Offline Goal Conditioned Reinforcement Learning with Temporal Distance Representations

Anonymous Author(s)
Affiliation
Address
email

Abstract

Learned successor features provide a powerful framework for learning goal-reaching policies. These representations are constructed such that similarity in the representation space predicts future outcomes, allowing goal-reaching policies to be extracted. Representations learned for forward inference have some practical limitations — stitching of behaviors does not arise naturally with forward objectives like contrastive classification, and additional regularization is required to enable valid policy extraction. In this work, we propose a new representation learning objective that enables extraction of goal-reaching policies. Our key insight is that rather than learning representations of the future, we should really learn representations that can associate outcomes with preceding states. We show that when combined with existing quasimetric network parameterization and the right invariances, these representations let us learn optimal goal-reaching policies from offline data. On existing offline GCRL benchmarks, the hindsight classification objective improves performance with a simpler algorithm and fewer independent networks/parameters to learn relative to past methods.

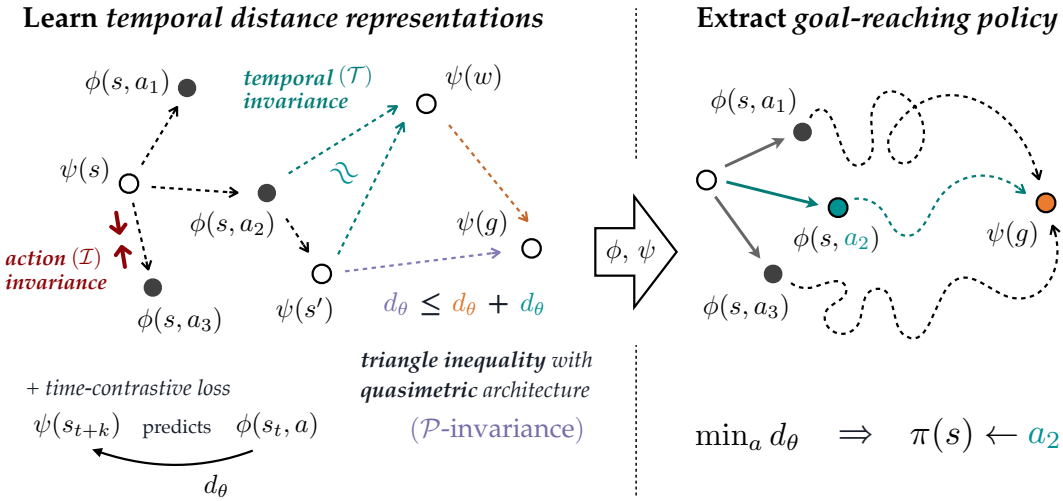

Figure 1: *(Left)* TMD learns a temporal distance d_θ that satisfies the triangle inequality and action invariance. It does this by minimizing the distance between the learned distance and the distance between the successor features of the states and actions in the dataset. *(Right)* The learned distance is used to extract a goal-conditioned policy.

Submitted to 39th Conference on Neural Information Processing Systems (NeurIPS 2025). Do not distribute.

1 Introduction

Learning temporal distances lies at the heart of many important problems in both control theory and reinforcement learning. In control theory, such distances form important Lyapunov functions [1] and control barrier functions [2], and are at the core of reachability analysis [3] and safety filtering [4] In reinforcement learning (RL), such distances are important not just for safe RL [5], but also for forming value functions in tasks ranging from navigation [6] to combinatorial reasoning [7] to robotic manipulation [8, 9]. Ideally, these learned distances have two important properties: *(i)* they can encode paths that are shorter than those demonstrated in the data (i.e., stitching); and *(i)* then can capture long-horizon distances with low variance.

Today, when users are selecting a method for learning temporal distances, they typically have to decide which of these desiderata they care more about and forgo the other. Methods based on Q-learning [10, 11, 12] use TD learning to stitch trajectories and find shortest paths, yet TD learning results in compounding errors that make it challenging to apply to long-horizon tasks [13]. In contrast, Monte Carlo methods [14] can directly learn to estimate the distances between temporally-separated states, yet their ability to find *shortest* paths remains limited.

