INVARIANCE TO PLANNING IN GOAL-CONDITIONED RL

Anonymous authors
Paper under double-blind review

ABSTRACT

We study goal-conditioned RL through the lens of generalization, but not in the traditional sense of random augmentations and domain randomization. Rather, we aim to learn goal-directed policies that generalize with respect to the horizon: after training to reach nearby goals (which are easy to learn), these policies should succeed in reaching distant goals (which are quite challenging to learn). In the same way that invariance is closely linked with generalization is other areas of machine learning (e.g., normalization layers make a network invariant to scale, and therefore generalize to inputs of varying scales), we show that this notion of horizon generalization is closely linked with invariance to planning: a policy navigating towards a goal will select the same actions as if it were navigating to a waypoint en route to that goal, implying that a policy trained to reach nearby goals would succeed at reaching arbitrarily distant goals. Our theoretical analysis proves that both horizon generalization and planning invariance are possible, under some assumptions. We present new experimental results and recall findings from prior work in support of our theoretical results. Taken together, our results open the door to studying how techniques for invariance and generalization developed in other areas of machine learning might be adapted to achieve this alluring property.

1 INTRODUCTION

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 (Kakade, 2003). 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.

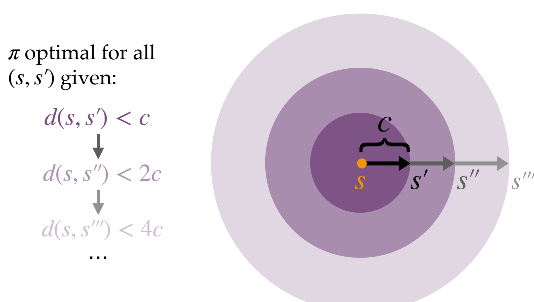


Figure 1: **Horizon generalization.** A policy generalizes over the horizon if optimality over all start-goal pairs (s, s') a small temporal distance $d(s, s') < c$ apart (say, in the training set) leads to optimality over all possible start-goal pairs.

The classical solution to the "curse of horizon" is dynamic programming (Bellman, 1966; Dijkstra, 1959) (i.e., temporal difference learning (Sutton, 2018)): 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 forgoing mental models associated with "standard" ML problems, such as generalization and invariance. This paper will discuss how these tools provide new ways of thinking about long-horizon problems.

While there is ample prior work studying generalization in RL, prior work almost exclusively focuses on either *(i) perceptual* changes (e.g., changes in lighting conditions) *(ii)* simple randomizations of simulator parameters, or *(iii)* mapping together states and actions with the same reward or value function. In this paper, 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. 1)? While these goals have been seen in different contexts before (e.g., reaching this goal from a different, closer state within the training set), they have never been used in learning long-horizon tasks. Horizon generalization is a type of extrapolation (Packer et al., 2018); 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. 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 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. 3). This reasoning also explains how a policy exhibiting horizon generalization must solve problems: by recursion, the policy maps 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.

The main contributions of this paper are precise definitions and proofs of existence for horizon generalization and invariance to planning. We theoretically show that policies defined with respect to a quasimetric are planning invariant and can exhibit horizon generalization. We support these theoretical results with experiments, where we demonstrate horizon generalization in learned, planning-invariant policies in a high-dimensional, standard RL benchmark: policies trained to navigate between nearby start-goal pairs can successfully navigate between far apart start-goal pairs, despite having never seen such long-horizon start-goal pairs during training.

The main takeaway from this paper is that there are rich notions of generalization *over the horizon* unique to the RL problem (and not the exclusive purview of TD methods). In addition, existing quasimetric methods *already* exhibit this form of generalization in high-dimensional settings. By theoretically and empirically linking planning with this form of generalization, our work suggests practical ways (i.e. quasimetric methods) to achieve powerful notions of generalization from short to long horizons.

2 Related Work

Our work builds upon prior work in goal-conditioned RL and generalization in RL. Section 5 returns to the discussion of prior work in light of our analysis.

Goal-conditioned RL. The problem of goal-conditioned RL, or learning goal-oriented behavior, dates to the early days of AI (Laird et al., 1987; Newell, 1959) but has received renewed attention in recent years (Chane-Sane et al., 2021; Chen et al., 2021; Colas et al., 2022; Janner et al., 2021; Ma et al., 2022; Schroecker and Isbell, 2020; Yang et al., 2022). Goal-conditioned RL relieves the burden of specifying rewards, allowing users to instead provide a single goal observation. Some of the excitement in goal-conditioned RL is a reflection of the recent success of self-supervised methods in computer vision (e.g., stable diffusion (Rombach et al., 2022)) and NLP (GPT-4 (OpenAI et al., 2024)): if these methods can achieve intriguing emergent properties, might a self-supervised approach to RL unlock emergent properties for RL?

Generalization in RL. Prior work on generalization in RL mostly focuses on variations in *perception* (Cobbe et al., 2019; Laskin et al., 2020; Stone et al., 2021) (or, similarly, e.g., across levels of a game (Farebrother et al., 2018; Justesen et al., 2018; Nichol et al., 2018; Zhang et al., 2018)). Similarly, work on robust RL (which measures a worst-case notion of generalization) usually randomly perturbs the physics parameters (Eysenbach and Levine, 2022; Igl et al., 2019; Moos et al., 2022; Packer et al., 2018; Tessler et al., 2019)). Our paper will study a different form of generalization: without changing the dynamics or the observations, can a policy trained on nearby goals succeed in reaching distant goals?

This form of generalization is related to yet distinct from other state abstractions for goal-conditioned learning such as bisimulation Castro and Precup (2010); Ferns et al. (2011); Hansen-Estruch et al. (2022); Zhang et al. (2021a), which assume a reward structure in the environment, and various state representation approaches Anand et al. (2019); Castro et al. (2021); Ghosh et al. (2019); Jain et al. (2023); Rakelly et al. (2021) which focus on learning representations helpful for selecting actions without consideration for long-horizon tasks. However, unlike prior work which maps MDPs with similar dynamics over some *fixed* horizon to the same latents, horizon generalization is from *short to long horizons* — by only training over short horizon tasks, can the policy generalize to long-horizon tasks over the covered state space? This form of generalization (to our knowledge) has not been directly addressed by other state abstraction methods. Prior work that has specifically looked at performing out-of-distribution long-horizon tasks have made assumptions about the environment, such as access to external planners Myers et al. (2024b); Shah and Levine (2022); Singh et al. (2023) or human demonstrations Mandlekar et al. (2021). Our contribution is to tackle the problem of generalization over the time-horizon in the context of modern, scalable deep RL methods without these additional structural assumptions. Instead of assuming structure in the environment, we study how planning invariance can be enforced over the state representation geometry used for decision-making.

3 PRELIMINARIES

We consider a controlled Markov process $\mathcal{M}$ with state space $\mathcal{S}$, action space $\mathcal{A}$, and dynamics $p(s' \mid s, a)$. The agent interacts with the environment by selecting actions according to a policy $\pi(a \mid s)$, 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 $\mathcal{S} \times \mathcal{S} \to \mathbb{R}$ must be positive everywhere except for a zero diagonal (positive definiteness). We will denote the set of all distances as $\mathcal{D}$:

$$\mathcal{D} \triangleq \{d : \mathcal{S} \times \mathcal{S} \to \mathbb{R} : d(s, s) = 0, d(s, s') > 0 \text{ for each } s, s' \in \mathcal{S} \text{ where } s \neq s'\}. \tag{1}$$

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

$$\mathcal{Q} \triangleq \{d \in \mathcal{D} : d(s, g) \leq d(s, w) + d(w, g) \text{ for all } s, g, w \in \mathcal{S}\}. \tag{2}$$

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 $s \to g$ may be a completely different task from navigating $g \to s$.

