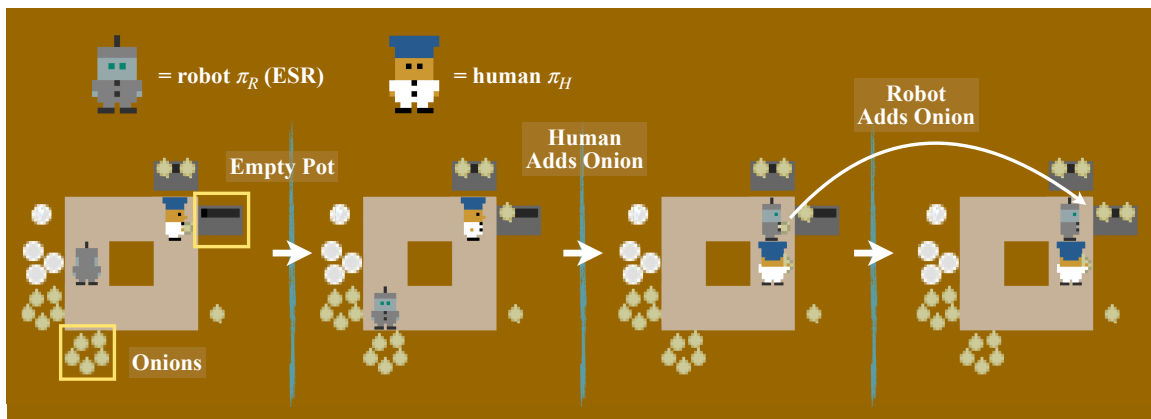


Figure 8.8: In Coordination Ring, our ESR agent learns to wait for the human to add an onion to the pot, and then adds one itself. There is another pot at the top which is nearly full, but the empowerment agent takes actions to maximize the impact of the human’s actions, and so follows the lead of the human by filling the empty pot.

cantly higher rewards than AvE, which performs worse than a random controller. These experiments support our claim that contrastive successor representations provide an effective means for estimating empowerment, and hint that ESR might be well suited for solving higher dimensional tasks.

Does our approach scale to tasks with image-based observations?

Our second set of experiments look at scaling ESR to the image-based Overcooked environment. Since contrastive learning is often applied to image domains, we conjectured that ESR would scale gracefully to this setting. We will evaluate our approach in assisting a human policy trained with behavioral cloning taken from Laidlaw and Dragan [336]. The human prepares dishes by picking up ingredients and cooking them on a stove, while the AI assistant moves ingredients and dishes

around the kitchen. We focus on two environments within this setting: a cramped room where the human must pass ingredients and dishes through a narrow corridor, and a coordination ring where the human must pass ingredients and dishes around a ring-shaped kitchen (Figs. 8.7b and 8.7c). As before, we compare with AvE as well as a naïve random controller. We report results in Table 8.1. On both tasks, we observe that our approach achieves higher rewards than AvE baseline, which performs no better than a random controller. In Fig. 8.8, we show an example of one of the collaborative behaviors learned by ESR. Taken together with the results in the previous setting, these results highlight the scalability of ESR to higher dimensional problems.

Table 8.1: Overcooked Results

Layout	ESR (Ours)	Reward Inference	AvE	Random
Asymmetric Advantages	<b>72.00 ± 5.37</b>	60.33 ± 0.26	36.71 ± 1.71	59.36
Coordination Ring	<b>8.40 ± 0.69</b>	5.96 ± 0.20	5.69 ± 0.93	6.02
Cramped Room	<b>91.33 ± 4.08</b>	39.24 ± 0.35	5.13 ± 1.31	69.26

### Additional Ablations and Details

See Appendix C.4 for additional ablations and details on the experiments.

## 8.4 ADDITIONAL ABLATIONS AND QUALITATIVE RESULTS

In this section we evaluate additional ablations and qualitative results for the ESR method.

### Learning Representations without the Robot Action

In our estimation of empowerment (Eq. 8.29) we supply the robot action  $a^R$  when learning both  $\phi$  and  $\psi$ , however, the theoretical empowerment formulation in Section 8.1 does not require it.

To evaluate the impact of including  $a^R$ , we run an additional ablation without it on the gridworld environment, shown in Fig. 8.3. This ablation shows that the inclusion of  $a^R$  is most impactful in more challenging (higher number of boxes) environments. We hypothesize that conditioning the representations on the robot action reduces the noise in the mutual information estimation, and also reduces the difficulty of classifying true future states.

### Greedy Empowerment Policy

All of our experiments used Soft-Q learning to learn a policy from the empowerment estimation. Here, we additionally study a greedy empowerment policy which takes



the most empowering action at each step. We model this by setting the q-learning gamma to 0 to fully discount future rewards.

Results for this ablation are shown in Fig. 8.4. Unsurprisingly, the greedy optimization vastly underperforms the policy with  $\gamma = 0.9$ .

### ESR Training Example

In Fig. 8.5, we show the mutual information during training of the ESR agent in the gridworld environment with 5 obstacles. The mutual information quickly becomes positive and remains so throughout training. As long as the mutual information is positive, the classifier is able to reward the agent for taking actions that empower the human.

### Simplifying the Objective

The reward function in Eq. (8.35) is itself a random variable because it depends on future states  $g$ . This subsection describes how this randomness can be removed. To do this, we follow prior work [161, 11] in arguing that the learned representations  $\psi(g)$  follow a Gaussian distribution:

**Assumption 8.3** (Based on Wang and Isola [161]). *The representations of future states  $\psi(g)$  learned by contrastive learning have a marginal distribution that is Gaussian:*

$$P(\psi) = \int P(g)\delta(\psi = \psi(g)) dg \stackrel{d}{=} \mathcal{N}(0, I). \quad (8.36)$$

With this assumption, we can remove the random sampling of  $g$  from the reward function. We start by noting that the learned representations tell us the *relative* likelihood of seeing a future state Eq. (8.29)). Assumption 8.3 will allow us to convert these relative likelihoods into likelihoods.

$$\begin{aligned} \mathbb{E}_{P(\mathfrak{s}^+|s, a^{\mathbf{R}}, a^{\mathbf{H}})}[r(s, a^{\mathbf{R}})] &= \mathbb{E}_{P(\mathfrak{s}^+)} \left[ \frac{P(\mathfrak{s}^+|s, a^{\mathbf{R}}, a^{\mathbf{H}})}{P(\mathfrak{s}^+)} r(s, a^{\mathbf{R}}) \right] \\ &= \mathbb{E}_{P(\mathfrak{s}^+)} \left[ C_1 e^{\phi(s, a^{\mathbf{R}}, a^{\mathbf{H}})^T \phi(\mathfrak{s}^+)} r(s, a^{\mathbf{R}}) \right] \\ &= C_1 \mathbb{E}_{\psi \sim P(\phi(\mathfrak{s}^+))} \left[ e^{\phi(s, a^{\mathbf{R}}, a^{\mathbf{H}})^T \psi} (\phi(s, a^{\mathbf{R}}, a^{\mathbf{H}}) - \phi(s, a^{\mathbf{R}}))^T \psi \right] \\ &= C_1 (\phi(s, a^{\mathbf{R}}, a^{\mathbf{H}}) - \phi(s, a^{\mathbf{R}}))^T \\ &\quad \int \frac{1}{(2\pi)^{d/2}} e^{-\frac{1}{2} \|\psi\|_2^2 + \phi(s, a^{\mathbf{R}}, a^{\mathbf{H}})^T \psi} \psi \, d\psi \\ &= C_1 (\phi(s, a^{\mathbf{R}}, a^{\mathbf{H}}) - \phi(s, a^{\mathbf{R}}))^T e^{\frac{1}{2} \|\phi(s, a^{\mathbf{R}}, a^{\mathbf{H}})\|_2^2} \\ &\quad \int \frac{1}{(2\pi)^{d/2}} e^{-\frac{1}{2} \|\psi\|_2^2 + \phi(s, a^{\mathbf{R}}, a^{\mathbf{H}})^T \psi - \frac{1}{2} \|\phi(s, a^{\mathbf{R}}, a^{\mathbf{H}})\|_2^2} \psi \, d\psi \end{aligned}$$

$$\begin{aligned}
&= C_1 (\phi(s, a^{\mathbf{R}}, a^{\mathbf{H}}) - \phi(s, a^{\mathbf{R}}))^T \\
&\quad e^{\frac{1}{2} \|\phi(s, a^{\mathbf{R}}, a^{\mathbf{H}})\|_2^2} \mathbb{E}_{\psi \sim \mathcal{N}(\mu = \phi(s, a^{\mathbf{R}}, a^{\mathbf{H}}), \Sigma = I)} [\psi] \\
&= C_1 e^{\frac{1}{2} \|\phi(s, a^{\mathbf{R}}, a^{\mathbf{H}})\|_2^2} (\phi(s, a^{\mathbf{R}}, a^{\mathbf{H}}) - \phi(s, a^{\mathbf{R}}))^T \phi(s, a^{\mathbf{R}}, a^{\mathbf{H}}).
\end{aligned} \tag{8.37}$$

This simplification may be attractive in cases where the computed empowerment bonuses have high variance, or when the empowerment horizon is large (i.e.,  $\gamma \rightarrow 1$ , as in Section 8.1). Empirically, we found this version of the objective to be less effective in practice due to the additional representation structure required by Assumption 8.3.

## 8.5 DISCUSSION

One of the most important problems in AI today is equipping AI agents with the capacity to assist humans achieve their goals. While much of the prior work in this area requires inferring the human’s intention, our work builds on prior work in studying how an assistive agent can *empower* a human user without inferring their intention. Relative to prior methods, we demonstrate how empowerment can be readily estimated using contrastive learning, paving the way for deploying these techniques on high-dimensional problems.

**Limitations.** One of the main limitations of our approach is the assumption that the assistive agent has access to the human’s actions, which could be challenging to observe in practice. Automatically inferring the human’s actions remains an important problem for future work. A second limitation is that the method is currently an on-policy method, in the sense that the assistive agent has to learn by trial and error. A third limitation is that the ESR formulation assumes that both agents share the same state space. In many cases the empowerment objective will still lead to desirable behavior, however, care must be taken in cases where the agent can restrict the information in its own observations, which could lead to reward hacking. Finally, our experiments do not test our method against real humans, whose policies may differ from the simulated policies. In the future, we plan to investigate techniques from off-policy evaluation and cooperative game theory to enable faster learning of assistive agents with fewer trials. We also plan to test the ESR objective in environments with partial observability over the human’s state.

**Safety risks.** Perhaps the main risk involved with maximizing empowerment is that it may be at odds with a human’s agents goal, especially in contexts where the pursuit of that goal limits the human’s capacity to pursue other goals. For example, a family choosing to have a kid has many fewer options over where they can travel for vacation, yet we do not want assistive agents to stymie families from having children.

One key consideration is *whom* should be empowered. The present paper assumes there is a single human agent. Equivalently, this can be seen as maximizing the empowerment of all exogenous agents. However, it is easy to adapt the proposed method to maximize the empowerment of a single target individual. Given historical inequities in the distribution of power, practitioners must take care when considering whose empowerment to maximize. Similarly, while we focused on *maximizing* empowerment, it is trivial to change the sign so that an “assistive” agent minimizes empowerment. One could imagine using such a tool in policies to handicap one’s political opponents.

## 8.6 RELATED WORK

Our approach broadly connects ideas from contrastive representation learning and intrinsic motivation to the problem of assisting humans.

**Assistive Agents.** There are two lines of past work on assistive agents that are most relevant.

The first line of work focuses on the setting of an assistance game [316], where a robot (AI) agent tries to optimize a human reward of which it is initially unaware. Practically, inverse reinforcement learning (IRL) can be used in such a setting to infer the human’s reward function and assist the human in achieving their goals [317]. The key challenge with this approach is that it requires modeling the human’s reward function. This can be difficult in practice, especially if the human’s behavior is not well-modeled by the reward architecture. Slightly misspecified reward functions can lead to catastrophic outcomes (i.e., directly harmful behavior in the assistance context) [337–339]. By contrast, our approach does not require modeling the human’s reward function.

The second line of work focuses on empowerment-like objectives for assistance and shared autonomy. Empowerment generally refers to a measure of an agent’s ability to influence the environment [323, 340]. In the context of assistance, Du et al. [4] show one such approximation of empowerment (AvE) can be approximated in simple environments through random rollouts to assist humans. Meanwhile, empowerment-like objectives have been used in shared autonomy settings to assist humans with teleoperation [341] and general assistive interfaces [342]. A key limitation of these approaches for general assistance is they only model empowerment over one time step. Our approach enables a more scalable notion of empowerment that can be computed over multiple time steps.

**Intrinsic Motivation.** Intrinsic motivation broadly refers to agents that accomplish behaviors in the absence of an externally-specified reward or task [343]. Common applications of intrinsic motivation in single-agent reinforcement learning include exploration and skill discovery [344–346], empowerment [340, 323], and surprise minimization [347, 348, 340]. When applied to settings with humans, these

objectives may lead to antisocial behavior [319]. Our approach applies intrinsic motivation to the setting of assisting humans, where the agent’s goal is an empowerment objective—to maximize the human’s ability to change the environment.

**Information-theoretic Decision Making.** Information-theoretic approaches have seen broad applicability across unsupervised reinforcement learning [236, 340, 344]. These methods have been applied to goal-reaching [211], skill discovery [349, 335, 345, 350, 351], and exploration [346, 352, 353]. In the context of assisting humans, information-theoretic methods have primarily been used to reason about the human’s goals or rewards [354–356].

Our approach is made possible by advances in contrastive representation learning for efficient estimation of the mutual information of sequence data [152]. While these methods have been widely used for representation learning [159, 224] and reinforcement learning [357, 121, 124, 358], to the best of our knowledge prior work has not used these contrastive techniques for learning assistive agents.

## 9

THE REPRESENTATION GEOMETRY OF  
INTRINSIC MOTIVATION

When learning contrastive representations for decision making, we typically try and construct state representations  $\phi$  and goal representations  $\psi$  such that the dot product  $\phi(s)^T \psi(g)$  can classify future goals from other states. Wang and Isola [161] argue that contrastive representations learn uniform marginal distributions when the goal representation is constrained to lie on the unit hypersphere  $S^{n-1}$ . This is a useful property for decision-making, as well-structured marginal distributions for states and goals enable easy planning and inference [11], reasoning about information-theoretic quantities like surprise [359], and allow for easy sampling of goals and states.

In this section, I'll discuss some ideas on how to make time-contrastive representations compatible with satisfying analytic marginal properties. This structure lets us reason about the relationship between the state and attainable goals geometrically. We focus on two quantities that are relevant for exploration and skill-discovery:

$$D_{\text{KL}} [p(g | s) || p(g)] \text{ and } I(s; g). \quad (9.1)$$

Depending on the geometry of our contrastive representations, we can derive closed-form expressions for these quantities in terms of the representation norms and/or the learned temperature.

**Notation.** We consider the setting of an infinite-horizon controlled Markov process  $\{s_t\}_{t=0}^{\infty} \in \mathcal{S}$  that starts at its stationary distribution (i.e., the marginal  $p(s_t) = p(s_0)$  for each  $t$ ). We say the state and goal are jointly distributed as  $p(s, g) = \Pr(s_0 = s, s^+ = g)$  for  $s^+ = s_k, k \sim \text{Geom}(1 - \gamma)$ . We will use  $\phi(s)$  and  $\psi(g)$  to denote the state and goal representations, respectively, and  $\Pr(s^+ = g | s_0 = s)$  to denote the conditional probability of the goal given the state.

In practice, none of the math in this note uses the Markov / time-series structure, though the derived properties are most relevant for decision making (e.g., Eysenbach et al. [11]). Given the assumptions about the marginal structure of the representations we make below, the results will hold for any joint distribution  $p(s, g)$  modeled with contrastive learning.

## 9.1 BINARY NCE

In the binary NCE setting, we learn to classify future goals from other states with a dot product that represents normalized log-likelihood ratios:

$$\max_{\phi, \psi} \mathbb{E}_{p(s, g)p(g^-)} \left[ \log \sigma(\phi(s)^T \psi(g)) + \log \sigma(-\phi(s)^T \psi(g^-)) \right]. \quad (9.2)$$

Eysenbach et al. [121] argue that contrastive representations learn to satisfy the following property when learned with this loss:

$$\phi(s)^T \psi(g) = \log \frac{\Pr(s^+ = g \mid s_0 = s)}{\Pr(s^+ = g)}. \quad (9.3)$$

If we restrict  $\psi$  to lie on the unit hypersphere  $S^{n-1}$ , Wang and Isola [161] separately argue that contrastive learning should produce representations that are uniform over the unit hypersphere  $S^{n-1}$ , i.e.,

$$\Pr(\psi(s_t)) = C \quad (\text{const.}) \quad (9.4)$$

We then run into an issue. For simplicity, suppose  $\phi, \psi$  are diffeomorphisms with inverses  $\phi^{-1}, \psi^{-1}$ . We will convert to an expectation to make the change of variables clear, using the fact that the marginal  $p(\psi(g))$  is uniform over  $S^{n-1}$ :

$$\begin{aligned} D_{\text{KL}}[p(g) \parallel p(g \mid s)] &= \int_S \log \frac{p(g)}{p(g \mid s)} p(g) \, dg \\ &= - \int_S \log \frac{p(g \mid s)}{p(g)} p(g) \, dg \\ &= - \int_S (\phi(s)^T \psi(g)) p(g) \, dg \\ &= - \int_S (\phi(s)^T \psi(g)) \, d(p(g)) \\ &= - \int_{S^{n-1}} (\phi(s)^T \psi(g)) \, d(p(\psi(g))) \\ &= -C \int_{S^{n-1}} \phi(s)^T \psi(g) \, d(\psi(g)) \\ &= -\frac{C}{2} \int_{S^{n-1}} \phi(s)^T \psi(g) \, d(\psi(g)) - \frac{C}{2} \int_{S^{n-1}} \phi(s)^T \psi(g) \, d(-\psi(g)) \\ &\hspace{15em} (\text{symmetry}) \\ &= -\frac{C}{2} \int_{S^{n-1}} \phi(s)^T \psi(g) \, d(\psi(g)) + \frac{C}{2} \int_{S^{n-1}} \phi(s)^T \psi(g) \, d(\psi(g)) \\ &= 0. \end{aligned} \quad (9.5)$$

But this implies that  $p(g \mid s) = p(g)$  everywhere, i.e.,  $I(s; g) = 0$  and states have no influence on the future states, a contradiction. This suggests our construction is

over-constrained in the dot-product parameterization of contrastive RL (if we also want nice marginal distributions over the goal representation). How can we solve this?