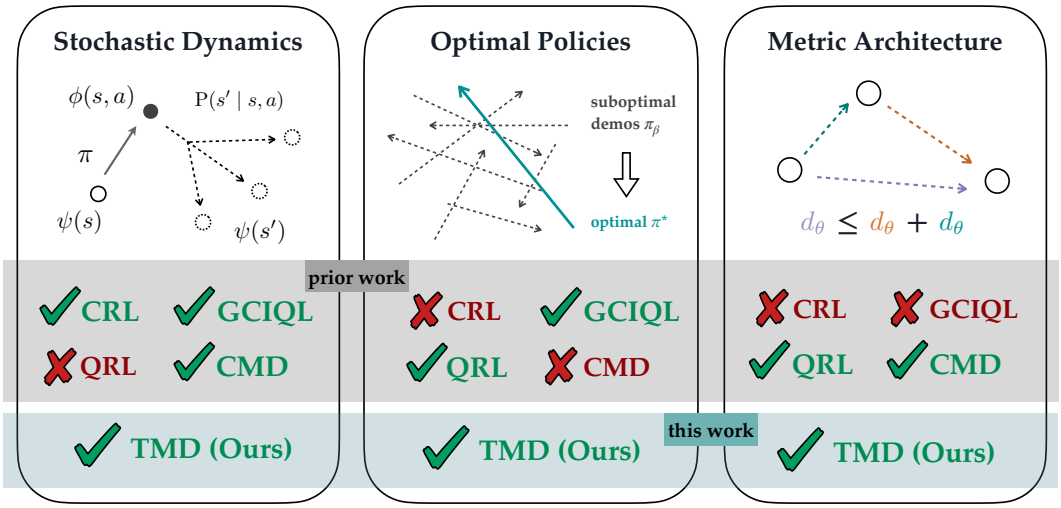

Figure 2: TMD enables key capabilities over prior work: *(Left)* handling stochastic transition dynamics, *(Center)* learning optimal policies from offline data, and *(Right)* stitching behaviors as a property of network architecture.

The aim of this paper is to build a method for learning temporal distances that retains the long-horizon estimation capabilities of Monte Carlo methods but nonetheless is able to compute shortest paths. We take an invariance perspective to do this. Temporal distances satisfy various invariance properties. Because they are value functions, they satisfy the Bellman equations. Prior work has also shown that they satisfy the triangle inequality, even in stochastic settings [15, 16]. The triangle inequality, also a form of invariance [17], is powerful because it lets us architecturally winnow down the hypothesis space of temporal distances by only considering neural network architectures that satisfy the triangle inequality [18, 19]. Importantly, the fact that temporal distances satisfy the triangle inequality holds for *any* temporal distance, including both optimal temporal distances and those learned by Monte Carlo methods. This raises an important question: might there be an *additional* invariance that is satisfied by optimal temporal distances, but not those learned by Monte Carlo methods? Identifying such invariance properties that would enable us to use Monte Carlo methods to architecturally winnow the hypothesis space, and then use this additional invariance property to identify optimal temporal distances within that space.

The key contribution of this paper is a method for learning temporal distances that relies primarily on Monte Carlo learning, but nonetheless provably converges to optimal shortest paths. To enable this, we identify two additional invariance properties that apply at the *transition* level. These invariance properties are structurally similar to the Bellman equations, but importantly forgo the need to define reward functions. We translate these invariance properties into a practical method for learning temporal distances. We demonstrate an application of those distances to goal-conditioned RL tasks. On benchmark tasks up to 21-dimensions as well as visual observations, we demonstrate that our method achieves results that considerably outperforms that of similar baselines. Additional

experiments reveal the importance of the enforced invariances and contrastive learning objective. Given the importance of long-horizon reasoning in many potential applications of RL today, we believe our work is useful for thinking about how to learn optimal temporal distances.

2 Related Work

We provide a unifying framework that connects metric learning to (optimal) offline goal-conditioned reinforcement learning (GCRL).

2.1 Metric Learning

We build on prior approaches to learning *temporal* distances, which reflect the reachability of states [16]. Temporal distances are usually defined as the expected number of time steps to transit from one state to another [20, 21]. Recent work has provided probabilistic definitions that are also compatible with continuous state spaces and stochastic transition dynamics [16]. A key consideration when thinking about temporal distance is *which* policy they reflect: is this an estimate of the number of time steps under our current policy or under the optimal policy? We will use *optimal temporal distance* to mean the temporal distance under the optimal (distance-minimizing) policy. Algorithmically, this choice is often reflected in the algorithm one uses for learning temporal distances. Methods based on Q-learning typically estimate optimal temporal distances [11, 22, 23], and are often structurally similar to popular actor-critic methods. Prior work has shown that temporal distance learning can be important for finding paths that are better than those demonstrated in the data, and can enable significantly more data efficient learning [24] (akin to standard results in the theory of Q-learning [25]). Methods based on Monte Carlo learning typically operate by sampling pairs of states that occur nearby in time (though not necessarily temporally-adjacent); distances are minimized for such positive pairs, and maximize for pairs of states that appear on different trajectories [26, 27]. These Monte Carlo methods typically estimate the temporal distance corresponding to the policy that collected the data. Methods for goal-conditioned behavioral cloning [28, 29], though not directly estimating temporal distances, are effectively working with this same behavioral temporal distance [7]. Despite the fact that these Monte Carlo methods do not estimate optimal temporal distances, they often outperform their Q-learning counterparts, suggesting that it is at least unclear whether the errors from learning the behavioral (rather than optimal) temporal distance are larger or smaller than those introduced by TD learning's compounding errors. Our work aims to bridge these two types of temporal distances, providing a method that does learn optimal temporal distances while reducing the reliance on TD learning to propagate values (and accumulate errors).