A particular quasimetric of note here is the *successor state distance* (Myers et al., 2024a), d_{SD}^{γ}, defined as

$$d_{\text{SD}}^{\gamma}(s, g) \triangleq \min_{\pi} \left[\log \frac{p_{\gamma}^{\pi}(\mathfrak{s}_K = g \mid \mathfrak{s}_0 = g)}{p_{\gamma}^{\pi}(\mathfrak{s}_K = g \mid \mathfrak{s}_0 = s)} \right], \text{ where } K \sim \text{Geom}(1 - \gamma). \tag{3}$$

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

$$p_{\gamma}^{\pi}(\mathfrak{s}_K = g \mid \mathfrak{s}_0 = s) \triangleq \sum_{t=0}^{\infty} \gamma^t p^{\pi}(\mathfrak{s}_t = g \mid \mathfrak{s}_0 = s). \tag{4}$$

The distance d_{SD}^{γ} 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 γ. 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_{SD}^{γ}-minimizing policy is the optimal policy. The related *successor distance with actions* $d_{\text{SD}}^{\gamma}(s, a, g)$ (Myers et al., 2024a) 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:

$$d_{\text{SD}}^{\gamma}(s, a, g) \triangleq \min_{\pi} \left[\log \frac{p_{\gamma}^{\pi}(\mathfrak{s}_K = g \mid \mathfrak{s}_0 = g)}{p_{\gamma}^{\pi}(\mathfrak{s}_K = g \mid \mathfrak{s}_0 = s, a)} \right], \text{ where } K \sim \text{Geom}(1 - \gamma) \tag{5}$$

where the *discounted state occupancy measure with actions* is defined as

$$p_{\gamma}^{\pi}(\mathfrak{s}_K = g \mid \mathfrak{s}_0 = s, a) \triangleq \sum_{t=0}^{\infty} \gamma^t p^{\pi}(\mathfrak{s}_t = g \mid \mathfrak{s}_0 = s, a). \tag{6}$$

4 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.

4.1 Intuition for Planning Invariance and Horizon Generalization

Many prior works have found that augmenting goal-conditioned policies with planning can significantly boost performance (Park et al., 2024; Savinov et al., 2018): 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. 2).

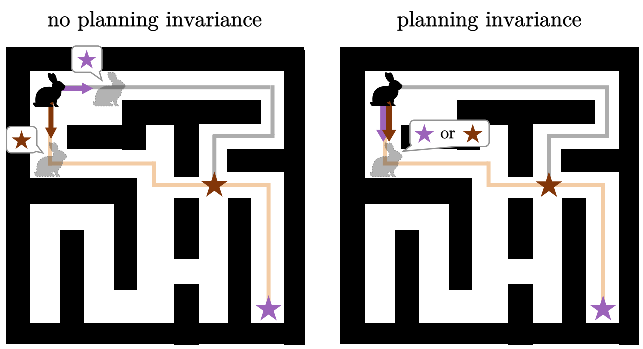


Figure 2: **Visualizing planning invariance.** Planning invariance (Definition 1) means that a policy should take similar actions when directed towards a goal (purple arrow and purple star) as when directed towards an intermediate waypoint (brown arrow and brown star). We visualize a policy with *(Right)* 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.

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. 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); these invariances lead to generalization over the transformed inputs (Benton et al. (2020); Cohen and Welling (2016); Rowley et al. (1998)). 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. 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, explaining 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 (Liu et al., 2023; Wang and Isola, 2022a;b) and show that quasimetric policies, where latents obey the triangle inequality, are invariant to planning.

4.2 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

$$\textbf{plan} \triangleq \{\textsc{Plan} : \mathcal{S} \times \mathcal{A} \times \mathcal{P}(\mathcal{S}) \mapsto \mathcal{P}(\mathcal{S})\} \tag{7}$$

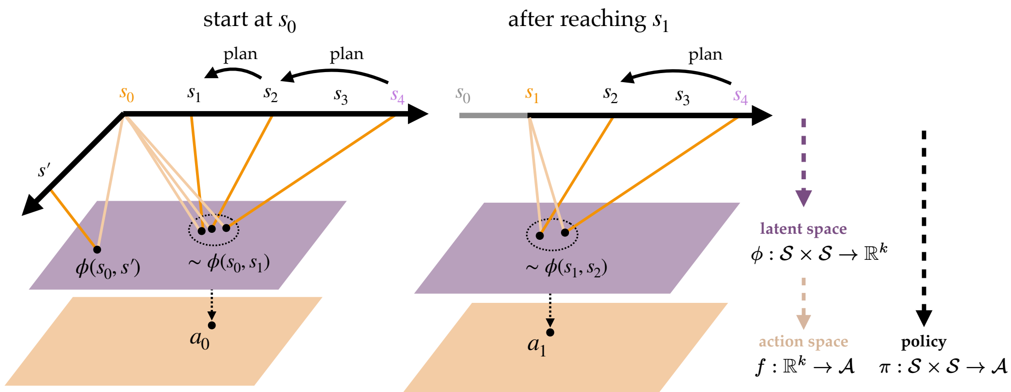


Figure 3: **Invariance to planning leads to horizon generalization.** *(Left)* Invariance to planning maps $(s_0, \{s_1, s_2, s_4\})$ together in latent space, which results in a shared optimal action. *(Right)* After successfully reaching the closest waypoint s_1 in 1 step, the next optimal action is also shared, meaning any trajectory of length 2 is optimal. We can repeat this argument for trajectories of length $4, 8, \ldots$ until the entire reachable state space is covered.

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

$$\mathbf{plan}^{\text{FIX}} \triangleq \{\textsc{Plan}^{\text{FIX}} : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \mapsto \mathcal{S}\} \subset \mathbf{plan}\,. \tag{8}$$

Our analysis in the rest of this section will focus on the simpler "fixed" setting of $\textsc{Plan}^{\text{FIX}} \in \mathbf{plan}^{\text{FIX}}$. We will use w or $w_{\textsc{Plan}}$ to denote the waypoint produced by $\textsc{Plan}^{\text{FIX}}(s, g)$. 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 Appendix C.

There are several different types of planning algorithms one might consider (e.g., Dijkstra's algorithm (Dijkstra, 1959), A* (Hart et al., 1968), RRT (LaValle and Kuffner, 2001)). Importantly, the constraints of a quasimetric (see Section 3) 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. 2):

Definition 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 $\pi(a \mid s, G)$ where $G \in \mathcal{P}(\mathcal{S})$, we say that $\pi(a \mid s, G)$ is invariant under planning operator* $\textsc{Plan} \in \mathbf{plan}$ *if and only if*

$$\pi(a \mid s, G) = \pi(a \mid s, W), \text{ where } W \sim \textsc{Plan}(s, a, G). \tag{9}$$

In the single-goal, controlled ("fixed") case,

$$\pi(a \mid s, g) = \pi(a \mid s, w), \text{ where } w = \textsc{Plan}^{\text{FIX}}(s, g). \tag{10}$$

Our second key definition is horizon generalization (see Fig. 1):

Definition 2 (Horizon generalization)**.** *In the single-goal, controlled ("fixed") case, a policy $\pi(a \mid s, g)$* **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*.

4.3 Existence of planning invariance

With these notions of planning invariance and horizon generalization in hand, we will consider planning algorithms $\textsc{Plan}_d^{\text{FIX}} \in \mathbf{plan}^{\text{FIX}}$ that acquire a quasimetric $d(s, g)$ and output a single waypoint $w \in \mathcal{S}$:

$$\textsc{Plan}_d^{\text{FIX}}(s, g) = w_{\textsc{Plan}} \in \arg\min_{w \in \mathcal{S}} d(s, w) + d(w, g). \tag{11}$$

where, by the triangle inequality, we have $d(s, w_{\textsc{Plan}}) + d(w_{\textsc{Plan}}, g) = d(s, g)$.