## 9.2 SYMMETRIC INFO NCE WITH A LEARNED TEMPERATURE

One idea is to add an additional degree of freedom to the contrastive critic, switching to a (symmetrized) infoNCE loss, constraining both  $\phi$  and  $\psi$  to lie on the unit hypersphere  $S^{n-1}$ , and applying a learned precision  $\tau$  to the dot product, similar to the CLIP loss [17, 11, 53]:

$$\max_{\phi, \psi} \mathbb{E}_{\{p(s_i, g_i)\}_{i=1}^N} \left[ \sum_{i=1}^N \log \left( \frac{e^{\tau \phi(s_i)^T \psi(g_i)}}{\sum_{j=1}^N e^{\tau \phi(s_i)^T \psi(g_j)}} \right) + \log \left( \frac{e^{\tau \phi(s_i)^T \psi(g_i)}}{\sum_{j=1}^N e^{\tau \phi(s_j)^T \psi(g_i)}} \right) \right]. \quad (9.6)$$

We then recover the following property at convergence for some constant  $C$  in the limit  $N \rightarrow \infty$ :

$$\tau \phi(s)^T \psi(g) = \log \frac{\Pr(s^+ = g \mid s_0 = s)}{C \Pr(s^+ = g)}. \quad (9.7)$$

We can compute  $D_{\text{KL}}[p(g \mid s) \parallel p(g)]$  by integrating over the hypersphere:

$$\begin{aligned} D_{\text{KL}}[p(g \mid s) \parallel p(g)] &= \int_{\mathcal{S}} p(g \mid s) \log \frac{p(g \mid s)}{p(g)} \, dg \\ &= \int_{\mathcal{S}} \frac{p(g \mid s)}{p(g)} \log \left( \frac{p(g \mid s)}{p(g)} \right) p(g) \, dg \\ &= \int_{\mathcal{S}} C e^{\tau \phi(s)^T \psi(g)} (\tau \phi(s)^T \psi(g) + \log C) p(g) \, dg \\ &= \int_{\mathcal{S}} C e^{\tau \phi(s)^T \psi(g)} (\tau \phi(s)^T \psi(g) + \log C) \, d(p(g)) \\ &= C \int_{S^{n-1}} e^{\tau \phi(s)^T \psi(g)} (\tau \phi(s)^T \psi(g) + \log C) \, d(p(\psi(g))) \\ &\hspace{15em} \text{(change of variables)} \\ &= C \int_{S^{n-1}} e^{\tau \phi(s)^T \psi(g)} (\tau \phi(s)^T \psi(g) + \log C) C_0 \, d(\psi(g)). \\ &\hspace{15em} \text{(uniform marginal (9.4))} \end{aligned}$$

We then convert to spherical coordinates relative to  $\phi(s)$  by defining a rotation matrix  $R$  whose first row is  $\phi(s) / \|\phi(s)\|$ , and coordinates

$$R\psi(g) = \begin{pmatrix} \cos(\theta_1) \\ \sin(\theta_1) \cos(\theta_2) \\ \sin(\theta_1) \sin(\theta_2) \cos(\theta_3) \\ \vdots \\ \sin(\theta_1) \sin(\theta_2) \cdots \sin(\theta_{n-2}) \cos(\theta_{n-1}) \end{pmatrix},$$

For some  $du$  that is orthogonal to  $d\theta_1$  and  $dr$ , we can write [360]

$$d(R\psi(g)) = \sin^{n-2}(\theta_1) d\theta_1 \wedge du.$$

Continuing from the previous expression, we have

$$\begin{aligned}
&= C \int_{S^{n-1}} e^{\tau\phi(s)^T\psi(g)} (\tau\phi(s)^T\psi(g) + \log C) C_0 d(\psi(g)) \\
&= C \int_{S^{n-1}} e^{\tau\phi(s)^T\psi(g)} (\tau\phi(s)^T\psi(g)) C_0 d(\psi(g)) + C \int_{S^{n-1}} e^{\tau\phi(s)^T\psi(g)} C_0 \log C d(\psi(g)) \\
&= C \int_{S^{n-1}} e^{\tau\phi(s)^T\psi(g)} (\tau\phi(s)^T\psi(g)) C_0 d(\psi(g)) + C_0 C \log C \left( \frac{(2\pi)^{n/2} I_{n/2-1}(\tau)}{\tau^{n/2-1}} \right) \\
&= C_0 C \log C \left( \frac{(2\pi)^{n/2} I_{n/2-1}(\tau)}{\tau^{n/2-1}} \right) + C \int_{S^{n-1}} e^{\tau \cos \theta_1} (\tau \cos \theta_1) C_0 d(\psi(g)) \\
&= C_0 C \log C \left( \frac{(2\pi)^{n/2} I_{n/2-1}(\tau)}{\tau^{n/2-1}} \right) + C \int_{S^{n-1}} e^{\tau \cos \theta_1} (\tau \cos \theta_1) C_0 d(R\psi(g)) \\
&\hspace{15em} \text{(rotational symmetry of } S^{n-1}) \\
&= C_0 C \log C \left( \frac{(2\pi)^{n/2} I_{n/2-1}(\tau)}{\tau^{n/2-1}} \right) + C \int_{S^{n-1}} e^{\tau \cos \theta_1} (\tau \cos \theta_1) C_0 \sin^{n-2}(\theta_1) d\theta_1 \wedge du \\
&= C_0 C \log C \left( \frac{(2\pi)^{n/2} I_{n/2-1}(\tau)}{\tau^{n/2-1}} \right) + C_0 C \text{vol}(S^{n-2}) \int_0^\pi e^{\tau \cos \theta} (\tau \cos \theta) \sin^{n-2}(\theta) d\theta \\
&= C_0 C \log C \left( \frac{(2\pi)^{n/2} I_{n/2-1}(\tau)}{\tau^{n/2-1}} \right) + C_0 C \left( \frac{2\pi^{(n-1)/2}}{\Gamma((n-1)/2)} \right) \int_0^\pi e^{\tau \cos \theta} (\tau \cos \theta) \sin^{n-2}(\theta) d\theta \\
&= C_0 C \log C \left( \frac{(2\pi)^{n/2} I_{n/2-1}(\tau)}{\tau^{n/2-1}} \right) + C_0 C \left( \frac{2\pi^{(n-1)/2}}{\Gamma((n-1)/2)} \right) \left( \frac{2^{\frac{n}{2}-1} \sqrt{\pi} \Gamma(\frac{n-1}{2}) I_{n/2}(\tau)}{\tau^{n/2-2}} \right) \\
&= C_0 C \log C \left( \frac{(2\pi)^{n/2} I_{n/2-1}(\tau)}{\tau^{n/2-1}} \right) + C_0 C \left( \frac{(2\pi)^{n/2} I_{n/2}(\tau)}{\tau^{n/2-2}} \right) \\
&\propto \frac{I_{n/2-1}(\tau) \log C + \tau I_{n/2}(\tau)}{\tau^{n/2-1}}. \tag{9.8}
\end{aligned}$$

where  $I_{n/2}$ ,  $I_{n/2-1}$  are modified Bessel functions of the first kind. We can see that this last expression is independent of  $s$ , implying that  $D_{\text{KL}}[p(g | s) \| p(g)]$  is independent of  $s$ . This is also undesirable, as this quantity is determined by the underlying Markov process — some states may be more informative about the future than others.

Since we only used the assumption of uniformity for the goal representation over  $S^{n-1}$ , one possible solution is to relax  $\phi(s)$  to lie anywhere in  $\mathbb{R}^n$ . This is equivalent to making the learned temperature  $\tau$  a function of the state representation  $s$  — specifically  $\tau(s) = \|\phi(s)\|$ . With this adjustment, we can say

$$D_{\text{KL}}[p(g | s) \| p(g)] \propto \frac{I_{n/2-1}(\|\phi(s)\|) \log C + \|\phi(s)\| I_{n/2}(\|\phi(s)\|)}{\|\phi(s)\|^{n/2-1}}. \tag{9.9}$$

Now,  $D_{\text{KL}}[p(g | s) \| p(g)]$  is a monotonic function of  $\|\phi(s)\|$ . As  $\|\phi(s)\| \rightarrow \infty$ , the conditional distribution  $p(g | s)$  more distinct from the marginal (see Figure 9.1).



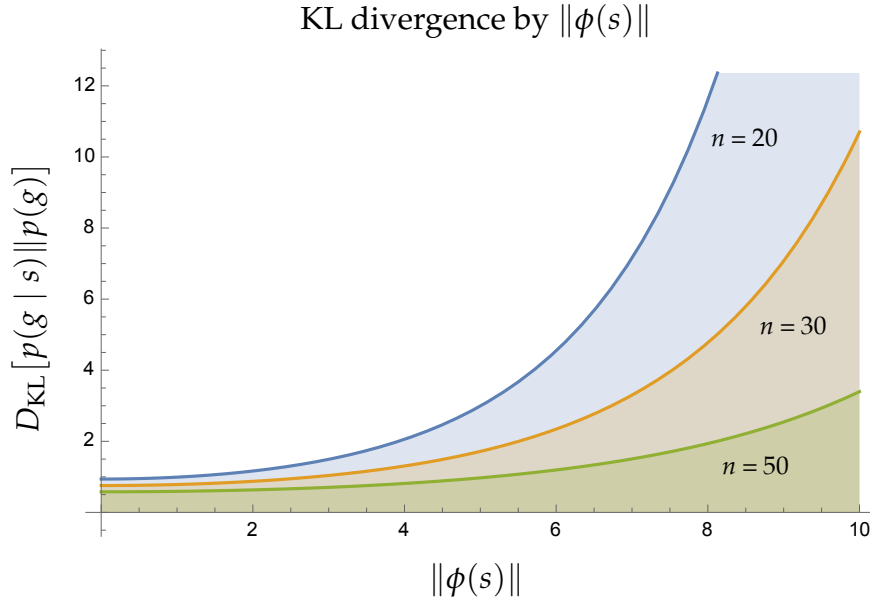


Figure 9.1: KL divergence between conditional and marginal goal distributions as a function of  $\|\phi(s)\|$  for different representation dimensions  $n$  with  $C = 5$  (up to a constant). The KL divergence is monotonically increasing with  $\|\phi(s)\|$ .

### 9.3 EUCLIDEAN INFONCE

Another option is to switch to an  $\ell_2$  parameterization  $\|\phi(s) - \psi(g)\|$  and assume Gaussian marginals. We train the following loss for  $\phi(s), \psi(g) \in \mathbb{R}^n$  learned:

$$\max_{\phi, \psi} \mathbb{E}_{\{p(s_i, g_i)\}_{i=1}^N} \left[ \sum_{i=1}^N \log \left( \frac{e^{-\|\phi(s_i) - \psi(g_i)\|^2}}{\sum_{j=1}^N e^{-\|\phi(s_i) - \psi(g_j)\|^2}} \right) + \log \left( \frac{e^{-\|\phi(s_i) - \psi(g_i)\|^2}}{\sum_{j=1}^N e^{-\|\phi(s_j) - \psi(g_i)\|^2}} \right) \right]. \quad (9.10)$$

This loss will converge to the following [11]:

$$\|\phi(s) - \psi(g)\|^2 = \log \left( \frac{C \Pr(s^+ = g)}{\Pr(s^+ = g \mid s_0 = s)} \right). \quad (9.11)$$

We additionally structure the marginals as Gaussians with precision controlled by  $\tau$  [11]:

$$\begin{aligned} \Pr(\psi(s_t)) &= \left(\frac{\tau}{\pi}\right)^{n/2} e^{-\tau \|\psi(s_t)\|^2} \\ \Pr(\phi(s_t)) &= \left(\frac{\tau}{\pi}\right)^{n/2} e^{-\tau \|\phi(s_t)\|^2}. \end{aligned} \quad (9.12)$$

We can now compute:

$$D_{\text{KL}} [p(g \mid s) \parallel p(g)] = \int_{\mathcal{S}} p(g \mid s) \log \frac{p(g \mid s)}{p(g)} dg$$

$$\begin{aligned}
&= \int_S \frac{p(g | s)}{p(g)} \log \frac{p(g | s)}{p(g)} p(g) \, dg \\
&= \int_S \frac{p(g | s)}{p(g)} \log \frac{p(g | s)}{p(g)} \, d(p(g)) \\
&= C \int_{\mathbb{R}^n} e^{-\|\phi(s) - \psi(g)\|^2} \left( \log C - \|\phi(s) - \psi(g)\|^2 \right) p(\psi(g)) \, d(\psi(g)) \\
&= C \int_{\mathbb{R}^n} e^{-\|\phi(s) - \psi(g)\|^2} \left( \log C - \|\phi(s) - \psi(g)\|^2 \right) \left( \frac{\tau}{\pi} \right)^{n/2} e^{-\tau \|\psi(g)\|^2} \, d(\psi(g)) \\
&= \left( \frac{\tau}{\pi} \right)^{n/2} C \int_{\mathbb{R}^n} e^{2\phi(s)^T \psi(g) - \phi(s)^T \phi(s) - (1+\tau) \psi(g)^T \psi(g)} \left( \log C - \|\phi(s) - \psi(g)\|^2 \right) \, d(\psi(g)) \\
&= \left( \frac{\tau}{\pi} \right)^{n/2} C \int_{\mathbb{R}^n} e^{-(1+\tau) \|\frac{\phi(s)}{1+\tau} - \psi(g)\|^2} e^{\frac{-\tau}{1+\tau} \|\phi(s)\|^2} \left( \log C - \|\phi(s) - \psi(g)\|^2 \right) \, d(\psi(g)) \\
&= \left( \frac{\tau}{\pi} \right)^{n/2} C e^{\frac{-\tau}{1+\tau} \|\phi(s)\|^2} \left( \int_{\mathbb{R}^n} \overbrace{\nabla_{\psi(g)} \cdot \left( \left( \frac{2\tau\phi(s) - (1+\tau)\psi(g)}{2(1+\tau)^2} \right) e^{-(1+\tau) \|\frac{\phi(s)}{1+\tau} - \psi(g)\|^2} \right)}^{\rightarrow 0 \text{ as } \|\psi(g)\| \rightarrow \infty} \, d(\psi(g)) \right. \\
&\quad \left. + \int_{\mathbb{R}^n} \left( \frac{\tau(\tau-2)}{(1+\tau)^2} \|\phi(s)\|^2 + \frac{n}{2(1+\tau)} + \log C \right) e^{-(1+\tau) \|\frac{\phi(s)}{1+\tau} - \psi(g)\|^2} \, d(\psi(g)) \right) \\
&= \left( \frac{\tau}{\pi} \right)^{n/2} C e^{\frac{-\tau}{1+\tau} \|\phi(s)\|^2} \int_{\mathbb{R}^n} \left( \frac{\tau(\tau-2)}{(1+\tau)^2} \|\phi(s)\|^2 + \frac{n}{2(1+\tau)} + \log C \right) e^{-(1+\tau) \|\frac{\phi(s)}{1+\tau} - \psi(g)\|^2} \, d(\psi(g)) \\
&= \left( \frac{\tau}{\pi} \right)^{n/2} C e^{\frac{-\tau}{1+\tau} \|\phi(s)\|^2} \left( \frac{\tau(\tau-2)}{(1+\tau)^2} \|\phi(s)\|^2 + \frac{n}{2(1+\tau)} + \log C \right) \int_{\mathbb{R}^n} e^{-(1+\tau) \|\frac{\phi(s)}{1+\tau} - \psi(g)\|^2} \, d(\psi(g)) \\
&= \left( \frac{\tau}{\pi} \right)^{n/2} C e^{\frac{-\tau}{1+\tau} \|\phi(s)\|^2} \left( \frac{\tau(\tau-2)}{(1+\tau)^2} \|\phi(s)\|^2 + \frac{n}{2(1+\tau)} + \log C \right) \int_{\mathbb{R}^n} e^{-(1+\tau) \|\psi(g)\|^2} \, d(\psi(g)) \\
&= \left( \frac{\tau}{\pi} \right)^{n/2} C e^{\frac{-\tau}{1+\tau} \|\phi(s)\|^2} \left( \frac{\tau(\tau-2)}{(1+\tau)^2} \|\phi(s)\|^2 + \frac{n}{2(1+\tau)} + \log C \right) \left( \frac{\pi}{1+\tau} \right)^{n/2} \\
&= \left( \frac{\tau}{\tau+1} \right)^{n/2} C e^{\frac{-\tau}{1+\tau} \|\phi(s)\|^2} \left( \frac{\tau(\tau-2)}{(1+\tau)^2} \|\phi(s)\|^2 + \frac{n}{2(1+\tau)} + \log C \right) \tag{9.13} \\
&\propto \|\phi(s)\|^2 e^{\frac{-\tau}{1+\tau} \|\phi(s)\|^2}.
\end{aligned}$$

Now, we have a term that depends on the norm of  $\phi(s)$ , which is more reasonable. This suggests that the Euclidean parameterization of the contrastive critic may be more appropriate for learning contrastive representations with a well-structured goal marginal.

When  $\|\phi(s)\|$  is small, the conditional distribution  $p(g | s)$  is close to the marginal  $p(g)$ , and the state is less informative about the future.