2.2 Offline Reinforcement Learning

Our investigation into temporal distances closely mirrors discussions in the offline RL literature about 1-step RL methods [30], which often use Monte Carlo value estimation, versus multi-step RL methods [31], which often use Q-learning value estimation. These 1-step RL methods avoid the compounding errors of Q-learning, yet are limited by their capacity to learn $Q^\star$ rather than Q^β. However, their strong performance over the years [32] continues to suggest that it remains an open question whether the compounding errors of Q-learning outweigh the benefits from learning the behavioral value function, rather than the value function of the optimal policy.

3 Temporal Metric Distillation (TMD)

In this section, we formally define TMD in terms of the invariances it must enforce to recover optimal distances, and by extension, the optimal policy. In Section 4 we will then show how these invariances can be converted into losses which can be optimized with a quasimetric architecture that enforces the triangle inequality.

3.1 Notation

We consider a controlled Markov process $\mathcal{M}$ with state space $\mathcal{S}$, action space $\mathcal{A}$, and dynamics $\mathrm{P}(s' \mid s, a)$. 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.

Policies $\pi \in \Pi$ are defined as distributions $\pi(a \mid s)$ for $s \in \mathcal{S}, a \in \mathcal{A}$. When applicable, for a fixed policy π, we can denote the state and action at step t as random variables $\mathfrak{s}_t$ and $\mathfrak{a}_t$, respectively. We will also use the shorthand

$$\mathfrak{s}_t^+ \triangleq \mathfrak{s}_{t+K} \text{ for } K \sim \text{Geom}(1-\gamma). \tag{1}$$

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 and have a zero diagonal. We will denote the set of all distances as $\mathcal{D}$, defined as

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

A desirable property for distances to satisfy is the triangle inequality, which states that the distance between two states is no greater than the sum of the distances between the states and a waypoint [15]. A distance satisfying this property is known as a *quasimetric*. Formally, we construct

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

If we further restrict distances to be symmetric ($d(x,y) = d(y,x)$), we obtain the set of traditional metrics over $\mathcal{S}$.

3.2 TMD Operators

TMD learns a distance parameterization that is made to satisfy two constraints: (i) the *triangle inequality*,

$$d(x,z) \leq d(x,y) + d(y,z) \text{ for any } x, y, z \in \mathcal{S} \times \mathcal{A} \cup \mathcal{S}, \tag{2}$$

and (ii) *action invariance*,

$$d\big(s,(s,a)\big) = 0 \text{ for any } s \in \mathcal{S} \text{ and } a \in \mathcal{A}. \tag{3}$$

We will show that to ensure that we recover the optimal distance d_{SD} [16] given the learned (backward NCE) contrastive critic distance, the missing additional constraint is the (exponentiated) SARSA version of Bellman consistency. So, what TMD is doing with these additional constraints is weakening the form of Bellman consistency that is required to recover the optimal distance from the standard $\max_{a \in \mathcal{A}}$ Bellman operator to the weaker on-policy SARSA Bellman operator. TMD thus turns on-policy SARSA into an off-policy algorithm through the metric constraints.

We can define this additional constraint as the fixed point of the following operator:

$$\mathcal{T}(d)(x,y) = \begin{cases} -\log \mathbb{E}_{\mathrm{P}(s'|s,a)}\left[e^{-d(s',y)}\right] - \log\gamma & \text{if } x = (s,a) \in \mathcal{S} \times \mathcal{A}, \\ d(x,y) & \text{otherwise.} \end{cases} \tag{4}$$

The triangle inequality Eq. (2) and action invariance Eq. (3) properties can also be written in terms of operator fixed points:

$$\mathcal{P}(d)(x,z) \triangleq \min_{y \in \mathcal{S}}\big[d(x,y) + d(y,z)\big] \tag{5}$$

$$\mathcal{I}d(s,x) \triangleq \begin{cases} 0 & \text{if } x = (s,a) \\ \mathcal{I}(d)(s,x) & \text{otherwise.} \end{cases} \tag{6}$$

3.3 Properties of path relaxation

Path relaxation $\mathcal{P}$ [17] (Eq. 5) enforces invariance to the triangle inequality, i.e., $\mathcal{P}(d) = d$ if and only if $d \in \mathcal{Q}$. P

Theorem 1. *Take $d \in \mathcal{D}$ and consider the sequence*

$$d_n = \mathcal{P}^n(d).$$

Then, d_n converges uniformly to a fixed point $d_\infty \in \mathcal{Q}$.