Theorem 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 $\pi_d^{\text{FIX}}(a \mid s, g)$ and planning operator $\textsc{Plan}_d^{\text{FIX}} \in \textbf{plan}^{\text{FIX}}$ such that $\pi_d^{\text{FIX}}(a \mid s, g) = \pi_d^{\text{FIX}}\big(a \mid s, w \text{ for } w = \textsc{Plan}_d^{\text{FIX}}(s, g)\big)$.*

The proof is in Appendix C.1. 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 $\pi_d(a \mid s, g) = \pi_d(a \mid s, w_{\textsc{Plan}})$; rather, the weaker condition $d(s, \pi_d(a \mid s, g), g) = d(s, \pi_d(a \mid s, w_{\textsc{Plan}}), w_{\textsc{Plan}})$ 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 Appendix C.3.

4.4 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 3 (Quasimetric policy)**.** *We define the quasimetric policy as some policy $\pi_d^{\text{FIX}}(a \mid s, g)$ where*

$$\pi_d^{\text{FIX}}(a \mid s, g) \in \textsc{Opt}_d(s, g) \triangleq \arg\min_{a \in \mathcal{A}} d(s, a, g)$$

and $d(s, a, g)$ is the successor distance with actions (Eq. 5). We can extend this definition to stochastic settings (see Definition 8) where $\pi_d(a \mid s, G)$ is defined over state-goal distribution pairs.

Theorem 2 (Horizon generalization exists)**.** *A quasimetric policy $\pi_d^{\text{FIX}}(a \mid s, g)$ 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 $\pi_d^{\text{FIX}}(a \mid s, \cdot)$ 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 Appendix C.4.

Importantly, this property is *not* guaranteed for any arbitrary optimal goal-reaching policy on some restricted horizon:

Remark 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 Appendix C.5.

Combined, these results show that (1) planning invariance and horizon generalization, as defined in Section 4.2, 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.

4.5 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 6 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 ($c_0 \to 2c_0 \to 4c_0 \to \cdots$), the success rate decreases by a factor of η. We will refer to η 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,

$$\text{REACH}_{wc} = 1 + \eta(2-1) + \eta^2(4-2) + \eta^3(8-4) + \cdots = \begin{cases} 1 + \eta\frac{1}{1-2\eta} & \text{if } 0 < \eta < 1/2 \\ \infty & \text{if } \eta \geq 1/2 \end{cases}. \quad (12)$$

We visualize this simple analysis in Fig. 8. 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. 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.

5 METHODS FOR PLANNING INVARIANCE: OLD AND NEW

In this section we discuss how planning invariance relates to several classes of RL algorithms. Appendix D discusses several new directions for designing RL algorithms that are invariant to planning. Appendix F 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" (Ziebart et al., 2008) 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 (Cormen et al., 2022, 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 (Myers et al., 2024a; Pitis et al., 2020; Wang et al., 2023) 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 ($d(s,g) \leftarrow \min_w d(s,w) + d(w,g)$) (Cormen et al., 2022, p. 609). This path relaxation is exactly the notion of planning used in our theoretical construction (Theorem 1). Thus, these architectures are, by construction, invariant to planning! We use these architectures in our experiments in Section 6.

While quasimetric architectures are invariant to path relaxation, other prior methods (Lee et al., 2018; Tamar et al., 2016) 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.

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 (Beker et al., 2022; Chane-Sane et al., 2021; Savinov et al., 2018; Zhang et al., 2021b), model-based methods (Chua et al., 2018; Lowrey et al., 2018; Nagabandi et al., 2018; Sutton, 1991;

Table 1: Summary of methods and modifications tested

Method	**Description**	**Losses**	**Critics**
CRL	Contrastive RL (Eysenbach et al., 2022)	$\{\mathcal{L}_{\text{fwd}}, \mathcal{L}_{\text{bwd}}, \mathcal{L}_{\text{sym}}\}$	$\{d_{\ell_2}, d_{\text{MLP}}\}$
SAC	Soft Actor-Critic (Haarnoja et al., 2018)	$\{\mathcal{L}_{\text{sac}}\}$	$\{Q_{\text{MLP}}\}$
CMD-1	Contrastive metric distillation (Myers et al., 2024a)	$\{\mathcal{L}_{\text{bwd}}\}$	$\{d_{\text{MRN}}\}$

(a) Losses

$\mathcal{L}_{\text{fwd}}$	InfoNCE loss: predict goal g from current state-action (s, a) pair (Sohn, 2016)
$\mathcal{L}_{\text{bwd}}$	Backward InfoNCE loss: predict current state and action (s, a) from future state g (Bortkiewicz et al., 2024)
$\mathcal{L}_{\text{sym}}$	Symmetric contrastive loss: combine the forward and backward contrastive losses (Radford et al., 2021)
$\mathcal{L}_{\text{sac}}$	Temporal difference loss (Haarnoja et al., 2018)

(b) Architectures

d_{ℓ_2}	L2-distance parameterized architecture, uses $\|\phi(s) - \psi(g)\|$ as a distance/critic (Eysenbach et al., 2024)
d_{MLP}	Uses multi-layer perceptron (MLP) to parameterize the distance/critic (Burr, 1986; Rosenblatt, 1961)
d_{MRN}	Metric residual network, uses a quasimetric architecture to parameterize the distance/critic (Liu et al., 2023)
Q_{MLP}	MLP-parameterized Q-function (Haarnoja et al., 2018)

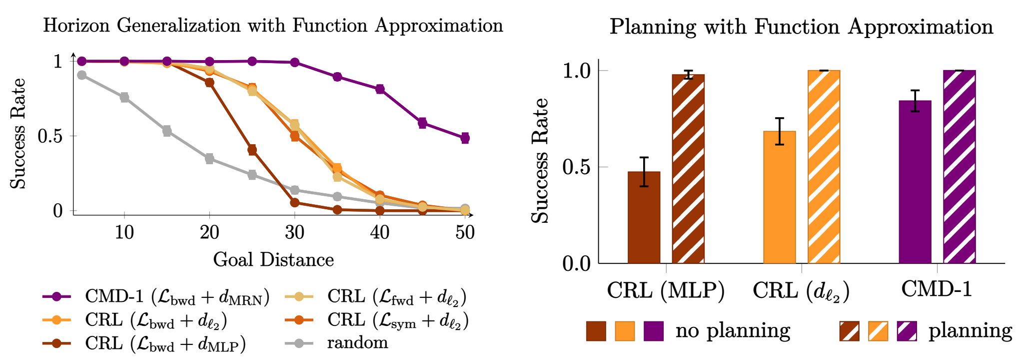


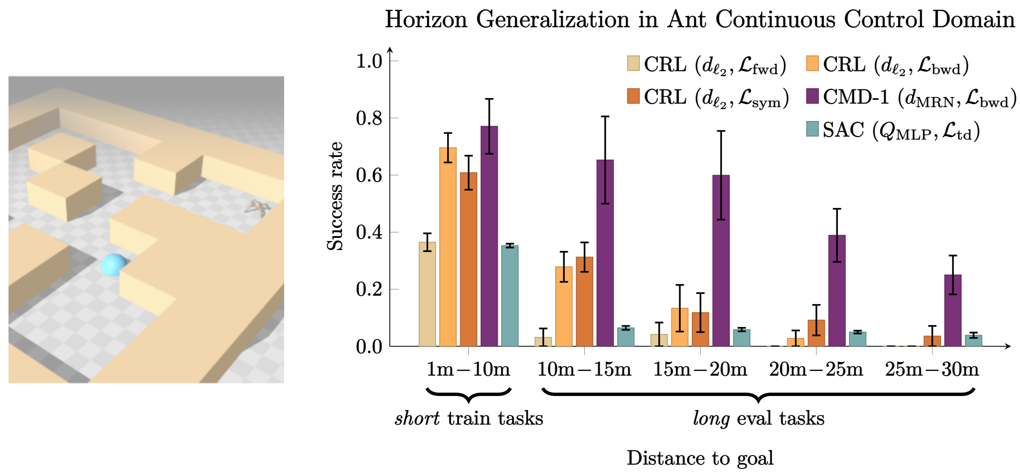
Figure 4: **Quantifying horizon generalization and invariance to planning.** On a simple navigation task, we collect short trajectories and train two goal-conditioned policies, comparing both to a random policy. *(Left)* We evaluate on (s, g) pairs of varying distances, observing that metric regression 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. Appendix Fig. 6 shows a version of this plot that also includes the tabular setting.