We can also reason directly about the mutual information between the state and goal:

$$\begin{aligned}
I(s; g) &= \mathbb{E}_{p(s, g)} \left[ \log \frac{p(s, g)}{p(s)p(g)} \right] \\
&= \mathbb{E}_{p(s)} D_{\text{KL}} [p(g | s) \| p(g)]
\end{aligned}$$

$$\begin{aligned}
 &= \int_{\mathbb{R}^n} \left( \left( \frac{\tau}{1+\tau} \right)^{n/2} C e^{\frac{-\tau}{1+\tau} \|\phi(s)\|^2} \left( \frac{\tau(\tau-2)}{(1+\tau)^2} \|\phi(s)\|^2 + \frac{n}{2(1+\tau)} + \log C \right) \right) d(p(\phi(s))) \\
 &= \left( \frac{\tau}{1+\tau} \right)^{n/2} \int_{\mathbb{R}^n} \left( C e^{\frac{-\tau}{1+\tau} \|\phi(s)\|^2} \left( \frac{\tau(\tau-2)}{(1+\tau)^2} \|\phi(s)\|^2 + \frac{n}{2(1+\tau)} + \log C \right) \right) \left( \frac{\tau}{\pi} \right)^{n/2} e^{-\tau \|\phi(s)\|^2} d(\phi(s)) \\
 &= \left( \frac{\tau}{\pi} \right)^{n/2} \left( \frac{\tau}{1+\tau} \right)^{n/2} \int_{\mathbb{R}^n} \left( C e^{\frac{-2\tau-\tau^2}{1+\tau} \|\phi(s)\|^2} \left( \frac{\tau(\tau-2)}{(1+\tau)^2} \|\phi(s)\|^2 + \frac{n}{2(1+\tau)} + \log C \right) \right) d(\phi(s)) \\
 &= C \left( \frac{\tau}{\pi} \right)^{n/2} \left( \frac{\tau}{1+\tau} \right)^{n/2} \frac{\tau(\tau-2)}{(1+\tau)^2} \int_{\mathbb{R}^n} \|\phi(s)\|^2 e^{\frac{-2\tau-\tau^2}{1+\tau} \|\phi(s)\|^2} d(\phi(s)) \\
 &\quad + C \left( \frac{\tau}{\pi} \right)^{n/2} \left( \frac{\tau}{1+\tau} \right)^{n/2} \left( \frac{n}{2(1+\tau)} + \log C \right) \int_{\mathbb{R}^n} e^{-\left( \frac{2\tau+\tau^2}{1+\tau} \right) \|\phi(s)\|^2} d(\phi(s)) \\
 &= C \left( \frac{\tau}{\pi} \right)^{n/2} \left( \frac{\tau}{1+\tau} \right)^{n/2} \frac{\tau(\tau-2)}{(1+\tau)^2} \int_{\mathbb{R}^n} \overbrace{\nabla_{\phi(s)} \cdot \left( \frac{1+\tau}{-2\tau-\tau^2} \right) \phi(s) e^{\frac{-2\tau-\tau^2}{1+\tau} \|\phi(s)\|^2}}^{\rightarrow 0 \text{ as } \|\phi(s)\| \rightarrow \infty} d(\phi(s)) \\
 &\quad - C \left( \frac{\tau}{\pi} \right)^{n/2} \left( \frac{\tau}{1+\tau} \right)^{n/2} \frac{\tau(\tau-2)}{(1+\tau)^2} \int_{\mathbb{R}^n} n \left( \frac{1+\tau}{-2\tau-\tau^2} \right) e^{\frac{-2\tau-\tau^2}{1+\tau} \|\phi(s)\|^2} d(\phi(s)) \\
 &\quad + C \left( \frac{\tau}{\pi} \right)^{n/2} \left( \frac{\tau}{1+\tau} \right)^{n/2} \left( \frac{n}{2(1+\tau)} + \log C \right) \int_{\mathbb{R}^n} e^{-\left( \frac{2\tau+\tau^2}{1+\tau} \right) \|\phi(s)\|^2} d(\phi(s)) \\
 &= C \left( \frac{\tau}{\pi} \right)^{n/2} n \left( \frac{\tau}{1+\tau} \right)^{n/2} \left( \frac{\tau(\tau-2)}{(1+\tau)^2} \left( \frac{1+\tau}{2\tau+\tau^2} \right) + \frac{1}{2(1+\tau)} + \log C \right) \\
 &\quad \int_{\mathbb{R}^n} e^{-\left( \frac{2\tau+\tau^2}{1+\tau} \right) \|\phi(s)\|^2} d(\phi(s)) \\
 &= C \left( \frac{\tau}{\pi} \right)^{n/2} n \left( \frac{\tau}{1+\tau} \right)^{n/2} \left( \frac{\tau^2+2\tau+2\tau(\tau-2)}{2\tau(\tau^2+3\tau+2)} + \log C \right) \int_{\mathbb{R}^n} e^{-\left( \frac{2\tau+\tau^2}{1+\tau} \right) \|\phi(s)\|^2} d(\phi(s)) \\
 &= C \left( \frac{\tau}{\pi} \right)^{n/2} n \left( \frac{\tau}{1+\tau} \right)^{n/2} \left( \frac{\tau^2+2\tau+2\tau(\tau-2)}{2\tau(\tau^2+3\tau+2)} + \log C \right) \left( \frac{\pi(1+\tau)}{\tau(2+\tau)} \right)^{n/2} \\
 &= C n \left( \frac{\tau}{\tau+2} \right)^{n/2} \left( \frac{\tau^2+2\tau+2\tau(\tau-2)}{2\tau(\tau^2+3\tau+2)} + \log C \right). \tag{9.14}
 \end{aligned}$$

We plot  $I(s; g)$  in Figure 9.2 for  $n = 20$  as a function of  $\tau$  and  $C$ .

How can we pick  $C$ ? Equation (9.14) seems to indicate that for a fixed temperature  $\tau$ , the representation ratio  $C$  has a fixed value. If we jointly tune  $\tau$  with the representations though in general we can pick  $C$  to be any positive value. To use an equation like Equation (9.13) in practice, it may be worthwhile constraining  $C$  to be related to  $\tau$  or directly learned so there is a consistent relationship between  $D_{\text{KL}}[p(g | s) \| p(g)]$  and the representation norm  $\|\phi(s)\|$ .

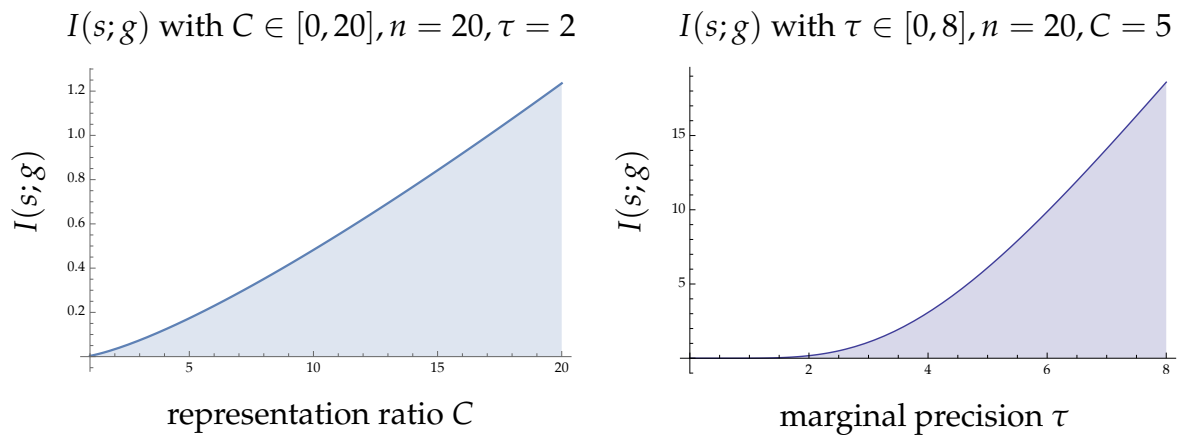


Figure 9.2: Mutual information  $I(s;g)$  between the state and goal representations in the Euclidean infoNCE setting. Both the representation ratio  $C$  and the marginal precision  $\tau$  scale monotonically with the mutual information. The mutual information  $I(s;g)$  is positive only when  $C > 1$ , and so we can additionally interpret this as a constraint on the representation ratio.

## BIBLIOGRAPHY

---

- [1] Raj Ghugare, Matthieu Geist, Glen Berseth, and Benjamin Eysenbach. Closing the Gap Between TD Learning and Supervised Learning – a Generalisation Point of View. In *International Conference on Learning Representations*. 2024.
- [2] Justin Fu, Aviral Kumar, Ofir Nachum, George Tucker, and Sergey Levine. D4RL: Datasets for Deep Data-Driven Reinforcement Learning. 2020. arXiv:2004.07219.
- [3] Laurens van der Maaten and Geoffrey Hinton. Visualizing Data Using T-Sne. *Journal of Machine Learning Research*, 9(11):2579–2605, 2008.
- [4] Yuqing Du, Stas Tiomkin, Emre Kiciman, Daniel Polani, Pieter Abbeel, and Anca Dragan. Ave: Assistance via Empowerment. In *Neural Information Processing Systems*, volume 33, pp. 4560–4571. 2020.
- [5] Divyansh Garg, Shuvam Chakraborty, Chris Cundy, Jiaming Song, and Stefano Ermon. Iq-Learn: Inverse Soft-Q Learning for Imitation. In *Neural Information Processing Systems*, volume 34, pp. 4028–4039. 2021.
- [6] Micah Carroll, Rohin Shah, Mark K. Ho, Thomas L. Griffiths, Sanjit A. Seshia, Pieter Abbeel, and Anca Dragan. On the Utility of Learning About Humans for Human-Ai Coordination. In *Neural Information Processing Systems*. 2019.
- [7] Vivek Myers, Andre Wang He, Kuan Fang, Homer Rich Walke, Philippe Hansen-Estruch, Ching-An Cheng, Mihai Jalobeanu, Andrey Kolobov, Anca Dragan, and Sergey Levine. Goal Representations for Instruction Following: A Semi-Supervised Language Interface to Control. In *Conference on Robot Learning*, pp. 3894–3908. 2023.
- [8] Vivek Myers, Bill Chunyuan Zheng, Oier Mees, Sergey Levine, and Kuan Fang. Policy Adaptation via Language Optimization: Decomposing Tasks for Few-Shot Imitation. In *Conference on Robot Learning*. 2024.
- [9] Vivek Myers, Chongyi Zheng, Anca Dragan, Sergey Levine, and Benjamin Eysenbach. Learning Temporal Distances: Contrastive Successor Features Can Provide a Metric Structure for Decision-Making. In *International Conference on Machine Learning*. 2024.
- [10] Vivek Myers, Bill Chunyuan Zheng, Anca Dragan, Kuan Fang, and Sergey Levine. Successor Representations Enable Emergent Compositional Instruction Following. 2024.

- [11] Benjamin Eysenbach, Vivek Myers, Ruslan Salakhutdinov, and Sergey Levine. Inference via Interpolation: Contrastive Representations Provably Enable Planning and Inference. In *Neural Information Processing Systems*. 2024.
- [12] Vivek Myers, Cathy Ji, and Benjamin Eysenbach. Invariance to Planning in Goal-Conditioned RL. 2024.
- [13] Vivek Myers, Evan Ellis, Sergey Levine, Benjamin Eysenbach, and Anca Dragan. Learning to Assist Humans Without Inferring Rewards. In *Neural Information Processing Systems*. 2024.
- [14] Marcin Andrychowicz, Filip Wolski, Alex Ray, Jonas Schneider, Rachel Fong, Peter Welinder, Bob McGrew, Josh Tobin, OpenAI Pieter Abbeel, and Wojciech Zaremba. Hindsight Experience Replay. In *Neural Information Processing Systems*, volume 30. 2017.
- [15] Dibya Ghosh, Abhishek Gupta, Ashwin Reddy, Justin Fu, Coline Manon Devin, Benjamin Eysenbach, and Sergey Levine. Learning to Reach Goals via Iterated Supervised Learning. 2020. OpenReview.net:arXiv:1912.06088.
- [16] Suraj Nair, Aravind Rajeswaran, Vikash Kumar, Chelsea Finn, and Abhinav Gupta. R3m: A Universal Visual Representation for Robot Manipulation. In *Conference on Robot Learning*, pp. 892–909. 2022.
- [17] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning Transferable Visual Models From Natural Language Supervision. In *International Conference on Machine Learning*, arXiv:2103.00020. 2021.
- [18] Terry Winograd. Procedures as a Representation for Data in a Computer Program for Understanding Natural Language. 1971.
- [19] Kai-yuh Hsiao, Stefanie Tellex, Soroush Vosoughi, Rony Kubat, and Deb Roy. Object Schemas for Grounding Language in a Responsive Robot. *Connection Science*, 20(4):253–276, 2008.
- [20] M. Skubic, D. Perzanowski, S. Blisard, A. Schultz, W. Adams, M. Bugajska, and D. Brock. Spatial Language for Human-Robot Dialogs. *IEEE Transactions on Systems*, 34(2):154–167, 2004.
- [21] Stefanie Tellex, Thomas Kollar, Steven Dickerson, Matthew Walter, Ashis Banerjee, Seth Teller, and Nicholas Roy. Understanding Natural Language Commands for Robotic Navigation and Mobile Manipulation. *AAAI Conference on Artificial Intelligence*, 25(1):1507–1514, 2011.
- [22] Simon Stepputtis, Joseph Campbell, Mariano Phielipp, Stefan Lee, Chitta Baral, and Heni Ben Amor. Language-Conditioned Imitation Learning for Robot Manipulation Tasks. *Neural Information Processing Systems*, 33:13139–13150, 2020.

- [23] Eric Jang, Alex Irpan, Mohi Khansari, Daniel Kappler, Frederik Ebert, Corey Lynch, Sergey Levine, and Chelsea Finn. BC-Z: Zero-Shot Task Generalization With Robotic Imitation Learning. *Conference on Robot Learning*, p. 12, 2021.
- [24] Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Joseph Dabis, Chelsea Finn, Keerthana Gopalakrishnan, Karol Hausman, et al. RT-1: Robotics Transformer for Real-World Control at Scale. In *Robotics - Science and Systems*. 2023.
- [25] Danny Driess, Fei Xia, Mehdi S. M. Sajjadi, Corey Lynch, Aakanksha Chowdhery, Brian Ichter, Ayzaan Wahid, Jonathan Tompson, et al. PaLM-E: An Embodied Multimodal Language Model. In *International Conference on Machine Learning*. 2023.
- [26] Yunfan Jiang, Agrim Gupta, Zichen Zhang, Guanzhi Wang, Yongqiang Dou, Yanjun Chen, Li Fei-Fei, Anima Anandkumar, Yuke Zhu, and Linxi Fan. VIMA: General Robot Manipulation With Multimodal Prompts. In *International Conference on Machine Learning*. 2023.
- [27] Kevin Lin, Christopher Agia, Toki Migimatsu, Marco Pavone, and Jeannette Bohg. Text2Motion: From Natural Language Instructions to Feasible Plans. 2023. arxiv:2303.12153.
- [28] Weiyu Liu, Chris Paxton, Tucker Hermans, and Dieter Fox. StructFormer: Learning Spatial Structure for Language-Guided Semantic Rearrangement of Novel Objects. *International Conference on Robotics and Automation*, pp. 6322–6329, 2021.
- [29] Mohit Shridhar, Lucas Manuelli, and Dieter Fox. CLIPort: What and Where Pathways for Robotic Manipulation. In *Conference on Robot Learning*. 2021.
- [30] Hongyuan Mei, Mohit Bansal, and Matthew Walter. Listen, Attend, and Walk: Neural Mapping of Navigational Instructions to Action Sequences. *AAAI Conference on Artificial Intelligence*, 30(1), 2016.
- [31] Peter Anderson, Qi Wu, Damien Teney, Jake Bruce, Mark Johnson, Niko Sünderhauf, Ian Reid, Stephen Gould, and Anton van den Hengel. Vision-And-Language Navigation: Interpreting Visually-Grounded Navigation Instructions in Real Environments. In *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3674–3683. 2018.
- [32] Daniel Fried, Ronghang Hu, Volkan Cirik, Anna Rohrbach, Jacob Andreas, Louis-Philippe Morency, Taylor Berg-Kirkpatrick, Kate Saenko, Dan Klein, and Trevor Darrell. Speaker-Follower Models for Vision-And-Language Navigation. In *Neural Information Processing Systems*, volume 31. 2018.
- [33] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language Models Are Few-Shot Learners. 2020.

- [34] Michael Ahn, Anthony Brohan, Noah Brown, Yevgen Chebotar, Omar Cortes, Byron David, Chelsea Finn, Chuyuan Fu, Keerthana Gopalakrishnan, Karol Hausman, et al. Do as I Can, Not as I Say: Grounding Language in Robotic Affordances. 2022. arxiv:2204.01691.
- [35] Maria Attarian, Advaya Gupta, Ziyi Zhou, Wei Yu, Igor Gilitschenski, and Animesh Garg. See, Plan, Predict: Language-Guided Cognitive Planning With Video Prediction. 2022. arXiv:2210.03825.
- [36] Mohit Shridhar, Jesse Thomason, Daniel Gordon, Yonatan Bisk, Winson Han, Roozbeh Mottaghi, Luke Zettlemoyer, and Dieter Fox. ALFRED: A Benchmark for Interpreting Grounded Instructions for Everyday Tasks. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10740–10749. 2020.
- [37] Jimmy Wu, Rika Antonova, Adam Kan, Marion Lepert, Andy Zeng, Shuran Song, Jeannette Bohg, Szymon Rusinkiewicz, and Thomas A. Funkhouser. TidyBot: Personalized Robot Assistance With Large Language Models. 2023. arXiv:2305.05658.
- [38] Ishika Singh, Valts Blukis, Arsalan Mousavian, Ankit Goyal, Danfei Xu, Jonathan Tremblay, Dieter Fox, Jesse Thomason, and Animesh Garg. Prog-Prompt: Generating Situated Robot Task Plans Using Large Language Models. In *International Conference on Robotics and Automation*. 2023.
- [39] Oier Mees, Lukas Hermann, Erick Rosete-Beas, and Wolfram Burgard. CALVIN: A Benchmark for Language-Conditioned Policy Learning for Long-Horizon Robot Manipulation Tasks. *IEEE Robotics and Automation Letters*, 7(3):7327–7334, 2022.
- [40] Jiahui Yu, Zirui Wang, Vijay Vasudevan, Legg Yeung, Mojtaba Seyedhosseini, and Yonghui Wu. CoCa: Contrastive Captioners Are Image-Text Foundation Models. 2022. arxiv:2205.01917.
- [41] Kristen Grauman, Andrew Westbury, Eugene Byrne, Zachary Chavis, Antonino Furnari, Rohit Girdhar, Jackson Hamburger, Hao Jiang, Miao Liu, Xingyu Liu, et al. Ego4D: Around the World in 3,000 Hours of Egocentric Video. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022.
- [42] Raghav Goyal, Samira Ebrahimi Kahou, Vincent Michalski, Joanna Materzyńska, Susanne Westphal, Heuna Kim, Valentin Haenel, Ingo Fruend, Peter Yianilos, Moritz Mueller-Freitag, et al. The "Something Something" Video Database for Learning and Evaluating Visual Common Sense. 2017. arxiv:1706.04261.
- [43] Ilija Radosavovic, Tete Xiao, Stephen James, P. Abbeel, Jitendra Malik, and Trevor Darrell. Real-World Robot Learning With Masked Visual Pre-Training. 2022. arXiv:2210.03109.



- [44] Linxi Fan, Guanzhi Wang, Yunfan Jiang, Ajay Mandlekar, Yuncong Yang, Haoyi Zhu, Andrew Tang, De-An Huang, Yuke Zhu, and Anima Anandkumar. MineDojo: Building Open-Ended Embodied Agents With Internet-Scale Knowledge. In *Neural Information Processing Systems*. 2022.
- [45] Lin Shao, Toki Migimatsu, Qiang Zhang, Karen Yang, and Jeannette Bohg. Concept2Robot: Learning Manipulation Concepts From Instructions and Human Demonstrations. *International Journal of Robotics Research*, 40(12-14):1419–1434, 2021.
- [46] Yecheng Jason Ma, William Liang, Vaidehi Som, Vikash Kumar, Amy Zhang, Osbert Bastani, and Dinesh Jayaraman. LIV: Language-Image Representations and Rewards for Robotic Control. In *International Conference on Machine Learning*. 2023.
- [47] Yuchen Cui, Scott Niekum, Abhinav Gupta, Vikash Kumar, and Aravind Rajeswaran. Can Foundation Models Perform Zero-Shot Task Specification for Robot Manipulation? In *L4DC*. 2022.
- [48] Suraj Nair, Eric Mitchell, Kevin Chen, Brian Ichter, Silvio Savarese, and Chelsea Finn. Learning Language-Conditioned Robot Behavior From Offline Data and Crowd-Sourced Annotation. In *Conference on Robot Learning*, pp. 1303–1315. 2022.
- [49] Leslie Pack Kaelbling. Learning to Achieve Goals. In *International Joint Conference on Artificial Intelligence*. 1993.
- [50] Tom Schaul, Daniel Horgan, Karol Gregor, and David Silver. Universal Value Function Approximators. In *International Conference on Machine Learning*, pp. 1312–1320. 2015.
- [51] Corey Lynch and Pierre Sermanet. Language Conditioned Imitation Learning Over Unstructured Data. In *Robotics: Science and Systems XVII*. 2021.
- [52] Zhuang Ma and Michael Collins. Noise Contrastive Estimation and Negative Sampling for Conditional Models: Consistency and Statistical Efficiency. 2018. arXiv:1809.01812.
- [53] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation Learning With Contrastive Predictive Coding. 2019. arXiv:1807.03748.
- [54] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual Instruction Tuning. 2023. arxiv:2304.08485.
- [55] Kenan Jiang, Xuehai He, Ruizhe Xu, and Xin Eric Wang. ComCLIP: Training-Free Compositional Image and Text Matching. 2022. arxiv:2211.13854.
- [56] Martha Lewis, Nihal V. Nayak, Peilin Yu, Qinan Yu, Jack Merullo, Stephen H. Bach, and Ellie Pavlick. Does CLIP Bind Concepts? Probing Compositionality in Large Image Models. In *Conference of the European Chapter of the Association for Computational Linguistics*. 2024.