In light of Theorem 1 we denote by $\mathcal{P}_* = \lim_{n\to\infty} \mathcal{P}^n$ the fixed point operator of $\mathcal{P}$, and note that $\mathcal{P}_*$ is in fact a projection operator onto $\mathcal{Q}$.

Proofs of Lemmas 5 to 7 and Theorem 1 can be found in Appendix C.

3.4 The modified successor distance

The *modified successor distance* $d^{\pi}_{\text{SD}} \in D$ can be defined by [16]: 11.5

$$d^{\pi}_{\text{SD}}(x,y) \triangleq \begin{cases} 0 & \text{if } x = y, \\ -\log p^{\pi}\left(\frac{\mathrm{P}(\mathfrak{s}^{+}=g|\mathfrak{s}=s,\mathfrak{a}=a)}{\mathrm{P}(\mathfrak{s}^{+}=g|\mathfrak{s}=g)}\right) \text{ for } K \sim \gamma & \text{if } x = (s,a) \in \mathcal{S}\times\mathcal{A}, y = g \in \mathcal{S} \\ -\log \mathbb{E}_{\pi(a|s)}\left[e^{-d^{\pi}_{\text{SD}}((s,a),g)}\right] - \log\gamma & \text{if } x = s \in \mathcal{S}, x \neq y \\ d^{\pi}_{\text{SD}}(s,g) - \log \pi(a \mid g) & \text{if } y = (g,a) \in \mathcal{S}\times\mathcal{A}. \end{cases} \tag{7}$$

The *optimal successor distance* d^{*}_{SD} can then be stated as

$$d^{*}_{\text{SD}}(x,y) \triangleq \min_{\pi\in\Pi} d^{\pi}_{\text{SD}}(x,y). \tag{8}$$

This distance is useful since it lets us recover optimal goal-reaching policies. For any $s, g \in \mathcal{S}, a \in \mathcal{A}$, the distance is proportional to the optimal goal-reaching value function

$$d^{*}_{\text{SD}}((s,a),g) \propto_a -Q^{*}_{g}(s,a) \tag{9}$$

where Q^{*}_{g} is defined as the standard optimal Q-function for reaching goal g [27]:

$$Q^{*}_{g}(s,a) \triangleq \max_{\pi\in\Pi} \mathbb{E}_{\{\mathfrak{s}_i,\mathfrak{a}_i\}\sim\pi}\Big[\sum_{t=0}^{\infty}\gamma^{t}\,\mathrm{P}(\mathfrak{s}_t = g \mid \mathfrak{s}_0 = s, \mathfrak{a}_0 = a)\Big]. \tag{10}$$

and

$$V^{*}_{g}(s) \triangleq \max_{a\in\mathcal{A}} Q^{*}_{g}(s,a). \tag{11}$$

In fact, we can equivalently define $d^{*}_{\text{SD}}\big((s,a),g\big)$ in terms of Q^{*}:

$$d^{*}_{\text{SD}}\big((s,a),g\big) = \log V^{*}_{g}(g) - \log Q^{*}_{g}(s,a). \tag{12}$$

Similar to Myers et al. [16], we argue that contrastive learning can recover these distances, i.e.,

$$\mathcal{C}(\pi) = d_{\text{SD}} \tag{13}$$

Then, through the operators in Section 3.2, we will extend this to the optimal distance d^{*}_{SD}.

For convenience, we also define the set of realized successor distances

$$\widetilde{\mathcal{D}} \triangleq \{d^{\pi}_{\text{SD}} : \pi \in \Pi\}. \tag{14}$$

Note that $\widetilde{\mathcal{D}}$ does not necessarily contain the optimal distance d^{*}_{SD}, as no single policy is generally optimal for reaching all goals.

Remark 2. *The optimal successor distance d^{*}_{SD} satisfies*

$$d_{\text{SD}}(s,(s,a)) = 0 \textit{ for all } s \in \mathcal{S} \textit{ and } a \in \mathcal{A}.$$

3.5 Convergence to the optimal successor distance

Applying the invariances in Section 3.2 to the contrastive distance Eq. (13), the TMD algorithm can be defined symbolically as

$$\mathcal{M}(\pi) \triangleq (\mathcal{P}_{*} \circ \mathcal{T} \circ \mathcal{I})^{\infty}\,\mathcal{C}(\pi). \tag{15}$$

In other words, TMD computes the initial π^{β} distance $\mathcal{C}(\pi)$, and then enforces the invariance (architecturally or explicitly), as expressed with the iterative application of $\mathcal{T} \circ \mathcal{I}$ followed by projection onto $\mathcal{Q}$ by $\mathcal{P}_{*}$.