Williams et al., 2017), collocation methods (Rybkin et al., 2021), and hierarchical methods (Kulkarni et al., 2016; Nasiriany et al., 2019; Parascandolo et al., 2020; Pertsch et al., 2020). 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. 10). 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 (Chane-Sane et al., 2021; Ghugare et al., 2024) has argued that data augmentation provides another avenue for achieving the benefits typically associated with planning or dynamic programming.

6 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



(a) Ant Environment

(b) Success rates stratified by distance to goal

Figure 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 η from Section 4.5. Results are plotted with standard errors across random seeds.

why horizon generalization is a useful notion even when considering temporal difference methods (Section 6.2).

We start with a didactic, tabular navigation task (Fig. 11), 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. 6 *(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. 6 *(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 of each policy (on distant start-goal pairs) when the policy is conditioned on the goal versus on a waypoint. See Appendix G for details. As shown in Fig. 6 *(Top Right)*, the "metric regression with quasimetric" policy exhibits stronger planning invariance, supporting our theoretical claim that (Theorem 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 (Eysenbach et al., 2022) for estimating the distances, comparing different architectures and loss functions. The results in Fig. 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 (Myers et al., 2024a). Intuitively this makes sense, as this method was explicitly designed to exploit the triangle inequality, which is closely linked to planning invariance. Fig. 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. 4 *(Left)* to estimate the horizon generalization parameter η, and used the data from the *(Right)* to compute the ratio of performance with and without planning. Fig. 7 shows these data as a scatter plot. These two quantities are well correlated, supporting Theorem 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 4.5). 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. 4, note that the success rate on the easiest goals is strictly less than 1).

6.1 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. 5). We start by running a series of experiments to compare the horizon generalization of different learning algorithms (CRL (Eysenbach et al., 2022) and SAC (Haarnoja et al., 2018)) and distance metric architectures (details in Appendix G). The results in Fig. 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 5 for a summary of the methods used in these experiments.

6.2 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. 4), with two different architectures. The results in Fig. 9 show that the architecture that exhibits stronger horizon generalization (d_{ℓ_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 nonetheless prove useful for designing architectures that enable faster convergence of TD methods.

7 Conclusion

The aim of this paper 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 (Watkins and Dayan, 1992) 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 6 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 Appendix E, 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 5 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 meany 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.

REFERENCES

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.

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.

Richard Bellman. Dynamic Programming. *Science*, 153(3731):34–37, 1966.

Gregory Benton, Marc Finzi, Pavel Izmailov, and Andrew Gordon Wilson. Learning Invariances in Neural Networks. arXiv:2010.11882, 2020.

Michał Bortkiewicz, Władek Pałucki, Vivek Myers, Tadeusz Dziarmaga, Tomasz Arczewski, Łukasz Kuciński, and Benjamin Eysenbach. Accelerating Goal-Conditioned RL Algorithms and Research. arXiv:2408.11052, 2024.

David J. Burr. A Neural Network Digit Recognizer. *Proc. IEEE SMC*, pp. 1621–1625, 1986.

Pablo Castro and Doina Precup. Using Bisimulation for Policy Transfer in MDPs. In *AAAI Conference on Artificial Intelligence*, volume 24, pp. 1065–1070. 2010.

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.

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.

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.

Kurtland Chua, Roberto Calandra, Rowan McAllister, and Sergey Levine. Deep Reinforcement Learning in a Handful of Trials Using Probabilistic Dynamics Models. *Neural Information Processing Systems*, 31, 2018.

Karl Cobbe, Oleg Klimov, Chris Hesse, Taehoon Kim, and John Schulman. Quantifying Generalization in Reinforcement Learning. In *International Conference on Machine Learning*, pp. 1282–1289. 2019.

Taco S. Cohen and Max Welling. Group Equivariant Convolutional Networks. arXiv:1602.07576, 2016.

Cédric Colas, Tristan Karch, Olivier Sigaud, and Pierre-Yves Oudeyer. Intrinsically Motivated Goal-Conditioned Reinforcement Learning: A Short Survey. In *J. Artif. Intell. Res*. 2022.

Thomas H Cormen, Charles E Leiserson, Ronald L Rivest, and Clifford Stein. *Introduction to Algorithms*. MIT press, 2022.

Ew Dijkstra. A Note on Two Problems in Connexion With Graphs. *Numerische Mathematik*, 1:269–271, 1959.

Benjamin Eysenbach and Sergey Levine. Maximum Entropy RL (Provably) Solves Some Robust RL Problems. In *International Conference on Learning Representations*. 2022.

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.

Benjamin Eysenbach, Tianjun Zhang, Ruslan Salakhutdinov, and Sergey Levine. Contrastive Learning as Goal-Conditioned Reinforcement Learning. In *Neural Information Processing Systems*, volume 35, pp. 35603–35620. 2022.

Jesse Farebrother, Marlos C Machado, and Michael Bowling. Generalization and Regularization in DQN. arXiv:1810.00123, 2018.

Norm Ferns, Prakash Panangaden, and Doina Precup. Bisimulation Metrics for Continuous Markov Decision Processes. *SIAM Journal on Computing*, 40(6):1662–1714, 2011.

Dibya Ghosh, Abhishek Gupta, and Sergey Levine. Learning Actionable Representations With Goal Conditioned Policies. In *International Conference on Learning Representations*. 2019.

Raj Ghugare, Matthieu Geist, Glen Berseth, and Benjamin Eysenbach. Closing the Gap Between TD Learning and Supervised Learning - a Generalisation Point of View. In *Twelfth International Conference on Learning Representations*. 2024.

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.

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*. 2022.

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.

E. Hellinger. Neue Begründung Der Theorie Quadratischer Formen Von Unendlichvielen Veränderlichen. *Journal Für Die Reine Und Angewandte Mathematik*, 1909(136):210–271, 1909.

Jiaxin Huang, Shixiang Shane Gu, Le Hou, Yuexin Wu, Xuezhi Wang, Hongkun Yu, and Jiawei Han. Large Language Models Can Self-Improve. arXiv:2210.11610, 2022.

Maximilian Igl, Kamil Ciosek, Yingzhen Li, Sebastian Tschiatschek, Cheng Zhang, Sam Devlin, and Katja Hofmann. Generalization in Reinforcement Learning With Selective Noise Injection and Information Bottleneck. *Neural Information Processing Systems*, 32, 2019.

Geoffrey Irving, Paul Christiano, and Dario Amodei. AI Safety via Debate. arXiv:1805.00899, 2018.

Arnav Kumar Jain, Lucas Lehnert, Irina Rish, and Glen Berseth. Maximum State Entropy Exploration Using Predecessor and Successor Representations. In *Neural Information Processing Systems*. 2023.

Michael Janner, Qiyang Li, and Sergey Levine. Offline Reinforcement Learning as One Big Sequence Modeling Problem. *Neural Information Processing Systems*, 34:1273–1286, 2021.

Niels Justesen, Ruben Rodriguez Torrado, Philip Bontrager, Ahmed Khalifa, Julian Togelius, and Sebastian Risi. Illuminating Generalization in Deep Reinforcement Learning Through Procedural Level Generation. arXiv:1806.10729, 2018.

Sham Machandranath Kakade. *On the Sample Complexity of Reinforcement Learning*. University of London, University College London (United Kingdom), 2003.

Tejas D Kulkarni, Karthik Narasimhan, Ardavan Saeedi, and Josh Tenenbaum. Hierarchical Deep Reinforcement Learning: Integrating Temporal Abstraction and Intrinsic Motivation. *Neural Information Processing Systems*, 29, 2016.

John E Laird, Allen Newell, and Paul S Rosenbloom. Soar: An Architecture for General Intelligence. *Artificial Intelligence*, 33(1):1–64, 1987.