- [57] Samir Yitzhak Gadre, Mitchell Wortsman, Gabriel Ilharco, Ludwig Schmidt, and Shuran Song. CoWs on Pasture: Baselines and Benchmarks for Language-Driven Zero-Shot Object Navigation. 2022. arxiv:2203.10421.
- [58] Yinfei Yang, Daniel Cer, Amin Ahmad, Mandy Guo, Jax Law, Noah Constant, Gustavo Hernandez Abrego, Steve Yuan, Chris Tar, Yun-Hsuan Sung, Brian Strope, and Ray Kurzweil. Multilingual Universal Sentence Encoder for Semantic Retrieval. In *ACL*. 2020.
- [59] Annie Xie, Lisa Lee, Ted Xiao, and Chelsea Finn. Decomposing the Generalization Gap in Imitation Learning for Visual Robotic Manipulation. In *International Conference on Robotics and Automation*. 2024.
- [60] Oier Mees, Lukas Hermann, and Wolfram Burgard. What Matters in Language Conditioned Robotic Imitation Learning Over Unstructured Data. *IEEE Robotics and Automation Letters*, 7(4):11205–11212, 2022.
- [61] Oier Mees and Wolfram Burgard. Composing Pick-And-Place Tasks by Grounding Language. In *International Symposium on Experimental Robotics*. 2021.
- [62] Homer Rich Walke, Kevin Black, Tony Z. Zhao, Quan Vuong, Chongyi Zheng, Philippe Hansen-Estruch, Andre Wang He, Vivek Myers, et al. BridgeData V2: A Dataset for Robot Learning at Scale. In *Conference on Robot Learning*, pp. 1723–1736. 2023.
- [63] Brenden M. Lake, Tal Linzen, and Marco Baroni. Human Few-Shot Learning of Compositional Instructions. In *CogSci*. 2019.
- [64] Kevin Ellis. Human-Like Few-Shot Learning via Bayesian Reasoning Over Natural Language. In *International Conference on Neural Information Processing Systems*. 2023.
- [65] Jessy Lin, Yuqing Du, Olivia Watkins, Danijar Hafner, Pieter Abbeel, Dan Klein, and Anca Dragan. Learning to Model the World With Language. In *Forty-First International Conference on Machine Learning*. 2024.
- [66] Andy Zeng, Maria Attarian, Brian Ichter, Krzysztof Choromanski, Adrian Wong, Stefan Welker, Federico Tombari, Aveek Purohit, Michael Ryoo, Vikas Sindhwani, et al. Socratic Models: Composing Zero-Shot Multimodal Reasoning With Language. 2022. arXiv:2204.00598.
- [67] Dibya Ghosh, Homer Walke, Karl Pertsch, Kevin Black, Oier Mees, Sudeep Dasari, Joey Hejna, Tobias Kreiman, et al. Octo: An Open-Source Generalist Robot Policy. In *Robotics: Science and Systems*. 2024.
- [68] Wei-Yu Chen, Yen-Cheng Liu, Zsolt Kira, Yu-Chiang Frank Wang, and Jia-Bin Huang. A Closer Look at Few-Shot Classification. In *International Conference on Learning Representations*. 2019.
- [69] Oriol Vinyals, Charles Blundell, Timothy Lillicrap, koray kavukcuoglu, and

- Daan Wierstra. Matching Networks for One Shot Learning. In *Neural Information Processing Systems*, volume 29. 2016.
- [70] Yisheng Song, Ting Wang, Puyu Cai, Subrota K. Mondal, and Jyoti Prakash Sahoo. A Comprehensive Survey of Few-Shot Learning: Evolution, Applications, Challenges, and Opportunities. *ACM Comput. Surv.*, 55(13s):271:1–271:40, 2023.
- [71] Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, et al. Flamingo: A Visual Language Model for Few-Shot Learning. In *Neural Information Processing Systems*. 2022.
- [72] Ali Ghadirzadeh, Xi Chen, Petra Poklukar, Chelsea Finn, Mårten Björkman, and Danica Kragic. Bayesian Meta-Learning for Few-Shot Policy Adaptation Across Robotic Platforms. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1274–1280. 2021.
- [73] Yurong Guo, Ruoyi Du, Yuan Dong, Timothy Hospedales, Yi-Zhe Song, and Zhanyu Ma. Task-Aware Adaptive Learning for Cross-Domain Few-Shot Learning. In *IEEE/CVF International Conference on Computer Vision*, pp. 1590–1599. 2023.
- [74] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks. In *International Conference on Machine Learning*. 2017.
- [75] Alex Nichol, Joshua Achiam, and John Schulman. On First-Order Meta-Learning Algorithms. 2018. arXiv:1803.02999.
- [76] Dong Chen, Lingfei Wu, Siliang Tang, Xiao Yun, Bo Long, and Yueting Zhuang. Robust Meta-Learning With Sampling Noise and Label Noise via Eigen-Reptile. In *International Conference on Machine Learning*, pp. 3662–3678. 2022.
- [77] Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical Networks for Few-Shot Learning. In *Neural Information Processing Systems*, volume 30. 2017.
- [78] Marcin Sendera, Jacek Tabor, Aleksandra Nowak, Andrzej Bedychaj, Massimiliano Patacchiola, Tomasz Trzcinski, Przemyslaw Spurek, and Maciej Zieba. Non-Gaussian Gaussian Processes for Few-Shot Regression. In *Neural Information Processing Systems*, volume 34, pp. 10285–10298. 2021.
- [79] Petru Tighineanu, Lukas Grossberger, Paul Baireuther, Kathrin Skubch, Stefan Falkner, Julia Vinogradska, and Felix Berkenkamp. Scalable Meta-Learning With Gaussian Processes. In *International Conference on Artificial Intelligence and Statistics*, pp. 1981–1989. 2024.
- [80] Ze Wang, Zichen Miao, Xiantong Zhen, and Qiang Qiu. Learning to Learn Dense Gaussian Processes for Few-Shot Learning. In *Neural Information Processing Systems*, volume 34, pp. 13230–13241. 2021.

- [81] Sang Michael Xie, Aditi Raghunathan, Percy Liang, and Tengyu Ma. An Explanation of In-Context Learning as Implicit Bayesian Inference. In *International Conference on Learning Representations*. 2022.
- [82] Qingxiu Dong, Lei Li, Damai Dai, Ce Zheng, Zhiyong Wu, Baobao Chang, Xu Sun, Jingjing Xu, Lei Li, and Zhifang Sui. A Survey for In-Context Learning. 2023. arXiv:2301.00234.
- [83] Jiaqiang Ye Zhu, Carla Gomez Cano, David Vazquez Bermudez, and Michal Drozdal. InCoRo: In-Context Learning for Robotics Control With Feedback Loops. 2024. arXiv:2402.05188.
- [84] Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Joseph Dabis, Chelsea Finn, Keerthana Gopalakrishnan, Karol Hausman, Alex Herzog, Jasmine Hsu, et al. RT-1: Robotics Transformer for Real-World Control at Scale. 2023. Curran Associates, Inc.:arXiv:2212.06817.
- [85] Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Xi Chen, Krzysztof Choromanski, Tianli Ding, Danny Driess, et al. RT-2: Vision-Language-Action Models Transfer Web Knowledge to Robotic Control. In *Conference on Robot Learning*. 2023.
- [86] Corey Lynch, Ayzaan Wahid, Jonathan Tompson, Tianli Ding, James Betker, Robert Baruch, Travis Armstrong, and Pete Florence. Interactive Language: Talking to Robots in Real Time. *IEEE Robotics and Automation Letters*, pp. 1–8, 2023.
- [87] Kevin Black, Mitsuhiro Nakamoto, Pranav Atreya, Homer Walke, Chelsea Finn, Aviral Kumar, and Sergey Levine. Zero-Shot Robotic Manipulation With Pre-Trained Image-Editing Diffusion Models. In *International Conference on Learning Representations*. 2024.
- [88] Fangchen Liu, Kuan Fang, Pieter Abbeel, and Sergey Levine. MOKA: Open-Vocabulary Robotic Manipulation Through Mark-Based Visual Prompting. In *Robotics: Science and Systems*. 2024.
- [89] Wenlong Huang, Chen Wang, Ruohan Zhang, Yunzhu Li, Jiajun Wu, and Li Fei-Fei. VoxPoser: Composable 3D Value Maps for Robotic Manipulation With Language Models. 2023. arXiv:2307.05973.
- [90] Michael Ahn, Anthony Brohan, Noah Brown, Yevgen Chebotar, Omar Cortes, Byron David, Chelsea Finn, Chuyuan Fu, et al. Do as I Can, Not as I Say: Grounding Language in Robotic Affordances. In *Conference on Robot Learning*. 2022.
- [91] Suneel Belkhale, Tianli Ding, Ted Xiao, Pierre Sermanet, Quon Vuong, Jonathan Tompson, Yevgen Chebotar, Debidatta Dwibedi, and Dorsa Sadigh. RT-H: Action Hierarchies Using Language. 2024. arXiv:2403.01823.
- [92] Lucy Xiaoyang Shi, Zheyuan Hu, Tony Z. Zhao, Archit Sharma, Karl Pertsch,

- Jianlan Luo, Sergey Levine, and Chelsea Finn. Yell at Your Robot: Improving On-the-Fly From Language Corrections. 2024. arXiv:2403.12910.
- [93] Oier Mees, Jessica Borja-Diaz, and Wolfram Burgard. Grounding Language With Visual Affordances Over Unstructured Data. In *IEEE International Conference on Robotics and Automation*. 2023.
- [94] OpenAI, Josh Achiam, Steven Adler, et al. GPT-4 Technical Report. 2024. arXiv:2303.08774.
- [95] Kristen Grauman, Andrew Westbury, Eugene Byrne, Zachary Chavis, Antonino Furnari, Rohit Girdhar, Jackson Hamburger, Hao Jiang, et al. Ego4D: Around the World in 3,000 Hours of Egocentric Video. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022.
- [96] Rishi Bommasani, Drew A. Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S. Bernstein, Jeannette Bohg, et al. On the Opportunities and Risks of Foundation Models. 2022. arXiv:2108.07258.
- [97] Rutav Shah, Roberto Martín-Martín, and Yuke Zhu. MUTEX: Learning Unified Policies From Multimodal Task Specifications. In *Conference on Robot Learning*. 2023.
- [98] Abby O’Neill, Abdul Rehman, Abhiram Maddukuri, Abhishek Gupta, Abhishek Padalkar, Abraham Lee, Acorn Pooley, Agrim Gupta, et al. Open X-Embodiment: Robotic Learning Datasets and RT-X Models. In *International Conference on Robotics and Automation*. 2024.
- [99] Ria Doshi, Homer Walke, Oier Mees, Sudeep Dasari, and Sergey Levine. Scaling Cross-Embodied Learning: One Policy for Manipulation, Navigation, Locomotion and Aviation. 2024. arXiv:2408.11812.
- [100] Tianhe Yu, Ted Xiao, Austin Stone, Jonathan Tompson, Anthony Brohan, Su Wang, Jaspiar Singh, Clayton Tan, Dee M, Jodilyn Peralta, et al. Scaling Robot Learning With Semantically Imagined Experience. 2023. arxiv:2302.11550.
- [101] Qiuyu Chen, Shosuke Kiami, Abhishek Gupta, and Vikash Kumar. GenAug: Retargeting Behaviors to Unseen Situations via Generative Augmentation. In *Robotics: Science and Systems XIX*. 2023.
- [102] Chenguang Huang, Oier Mees, Andy Zeng, and Wolfram Burgard. Audio Visual Language Maps for Robot Navigation. In *International Symposium on Experimental Robotics*. 2023.
- [103] Jesse Zhang, Karl Pertsch, Jiahui Zhang, and Joseph J. Lim. SPRINT: Scalable Policy Pre-Training via Language Instruction Relabeling. 2023. arXiv:2306.11886.
- [104] William Chen, Oier Mees, Aviral Kumar, and Sergey Levine. Vision-Language Models Provide Promptable Representations for Reinforcement Learning. 2024. arXiv:2402.02651.

- [105] Dhruv Shah, Blazej Osinski, Brian Ichter, and Sergey Levine. LM-Nav: Robotic Navigation With Large Pre-Trained Models of Language, Vision, and Action. In *Conference on Robot Learning*. 2022.
- [106] Chenguang Huang, Oier Mees, Andy Zeng, and Wolfram Burgard. Visual Language Maps for Robot Navigation. In *IEEE International Conference on Robotics and Automation*. 2023.
- [107] Minae Kwon, Hengyuan Hu, Vivek Myers, Siddharth Karamcheti, A. Dragan, and Dorsa Sadigh. Toward Grounded Commonsense Reasoning. In *International Conference on Robotics and Automation*. 2023.
- [108] O. Catoni. A PAC-Bayesian Approach to Adaptive Classification. 2004.
- [109] Pierre Alquier. User-Friendly Introduction to PAC-Bayes Bounds. *Foundations and Trends in Machine Learning*, 17(2):174–303, 2024.
- [110] L. G. Valiant. A Theory of the Learnable. *Commun. ACM*, 27(11):1134–1142, 1984.
- [111] Imre Csiszár and János Körner. *Information Theory: Coding Theorems for Discrete Memoryless Systems*. Cambridge University Press, 2011.
- [112] Wassily Hoeffding. Probability Inequalities for Sums of Bounded Random Variables. *Journal of the American Statistical Association*, 58(301):13–30, 1963.
- [113] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. In *Computer Vision and Pattern Recognition*. 2016.
- [114] Ethan Perez, Florian Strub, Harm de Vries, Vincent Dumoulin, and Aaron Courville. FiLM: Visual Reasoning With a General Conditioning Layer. In *AAAI Conference on Artificial Intelligence*. 2018.
- [115] Jyothish Pari, Nur Muhammad (Mahi) Shafiullah, Sridhar Pandian Arunachalam, and Lerrel Pinto. The Surprising Effectiveness of Representation Learning for Visual Imitation. In *Robotics: Science and Systems XVIII*. 2022.
- [116] Srinivas Venkattaramanujam, Eric Crawford, Thang Doan, and Doina Precup. Self-Supervised Learning of Distance Functions for Goal-Conditioned Reinforcement Learning. 2019. arXiv:1907.02998.
- [117] Nikolay Savinov, Alexey Dosovitskiy, and Vladlen Koltun. Semi-Parametric Topological Memory for Navigation. In *International Conference on Learning Representations*. 2018.
- [118] Ishan Durugkar, Mauricio Tec, Scott Niekum, and Peter Stone. Adversarial Intrinsic Motivation for Reinforcement Learning. In *Neural Information Processing Systems*. 2021.
- [119] Yecheng Jason Ma, Shagun Sodhani, Dinesh Jayaraman, Osbert Bastani, Vikash Kumar, and Amy Zhang. VIP: Towards Universal Visual Reward and Representation via Value-Implicit Pre-Training. In *International Conference on Learning Representations*. 2023.

- [120] Kristian Hartikainen, Xinyang Geng, Tuomas Haarnoja, and Sergey Levine. Dynamical Distance Learning for Semi-Supervised and Unsupervised Skill Discovery. In *International Conference on Learning Representations*. 2020.
- [121] Benjamin Eysenbach, Tianjun Zhang, Sergey Levine, and Russ R Salakhutdinov. Contrastive Learning as Goal-Conditioned Reinforcement Learning. *Neural Information Processing Systems*, 35:35603–35620, 2022.
- [122] Benjamin Eysenbach, Vivek Myers, Sergey Levine, and Ruslan Salakhutdinov. Contrastive Representations Make Planning Easy. In *NeurIPS 2023 Workshop on Generalization in Planning*. 2023.
- [123] K. R. Popper. The Arrow of Time. *Nature*, 177(4507):538–538, 1956.
- [124] Peter Dayan. Improving Generalization for Temporal Difference Learning: The Successor Representation. *Neural Computation*, 5(4):613–624, 1993.
- [125] Max Schwarzer, Ankesh Anand, Rishab Goel, R Devon Hjelm, Aaron Courville, and Philip Bachman. Data-Efficient Reinforcement Learning With Self-Predictive Representations. In *International Conference on Learning Representations*. 2021.
- [126] Tongzhou Wang, Antonio Torralba, Phillip Isola, and Amy Zhang. Optimal Goal-Reaching Reinforcement Learning via Quasimetric Learning. In *International Conference on Machine Learning*. 2023.
- [127] Bo Liu, Yihao Feng, Qiang Liu, and Peter Stone. Metric Residual Network for Sample Efficient Goal-Conditioned Reinforcement Learning. In *AAAI Conference on Artificial Intelligence*, volume 37, pp. 8799–8806. 2023.
- [128] Kihyuk Sohn. Improved Deep Metric Learning With Multi-Class N-Pair Loss Objective. In *Neural Information Processing Systems*, volume 29. 2016.
- [129] Ben Poole, Sherjil Ozair, Aaron van den Oord, Alexander A. Alemi, and George Tucker. On Variational Bounds of Mutual Information. 2019. arxiv:1905.06922.
- [130] Tongzhou Wang and Phillip Isola. Improved Representation of Asymmetrical Distances With Interval Quasimetric Embeddings. In *NeurIPS 2022 NeurReps Workshop Proceedings Track*. 2022.
- [131] Scott Fujimoto and Shixiang Shane Gu. A Minimalist Approach to Offline Reinforcement Learning. *Neural Information Processing Systems*, 34:20132–20145, 2021.
- [132] Aviral Kumar, Aurick Zhou, George Tucker, and Sergey Levine. Conservative Q-Learning for Offline Reinforcement Learning. In *Neural Information Processing Systems*, volume 33, pp. 1179–1191. 2020.
- [133] Aviral Kumar, Justin Fu, George Tucker, and Sergey Levine. Stabilizing Off-Policy Q-Learning via Bootstrapping Error Reduction. 2019. arXiv:1906.00949.