Theorem 3. *The TMD algorithm converges pointwise to the optimal successor distance d^{*}_{SD} for any policy π with full state and action coverage, i.e.,*

$$\lim_{n\to\infty} (\mathcal{P}_{*} \circ \mathcal{T} \circ \mathcal{I})^{n}\,\mathcal{C}(\pi) = d^{*}_{\text{SD}}. \tag{16}$$

Our approach for proving Theorem 3 will be to analyze the convergence properties of $(\mathcal{P}_* \circ \mathcal{T} \circ \mathcal{I})$ over the space of "suboptimal" distances $\mathcal{D}_+^*$, defined as

$$\mathcal{D}_+^* \triangleq \{d \in \mathcal{D} : d(x, y) \geq d_{\text{SD}}^*(x, y) \text{ for all } x, y \in \mathcal{S} \times \mathcal{A} \cup \mathcal{S}\}. \tag{17}$$

Unfortunately, $(\mathcal{P}_* \circ \mathcal{T} \circ \mathcal{I})$ is not a contraction on $\mathcal{D}_+^*$, so we cannot directly apply the Banach fixed-point theorem as we would for the standard Bellman (optimality) operator. Instead, we will show this operator induces a "more aggressive" form of tightening over $\mathcal{D}_+^*$, which will allow us to prove convergence to d_{SD}^*. We start by showing that d_{SD}^* is a fixed point of $(\mathcal{P}_* \circ \mathcal{T} \circ \mathcal{I})$ in Lemma 4.

Lemma 4. *The optimal successor distance d_{SD}^* is the unique fixed point of $\mathcal{P}_* \circ \mathcal{T} \circ \mathcal{I}$ on $\mathcal{D}_+^*$.*

Proofs are in Appendix B.

4 Implementing TMD

We show that the backward NCE contrastive learning algorithm can recover an initial estimate of $d_{\text{SD}}^{\pi_\beta}$. As justified by Theorem 3, we can then enforce the invariances to recover the optimal distance d_{SD}^*.

The algorithm learns a distance d_θ parameterized by a quasimetric neural network θ such as MRN [18]. By construction, this distance is a quasimetric that is invariant to $\mathcal{P}$, i.e., $\mathcal{P}d_\theta = d_\theta$.

4.1 Initializing the Distance with Contrastive Learning

Defining the critic

$$f(s, a, g) \triangleq -d_\theta\big((s, a), g\big),$$

the core contrastive objective is the backward NCE loss:

$$\mathcal{L}_{\text{NCE}}\left(\phi, \psi; \{s_i, a_i, s_i', g_i\}_{i=1}^N\right) = \sum_{i=1}^N \log\left(\frac{e^{f(s_i, a_i, g_i)}}{\sum_{j=1}^N e^{f(s_j, a_j, g_i)}}\right) \tag{18}$$

which is enforced across batches of triplets $\{s_i, a_i, s_{i+k}\}_{i=1}^N$ for $k \sim \text{Geom}(1-\gamma)$ sampled from the dataset generated by policy π_β.

The optimal solution to this objective is

$$f(s, a, g) = \log\Big(\frac{\text{P}(\mathfrak{s}^+ = g \mid \mathfrak{s} = s, \mathfrak{a} = a)}{\text{P}(\mathfrak{s}^+ = g)C(g)}\Big). \tag{19}$$

for some $C(g)$ [33].

The parameterization $f(s, a, g) = -d_\theta\big((s, a), g\big)$ where d_θ is a quasimetric-enforcing parameterization (see [18, 15]) ensures that

$$C(g) = \frac{\text{P}(\mathfrak{s}^+ = g \mid \mathfrak{s} = g)}{\text{P}(\mathfrak{s}^+ = g)},$$

so the only valid quasimetric satisfying Eq. (19) is $d_\theta = d_{\text{SD}}^{\pi_\beta}$.

Optimality of $\mathcal{L}$ in Eq. (18) implies that the learned distance $d_\theta = \mathcal{C}(\pi_\beta) = d_{\text{SD}}^{\pi_\beta}$.

The additional invariance constraints $\mathcal{I}$ and $\mathcal{T}$ can be directly enforced by regressing $\|d_\theta - \mathcal{I}d_\theta\|_\infty$ and $\|d_\theta - \mathcal{T}d_\theta\|_\infty$ to zero. Theorem 3 guarantees that if we can enforce those constraints and enforce invariance to $\mathcal{P}$ by using a quasimetric architecture, we can recover the optimal distance d_{SD}^*.