Misha Laskin, Kimin Lee, Adam Stooke, Lerrel Pinto, Pieter Abbeel, and Aravind Srinivas. Reinforcement Learning With Augmented Data. *Neural Information Processing Systems*, 33:19884–19895, 2020.

Steven M. LaValle and James J. Kuffner. Randomized Kinodynamic Planning. *International Journal of Robotics Research*, 20(5):378–400, 2001.

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.

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.

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.

Kendall Lowrey, Aravind Rajeswaran, Sham Kakade, Emanuel Todorov, and Igor Mordatch. Plan Online, Learn Offline: Efficient Learning and Exploration via Model-Based Control. arXiv:1811.01848, 2018.

Yecheng Jason Ma, Jason Yan, Dinesh Jayaraman, and Osbert Bastani. How Far I'll Go: Offline Goal-Conditioned Reinforcement Learning via f-Advantage Regression. arXiv:2206.03023, 2022.

Ajay Mandlekar, Danfei Xu, Roberto Martín-Martín, Silvio Savarese, and Li Fei-Fei. Learning to Generalize Across Long-Horizon Tasks From Human Demonstrations. arXiv:2003.06085, 2021.

Janosch Moos, Kay Hansel, Hany Abdulsamad, Svenja Stark, Debora Clever, and Jan Peters. Robust Reinforcement Learning: A Review of Foundations and Recent Advances. *Machine Learning and Knowledge Extraction*, 4(1):276–315, 2022.

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 *Forty-First International Conference on Machine Learning*. 2024a.

Vivek Myers, Chunyuan Zheng, Oier Mees, Kuan Fang, and Sergey Levine. Policy Adaptation via Language Optimization: Decomposing Tasks for Few-Shot Imitation. In *Conference on Robot Learning*. 2024b.

Ofir Nachum, Haoran Tang, Xingyu Lu, Shixiang Gu, Honglak Lee, and Sergey Levine. Why Does Hierarchy (Sometimes) Work So Well in Reinforcement Learning? arXiv:1909.10618, 2019.

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*, pp. 7559–7566. 2018.

Soroush Nasiriany, Vitchyr Pong, Steven Lin, and Sergey Levine. Planning With Goal-Conditioned Policies. *Neural Information Processing Systems*, 32, 2019.

Allen Newell. Report on a General Problem-Solving Program. In *IFIP Congress*. 1959.

Alex Nichol, Vicki Pfau, Christopher Hesse, Oleg Klimov, and John Schulman. Gotta Learn Fast: A New Benchmark for Generalization in RL. arXiv:1804.03720, 2018.

OpenAI, Josh Achiam, Steven Adler, et al. GPT-4 Technical Report. *ArXiv*, abs/2303.08774, 2024.

Charles Packer, Katelyn Gao, Jernej Kos, Philipp Krähenbühl, Vladlen Koltun, and Dawn Song. Assessing Generalization in Deep Reinforcement Learning. arXiv:1810.12282, 2018.

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. arXiv:2004.11410, 2020.

Seohong Park, Dibya Ghosh, Benjamin Eysenbach, and Sergey Levine. Hiql: Offline Goal-Conditioned RL With Latent States as Actions. *Neural Information Processing Systems*, 36, 2024.

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.

Silviu Pitis, Harris Chan, Kiarash Jamali, and Jimmy Ba. An Inductive Bias for Distances: Neural Nets That Respect the Triangle Inequality. arXiv:2002.05825, 2020.

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.

Kate Rakelly, Abhishek Gupta, Carlos Florensa, and Sergey Levine. Which Mutual-Information Representation Learning Objectives Are Sufficient for Control? *Neural Information Processing Systems*, 34:26345–26357, 2021.

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.

F. Rosenblatt. *Principles of Neurodynamics: Perceptrons and the Theory of Brain Mechanisms*. Spartan Books, 1961.

Henry A Rowley, Shumeet Baluja, and Takeo Kanade. Rotation invariant neural network-based face detection. In *Proceedings. 1998 IEEE computer society conference on computer vision and pattern recognition (Cat. No. 98CB36231)*, pp. 38–44. 1998.

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.

Nikolay Savinov, Alexey Dosovitskiy, and Vladlen Koltun. Semi-Parametric Topological Memory for Navigation. arXiv:1803.00653, 2018.

Yannick Schroecker and Charles Isbell. Universal Value Density Estimation for Imitation Learning and Goal-Conditioned Reinforcement Learning. arXiv:2002.06473, 2020.

Dhruv Shah and Sergey Levine. ViKiNG: Vision-Based Kilometer-Scale Navigation With Geographic Hints. In *Robotics: Science and Systems XVIII*. 2022.

Ishika Singh, Valts Blukis, Arsalan Mousavian, Ankit Goyal, Danfei Xu, Jonathan Tremblay, Dieter Fox, Jesse Thomason, and Animesh Garg. ProgPrompt: Generating Situated Robot Task Plans Using Large Language Models. In *International Conference on Robotics and Automation*. 2023.

Kihyuk Sohn. Improved Deep Metric Learning With Multi-Class N-Pair Loss Objective. In *Neural Information Processing Systems*, volume 29. 2016.

Austin Stone, Oscar Ramirez, Kurt Konolige, and Rico Jonschkowski. The Distracting Control Suite–a Challenging Benchmark for Reinforcement Learning From Pixels. arXiv:2101.02722, 2021.

Richard S Sutton. Dyna, an Integrated Architecture for Learning, Planning, and Reacting. *ACM Sigart Bulletin*, 2(4):160–163, 1991.

Richard S. Sutton. Reinforcement Learning: An Introduction. *A Bradford Book*, 2018.

Aviv Tamar, Yi Wu, Garrett Thomas, Sergey Levine, and Pieter Abbeel. Value Iteration Networks. *Neural Information Processing Systems*, 29, 2016.

Joshua B Tenenbaum, Vin de Silva, and John C Langford. A Global Geometric Framework for Nonlinear Dimensionality Reduction. *Science*, 290(5500):2319–2323, 2000.

Chen Tessler, Yonathan Efroni, and Shie Mannor. Action Robust Reinforcement Learning and Applications in Continuous Control. In *International Conference on Machine Learning*, pp. 6215–6224. 2019.

Tongzhou Wang and Phillip Isola. Improved Representation of Asymmetrical Distances With Interval Quasimetric Embeddings. In *NeurIPS 2022 Neurreps Workshop Proceedings Track*. 2022a.

Tongzhou Wang and Phillip Isola. On the Learning and Learnability of Quasimetrics. In *Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. 2022b.

Tongzhou Wang, Antonio Torralba, Phillip Isola, and Amy Zhang. Optimal Goal-Reaching Reinforcement Learning via Quasimetric Learning. In *International Conference on Machine Learning*. 2023.

Christopher JCH Watkins and Peter Dayan. Q-Learning. *Machine Learning*, 8:279–292, 1992.

Alfred North Whitehead and Bertrand Russell. *Principia Mathematica to* 56*, volume 2. Cambridge University Press, 1927.

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 *IEEE International Conference on Robotics and Automation*, pp. 1714–1721. 2017.

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.

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*. 2021a.

Chiyuan Zhang, Oriol Vinyals, Remi Munos, and Samy Bengio. A Study on Overfitting in Deep Reinforcement Learning. arXiv:1804.06893, 2018.

Tianjun Zhang, Benjamin Eysenbach, Ruslan Salakhutdinov, Sergey Levine, and Joseph E Gonzalez. C-Planning: An Automatic Curriculum for Learning Goal-Reaching Tasks. arXiv:2110.12080, 2021b.

Brian D Ziebart, Andrew L Maas, J Andrew Bagnell, Anind K Dey, et al. Maximum Entropy Inverse Reinforcement Learning. In *AAAI Conference on Artificial Intelligence*, volume 8, pp. 1433–1438. 2008.