- [134] Chongyi Zheng, Ruslan Salakhutdinov, and Benjamin Eysenbach. Contrastive Difference Predictive Coding. In *International Conference on Learning Representations*. 2023.
- [135] Ashvin Nair, Abhishek Gupta, Murtaza Dalal, and Sergey Levine. AWAC: Accelerating Online Reinforcement Learning With Offline Datasets. 2021. arxiv:2006.09359.
- [136] Timothy P Lillicrap, Jonathan J Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. Continuous Control With Deep Reinforcement Learning. In *International Conference on Learning Representations*. 2016.
- [137] Xingyu Lin, Harjatin Singh Baweja, and David Held. Reinforcement Learning Without Ground-Truth State. 2019. arXiv:1905.07866.
- [138] Gerhard Neumann and Jan Peters. Fitted Q-Iteration by Advantage Weighted Regression. *Neural Information Processing Systems*, 21, 2008.
- [139] Jan Peters and Stefan Schaal. Reinforcement Learning by Reward-Weighted Regression for Operational Space Control. In *International Conference on Machine Learning*, pp. 745–750. 2007.
- [140] Wenhao Ding, Haohong Lin, Bo Li, and Ding Zhao. Generalizing Goal-Conditioned Reinforcement Learning With Variational Causal Reasoning. 2023. arxiv:2207.09081.
- [141] Lili Chen, Kevin Lu, Aravind Rajeswaran, Kimin Lee, Aditya Grover, Misha Laskin, Pieter Abbeel, Aravind Srinivas, and Igor Mordatch. Decision Transformer: Reinforcement Learning via Sequence Modeling. In *Neural Information Processing Systems*, volume 34, pp. 15084–15097. 2021.
- [142] Xinlei Chen and Kaiming He. Exploring Simple Siamese Representation Learning. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 15750–15758. 2021.
- [143] Tongzhou Wang and Phillip Isola. On the Learning and Learnability of Quasimetrics. In *International Conference on Learning Representations*. 2022.
- [144] Ilya Kostrikov, Ashvin Nair, and Sergey Levine. Offline Reinforcement Learning With Implicit Q-Learning. In *International Conference on Learning Representations*. 2022.
- [145] Kurt Hornik, Maxwell Stinchcombe, and Halbert White. Multilayer Feedforward Networks Are Universal Approximators. *Neural Networks*, 2(5):359–366, 1989.
- [146] Maciej Paluszyński and Krzysztof Stempak. On Quasi-Metric and Metric Spaces. *American Mathematical Society*, 137:4307–4312, 1931.
- [147] Philippe Hansen-Estruch, Amy Zhang, Ashvin Nair, Patrick Yin, and Sergey Levine. Bisimulation Makes Analogies in Goal-Conditioned Reinforcement Learning. In *International Conference on Machine Learning*, pp. 8407–8426. 2022.



- [148] Norm Ferns, Prakash Panangaden, and Doina Precup. Bisimulation Metrics for Continuous Markov Decision Processes. *SIAM Journal on Computing*, 40(6):1662–1714, 2011.
- [149] Steve N’Guyen, Clément Moulin-Frier, and Jacques Droulez. Decision Making Under Uncertainty: A Quasimetric Approach. *Plos ONE*, 8(12):e83411, 2013.
- [150] Charline Le Lan, Marc G. Bellemare, and Pablo Samuel Castro. Metrics and Continuity in Reinforcement Learning. *AAAI Conference on Artificial Intelligence*, 35(9):8261–8269, 2021.
- [151] Joey Hejna, Jensen Gao, and Dorsa Sadigh. Distance Weighted Supervised Learning for Offline Interaction Data. In *International Conference on Machine Learning*, pp. 12882–12906. 2023.
- [152] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation Learning With Contrastive Predictive Coding. 2018. arXiv:1807.03748.
- [153] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed Representations of Words and Phrases and Their Compositionality. In *Neural Information Processing Systems*, volume 26. 2013.
- [154] Bogdan Mazoure, Remi Tachet des Combes, Thang Long Doan, Philip Bachman, and R Devon Hjelm. Deep Reinforcement and InfoMax Learning. In *Neural Information Processing Systems*, volume 33, pp. 3686–3698. 2020.
- [155] Chuhan Wu, Fangzhao Wu, and Yongfeng Huang. Rethinking InfoNCE: How Many Negative Samples Do You Need? 2021. arXiv:2105.13003.
- [156] Michael Gutmann and Aapo Hyvärinen. Noise-Contrastive Estimation: A New Estimation Principle for Unnormalized Statistical Models. In *Thirteenth International Conference on Artificial Intelligence and Statistics*, pp. 297–304. 2010.
- [157] Pierre Sermanet, Corey Lynch, Yevgen Chebotar, Jasmine Hsu, Eric Jang, Stefan Schaal, Sergey Levine, and Google Brain. Time-Contrastive Networks: Self-Supervised Learning From Video. In *IEEE International Conference on Robotics and Automation*, pp. 1134–1141. 2018.
- [158] Rui Qian, Tianjian Meng, Boqing Gong, Ming-Hsuan Yang, Huisheng Wang, Serge Belongie, and Yin Cui. Spatiotemporal Contrastive Video Representation Learning. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6964–6974. 2021.
- [159] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A Simple Framework for Contrastive Learning of Visual Representations. In *International Conference on Machine Learning*. 2020.
- [160] Ting Chen, Calvin Luo, and Lala Li. Intriguing Properties of Contrastive Losses. In *Neural Information Processing Systems*, volume 34, pp. 11834–11845. 2021.

- [161] Tongzhou Wang and Phillip Isola. Understanding Contrastive Representation Learning Through Alignment and Uniformity on the Hypersphere. In *International Conference on Machine Learning*, pp. 9929–9939. 2020.
- [162] Andre Barreto, Diana Borsa, John Quan, Tom Schaul, David Silver, Matteo Hessel, Daniel Mankowitz, Augustin Zidek, and Remi Munos. Transfer in Deep Reinforcement Learning Using Successor Features and Generalised Policy Improvement. In *International Conference on Machine Learning*, pp. 501–510. 2018.
- [163] Chongyi Zheng, Benjamin Eysenbach, Homer Walke, Patrick Yin, Kuan Fang, Ruslan Salakhutdinov, and Sergey Levine. Stabilizing Contrastive RL: Techniques for Robotic Goal Reaching From Offline Data. In *International Conference on Learning Representations*. 2024.
- [164] John E Laird, Allen Newell, and Paul S Rosenbloom. Soar: An Architecture for General Intelligence. *Artificial Intelligence*, 33(1):1–64, 1987.
- [165] Seohong Park, Dibya Ghosh, Benjamin Eysenbach, and Sergey Levine. HIQL: Offline Goal-Conditioned RL With Latent States as Actions. In *Neural Information Processing Systems*. 2023.
- [166] Rui Yang, Lin Yong, Xiaoteng Ma, Hao Hu, Chongjie Zhang, and Tong Zhang. What Is Essential for Unseen Goal Generalization of Offline Goal-Conditioned RL? In *International Conference on Machine Learning*, pp. 39543–39571. 2023.
- [167] Rui Yang, Yiming Lu, Wenzhe Li, Hao Sun, Meng Fang, Yali Du, Xiu Li, Lei Han, and Chongjie Zhang. Rethinking Goal-Conditioned Supervised Learning and Its Connection to Offline RL. In *International Conference on Learning Representations*. 2022.
- [168] Benjamin Eysenbach, Ruslan Salakhutdinov, and Sergey Levine. C-Learning: Learning to Achieve Goals via Recursive Classification. In *International Conference on Learning Representations*. 2021.
- [169] Dibya Ghosh, Chethan Bhateja, and Sergey Levine. Reinforcement Learning From Passive Data via Latent Intentions. 2023. arxiv:2304.04782.
- [170] Kuan Fang, Patrick Yin, Ashvin Nair, Homer Walke, Gengchen Yan, and Sergey Levine. Generalization With Lossy Affordances: Leveraging Broad Offline Data for Learning Visuomotor Tasks. In *Conference on Robot Learning*. 2022.
- [171] Elliot Chane-Sane, Cordelia Schmid, and Ivan Laptev. Goal-Conditioned Reinforcement Learning With Imagined Subgoals. In *International Conference on Machine Learning*, pp. 1430–1440. 2021.
- [172] Russell Mendonca, Oleh Rybkin, Kostas Daniilidis, Danijar Hafner, and Deepak Pathak. Discovering and Achieving Goals via World Models. In *Neural Information Processing Systems*, volume 34, pp. 24379–24391. 2021.

- [173] Seohong Park, Oleh Rybkin, and Sergey Levine. METRA: Scalable Unsupervised RL With Metric-Aware Abstraction. In *International Conference on Learning Representations*. 2024.
- [174] Ahmed Touati and Yann Ollivier. Learning One Representation to Optimize All Rewards. In *Neural Information Processing Systems*, volume 34, pp. 13–23. 2021.
- [175] K. S. Lashley. The Problem of Serial Order in Behavior. In *Cerebral Mechanisms in Behavior*, pp. 112–136. 1951.
- [176] Tejas D Kulkarni, Karthik Narasimhan, Ardavan Saeedi, and Josh Tenenbaum. Hierarchical Deep Reinforcement Learning: Integrating Temporal Abstraction and Intrinsic Motivation. In *Neural Information Processing Systems*, volume 29. 2016.
- [177] Julian Schrittwieser, Thomas Hubert, Amol Mandhane, Mohammadamin Barekatin, Ioannis Antonoglou, and David Silver. Online and Offline Reinforcement Learning by Planning With a Learned Model. In *Neural Information Processing Systems*, volume 34, pp. 27580–27591. 2021.
- [178] Cassidy Laidlaw, Banghua Zhu, Stuart Russell, and Anca Dragan. The Effective Horizon Explains Deep RL Performance in Stochastic Environments. In *International Conference on Learning Representations*. 2024.
- [179] Simon Ciranka, Juan Linde-Domingo, Ivan Padezhki, Clara Wicharz, Charley M. Wu, and Bernhard Spitzer. Asymmetric Reinforcement Learning Facilitates Human Inference of Transitive Relations. *Nature Human Behaviour*, 6(4):555–564, 2022.
- [180] Alison Gopnik, Shaun O’Grady, Christopher G. Lucas, Thomas L. Griffiths, Adrienne Wente, Sophie Bridgers, Rosie Aboody, Hoki Fung, and Ronald E. Dahl. Changes in Cognitive Flexibility and Hypothesis Search Across Human Life History From Childhood to Adolescence to Adulthood. *National Academy of Sciences*, 114(30):7892–7899, 2017.
- [181] Oliver M. Vikbladh, Michael R. Meager, John King, Karen Blackmon, Orrin Devinsky, Daphna Shohamy, Neil Burgess, and Nathaniel D. Daw. Hippocampal Contributions to Model-Based Planning and Spatial Memory. *Neuron*, 102(3):683–693, 2019.
- [182] Xue Bin Peng, Aviral Kumar, Grace Zhang, and Sergey Levine. Advantage-Weighted Regression: Simple and Scalable Off-Policy Reinforcement Learning. 2019. arXiv:1910.00177.
- [183] Aviral Kumar, Joey Hong, Anikait Singh, and Sergey Levine. Should I Run Offline Reinforcement Learning or Behavioral Cloning? In *International Conference on Learning Representations*. 2021.
- [184] Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gau-

- rav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, et al. PaLM: Scaling Language Modeling With Pathways. In *J. Mach. Learn. Res.* 2023.
- [185] Yevgen Chebotar, Quan Vuong, Karol Hausman, Fei Xia, Yao Lu, Alex Irpan, Aviral Kumar, Tianhe Yu, et al. Q-Transformer: Scalable Offline Reinforcement Learning via Autoregressive Q-Functions. In *Conference on Robot Learning*. 2023.
- [186] Andreea Bobu, Yi Liu, Rohin Shah, Daniel S. Brown, and Anca D. Dragan. SIREL: Similarity-Based Implicit Representation Learning. In *ACM/IEEE International Conference on Human-Robot Interaction*, pp. 565–574. 2023.
- [187] Yuchen Cui, Siddharth Karamcheti, Raj Palleeti, Nidhya Shivakumar, Percy Liang, and Dorsa Sadigh. No, to the Right: Online Language Corrections for Robotic Manipulation via Shared Autonomy. In *ACM/IEEE International Conference on Human-Robot Interaction*, pp. 93–101. 2023.
- [188] Stanislas Dehaene, Fosca Al Roumi, Yair Lakretz, Samuel Planton, and Mathias Sablé-Meyer. Symbols and Mental Programs: A Hypothesis About Human Singularity. *Trends in Cognitive Sciences*, 26(9):751–766, 2022.
- [189] Yiming Ding, Carlos Florensa, Pieter Abbeel, and Mariano Phielipp. Goal-Conditioned Imitation Learning. *Neural Information Processing Systems*, 32, 2019.
- [190] Valerio Rubino, Mani Hamidi, Peter Dayan, and Charley M. Wu. Compositionality Under Time Pressure. In *Cognitive Science Society*, volume 45. 2023.
- [191] Mark Steedman. Where Does Compositionality Come From? In *AAAI Technical Report*. 2004.
- [192] David W. Dickins. Transitive Inference in Stimulus Equivalence and Serial Learning. *European Journal of Behavior Analysis*, 12(2):523–555, 2011.
- [193] Ekin Akyürek, Afra Feyza Akyürek, and Jacob Andreas. Learning to Recombine and Resample Data for Compositional Generalization. In *International Conference on Learning Representations*. 2021.
- [194] Takuya Ito, Tim Klinger, Doug Schultz, John Murray, Michael Cole, and Mattia Rigotti. Compositional Generalization Through Abstract Representations in Human and Artificial Neural Networks. *Neural Information Processing Systems*, 35:32225–32239, 2022.
- [195] Aviral Kumar, Anikait Singh, Frederik Ebert, Yanlai Yang, Chelsea Finn, and Sergey Levine. Pre-Training for Robots: Offline RL Enables Learning New Tasks From a Handful of Trials. 2022. arXiv:2210.05178.
- [196] Kuan Fang, Yuke Zhu, Animesh Garg, Silvio Savarese, and Li Fei-Fei. Dynamics Learning With Cascaded Variational Inference for Multi-Step Manipulation. *Conference on Robot Learning*, 2019.

- [197] Ajay Mandlekar, Danfei Xu, Roberto Martín-Martín, Silvio Savarese, and Li Fei-Fei. Learning to Generalize Across Long-Horizon Tasks From Human Demonstrations. 2021. arXiv:2003.06085.
- [198] Soroush Nasiriany, Vitchyr Pong, Steven Lin, and Sergey Levine. Planning With Goal-Conditioned Policies. *Neural Information Processing Systems*, 32, 2019.
- [199] Kuan Fang, Patrick Yin, Ashvin Nair, and Sergey Levine. Planning to Practice: Efficient Online Fine-Tuning by Composing Goals in Latent Space. In *International Conference on Intelligent Robots and Systems*. 2022.
- [200] Tianjun Zhang, Benjamin Eysenbach, Ruslan Salakhutdinov, Sergey Levine, and Joseph E Gonzalez. C-Planning: An Automatic Curriculum for Learning Goal-Reaching Tasks. In *International Conference on Learning Representations*. 2022.
- [201] Zichen Zhang, Yunshuang Li, Osbert Bastani, Abhishek Gupta, Dinesh Jayaraman, Yecheng Jason Ma, and Luca Weihs. Universal Visual Decomposer: Long-Horizon Manipulation Made Easy. 2023. arXiv:2310.08581.
- [202] Siddharth Karamcheti, Suraj Nair, Annie S. Chen, Thomas Kollar, Chelsea Finn, Dorsa Sadigh, and Percy Liang. Language-Driven Representation Learning for Robotics. In *Robotics - Science and Systems*. 2023.
- [203] Liunian Harold Li, Pengchuan Zhang, Haotian Zhang, Jianwei Yang, Chunyuan Li, Yiwu Zhong, Lijuan Wang, Lu Yuan, Lei Zhang, Jenq-Neng Hwang, Kai-Wei Chang, and Jianfeng Gao. Grounded Language-Image Pre-Training. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10955–10965. 2022.
- [204] Rutav Shah and Vikash Kumar. RRL: Resnet as Representation for Reinforcement Learning. In *International Conference on Machine Learning*. 2021.
- [205] Ankesh Anand, Evan Racah, Sherjil Ozair, Yoshua Bengio, Marc-Alexandre Côté, and R. Devon Hjelm. Unsupervised State Representation Learning in Atari. In *Neural Information Processing Systems*. 2019.
- [206] Amy Zhang, Rowan McAllister, Roberto Calandra, Yarin Gal, and Sergey Levine. Learning Invariant Representations for Reinforcement Learning Without Reconstruction. In *International Conference on Learning Representations*. 2021.
- [207] André Barreto, Will Dabney, Rémi Munos, Jonathan J Hunt, Tom Schaul, Hado P van Hasselt, and David Silver. Successor Features for Transfer in Reinforcement Learning. In *Neural Information Processing Systems*, volume 30. 2017.
- [208] Léonard Blier, Corentin Tallec, and Yann Ollivier. Learning Successor States and Goal-Dependent Values: A Mathematical Viewpoint. 2021. arXiv:2101.07123.

- [209] Peter Dayan. Improving Generalisation for Temporal Difference Learning: The Successor Representation. *Neural Computation*, 1993.
- [210] Alexey Dosovitskiy and Vladlen Koltun. Learning to Act by Predicting the Future. In *International Conference on Learning Representations*. 2017.
- [211] Jongwook Choi, Archit Sharma, Honglak Lee, Sergey Levine, and Shixiang Shane Gu. Variational Empowerment as Representation Learning for Goal-Conditioned Reinforcement Learning. In *International Conference on Machine Learning*, pp. 1953–1963. 2021.
- [212] Danijar Hafner, Timothy Lillicrap, Ian Fischer, Ruben Villegas, David Ha, Honglak Lee, and James Davidson. Learning Latent Dynamics for Planning From Pixels. 2018. arXiv:1811.04551.
- [213] Evangelos Theodorou, Jonas Buchli, and Stefan Schaal. A Generalized Path Integral Control Approach to Reinforcement Learning. *Journal of Machine Learning Research*, 11:3137–3181, 2010.
- [214] Hannes Jónsson, Greg Mills, and Karsten W Jacobsen. Nudged Elastic Band Method for Finding Minimum Energy Paths of Transitions. In *Classical and Quantum Dynamics in Condensed Phase Simulations*, pp. 385–404. World Scientific, 1998.
- [215] Wouter Saelens, Robrecht Cannoodt, Helena Todorov, and Yvan Saeys. A Comparison of Single-Cell Trajectory Inference Methods. *Nature Biotechnology*, 37(5):547–554, 2019.
- [216] Steven R Majewski, Ricardo P Schiavon, Peter M Frinchaboy, Carlos Allende Prieto, Robert Barkhouser, Dmitry Bizyaev, Basil Blank, Sophia Brunner, Adam Burton, Ricardo Carrera, et al. The Apache Point Observatory Galactic Evolution Experiment (APOGEE). *Astronomical Journal*, 154(3):94, 2017.
- [217] Shengjia Zhao, Jiaming Song, and Stefano Ermon. Towards Deeper Understanding of Variational Autoencoding Models. 2017. arXiv:1702.08658.
- [218] Yizhe Zhu, Martin Renqiang Min, Asim Kadav, and Hans Peter Graf. S3vae: Self-Supervised Sequential VAE for Representation Disentanglement and Data Generation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6538–6547. 2020.
- [219] Alireza Makhzani, Jonathon Shlens, Navdeep Jaitly, Ian Goodfellow, and Brendan Frey. Adversarial Autoencoders. 2015. arXiv:1511.05644.
- [220] Vincent Dumoulin, Ishmael Belghazi, Ben Poole, Alex Lamb, Martin Arjovsky, Olivier Mastropietro, and Aaron Courville. Adversarially Learned Inference. In *International Conference on Learning Representations*. 2016.
- [221] Ali Razavi, Aaron Van den Oord, and Oriol Vinyals. Generating Diverse High-Fidelity Images With Vq-VAE-2. *Neural Information Processing Systems*, 32, 2019.