In practice, we will directly enforce the constraints across the batches used in our contrastive loss. We will use the MRN parameterization for d_θ for $\theta = (\psi, \phi)$ on learned representations of states (ψ) and state-action pairs (ϕ):

$$d_\theta\big(s, g\big) \triangleq d_{\text{MRN}}(\psi(s), \psi(g)) \qquad d_\theta\big((s, a), g\big) \triangleq d_{\text{MRN}}(\phi(s, a), \psi(g))$$
$$d_\theta\big(s, (s, a)\big) \triangleq d_{\text{MRN}}(\psi(s), \phi(s, a)) \qquad d_\theta\big((s, a), (s', a')\big) \triangleq d_{\text{MRN}}(\phi(s, a), \phi(s', a'))$$

where

$$d_{\text{MRN}}(x, y) \triangleq \frac{1}{K}\sum_{k=1}^K \max_{m=1\ldots M} \max(0, x_{kM+m} - y_{kM+m}) \tag{20}$$

4.2 Action $\mathcal{I}$-Invariance

Invariance to the $\mathcal{I}$ backup operator in Eq. (6) gives the following update across $s, a \in \mathcal{S} \times \mathcal{A}$

$$d_\theta\big(\psi(s), \phi(s,a)\big) \leftarrow 0, \tag{21}$$

which can be directly enforced with the following loss across the batch:

$$\mathcal{L}_\mathcal{I}\left(\phi, \psi; \{s_i, a_i, s_i', g_i\}_{i=1}^N\right) = \sum_{i,j=1}^N d_{\text{MRN}}\big(\psi(s_i), \phi(s_i, a_j)\big). \tag{22}$$

4.3 Temporal $\mathcal{T}$-Invariance

Invariance to the $\mathcal{T}$ backup operator in Eq. (4) corresponds to the following update performed with respect to $\phi(s,a)$:

$$e^{-d_{\text{MRN}}(\phi(s,a),\psi(g))} \leftarrow \mathbb{E}_{\text{P}(s'|s,a)} e^{\log\gamma - d_{\text{MRN}}(\psi(s'),\psi(g))}. \tag{23}$$

This update is enforced by minimizing a divergence between the LHS and samples from the RHS expectation. Classic approaches for backups in deep RL include the ℓ_2 distance to the target (RHS) [34], or when values can be interpreted as probabilities, a binary cross-entropy loss [35].

We use the following Bregman divergence [36], which we find empirically is more stable for learning the update in Eq. (23) (c.f. the Itakura-Saito distance [37]).

$$D_T(d, d') \triangleq \exp(d - d') - d. \tag{24}$$

We discuss this divergence and prove correctness in Appendix E. With the divergence, the $\mathcal{T}$-invariance loss is:

$$\mathcal{L}_\mathcal{T}\left(\phi, \psi; \{s_i, a_i, s_i', g_i\}_{i=1}^N\right) = \sum_{i,j=1}^N D_T\big(d_{\text{MRN}}(\phi(s_i,a_i), \psi(g_j)), d_{\text{MRN}}(\psi(s_i'), \psi(g_j)) - \log\gamma\big) \tag{25}$$

We minimize this loss only with respect to ϕ, stopping the gradient through ψ. This avoids the moving target that classically necessitates learning separate target networks in RL [34].

4.4 The Overall Distance Learning Objective

We can express the overall critic loss as:

$$\mathcal{L}_{\text{TMD}}\big(\phi, \psi; \overline{\psi}, \mathcal{B}\big) = \mathcal{L}_{\text{NCE}}\left(\phi, \psi; \mathcal{B}\right) + \zeta\Big(\mathcal{L}_\mathcal{I}\left(\phi, \psi; \mathcal{B}\right) + \mathcal{L}_\mathcal{T}\left(\phi, \overline{\psi}; \mathcal{B}\right)\Big) \tag{26}$$
$$\text{for batch} \quad \mathcal{B} \sim p^{\pi_\beta} = \{s_i, a_i, s_i', g_i\}_{i=1}^N$$

We minimize Eq. (26) with respect to ϕ and ψ, where $\overline{\psi}$ is a separate copy of the representation network ψ (stop-gradient). Here, ζ controls the weight of the contrastive loss and invariance constraints, and batches are sampled

$$\{s_i, a_i, s_i', g_i\}_{i=1}^N \sim \pi_\beta,$$

for s_i' the state following s_i, and g_i the state K steps ahead of s_i for $K \sim \text{Geom}(1-\gamma)$. In theory, ζ^{-1} should be annealed between 1 at the start of training (to extract the distance $\mathcal{C}(\pi)$), toward 0 at the end of training to enforce invariance to $(\mathcal{T} \circ \mathcal{I})$.