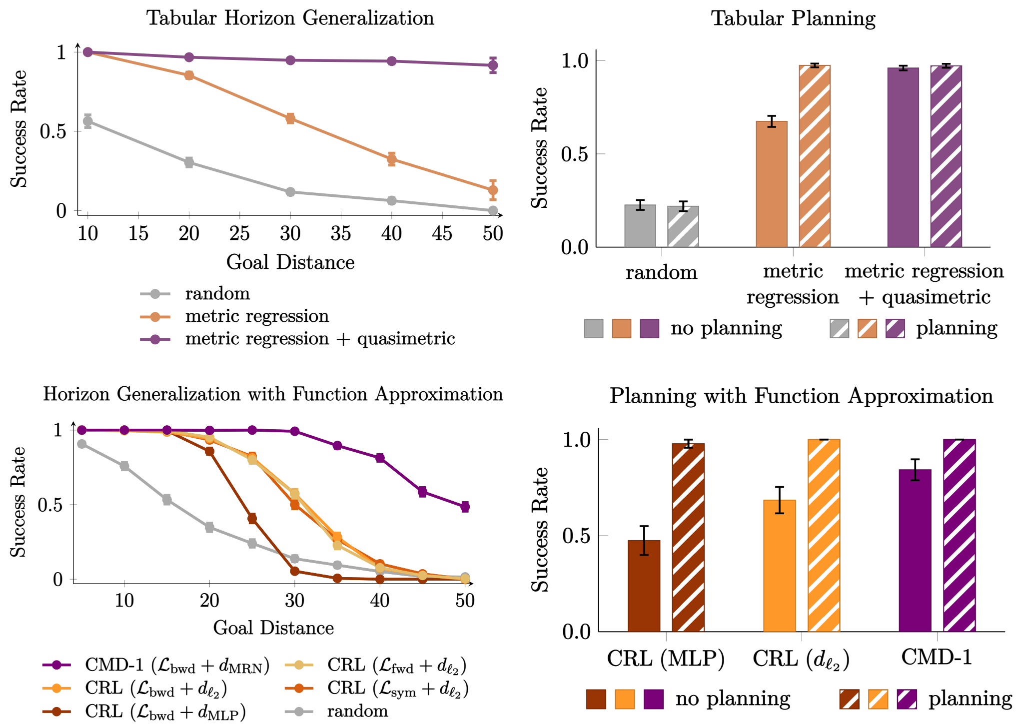


Figure 6: **Quantifying horizon generalization and invariance to planning.** On a simple navigation task, we collect short trajectories and train two goal-conditioned policies, comparing both to a random policy. *(Top Left)* We evaluate on (s, g) pairs of varying distances, observing that metric regression 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.

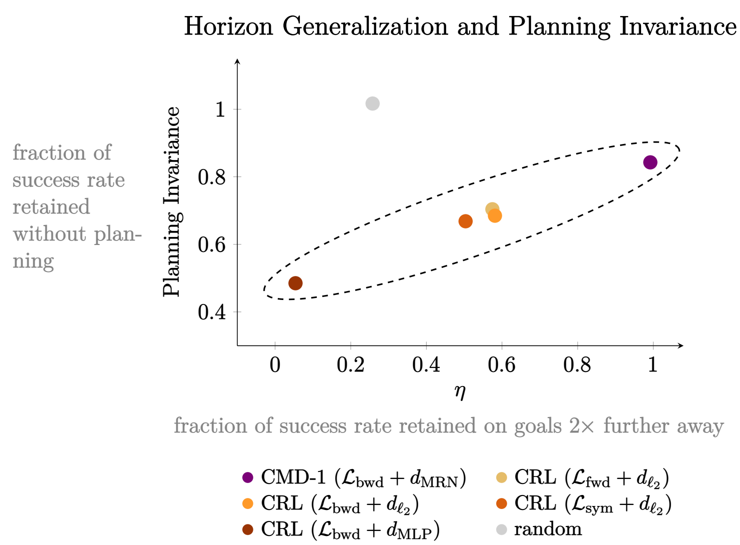


Figure 7: Quantifying horizon generalization (x-axis) and planning invariance (y-axis). See text Section 6 for more details.

A Additional Figures

For brevity, we include some of the figures referenced in the main text within this section.

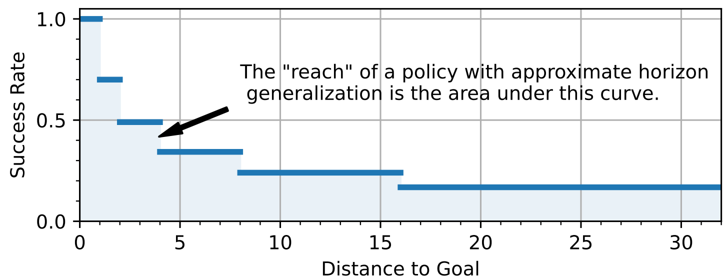


Figure 8: ***Approximate* horizon generalization is still useful:** SUCCESS when there is horizon generalization. When the success attenuation factor $\eta \geq 0.5$, the REACH goes to ∞. For a policy with no horizon generalization ($\eta = 0$), its REACH $= 1$.

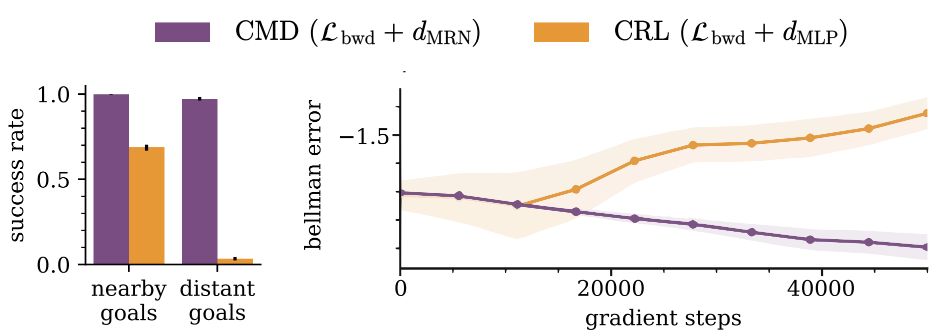


Figure 9: **Impact of horizon generalization on Bellman errors.** *(Left)* Two goal-reaching methods exhibit different horizon generalization. *(Right)* Despite neither method being trained with the Bellman loss, we observe that the method with stronger horizon generalization has a lower Bellman loss. Thus, understanding horizon generalization may be important even when using TD methods (which guarantee horizon generalization at convergence).

B Definition of Path Relaxation

Definition 4 (Path relaxation operator). *Let* $\textsc{Path}_d(s, G)$ *be the path relaxation operator over quasimetric* $d(s, G)$*. For any triplet of state and state distributions* $(s, W, G) \in \mathcal{S} \times \mathcal{P}(\mathcal{S}) \times \mathcal{P}(\mathcal{S})$*,*

$$\textsc{Path}_d(s, G) \triangleq \min_W d(s, W) + d(W, G). \tag{13}$$

In the controlled, fixed goal setting, define

$$\textsc{Path}_d^{\text{FIX}}(s, g) \triangleq \min_w d(s, w) + d(w, g). \tag{14}$$

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 (Liu et al., 2023; Wang and Isola, 2022a;b) innately satisfies the triangle inequality before seeing any training data.

Definition 5 (Path relaxation operator with actions). *Let* $\textsc{Path}_d(s, a, G)$ *be the path relaxation operator over quasimetric* $d(s, a, G)$*. For any triplet of state and state distributions* $(s, W, G) \in \mathcal{S} \times \mathcal{S} \times \mathcal{S}$*,*

$$\textsc{Path}_d(s, G) \triangleq \min_w d(s, W) + d(W, G). \tag{15}$$

In the controlled, fixed goal setting, define

$$\textsc{Path}_d^{\text{FIX}}(s, g) \triangleq \min_w d(s, w) + d(w, g). \tag{16}$$

C Proofs

In this section, we prove results discussed in Section 4.3 and versions of results in Section 4 for the general stochastic, distributional setting.