- [222] Xinlei Chen, Haoqi Fan, Ross Girshick, and Kaiming He. Improved Baselines With Momentum Contrastive Learning. 2020. arXiv:2003.04297.
- [223] Yonglong Tian, Dilip Krishnan, and Phillip Isola. Contrastive Multiview Coding. In *Computer Vision—ECCV 2020: 16th European Conference*, pp. 776–794. 2020.
- [224] Zhirong Wu, Yuanjun Xiong, Stella X Yu, and Dahua Lin. Unsupervised Feature Learning via Non-Parametric Instance Discrimination. In *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3733–3742. 2018.
- [225] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient Estimation of Word Representations in Vector Space. 2013. arXiv:1301.3781.
- [226] Matthew Botvinick and Marc Toussaint. Planning as Inference. *Trends in Cognitive Sciences*, 16(10):485–488, 2012.
- [227] Hagai Attias. Planning by Probabilistic Inference. In *International Workshop on Artificial Intelligence and Statistics*, pp. 9–16. 2003.
- [228] George E Uhlenbeck and Leonard S Ornstein. On the Theory of the Brownian Motion. *Physical Review*, 36(5):823, 1930.
- [229] George EP Box, Gwilym M Jenkins, Gregory C Reinsel, and Greta M Ljung. *Time Series Analysis: Forecasting and Control*. John Wiley & Sons, 2015.
- [230] Dmitry M Malioutov, Jason K Johnson, and Alan S Willsky. Walk-Sums and Belief Propagation in Gaussian Graphical Models. *Journal of Machine Learning Research*, 7:2031–2064, 2006.
- [231] Yair Weiss and William Freeman. Correctness of Belief Propagation in Gaussian Graphical Models of Arbitrary Topology. *Neural Information Processing Systems*, 12, 1999.
- [232] Sanjeev Arora, Hrishikesh Khandeparkar, Mikhail Khodak, Orestis Plevrakis, and Nikunj Saunshi. A Theoretical Analysis of Contrastive Unsupervised Representation Learning. 2019. arXiv:1902.09229.
- [233] Claude Elwood Shannon. A Mathematical Theory of Communication. *Bell System Technical Journal*, 27(3):379–423, 1948.
- [234] E. T. Jaynes. Information Theory and Statistical Mechanics. *Physical Review*, 106(4):620–630, 1957.
- [235] Keith Conrad. Probability Distributions and Maximum Likelihood. 2010.
- [236] Ben Poole, Sherjil Ozair, Aaron Van Den Oord, Alex Alemi, and George Tucker. On Variational Bounds of Mutual Information. In *International Conference on Machine Learning*. 2019.
- [237] Nick Higham. What Is the Second Difference Matrix? <https://nhigham.com/2022/01/31/what-is-the-second-difference-matrix/>, 2022.

- [238] Morris Newman and John Todd. The Evaluation of Matrix Inversion Programs. *Journal of the Society for Industrial and Applied Mathematics*, 6(4):466–476, 1958.
- [239] Seongmin Park and Jihwa Lee. Finetuning Pretrained Transformers Into Variational Autoencoders. 2021. arXiv:2108.02446.
- [240] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-Training of Deep Bidirectional Transformers for Language Understanding. 2019. arXiv:1810.04805.
- [241] Micah Carroll, Orr Paradise, Jessy Lin, Raluca Georgescu, Mingfei Sun, David Bignell, Stephanie Milani, Katja Hofmann, Matthew Hausknecht, Anca Dragan, et al. UniMASK: Unified Inference in Sequential Decision Problems. 2022. arXiv:2211.10869.
- [242] Junyoung Chung, Kyle Kastner, Laurent Dinh, Kratarth Goel, Aaron C Courville, and Yoshua Bengio. A Recurrent Latent Variable Model for Sequential Data. In *Neural Information Processing Systems*, volume 28. 2015.
- [243] Rui Shu, Tung Nguyen, Yinlam Chow, Tuan Pham, Khoat Than, Mohammad Ghavamzadeh, Stefano Ermon, and Hung Bui. Predictive Coding for Locally-Linear Control. In *International Conference on Machine Learning*, pp. 8862–8871. 2020.
- [244] Manuel Watter, Jost Springenberg, Joschka Boedecker, and Martin Riedmiller. Embed to Control: A Locally Linear Latent Dynamics Model for Control From Raw Images. *Neural Information Processing Systems*, 28, 2015.
- [245] Ershad Banijamali, Rui Shu, Hung Bui, Ali Ghodsi, et al. Robust Locally-Linear Controllable Embedding. In *International Conference on Artificial Intelligence and Statistics*, pp. 1751–1759. 2018.
- [246] Brandon Cui, Yinlam Chow, and Mohammad Ghavamzadeh. Control-Aware Representations for Model-Based Reinforcement Learning. In *International Conference on Learning Representations*. 2020.
- [247] Tung D Nguyen, Rui Shu, Tuan Pham, Hung Bui, and Stefano Ermon. Temporal Predictive Coding for Model-Based Planning in Latent Space. In *International Conference on Machine Learning*, pp. 8130–8139. 2021.
- [248] Tung Nguyen, Rui Shu, Tuan Pham, Hung Bui, and Stefano Ermon. Non-Markovian Predictive Coding for Planning in Latent Space. 2020.
- [249] Ross Goroshin, Michael F Mathieu, and Yann LeCun. Learning to Linearize Under Uncertainty. *Neural Information Processing Systems*, 28, 2015.
- [250] Dinesh Jayaraman and Kristen Grauman. Slow and Steady Feature Analysis: Higher Order Temporal Coherence in Video. In *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3852–3861. 2016.
- [251] Mengda Xu, Zhenjia Xu, Cheng Chi, Manuela Veloso, and Shuran Song. XSkill: Cross Embodiment Skill Discovery. 2023. arXiv:2307.09955.



- [252] Laurenz Wiskott and Terrence J. Sejnowski. Slow Feature Analysis: Unsupervised Learning of Invariances. *Neural Computation*, 14(4):715–770, 2002.
- [253] Jia-Wei Yan, Ci-Siang Lin, Fu-En Yang, Yu-Jhe Li, and Yu-Chiang Frank Wang. Semantics-Guided Representation Learning With Applications to Visual Synthesis. In *International Conference on Pattern Recognition*, pp. 7181–7187. 2021.
- [254] Alon Oring, Zohar Yakhini, and Yacov Hel-Or. Autoencoder Image Interpolation by Shaping the Latent Space. 2020. arXiv:2008.01487.
- [255] Ying-Cong Chen, Xiaogang Xu, Zhuotao Tian, and Jiaya Jia. Homomorphic Latent Space Interpolation for Unpaired Image-to-Image Translation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2403–2411. 2019.
- [256] Xiaofeng Liu, Yang Zou, Lingsheng Kong, Zhihui Diao, Junliang Yan, Jun Wang, Site Li, Ping Jia, and Jane You. Data Augmentation via Latent Space Interpolation for Image Classification. In *International Conference on Pattern Recognition*, pp. 728–733. 2018.
- [257] Omer Levy and Yoav Goldberg. Linguistic Regularities in Sparse and Explicit Word Representations. In *Eighteenth Conference on Computational Natural Language Learning*, pp. 171–180. 2014.
- [258] Sanjeev Arora, Yuanzhi Li, Yingyu Liang, Tengyu Ma, and Andrej Risteski. A Latent Variable Model Approach to Pmi-Based Word Embeddings. *Transactions of the Association for Computational Linguistics*, 4:385–399, 2016.
- [259] Carl Allen and Timothy Hospedales. Analogies Explained: Towards Understanding Word Embeddings. In *International Conference on Machine Learning*, pp. 223–231. 2019.
- [260] Tatsunori B Hashimoto, David Alvarez-Melis, and Tommi S Jaakkola. Word Embeddings as Metric Recovery in Semantic Spaces. *Transactions of the Association for Computational Linguistics*, 4:273–286, 2016.
- [261] Allen Newell. Report on a General Problem-Solving Program. In *IFIP Congress*. 1959.
- [262] Cédric Colas, Tristan Karch, Olivier Sigaud, and Pierre-Yves Oudeyer. Intrinsically Motivated Goal-Conditioned Reinforcement Learning: A Short Survey. 2021.
- [263] Yecheng Jason Ma, Jason Yan, Dinesh Jayaraman, and Osbert Bastani. How Far I’ll Go: Offline Goal-Conditioned Reinforcement Learning via F-Advantage Regression. 2022.
- [264] Yannick Schroecker and Charles Isbell. Universal Value Density Estimation for Imitation Learning and Goal-Conditioned Reinforcement Learning. 2020.
- [265] Michael Janner, Qiyang Li, and Sergey Levine. Offline Reinforcement Learning as One Big Sequence Modeling Problem. *Neural Information Processing Systems*, 34:1273–1286, 2021.

- [266] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-Resolution Image Synthesis With Latent Diffusion Models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10684–10695. 2022.
- [267] Chongyi Zheng, Benjamin Eysenbach, Homer Walke, Patrick Yin, Kuan Fang, Ruslan Salakhutdinov, and Sergey Levine. Stabilizing Contrastive RL: Techniques for Offline Goal Reaching. 2023. arXiv:2306.03346.
- [268] Alekh Agarwal, Nan Jiang, Sham M Kakade, and Wen Sun. Reinforcement Learning: Theory and Algorithms. *CS Dept*, pp. 10–4, 2019.
- [269] Yunhao Tang, Zhaohan Daniel Guo, Pierre Harvey Richemond, Bernardo Ávila Pires, Yash Chandak, Rémi Munos, Mark Rowland, Mohammad Gheshlaghi Azar, Charline Le Lan, Clare Lyle, et al. Understanding Self-Predictive Learning for Reinforcement Learning. In *International Conference on Machine Learning*. 2023.
- [270] Adam Stooke, Kimin Lee, Pieter Abbeel, and Michael Laskin. Decoupling Representation Learning From Reinforcement Learning. In *International Conference on Machine Learning*, pp. 9870–9879. 2021.
- [271] Homanga Bharadhwaj, Mohammad Babaeizadeh, Dumitru Erhan, and Sergey Levine. Information Prioritization Through Empowerment in Visual Model-Based RL. In *International Conference on Learning Representations*. 2022.
- [272] Pablo Samuel Castro, Tyler Kastner, P. Panangaden, and Mark Rowland. MICo: Improved Representations via Sampling-Based State Similarity for Markov Decision Processes. In *Neural Information Processing Systems*. 2021.
- [273] Raj Ghugare, Homanga Bharadhwaj, Benjamin Eysenbach, Sergey Levine, and Ruslan Salakhutdinov. Simplifying Model-Based RL: Learning Representations, Latent-Space Models, and Policies With One Objective. In *International Conference on Learning Representations*. 2023.
- [274] Cameron S. Allen. Learning Markov State Abstractions for Deep Reinforcement Learning. In *Neural Information Processing Systems*. 2021.
- [275] Bogdan Mazoure, Benjamin Eysenbach, Ofir Nachum, and Jonathan Tompson. Contrastive Value Learning: Implicit Models for Simple Offline RL. In *Conference on Robot Learning*, pp. 1257–1267. 2023.
- [276] Zhaohan Daniel Guo, Shantanu Thakoor, Miruna Pislari, Bernardo Ávila Pires, Florent Altch'e, Corentin Tallec, Alaa Saade, Daniele Calandriello, et al. BYOL-Explore: Exploration by Bootstrapped Prediction. In *Neural Information Processing Systems*. 2022.
- [277] Yilun Du, Chuang Gan, and Phillip Isola. Curious Representation Learning for Embodied Intelligence. In *IEEE/CVF International Conference on Computer Vision*, pp. 10388–10397. 2021.

- [278] Lydia E Kavraki, Petr Svestka, J-C Latombe, and Mark H Overmars. Probabilistic Roadmaps for Path Planning in High-Dimensional Configuration Spaces. *IEEE Transactions on Robotics and Automation*, 12(4):566–580, 1996.
- [279] Steven M LaValle, James J Kuffner, BR Donald, et al. Rapidly-Exploring Random Trees: Progress and Prospects. *Algorithmic and Computational Robotics: New Directions*, 5:293–308, 2001.
- [280] Sep Thijssen and H. J. Kappen. Path Integral Control and State-Dependent Feedback. *Physical Review E*, 91(3):032104, 2015.
- [281] Grady Williams, Andrew Aldrich, and Evangelos Theodorou. Model Predictive Path Integral Control Using Covariance Variable Importance Sampling. 2015. arXiv:1509.01149.
- [282] Ben Eysenbach, Russ R Salakhutdinov, and Sergey Levine. Search on the Replay Buffer: Bridging Planning and Reinforcement Learning. *Neural Information Processing Systems*, 32, 2019.
- [283] Sham Machandranath Kakade. *On the Sample Complexity of Reinforcement Learning*. University of London, University College London (United Kingdom), 2003.
- [284] Richard Bellman. Dynamic Programming. *Science*, 153(3731):34–37, 1966.
- [285] Ew Dijkstra. A Note on Two Problems in Connexion With Graphs. *Numerische Mathematik*, 1:269–271, 1959.
- [286] Richard S. Sutton. Reinforcement Learning: An Introduction. *A Bradford Book*, 2018.
- [287] Charles Packer, Katelyn Gao, Jernej Kos, Philipp Krähenbühl, Vladlen Koltun, and Dawn Song. Assessing Generalization in Deep Reinforcement Learning. 2018. arXiv:1810.12282.
- [288] Peter E. Hart, Nils J. Nilsson, and Bertram Raphael. A Formal Basis for the Heuristic Determination of Minimum Cost Paths. *IEEE Transactions on Systems Science and Cybernetics*, 4(2):100–107, 1968.
- [289] Steven M. LaValle and James J. Kuffner. Randomized Kinodynamic Planning. *International Journal of Robotics Research*, 20(5):378–400, 2001.
- [290] Brian D Ziebart, Andrew L Maas, J Andrew Bagnell, Anind K Dey, et al. Maximum Entropy Inverse Reinforcement Learning. In *AAAI*, volume 8, pp. 1433–1438. 2008.
- [291] Thomas H Cormen, Charles E Leiserson, Ronald L Rivest, and Clifford Stein. *Introduction to Algorithms*. MIT press, 2022.
- [292] Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft Actor-Critic: Off-Policy Maximum Entropy Deep Reinforcement Learning With a Stochastic Actor. In *International Conference on Machine Learning*. 2018.

- [293] Michal Bortkiewicz, Wlodek Palucki, Vivek Myers, Tadeusz Dziarmaga, Tomasz Arczewski, Lukasz Kuciński, and Benjamin Eysenbach. Accelerating Goal-Conditioned RL Algorithms and Research. 2024. arXiv:2408.11052.
- [294] F. Rosenblatt. *Principles of Neurodynamics: Perceptrons and the Theory of Brain Mechanisms*. Spartan Books, 1961.
- [295] David J. Burr. A Neural Network Digit Recognizer. *Proc. IEEE SMC*, pp. 1621–1625, 1986.
- [296] Silviu Pitis, Harris Chan, Kiarash Jamali, and Jimmy Ba. An Inductive Bias for Distances: Neural Nets That Respect the Triangle Inequality. 2020. arXiv:2002.05825.
- [297] Aviv Tamar, Yi Wu, Garrett Thomas, Sergey Levine, and Pieter Abbeel. Value Iteration Networks. *Neural Information Processing Systems*, 29, 2016.
- [298] Lisa Lee, Emilio Parisotto, Devendra Singh Chaplot, Eric Xing, and Ruslan Salakhutdinov. Gated Path Planning Networks. In *International Conference on Machine Learning*, pp. 2947–2955. 2018.
- [299] Onur Beker, Mohammad Mohammadi, and Amir Zamir. PALMER: Perception-Action Loop With Memory for Long-Horizon Planning. *Neural Information Processing Systems*, 35:34258–34271, 2022.
- [300] Richard S Sutton. Dyna, an Integrated Architecture for Learning, Planning, and Reacting. *ACM Sigart Bulletin*, 2(4):160–163, 1991.
- [301] Kurtland Chua, Roberto Calandra, Rowan Mcallister, and Sergey Levine. Deep Reinforcement Learning in a Handful of Trials Using Probabilistic Dynamics Models. In *Neural Information Processing Systems*. 2018.
- [302] Anusha Nagabandi, Gregory Kahn, Ronald S Fearing, and Sergey Levine. Neural Network Dynamics for Model-Based Deep Reinforcement Learning With Model-Free Fine-Tuning. In *IEEE International Conference on Robotics and Automation*. 2018.
- [303] Kendall Lowrey, Aravind Rajeswaran, Sham Kakade, Emanuel Todorov, and Igor Mordatch. Plan Online, Learn Offline: Efficient Learning and Exploration via Model-Based Control. 2018. arXiv:1811.01848.
- [304] Grady Williams, Nolan Wagener, Brian Goldfain, Paul Drews, James M Rehg, Byron Boots, and Evangelos A Theodorou. Information Theoretic MPC for Model-Based Reinforcement Learning. In *International Conference on Robotics and Automation*. 2017.
- [305] Oleh Rybkin, Chuning Zhu, Anusha Nagabandi, Kostas Daniilidis, Igor Mordatch, and Sergey Levine. Model-Based Reinforcement Learning via Latent-Space Collocation. In *International Conference on Machine Learning*, pp. 9190–9201. 2021.
- [306] Giambattista Parascandolo, Lars Buesing, Josh Merel, Leonard Hasenclever, John Aslanides, Jessica B Hamrick, Nicolas Heess, Alexander Neitz, and

- Theophane Weber. Divide-And-Conquer Monte Carlo Tree Search for Goal-Directed Planning. 2020. arXiv:2004.11410.
- [307] Karl Pertsch, Oleh Rybkin, Jingyun Yang, Shenghao Zhou, Konstantinos Derpanis, Kostas Daniilidis, Joseph Lim, and Andrew Jaegle. Keyframing the Future: Keyframe Discovery for Visual Prediction and Planning. In *Learning for Dynamics and Control*, pp. 969–979. 2020.
- [308] Joshua B Tenenbaum, Vin de Silva, and John C Langford. A Global Geometric Framework for Nonlinear Dimensionality Reduction. *Science*, 290(5500):2319–2323, 2000.
- [309] Ofir Nachum, Haoran Tang, Xingyu Lu, Shixiang Gu, Honglak Lee, and Sergey Levine. Why Does Hierarchy (Sometimes) Work So Well in Reinforcement Learning? 2019. arXiv:1909.10618.
- [310] Jonathan Lee, Annie Xie, Aldo Pacchiano, Yash Chandak, Chelsea Finn, Ofir Nachum, and Emma Brunskill. Supervised Pretraining Can Learn In-Context Reinforcement Learning. *Neural Information Processing Systems*, 36, 2024.
- [311] Jiaxin Huang, Shixiang Shane Gu, Le Hou, Yuexin Wu, Xuezhi Wang, Hongkun Yu, and Jiawei Han. Large Language Models Can Self-Improve. 2022. arXiv:2210.11610.
- [312] Geoffrey Irving, Paul Christiano, and Dario Amodei. AI Safety via Debate. 2018. arXiv:1805.00899.
- [313] Alfred North Whitehead and Bertrand Russell. *Principia Mathematica to\* 56*, volume 2. Cambridge University Press, 1927.
- [314] Christopher JCH Watkins and Peter Dayan. Q-Learning. *Machine Learning*, 8:279–292, 1992.
- [315] Stephen Casper, Xander Davies, Claudia Shi, Thomas Krendl Gilbert, Jérémy Scheurer, Javier Rando, Rachel Freedman, Tomasz Korbak, David Lindner, Pedro Freire, et al. Open Problems and Fundamental Limitations of Reinforcement Learning From Human Feedback. 2023. arXiv:2307.15217.
- [316] Dylan Hadfield-Menell, Stuart J. Russell, Pieter Abbeel, and Anca Dragan. Cooperative Inverse Reinforcement Learning. *Neural Information Processing Systems*, 29, 2016.
- [317] Dylan Hadfield-Menell, Smitha Milli, Pieter Abbeel, Stuart J. Russell, and Anca Dragan. Inverse Reward Design. *Neural Information Processing Systems*, 30, 2017.
- [318] Micah Carroll, Dylan Hadfield-Menell, Stuart Russell, and Anca Dragan. Estimating and Penalizing Preference Shift in Recommender Systems. In *ACM Conference on Recommender Systems, RecSys '21*, pp. 661–667. 2021.
- [319] Alexander Matt Turner, Logan Smith, Rohin Shah, Andrew Critch, and Prasad Tadepalli. Optimal Policies Tend to Seek Power. In *Neural Information Processing Systems*. 2023.