In practice, we pick ζ based on how much stitching and stochasticity we expect in the environment — when ζ is large, we more aggressively try and improve on the initial distance $\mathcal{C}(\pi_\beta)$ describing the dataset policy π_β.

4.5 Policy Extraction

We finally extract the goal-conditioned policy $\pi(s,g) : \mathcal{S}^2 \to \mathcal{A}$ with the learned distance d_θ:

$$\min_\pi \mathbb{E}_{\{s_i,a_i,s_i',g_i\}_{i=1}^N \sim \pi_\beta}\Big[\sum_{i,j=1}^N d_\theta\big((s_i, \pi(s_i,g_j)), g_j\big)\Big]. \tag{27}$$

Table 1: OGBench Evaluation

Task	Methods					
	TMD	CRL	QRL	GCBC	GCIQL	GCIVL
humanoidmaze_medium_stitch	**60.5**$^{(\pm 1.6)}$	36.2$^{(\pm 0.9)}$	18.0$^{(\pm 0.7)}$	29.0$^{(\pm 1.7)}$	12.1$^{(\pm 1.1)}$	12.3$^{(\pm 0.6)}$
pointmaze_teleport_stitch	29.3$^{(\pm 2.2)}$	4.1$^{(\pm 1.1)}$	8.6$^{(\pm 1.9)}$	31.5$^{(\pm 3.2)}$	25.2$^{(\pm 1.0)}$	**44.4**$^{(\pm 0.7)}$
humanoidmaze_large_stitch	**23.0**$^{(\pm 1.5)}$	4.0$^{(\pm 0.2)}$	3.5$^{(\pm 0.5)}$	5.6$^{(\pm 1.0)}$	0.5$^{(\pm 0.1)}$	1.2$^{(\pm 0.2)}$
antmaze_teleport_explore	**49.6**$^{(\pm 1.5)}$	19.5$^{(\pm 0.8)}$	2.3$^{(\pm 0.7)}$	2.4$^{(\pm 0.4)}$	7.3$^{(\pm 1.2)}$	32.0$^{(\pm 0.6)}$
antmaze_large_stitch	**20.9**$^{(\pm 1.7)}$	10.8$^{(\pm 0.6)}$	**18.4**$^{(\pm 0.7)}$	3.4$^{(\pm 1.0)}$	7.5$^{(\pm 0.7)}$	**18.5**$^{(\pm 0.8)}$
scene_noisy	19.6$^{(\pm 1.7)}$	1.2$^{(\pm 0.3)}$	9.1$^{(\pm 0.7)}$	1.2$^{(\pm 0.2)}$	**25.9**$^{(\pm 0.8)}$	**26.4**$^{(\pm 1.7)}$
visual_antmaze_teleport_stitch	**38.5**$^{(\pm 1.5)}$	31.7$^{(\pm 3.2)}$	1.4$^{(\pm 0.8)}$	31.8$^{(\pm 1.5)}$	1.0$^{(\pm 0.2)}$	1.4$^{(\pm 0.4)}$
visual_antmaze_large_stitch	**26.6**$^{(\pm 2.8)}$	11.1$^{(\pm 1.3)}$	0.6$^{(\pm 0.3)}$	**23.6**$^{(\pm 1.4)}$	0.1$^{(\pm 0.0)}$	0.8$^{(\pm 0.3)}$

We **bold** the best performance. Success rate (%) is presented with the standard deviation across six seeds.

For conservatism [38], we augment Eq. (27) with a behavioral cloning loss against $\pi_\beta$via behavior-constrained deep deterministic policy gradient [39]. Using additional goals g_i sampled from the same trajectory as s_i in Eq. (27) could also be done through an extra tuned parameter (cf. Bortkiewicz et al. [40], Park et al. [41]). Denoting these hyperparameters as λ and α respectively, the overall policy extraction objective is:

$$\min_{\pi} \mathbb{E}_{\{s_i,a_i,s'_i,g_i\}_{i=1}^N \sim \pi_\beta} \left[\mathcal{L}_\pi\left(\pi; \phi, \psi, \{s_i, a_i, s'_i, g_i\}_{i=1}^N\right)\right] \tag{28}$$

$$\mathcal{L}_\pi \triangleq \sum_{i,j=1}^{N} (1-\lambda) d_{\text{MRN}}\left(\phi(s_i, \hat{a}_{ij}), \psi(g_j)\right), g_j\right) + \lambda d_{\text{MRN}}\left(\phi(s_i, \hat{a}_{ii}), \psi(g_i)\right) + \alpha \left\|\hat{a}_{ii} - a_i\right\|_2^2$$

$$\text{where } \hat{a}_{ij} = \pi(s_i, g_j). \tag{29}$$

Prior offline RL methods use similar α and λ hyperparameters, which must be tuned per environment [41].