C.1 Planning invariance exists

Theorem 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* $\pi_d^{\text{FIX}}(a \mid s, g)$ *and planning operator* $\textsc{Plan}_d^{\text{FIX}} \in \textbf{plan}^{\text{FIX}}$ *such that* $\pi_d^{\text{FIX}}(a \mid s, g) = \pi_d^{\text{FIX}}\big(a \mid s, w \textit{ for } w = \textsc{Plan}_d^{\text{FIX}}(s, g)\big)$*.*

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 constrastive successor distances (Eq. 3). Define the (deterministic) planned waypoint as

$$w_{\textsc{Plan}} \leftarrow \textsc{Plan}_d^{\text{FIX}}(s, g) \in \operatorname*{arg\,min}_{w \in \mathcal{S}} d(s, w) + d(w, g). \tag{17}$$

We can then construct the following policy:

$$\pi_d^{\text{FIX}}(a \mid s, g) \in \textsc{Opt}_d(s, g) \triangleq \operatorname*{arg\,min}_{a \in \mathcal{A}} d(s, a, g). \tag{18}$$

and later restrict the selection of the action to reach waypoint $w_{\textsc{Plan}}$ to get planning invariance, where $w_{\textsc{Plan}} \in \arg\min_{w \in \mathcal{S}} d(s, w) + d(w, g)$. Applying this policy to $(s, w_{\textsc{Plan}})$,

$$\begin{aligned}
\pi_d^{\text{FIX}}(a \mid s, w_{\textsc{Plan}}) \in \textsc{Opt}_d(s, w_{\textsc{Plan}}) &\triangleq \operatorname*{arg\,min}_{a \in \mathcal{A}} d(s, a, w_{\textsc{Plan}}) \\
&= \operatorname*{arg\,min}_{a \in \mathcal{A}} d(s, a, w_{\textsc{Plan}}) + d(w_{\textsc{Plan}}, g) \\
&= d(s, w_{\textsc{Plan}}) + d(w_{\textsc{Plan}}, g) \\
&\subseteq \operatorname*{arg\,min}_{a \in \mathcal{A}} d(s, a, g) \\
&= \textsc{Opt}_d(s, g).
\end{aligned} \tag{19}$$

Thus, for a given deterministic planning algorithm defined as in Eq. (17), there exists some deterministic policy $\pi_d^{\text{FIX}}(a \mid s, g) = \pi_d^{\text{FIX}}(a \mid s, w_{\textsc{Plan}}) \in \textsc{Opt}_d(s, w_{\textsc{Plan}}) \subseteq \textsc{Opt}_d(s, g)$ which is planning invariance. $\square$

C.2 QUASIMETRIC OVER DISTRIBUTIONS

Definition 6 (Quasimetric over distributions)**.** *For a given quasimetric $d_{\text{QM}} \in \mathcal{S}$, we define the quasimetic over distributions as*

$$d_{\text{QMD}}(L, M) = \Big(\int_{\mathcal{S}\times\mathcal{S}} p_L(l) p_M(m) d_{\text{QM}}(L, M) \,\mathrm{d}l \,\mathrm{d}m \Big) \times \Big(1 - \int_{\mathcal{S}} \sqrt{p_L(s) p_M(s)} \,\mathrm{d}s \Big). \quad (20)$$

We show Definition 6 is a valid quasimetric.

Proof. Note that we can rewrite $d(L, M) = f(L, M) \cdot g(L, M)$ where

$$f(L, M) = \int_{\mathcal{S}\times\mathcal{S}} p_L(l) p_M(m) d(L, M) \,\mathrm{d}l \,\mathrm{d}m \quad (21)$$

$$g(L, M) = 1 - \int_{\mathcal{S}} \sqrt{p_L(s) p_M(s)} \,\mathrm{d}s. \quad (22)$$

We note that $g(L, M)$ is also known as the Hellinger distance, which is a valid metric defined over probability distributions (Hellinger, 1909). Checking the quasimetric conditions,

1. **Positive semi-definiteness:** $d(M, M) = 0$ trivially because $g(M, M) = 0$. For $M \neq N$, $d(L, M) > 0$ given $d(L, M)$ is a quasimetric and $g(L, M)$ is a metric.
2. **Triangle inequality:** Both $g(L, M)$ and $f(L, M)$ satisfy the triangle inequality. So $g(L, M) + g(M, N) \geq g(L, N)$ and $f(L, M) + f(M, N) \geq f(L, N)$ because $d(s, g)$ is a quasimetric. Multiplying the two sides of these two inequalities, we get $d(L, M) + d(M, N) \geq d(L, N)$ as desired.

Note that, here, we could replace $g(L, M)$ with any metric defined over two probability distributions (i.e. Jensen-Shannon divergence) – the resulting $d(L, M)$ would still be a quasimetric. □

C.3 QUASIMETRICS, POLICIES, AND PLANNING INVARIANCE (STOCHASTIC SETTING)

Utilizing a *quasimetric over distributions $d(L, M)$* like one defined in Definition 6, we can define a *distributional quasimetric over actions*. This is the stochastic generalization of the *successor distance with actions* in Eq. (5).

Definition 7 (Quasimetric over actions in general stochastic setting)**.** *Assume $d(s, g)$ is the Contrastive Successor Distance Myers et al. (2024a). Define the stochastic-setting quasimetric over actions as*

$$d(s, a, G) \triangleq \int_{\mathcal{S}} p(s' \mid s, a)(d(s, s') + d(s', G)) \,\mathrm{d}s'$$

where $p(s' \mid s, a)$ is the distribution over next-step states after taking action a from state s.

Definition 8 (Quasimetric policy in general stochastic setting)**.** *Extending the deterministic quasimetric policy to stochastic settings,*

$$\pi_d(a \mid s, G) \in \textsc{Opt}_d(s, G) \triangleq \arg\min_a d(s, a, G).$$

The existence of planning invariance in stochastic settings follows from these quasimetric definitions.

Lemma 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*

$$\pi_d(a \mid s, G) \in \arg\min_{a\in\mathcal{A}} d(s, a, G)$$

where $\pi_d(a \mid s, G) = \pi_d(a \mid s, W)$, and planning operator

$$\textsc{Plan}_d(s, a, G) = W_{\text{PLAN}} \in \arg\min_{W\in\mathcal{P}(\mathcal{S})} (d(s, a, W) + d(W, G)).$$

Proof. For any s, G pairs,

$$\min_a d(s,a,G) = \min_a \int_{\mathcal{S}} p(s' \mid s,a)(d(s,s') + d(s',G))\, \mathrm{d}s' \tag{23}$$

$$= \min_W \min_a \int_{\mathcal{S}} p(s'|s,a)(d(s,s') + d(s',W) + d(W,G))\, \mathrm{d}s' \qquad (\triangle\text{-ineq.})$$

$$= \min_a \min_{W \sim p(w|s)} \int_{\mathcal{S}} p(s'|s,a)(d(s,s') + d(s',W))\, \mathrm{d}s' + d(W,G) \tag{24}$$

$$= \min_a \min_{W \sim p(w|s)} d(s,a,W) + d(W,G) \tag{25}$$

Now, applying this policy to state-waypoint pair (s, W_{PLAN}),

$$\begin{aligned} \pi(a|s, W_{\text{PLAN}}) &\in \textsc{Opt}_d(s, W_{\text{PLAN}}) \\ &\triangleq \arg\min_{a \in \mathcal{A}} d(s,a,W_{\text{PLAN}}) \\ &= \arg\min_{a \in \mathcal{A}} d(s,a,W_{\text{PLAN}}) + d(W_{\text{PLAN}}, G) \\ &\subseteq \arg\min_{a \in \mathcal{A}} d(s,a,G) \end{aligned}$$

as desired. Thus, for the given stochastic planning algorithm, there exists some policy $\pi_d(a \mid s, G) = \pi_d(a \mid s, W_{\text{PLAN}}) \in \textsc{Opt}_d(s, W_{\text{PLAN}}) \subseteq \textsc{Opt}_d(s, G)$, which is planning invariance. □

C.4 Horizon generalization exists

Theorem 2 (Horizon generalization exists)**.** *A quasimetric policy $\pi_d^{\text{FIX}}(a \mid s, g)$ 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 $\pi_d(a \mid s, G)$ defined over state-goal distribution pairs (s, G). See earlier sections in Appendix C.3 for quasimetric, policy, and planning definitions over distributions:

Lemma 5 (Horizon generalization exists, stochastic settings)**.** *A quasimetric policy $\pi_d(a \mid s, G)$ 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 $\pi_d(a \mid s, G)$.