- [320] Stuart Russell. Learning Agents for Uncertain Environments (Extended Abstract). In *Conference on Computational Learning Theory*, pp. 101–103. 1998.
- [321] Saurabh Arora and Prashant Doshi. A Survey of Inverse Reinforcement Learning: Challenges, Methods and Progress. *Artificial Intelligence*, 297:103500, 2021.
- [322] Christian Wirth, Riad Akrouf, Gerhard Neumann, and Johannes Fürnkranz. A Survey of Preference-Based Reinforcement Learning Methods. *Journal of Machine Learning Research*, 18(136):1–46, 2017.
- [323] Christoph Salge, Cornelius Glackin, and Daniel Polani. Empowerment—an Introduction. *Guided Self-Organization: Inception*, pp. 67–114, 2014.
- [324] Alexander S Klyubin, Daniel Polani, and Chrystopher L Nehaniv. Empowerment: A Universal Agent-Centric Measure of Control. In *IEEE Congress on Evolutionary Computation*, volume 1, pp. 128–135. 2005.
- [325] Siddharth Reddy, Anca D Dragan, and Sergey Levine. Shared Autonomy via Deep Reinforcement Learning. 2018. arXiv:1802.01744.
- [326] Nathan D. Ratliff, J. Andrew Bagnell, and Martin A. Zinkevich. Maximum Margin Planning. In *Rd International Conference on Machine Learning - ICML '06*, pp. 729–736. 2006.
- [327] Pieter Abbeel and Andrew Y Ng. Apprenticeship Learning via Inverse Reinforcement Learning. In *International Conference on Machine Learning*, p. 1. 2004.
- [328] Thomas M Cover. *Elements of Information Theory*. John Wiley & Sons, 1999.
- [329] Nihat Ay, Jürgen Jost, Hông Vân Lê, and Lorenz Schwachhöfer. *Information Geometry*, volume 64 of *Ergebnisse Der Mathematik Und Ihrer Grenzgebiete 34*. Springer International Publishing, 2017.
- [330] Frank Nielsen. An Elementary Introduction to Information Geometry. *Entropy*, 22(10):1100, 2020.
- [331] Benjamin Eysenbach, Ruslan Salakhutdinov, and Sergey Levine. The Information Geometry of Unsupervised Reinforcement Learning. In *International Conference on Learning Representations*. 2022.
- [332] Robert G. Gallager. Source Coding With Side Information and Universal Coding. 1979.
- [333] Boris Yakovlevich Ryabko. Coding of a Source With Unknown but Ordered Probabilities. *Problems of Information Transmission*, 15(2):134–138, 1979.
- [334] R. Duncan Luce. *Individual Choice Behavior*. Individual Choice Behavior. John Wiley, 1959.
- [335] Tobias Jung, Daniel Polani, and Peter Stone. Empowerment for Continuous Agent—Environment Systems. *Adaptive Behavior*, 19(1):16–39, 2011.

- [336] Cassidy Laidlaw and Anca Dragan. The Boltzmann Policy Distribution: Accounting for Systematic Suboptimality in Human Models. In *International Conference on Learning Representations*. 2022.
- [337] Alexander Pan, Kush Bhatia, and Jacob Steinhardt. The Effects of Reward Misspecification: Mapping and Mitigating Misaligned Models. In *International Conference on Learning Representations*, arXiv:2201.03544. 2022.
- [338] Jeremy Tien, Jerry Zhi-Yang He, Zackory Erickson, Anca D. Dragan, and Daniel S. Brown. Causal Confusion and Reward Misidentification in Preference-Based Reward Learning. In *International Conference on Learning Representations*. 2023.
- [339] Cassidy Laidlaw, Shivam Singhal, and Anca Dragan. Preventing Reward Hacking With Occupancy Measure Regularization. 2024. arXiv:2403.03185.
- [340] Ildefons Magrans de Abril and Ryota Kanai. A Unified Strategy for Implementing Curiosity and Empowerment Driven Reinforcement Learning. 2018. arXiv:1806.06505.
- [341] Sean Chen, Jensen Gao, Siddharth Reddy, Glen Berseth, Anca D. Dragan, and Sergey Levine. ASHA: Assistive Teleoperation via Human-in-the-Loop Reinforcement Learning. In *International Conference on Robotics and Automation*. 2022.
- [342] Siddharth Reddy, Sergey Levine, and Anca Dragan. First Contact: Unsupervised Human-Machine Co-Adaptation via Mutual Information Maximization. *Neural Information Processing Systems*, 35:31542–31556, 2022.
- [343] Andrew G. Barto. Intrinsic Motivation and Reinforcement Learning. In Gianluca Baldassarre and Marco Mirolli, editors, *Intrinsically Motivated Learning in Natural and Artificial Systems*, pp. 17–47. Springer, 2013.
- [344] Arthur Aubret, Laetitia Matignon, and Salima Hassas. A Survey on Intrinsic Motivation in Reinforcement Learning. In *Entropy*. 2023.
- [345] Benjamin Eysenbach, Abhishek Gupta, Julian Ibarz, and Sergey Levine. Diversity Is All You Need: Learning Skills Without a Reward Function. 2018. arXiv:1802.06070.
- [346] Yuri Burda, Harrison Edwards, Amos Storkey, and Oleg Klimov. Exploration by Random Network Distillation. *International Conference on Learning Representations*, 2019.
- [347] Karl Friston. The Free-Energy Principle: A Unified Brain Theory? *Nature Reviews Neuroscience*, 11(2):127–138, 2010.
- [348] Glen Berseth, Daniel Geng, Coline Devin, Nicholas Rhinehart, Chelsea Finn, Dinesh Jayaraman, and Sergey Levine. Smirl: Surprise Minimizing Reinforcement Learning in Unstable Environments. In *International Conference on Learning Representations*. 2019.

- [349] Shakir Mohamed and Danilo Jimenez Rezende. Variational Information Maximisation for Intrinsically Motivated Reinforcement Learning. In *Neural Information Processing Systems*, volume 28. 2015.
- [350] Michael Laskin, Hao Liu, Xue Bin Peng, Denis Yarats, Aravind Rajeswaran, and Pieter Abbeel. CIC: Contrastive Intrinsic Control for Unsupervised Skill Discovery. 2022. arXiv:2202.00161.
- [351] Seohong Park, Jongwook Choi, Jaekyeom Kim, Honglak Lee, and Gunhee Kim. Lipschitz-Constrained Unsupervised Skill Discovery. In *International Conference on Learning Representations*. 2021.
- [352] Susanne Still and Doina Precup. An Information-Theoretic Approach to Curiosity-Driven Reinforcement Learning. *Theory in Biosciences*, 131:139–148, 2012.
- [353] Nikolay Nikolov, Johannes Kirschner, Felix Berkenkamp, and Andreas Krause. Information-Directed Exploration for Deep Reinforcement Learning. In *International Conference on Learning Representations*. 2019.
- [354] Erdem Biyik, Dylan P. Losey, Malayandi Palan, Nicholas C. Landolfi, Gleb Shevchuk, and Dorsa Sadigh. Learning Reward Functions From Diverse Sources of Human Feedback: Optimally Integrating Demonstrations and Preferences. In *Int. J. Robotics Res.* 2022.
- [355] Vivek Myers, Erdem Biyik, Nima Anari, and Dorsa Sadigh. Learning Multi-modal Rewards From Rankings. In *Conference on Robot Learning*, pp. 342–352. 2021.
- [356] Neil Houlsby, Ferenc Huszár, Zoubin Ghahramani, and Máté Lengyel. Bayesian Active Learning for Classification and Preference Learning. 2011. arXiv:1112.5745.
- [357] Michael Laskin, Aravind Srinivas, and Pieter Abbeel. Curl: Contrastive Unsupervised Representations for Reinforcement Learning. In *International Conference on Machine Learning*, pp. 5639–5650. 2020.
- [358] Ida Momennejad, Evan M Russek, Jin H Cheong, Matthew M Botvinick, Nathaniel Douglass Daw, and Samuel J Gershman. The Successor Representation in Human Reinforcement Learning. *Nature Human Behaviour*, 1(9):680–692, 2017.
- [359] K. Friston, L. Da Costa, N. Sajid, Robert Conor Heins, K. Ueltzhoffer, G. A. Pavliotis, and T. Parr. The Free Energy Principle Made Simpler but Not Too Simple. *Physics Reports-Review Section of Physics Letters*, 1024:1–29, 2023.
- [360] Angel Muleshkov and Tan Nguyen. Easy Proof of the Jacobian for the N-Dimensional Polar Coordinates. *Pi Mu Epsilon Journal*, 14(4):269–273, 2016.
- [361] D. P. Kingma. Adam: A Method for Stochastic Optimization. In *International Conference on Learning Representations*. 2014.



- [362] Diederik P. Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization. 2017. arXiv:1412.6980.
- [363] Alexander Rutherford, Benjamin Ellis, Matteo Gallici, Jonathan Cook, Andrei Lupu, Gardar Ingvarsson, Timon Willi, Akbir Khan, et al. Jaxmarl: Multi-Agent RL Environments in JAX. In *International Conference on Autonomous Agents and Multiagent Systems*. 2024.
- [364] Andrew L. Maas, Awni Y. Hannun, and Andrew Y. Ng. Rectifier Nonlinearities Improve Neural Network Acoustic Models. In *International Conference on Machine Learning*, volume 30, p. 3. 2013.



## VIDEOS AND CODE

---

Videos and code for the following chapters can be found at the provided links:

**Chapter 2:** Goal Representations for Instruction Following

Website and code: <https://rail-berkeley.github.io/grif/>.

**Chapter 3:** Policy Adaptation via Language Optimization

Code: <https://github.com/vivekmyers/palo-robot>.

**Chapter 4:** A Metric Structure for Successor Representations

Code and experiments: [https://github.com/vivekmyers/contrastive\\_metrics](https://github.com/vivekmyers/contrastive_metrics).

**Chapter 6:** Planning with Contrastive Representations

Code and experiments: [https://github.com/vivekmyers/contrastive\\_planning](https://github.com/vivekmyers/contrastive_planning).

**Chapter 8:** Empowerment via Successor Representations

Website and code: <https://empowering-humans.github.io>.

# B LANGUAGE MODEL PROMPTING DETAILS

---

This section contains the prompts used for the various methods (Chapters 2, 3 and 5) that require language model prompting.

## B.1 GRIF INSTRUCTION AUGMENTATION

In order to increase the diversity of language annotations, we augment our natural language annotations using GPT-3.5. Through the API, we query the gpt-3.5-turbo model to generate paraphrases of instructions in our dataset. We generate 5 paraphrases per instruction and sample from them randomly during training. An example prompt and response are shown below. We found in preliminary experiments that using augmented instructions slightly improved language generalization, so we keep this augmentation for all models and baselines.

Prompt:

```
Generate 5 variations of the following command:  
"put the mushroom in the metal pot"  
Number them like 1. 2. 3.  
Be concise and use synonyms.
```

Response:

1. Place the fungus in the metallic container.
2. Insert the mushroom into the steel vessel.
3. Set the toadstool inside the iron cauldron.
4. Position the champignon within the tin pot.
5. Place the fungi in the metallic kettle.

## B.2 PALO PROMPTING DETAILS

We employ a keyword decomposition prompt in our augmentation method and a planning prompt to generate VLM outputs. They are listed below:

```
User: "You are presented with a text for high level instruction for  
a robot, and you need to extract keywords in the task description  
text."
```

In this instruction, the first keyword is the object being moved, and the second keyword, if applicable, what is the moving taking this to (either another object or a location) within the instruction.

Only return the first and second keyword, and they should be separated by a comma. If the instruction is in another language, write your response in English.

For example, if the text instruction says "Pick up the silver lid on the left to the middle of two burners", return "silver lid, middle of two burners".

Or if the instruction says: "Move the object to the top middle side of the table.", your response should be "object, top middle side of the table".

Or if the instruction says : "Move the red greenish thing on the towel to the right.", return "red greendish thing on the towel, the right".

Try your best to find the two key phrases, but if you can't find the second keyword within the instruction sentence, write "N/A".

For example, if the instruction is "Move the pot lid.", the response should be "pot lid, N/A".

There might be some other description regarding confidence at the end, you are safe to ignore it.\n The specific task description for you to analyze is: \n {instruction} \n "

User: Here is an image observed by the robot in a tabletop robot manipulation environment. The gripper situated at the top of the center of table and perpendicular to it.

Now plan for the list of subtasks and skills the robot needs to perform in order to {instrs}.

Each step in the plan can be selected from the available skills below:

\*movement direction:

- \*forward. This skill moves the robot gripper away from the camera by a small distance.
- \*backward. This skill moves the robot gripper towards the camera by a small distance.
- \*left. This skill moves the robot gripper to the left of the image by a small distance.
- \*right. This skill moves the robot gripper to the right of the image by a small distance.
- \*up. This skill moves the robot gripper upward until a safe height.
- \*down. This skill moves the robot gripper downward to the table surface.

\*rotation direction:

- \*left. This skill tilts the gripper to an angle to the left.

- \*right. This skill tilts the gripper to an angle to the right.
- \*down. This skill tilts the gripper to an angle facing up.
- \*up. This skill tilts the gripper to an angle facing down.
- \*clockwise. This skill rotates the gripper and the object it is holding clockwise.
- \*counterclockwise. This skill rotates the gripper and the object it is holding counterclockwise.

\*gripper movement:

- \*close the gripper. This skill controls the robot gripper to close to grasp an object.
- \*open the gripper. This skill controls the robot gripper to open and release the object in hand.

You may choose between using one of movement direction, rotation direction, or gripper movement.

If you were to choose to use movement direction, you may use one or two directions and include a target object, and you should format it like this:

"move the gripper x towards z" or "move the gripper x and y towards z" where x and y are the directions and z is the target object.

You also must start your command with "move the gripper".

Therefore, instead of saying something like "down" or "up", you should phrase it like "move the gripper down" and "move the gripper up". Make sure to include at least one direction in your command since otherwise this command format won't make sense.

If you were to choose to use gripper movement, you should format the command as "close the gripper to pick up x" or "open the gripper to release x", where x is the target object.

You may discard the target object if necessary. In that case use "close the gripper" or "open the gripper".

If you think the gripper is close to the target object, then you must choose to use gripper movement to grasp the target object to maintain efficiency.

If you were to choose gripper rotation, you should format the command as "rotate the gripper x", where x is the target rotation direction. You need to make sure that in pouring tasks, the opening of the container is aligned with the pot.

For example, if the object is aligned vertically but you want it to align it horizontally, then you should call "rotate the gripper counterclockwise". If you want to tilt the object in the gripper to pour it, you should call "rotate the gripper left"

Pay close attention to these factors:

- \*Which task are you doing.
- \*Whether the gripper is closed.
- \*Whether the gripper is holding the target object.
- \*How far the two target objects are. If they are across the table, then duplicate the commands with a copy of it.
- \*Where the gripper is. After the end of each subtask, it is reasonable to assume that the gripper will not be at where it originally was in the image, but somewhere close to the last target object.

Especially pay attention to the actual direction between the gripper and the target object. Remember that the robot's angle is roughly the same as the camera's angle.

To determine whether the gripper should move forward or backward, look into the edge of the table. If the target object is closer to the edge of the table that is near the top of the image, you should move forward, and if it is closer to the edge that is near the bottom of the image, you should move backward.

At the end of each subtask, you need to use the skill "move the gripper back to neutral. This will move the gripper back to the original position of the image after completing the task.

Start by looking at what objects are in the image, and then plan with the direction of the objects in mind. The tasks should be completed sequentially, therefore you need to consider the position of the gripper after each task before planning the next task.

You should return a json dictionary with the following fields:

- subtask: this should be the key of the dictionary. It should contain the only the verbal description of the subtask the robot needs to perform sequentially in order to finish the task, and they should be ordered in the same way the task is completed.
- list of skills: this should be the value of the dictionary. It should be a list of skills the robot needs to perform in order to finish the corresponding subtask.

Be concise, and do not return any other comments other than the dictionary mentioned above. Do not put "subtask: " or "list of skills: " in the key and value of the dictionary you generate. Remember only the description and list should be returned.

# C

## EXPERIMENT DETAILS

---

This chapter contains additional details for the experiments in the main text.

### C.1 PALO EVALUATION DETAILS

Experimental details for Chapter 3 are provided below.

#### Ablation Details

We ablate our experiment in progressive manners, going from full implementation to using only the barebone hierarchical policy network.

- PALO without high level instruction: while running PALO, we derive both high and low level instruction sets. However, during inference on robot, we mask out the high level instruction and feed in zero embeddings.
- PALO without low level instruction: mask out the low level instruction and replace them with zero embeddings during inference.
- Fixed Time During Optimization: for each trajectory that has corresponding length  $H_1, H_2, \dots, H_i$ , we choose fixed  $u_i = [\frac{H_i}{k}, \frac{2H_i}{k}, \dots, \frac{(k-1)H_i}{k}]$  during optimization. We implement no  $u$  sampling, which reduce PALO into an arg max operation.
- Zero-Shot Plan Generation: instead of sampling 15 plans, we sample only one plan from VLM and examine the behavior of the robot using that specific plan.
- No VLM Guidance: We use only  $\ell$  as our high level instruction, and mask out low level instruction with zero embeddings during inference.

### C.2 TRA IMPLEMENTATION

In this section, we provide some details on the implementatinon of temporal representation alignment (TRA) and its training process.

## Dataset Curation

We use an augmented version of BridgeData. We augment the dataset by generating 5 additional paraphrased instruction per language instruction. During training process, we randomly sample the instructions for each trajectory to ensure an equal coverage of texts.

During data loading process, for each observation that is being sampled with timestep  $k$ , we also sample  $k^+ \triangleq \min(k + x, H)$ ,  $x \sim \text{Geom}(1 - \gamma)$ , and load  $s_k$  along with  $s_{k^+}$ . We employ random cropping, resizing, and hue changes during training process image robustness.

## Policy Training

We use a ResNet-34 architecture for the policy network. We train our policy with one Google V4-8 TPU VM instance for 150,000 steps, which takes a total of 20 hours. We use a learning rate of  $3 \times 10^{-4}$ , 2000 linear warmup steps, and a MLP head of 3 layers of 256 dimensions after encoding the observation representations as well as goal representations.

## Baseline Implementations

We summarize the implementation details of the baselines discussed in Section 5.2.

**Octo.** We use the Octo-base 1.5 model publicly available on HuggingFace for evaluating Octo baselines. We use inference code that is readily available for both image- and language- conditioned tasks. During inference, we use an action chunking window of 4 and an execution horizon window of 4.

**Behavior Cloning.** We use the same architecture for LCBC as in Walke et al. [62], Myers et al. [7]. During the training process we use the same hyperparameters as TRA.

**Advantage Weighted Regression.** In order to train an AWR agent without separately implementing a reward critic, we follow Eysenbach et al. [121] and use a surrogate for advantage:

$$\mathcal{A}(s_t) = \mathcal{L}_{\text{NCE}}(f(s_t), f(g)) - \mathcal{L}_{\text{NCE}}(f(s_{t+1}), f(g)). \quad (\text{C.1})$$

Here,  $f$  can be any of the encoders  $\phi, \xi, \psi$ .  $\mathcal{L}$  is the same InfoNCE loss defined Section 5.1, and  $g$  is defined as either the goal observation or the goal language instruction, depending on the modality.

And we extract the policy using advantage weighted regression (AWR) [138]:

$$\pi \leftarrow \arg \max_{\pi} \mathbb{E}_{s,a \sim \mathcal{D}} \left[ \log \pi(a|s, z) \exp(A(s, a) / \beta) \right]. \quad (\text{C.2})$$

During training, we set  $\beta$  to 1, and we use a batch size of 128, the same value as policy training for our method.



## Experiment Details

In this section, we go through our experiment details and how they are set up. During evaluation, we randomly reset the positions of each item within the table, and perform 5 to 10 trials on each task, depending on whether this task is important within each scene. We examine tasks that are seen in BridgeData, which include conventionally less challenging tasks such as object manipulation, and challenging tasks to learn within the dataset such as cloth folding and drawer opening.

**List of Tasks.** Table C.1 describes each task within each scene, and the language annotation used when the policy is used for inference. Every task that is outside of the drawer scene are multiple step, and require compositional generalization.

Table C.1: Task Instructions

Scene	Count	Task Description	Instruction
Drawer	10	open the drawer	"open the drawer"
	10	put the mushroom in the drawer	"put the mushroom in the drawer"
	10	close the drawer	"close the drawer"
Task Generalization	5	put the spoons on the plates	"move the spoons onto the plates."
	5	put the spoons on the towels	"move the spoons on the towels"
	6	fold the cloth into the center from all corners	"fold the cloth into center"
	10	sweep the towels to the right	"sweep the towels to the right of the table"
Semantic Generalization	10	put the sushi and the corn on the plate	"put the food items on the plate"
	5	put the sushi and the mushroom in the bowl	"put the food items in the bowl"
	10	put the sushi, corn, and the banana in the bowl	"put everything in the bowl"
Tasks With Dependency	10	take mushroom out of drawer	"open the drawer and then take the mushroom out of the drawer"
	10	move bell pepper and sweep towel	"move the bell pepper to the bottom right corner of the table, and then sweep the towel to the top right corner of the table"
	10	put the corn on the plate, <i>and then</i> put the sushi in the pot	"put the corn on the plate and then put the sushi in the pot"

**Inference.** During inference, we use a maximum of 200 timesteps to account for long-horizon behaviors, which remains the same for all policies. We determine a task as successful when the robot completes the task it was instructed to within the timeframe. For evaluating baselines, we use 5 trials for each of the tasks.

**Validation MSE.** In addition to rolling out the policy on real-world robot settings, we additionally collected 9 additional tasks that are compositionally OOD for 5 trajectories each, and we use 3 randomly selected seeds to train policies to evaluate the MSE on the validation trajectories.

## Environment Details

We evaluate our method in a real-world tabletop manipulation setup. We use a 6DOF WidowX-250 robot interacting with various objects both inside and outside of our training distribution at 5 Hz. We use one 640×480 RGB camera mounted on top of the model as set up in BridgeData [62]. When computing observations we downsample images to 224 × 224.

We evaluate our method in the following scenes, which include:

**Sweep:** This scene involves an object manipulation as well as sweeping task unseen in the BridgeData’s initial training trajectories.

**mint:** Placing the mushroom in the pot, then sweep the mints on the right using the towel.

**skittles:** Instead of using mints and towel for sweeping, we use a swiffer and skittles instead.

**Drawer:** This scene involves using a drawer and perform manipulation within the space of the drawer.

**put in:** Open the drawer, and then put a purple object (beet/sweet potato) inside the drawer.

**pry away:** A pot is stored inside the drawer space, and the robot must use a ladle to pry away the pot within drawer.

**Bowl:** This scene involves object manipulation to a bowl and perform long-horizon or 6DOF manipulation.

**salad:** This task requires sequential object manipulation by putting a corn cob and a mushroom in the bowl.

**pouring:** This task requires the robot to grasp a scoop and pour almonds inside the scoop into the bowl.

**Rotation:** This scene involves rotating a spoon and a marker to fit into a white container not aligned with the pen/marker, and naive pick-and-place will not correctly align the object into the container.

**spoon:** Placing the spoon in the container placed on the left side of the table.

**marker:** Replacing the spoon with the marker and randomize location of the container while being misaligned.

We summarize the evaluation tasks in Table C.2, and show example rollouts in Fig. C.1.

## Training Details

We train on an augmented version of the BridgeDataV2 dataset [62], which features over 50k trajectories with 72k language annotations. We algorithmically augment the dataset with low-level instructions using heuristics designed over the proprioceptive states of the robot and incorporate language context by parsing the language

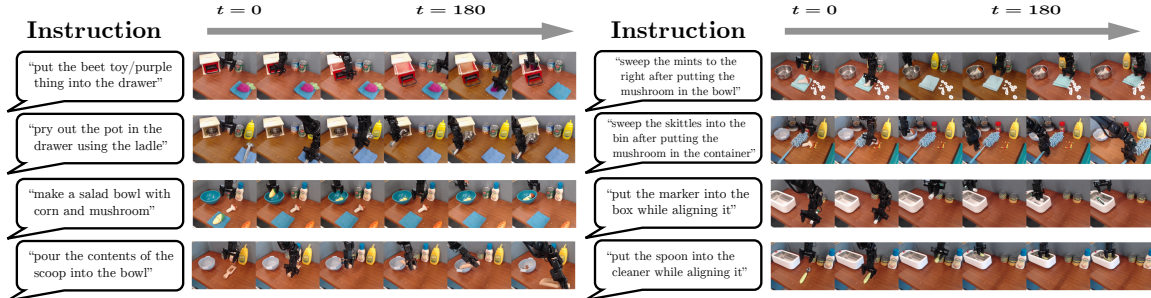


Figure C.1: Sample rollouts using PALO on unseen testing tasks.

Table C.2: Task Breakdown

Scene	Task	Long-Horizon?	6DOF?	Instruction
Drawer	put in	Yes	Yes	"put the beet toy /purple thing into the drawer."
	pry away	Yes	Yes	"pry out the pot in the drawer using the ladle."
Bowl	salad	Yes	No	"make a salad bowl with corn and mushroom."
	pour scoop	No	Yes	"pour the contents of the scoop into the bowl."
Sweep	mints	Yes	No	"sweep the mints to the right after putting the mushroom in the bowl."
	skittles	Yes	No	"sweep the skittles into the bin after putting the mushroom in the container."
Rotation	marker	No	Yes	"put the marker into the box while aligning it."
	spoon	No	Yes	"put the spoon into the cleaner while aligning it."

instruction using a language model. We use the Adam optimizer [361] to minimize the loss function in Eq. (C.3).

Instead of naively looping through Algorithm 1, we batch our implementation with the exception of the outermost for loop, thus reducing time consumption during optimization by a significant margin via vectorization. We record an empirical time consumption of 470 seconds for our language optimization module on computations ran on a V4 TPU module, in which only 200 seconds are required for sampling 20000 different partitions to complete the optimization for all of the 15 sets of language instructions. We save our optimal plans for future use, thus reducing overhead even more.

We encode both language instructions using a frozen MUSE model [58] before passing them into the main ResNet with FiLM layers [114].

### Hyperparameter Selection

We discuss the hyperparameters used in our method and baselines.

**Policy Training.** We set our learning rate for our Adam Optimizer [361] to  $3 \cdot 10^{-4}$  and a dropout rate of 0.1 in our policy head. We employ random resizing and cropping, contrast, brightness, saturation, and hue for input images. We train our policy for 300,000 steps, in which we use the checkpoint with the lowest validation MSE. The total training time takes 12 hours when trained on 4 TPU pods.

**Language Decomposition Optimization.** During optimization, we sample  $M = 15$  random instruction sets from GPT4-o, and we use  $N = 20,000$  sampling steps in order to find the best subtask decomposition.

In order to batch across demonstrations, which have different trajectory lengths, we pad our trajectories to a certain length  $H$  (200 for long-horizon tasks, 150 for non long-horizon tasks). We sum the squared difference between generated action and oracle action in evaluation, thus giving a consistent error metric analogous to Eq. (3.7).

### Baseline Details.

We finetune an Octo-small [67] model that is trained on BridgeData with a learning rate of  $3 \cdot 10^{-4}$  and finetune our model’s action head for 5000 steps. We use the hyperparameters set by Octo for the rest of the settings.

In order to perform tasks in long-horizon, we assign a language label for each task in order to transplant semantic understanding from human into Octo. The same language instruction for PALO evaluation is also used for Octo finetuning.

### Augmentation Details.

We train the policy by maximizing the likelihood of actions given high- and low-level instructions in the dataset  $\mathcal{D}_{\text{prior}}$ :

$$\begin{aligned} \mathcal{J}(\theta) = \mathbb{E}_{\mathcal{D}_{\text{prior}}} [ & \|a_t - \pi_{\theta}(s_t, (c^H, \mathbf{0}))\|^2 \\ & + \|a_t - \pi_{\theta}(s_t, (\mathbf{0}, c^L))\|^2 + \|a_t - \pi_{\theta}(s_t, (c^H, c^L))\|^2] \end{aligned} \quad (\text{C.3})$$

where  $s_1 \dots s_H \in \mathcal{S}$ ,  $a_1 \dots a_H \in \mathcal{A}$ ,  $c^L, c^H \in \mathcal{L} \cup \{\mathbf{0}\}$ , and  $\theta$  are the parameters of the policy network,  $\mathbf{0}$  is an additional point representing the absence of a high- or low-level instruction, which will be represented as an embedding vector of zero during training, and  $\tau = (s_0, a_0, \dots, s_H, a_H)$  is a trajectory sampled from the dataset.

This objective encourages the policy to learn to follow instructions at both levels of abstraction, marginalizing over missing instructions. We chunk actions within training data into segments of length 4 and evaluate the low level instruction within these segments and append them into the training data.

### Heuristics for Language Augmentation

We algorithmically enhance our training data by using heuristics generated by the proprioception of the robot and language context, which generates the low-level instructions. The labeled language instruction is passed into a language model to obtain manipulation keywords, and we combine the keywords with the proprioceptive information within that time span including translation, rotation, and gripper movement into coherent language commands.

**Proprioception.** We use standard deviation of each action against the metadata of BridgeData [62] and determine how to describe the proprioception of the label. We determine the largest direction in which the gripper is moving (up, down, left, right, forward, backward) and the orientation it is rotating (up, down, left, right, clockwise, counterclockwise), and determine whether the movement is unambiguous enough by checking the largest z-score in translation and rotation. We then combine the movement as well as the keywords extracted to form language primitive commands.

**Target Object.** We identify the target object using a prompt heuristic to be fed into GPT3.5-Turbo [33] by taking advantage of the fact that BridgeData consists of mainly object manipulation data. We extract two keywords: the object to be manipulated and the destination of the object, based on the fact that much of BridgeData is focused on object manipulation. The precise prompt can be found at Appendix B.2

**Data Filtering.** We filter low-level instruction on two occasions: when the movement itself is ambiguous and when the language model gives inconsistent results. We check the former by looking up the norm of the translation and the norm of

rotation, and we check the latter by using regular expression to see if the result was against the desired format and manually filtering out some common keywords of inadmissible GPT query. On the former occasion, we use an empty string as the low level instruction, and on the second occasion, we use only proprioceptive information for low-level instruction.

**Additional High-Level Language Augmentation.** We additionally augment the high-level language annotations by generating context-free rephrasings with GPT-3.5 [33]. For each trajectory with crowdsourced language annotations in the Bridge-Data v2 dataset, we generate 5 such augmented language strings following the approach of Myers et al. [7].

### C.3 CMD IMPLEMENTATION

We implement CMD, CRL (CPC / NCE), and GCBC using JAX building upon the official codebase of contrastive RL [121]. For the QRL baseline, we use the implementation provided by the author [143]. Whenever possible, we used the same hyperparameters as contrastive RL [121] and match the number of parameters in the model for different algorithms. We used 4 layers of 512 units of MLP as our neural network architectures and set batch size to 256. We find that using a smaller learning rate  $5 \cdot 10^{-6}$  for the contrastive network is useful for improving performance. In light of Lemma 4.1, when learning the  $d_{SD}$  critic in Eq. (4.14), we use a dummy action  $a'$  sampled from the marginal distribution over geometrically-discounted future actions.

We compared approaches in the offline settings across the best performance from 500k steps of training, consistent with past work [134, 121]. All approaches were tested with similar model sizes and runtime, and used tuned hyperparameters. Our code at <https://github.com/mnm-anonymous/qmd> features the precise configurations for the experiments.

### C.4 ESR DETAILS

We provide implementation details for the Empowerment via Successor Representations (ESR) method from Chapter 8.

#### Implementation Details

We ran all our experiments on NVIDIA RTX A6000 GPUs with 48GB of memory within an internal cluster. Each evaluation seed took around 5-10 hours to complete. Our losses (Eqs. 8.27 and 8.35) were computed and optimized in JAX with Adam [362]. We used a hardware-accelerated version of the Overcooked environment from the JaxMARL package [363]. The experimental results described in Section 8.3 were obtained by averaging over 5 seeds for the Overcooked coordination ring layout, 15 for the cramped room layout, and 20 for the obstacle gridworld environment.

Specific hyperparameter values can be found in our code, which is available at [https://github.com/vivekmyers/empowerment\\_successor\\_representations](https://github.com/vivekmyers/empowerment_successor_representations).

**Network Architecture.** In the obstacle grid environment, we used a network with 2 convolutional and 2 fully connected layers and SiLU activations. In Overcooked, we adapted the policy architecture from past work [318], using 3 convolutional layers followed by 4 MLP layers with Leaky ReLU activations [364]. We concatenate in  $a^R$  and  $a^H$  to the state as one-hot encoded channels, i.e. if the action is 5, 6 additional channels will be concatenated to the state with all set to 0s except the 5th channel which is set to 1s.

## C.5 PLANNING INVARIANCE AND HORIZON GENERALIZATION

We provide details for the environments studied in the figures within Chapter 7.

Figure 7.2

This task is a gridworld of size  $30 \times 30$ , with walls shown as in Fig. 7.2. The dynamics are deterministic. There are 5 actions, corresponding to the cardinal directions and a no-op action.

For this plot, we generated data from a random policy, using 1000 trajectories of length 200. We estimated distances using Monte Carlo regression. The left two subplots were generated by selecting actions uses these Monte Carlo distances. We computed the true distances by running Dijkstra’s algorithm. The right two subplots show actions selected using Dijkstra’s algorithm.

Figure 7.7 (Top)

This plot used the same environment as described in Appendix C.5. For this plot, we generated 3000 trajectories of length 50 using a random policy. Only 14% of start-goal pairs have any trajectory between them, meaning that the vast majority of start-goal pairs have never been seen together during training. Thus, this is a good setting for studying generalization.

We first estimated distances using Monte Carlo regression. We select actions using a Boltzmann policy with temperature 0.1 (i.e.,  $\pi(a | s, g) \propto e^{-0.1d(s,g)}$ ). Evaluation is done over 1000 randomly-sampled start-goal pairs. The X axis is binned based on the shortest path distance. The data are aggregated so that start-goal pairs with distance between (say) 20 and 30 get plotted at  $x = 30$ . The “metric regression + quasimetric” distances are obtained by performing path relaxation on these Monte Carlo distances until convergence. The corresponding policy is again a Boltzmann policy with temperature 0.1.

For the Top Right subplot, we perform planning using Dijkstra’s algorithm. We first identify a set of candidate midpoint states where  $d(s, w)$  and  $d(w, g)$  are both



within one unit of half the shortest path distance. We then randomly sample a midpoint state. This planning is done anew at every timestep.

Figure 7.7 (Bottom)

This plot used the same environment as described in Appendix C.5. The CRL method refers to [121] and CMD refers to [9]. We used a representation dimension of 16, a batch size of 256, neural networks with 2 hidden layers of width 32 and Swish activations,  $\gamma = 0.9$ , and Adam optimizer with learning rate  $3e-3$ . The loss functions and architectures are based on those from [293].

For the Bottom Right subplot, we performed planning in the same way as for the Top Right subplot.

Figure 7.5

For this task we directly used the ant maze task as well as loss functions and architectures from Bortkiewicz et al. [293]. All other hyperparameters are kept as the defaults from that paper. Training is done for 100M steps

Figure 7.6

For this experiment we used an S-shaped maze, shown in Fig. C.2.<sup>1</sup> The dynamics are the same as those of Fig. 7.2.



Figure C.2: S-shaped maze.

We collected 3000 trajectories of length 10 and applied CRL with a representation dimension of 16, a batch size of 256, neural networks with 2 hidden layers of width 32 and Swish activations, the backward NCE loss [293],  $\gamma = 0.9$ , using the Adam optimizer with learning rate  $3e-3$ . We measured the Bellman error as follows, where  $x_0, x_1, x_T$  are the current, immediate next, and future states:

```
pdist = metric_fn.apply(params, x0[:, None], xT[None])
pdist1 = metric_fn.apply(params, x1[:, None], xT[None])
td_target = (1 - gamma) * (x1 == xT[None, :, 0])
            + gamma * jax.nn.softmax(pdist1, axis=1)
bellman = optax.kl_divergence(
            td_target, jax.nn.softmax(pdist, axis=1)
        ).mean()
```

For the success rates in the Left subplot, we stratify goals into “easy” (less than 100 steps away, under an optimal policy) and “distant” (more than 100 steps away).

We repeated this experiment 10 times for generate the standard errors shown in both the Left and Right subplots.

<sup>1</sup>We used this maze in preliminary versions of other experiments, but opted for the larger maze in the other paper experiments because the results were easier to visualize.