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 $\pi_d(a \mid S, S')$, for some waypoint distribution $W_{\text{PLAN}} \in \arg\min_{W \in \mathcal{D}_n'} (d(s,a,W) + d(W,G))$ over distributions $G \in \mathcal{D}_n'$:

$$\pi_d(a \mid s, G) \in \arg\min_{a \in \mathcal{A}} d(s,a,W_{\text{PLAN}}).$$

To show that there always exists some planned waypoint distribution W_{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,

$$\begin{aligned} d(s,a,G) &= \min_{W \in \mathcal{D}'_n} (d(s,a,W) + d(W,G)) \\ &= d(s,a,W_{\text{PLAN}}) + d(W_{\text{PLAN}}, G) \\ &= \min_{W_{\text{OUT}} \in \mathcal{D}'_n \setminus \mathcal{D}_n} d(s,a,W_{\text{OUT}}) + d(W_{\text{OUT}}, G) \\ &= \min_{W_{\text{OUT}} \in \mathcal{D}'_n \setminus \mathcal{D}_n} \min_{W_{\text{IN}} \in \mathcal{D}_n} (d(s,a,W_{\text{IN}}) + d(W_{\text{IN}}, W_{\text{OUT}})) + d(W_{\text{OUT}}, G) \\ &= \min_{W_{\text{IN}} \in \mathcal{D}_n} \min_{W_{\text{OUT}} \in \mathcal{D}'_n \setminus \mathcal{D}_n} d(s,a,W_{\text{IN}}) + (d(W_{\text{IN}}, W_{\text{OUT}}) + d(W_{\text{OUT}}, G)) \\ &= \min_{W_{\text{IN}} \in \mathcal{D}_n} d(s,a,W_{\text{IN}}) + d(W_{\text{IN}}, G) \qquad (\triangle\text{-ineq}) \\ &= d(s,a,W_{\text{PLAN, IN}}) + d(W_{\text{PLAN, IN}}, G), \end{aligned}$$

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 $\pi_d(a \mid s, G)$,

$$\begin{aligned} \pi_d(a \mid (s, W_{\text{PLAN}}) \in \mathcal{B}_n) &\in \arg\min_{a \in \mathcal{A}} d(s,a,W_{\text{PLAN}}) && \text{(inductive assumption)} \\ &\subseteq \arg\min_{a \in \mathcal{A}} d(s,a,G). && \text{(Lemma 4: planning invariance)} \end{aligned}$$

Therefore,e, 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 2 follows. □

C.5 Horizon generalization is nontrivial

Remark 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 π^{H+1} where $\pi^{H+1}(a \mid s, g')$ returns an action in the suboptimal set $\mathcal{A} \setminus \mathcal{A}'$, and π^{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. □

D New Methods for Planning Invariance

While the aim of this paper 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. 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?[1] The fact that plans over representations sometimes correspond to geodesics (Eysenbach et al., 2024; Tenenbaum et al., 2000) hints that this may be possible.

[1]The construction in our proof is a degenerate case of this, where the internal representations are equal to the output actions.

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 hierarchicies may aid in exploration (Nachum et al., 2019), 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 (Chane-Sane et al., 2021; Lee et al., 2024). 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?

E 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 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 studied in this paper. 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.

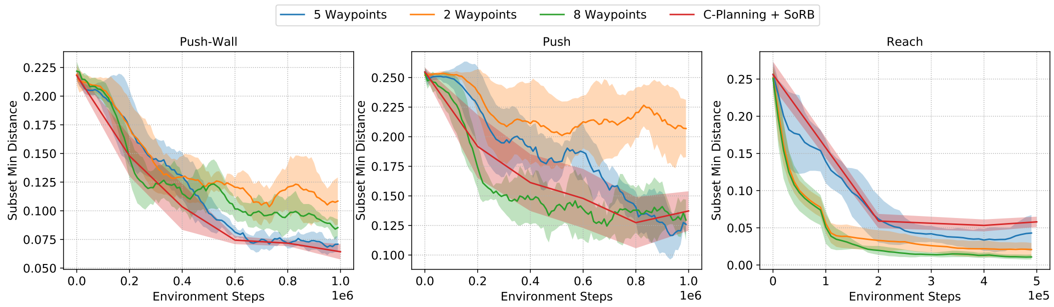
Do self-consistent models know what they know? What if we assume that quasimetric networks can generalize? That is, after learning that (say) s_1 and s_2 are 5 steps apart, it will predict that similar states s_1' and s_2' are also 5 steps apart. Because the model is architecturally constrained to be a quasimetric, this prediction (or "hallucination") could also result in changing the predictions for other s-g pairs. That is, this new "hallucinated" edge $s_1' \longrightarrow s_2'$ 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 (Huang et al., 2022; Irving et al., 2018). 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.

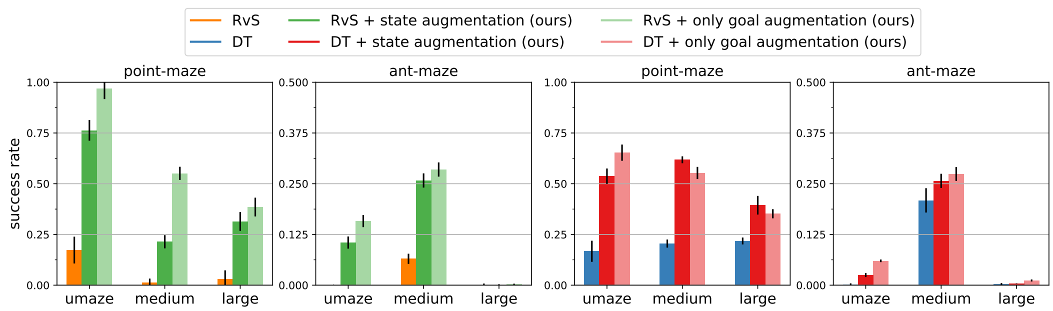
There is an important difference between this example and the quasimetrics. While the axiom used for checking self-consistency in quasimetrics was the triangle inequality, in this language modeling example self-consistency is checked using the predictions from the language model itself. In the example 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 (Whitehead and Russell, 1927).

F Evidence of Horizon Generalization and Planning Invariance from Prior Work

Not only do the experiments in Section 6 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. 10:



(a) Zhang et al. (2021b)



(b) Ghugare et al. (2024)

Figure 10: **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.

1. Zhang et al. (2021b) 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. (2024) 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.

G Experiment Details

The following subsections discuss the environment details for the figures in the main text.

G.1 Figure 2

This task is a gridworld of size 30 x 30, with walls shown as in Fig. 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.

G.2 Figure 6 (Top)

This plot used the same environment as described in Appendix G.1. 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., $\pi(a \mid s, g) \propto e^{-0.1 d(s,g)}$). 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.

G.3 Figure 6 (Bottom)

This plot used the same environment as described in Appendix G.1. The CRL method refers to (Eysenbach et al., 2022) and CMD refers to (Myers et al., 2024a). 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 (Bortkiewicz et al., 2024).

For the Bottom Right subplot, we performed planning in the same way as for the Top Right subplot.

G.4 Figure 5

For this task we directly used the ant maze task as well as loss functions and architectures from Bortkiewicz et al. (2024). All other hyperparameters are kept as the defaults from that paper. Training is done for 100M steps

G.5 Figure 9

For this experiment we used an S-shaped maze, shown in Fig. 11.[2] The dynamics are the same as those of Fig. 2.

Figure 11: 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 (Bortkiewicz et al., 2024), $\gamma = 0.9$, 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.

[2]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.