5 Experiments

In our experiments, we evaluate the performance of TMD on tasks from the OGBench benchmark [41]. We aim to answer the following questions:[1]

1. Do the invariance terms in Eq. (5) improve performance quantitatively in offline RL settings?
2. Is the contrastive loss in Eq. (18) necessary to facilitate learning these tasks?
3. What capabilities does TMD enable for compositional task learning?

5.1 Experimental Results

We evaluate TMD across evaluation tasks OGBench for the environments and datasets listed in Table 1, each containing 5 separate goal-reaching tasks, making a total of 40 tasks. We evaluate TMD with 6 seeds in all environments. Of particular interest are the "`teleport`" and "`stitch`" environments, which respectively test the ability to handle stochasticity and composition.

We compare against Goal-Conditioned Behavior Cloning (GCBC), Goal-Conditioned Implicit Q-Learning (GCIQL), Goal-Conditioned Implicit Value Learning

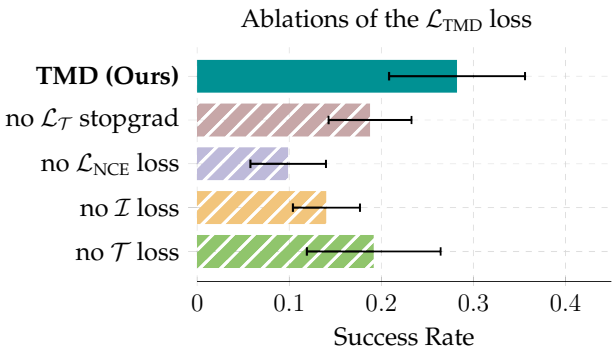

Figure 3: We ablate the loss components of TMD in the `pointmaze_teleport_stitch` environment.

[1]Anonymous code: `https://anonymous.4open.science/r/ogcrl-FCDC/`

(GCIVL), Contrastive RL (CRL), and Quasimetric RL (QRL) as our baselines. GCBC uses imitation learning to learn a policy that follows the given trajectories within a dataset [42]. CRL [27] performs policy improvement by fitting a value function via contrastive learning. QRL [15] learns a quasimetric value function via Bellman consistency. GCIQL and GCIVL use expectile regression to fit a value function [43].

Over stitching environments, TMD consistently outperforms QRL and CRL, especially in `humanoidmaze`, where it outperformed all baselines in comparison by up to a sixfold factor in the large humanoidmaze environment, and demonstrated the ability to solve the environment while the baselines displayed barely any positive success rate. In `teleport` environments, where there exists stochastic transitions, TMD outperforms both CRL and QRL by a considerable margin, especially in `pointmaze_teleport_stitch`, where TMD outperforms the other two methods by a factor of three fold. Although TMD is outperformed by GCIQL and GCIVL in some occasions, it shrinks the gap between it and these methods considerably than CRL and QRL. Finally, our results show that TMD performs consistently above baselines regardless of dataset composition, as it demonstrates strong performance in both `explore` and `noisy` datasets, where learning value functions become more important.

5.2 Ablation Study

We perform an ablation study on the `pointmaze_teleport_stitch` environment to evaluate the importance of the invariance terms and the contrastive initialization loss in TMD. We separately disable the contrastive, $\mathcal{T}$ invariance, and $\mathcal{I}$ invariance component during training and observe its effects. We also examine the empirical effects of stopping gradients when calculating $\mathcal{L}_{\mathcal{T}}$. We log the corresponding success rate for each of the ablations in 3.

Our ablation studies answer questions 2 and 3, in which we demonstrate that by removing some of the invariances or removing the contrastive loss, the performance of TMD decreases to levels similar to CRL and QRL. Similarly, we see the importance of keeping the contrastive objective, as the performance of TMD degrades even more despite the presence of other loss components. We also note the empirical performance of TMD is better when we stop gradients on $\mathcal{L}_{\mathcal{T}}$. We provide further ablation details in Appendix D.2.

6 Discussion

In this work, we introduce Temporal Metric Distillation (TMD), an offline goal-conditioned reinforcement learning method that learns representations which exploit the quasimetric structure of temporal distances. Our approach unifies quasimetric, temporal-difference, and Monte Carlo learning approaches to GCRL by enforcing a set of invariance properties on the learned distance function. To the best of our knowledge, TMD is the first method that can exploit the quasimetric structure of temporal distances to learn optimal policies from offline data, even in stochastic settings (see Fig. 2). On a standard suite of offline GCRL benchmarks, TMD outperforms prior methods, in particular on long-horizon tasks that require stitching together trajectories across noisy dynamics and visual observations.

6.1 Limitations and Future Work

Future work could examine more principled ways to set the ζ parameter in our method, or if there are ways to more directly integrate the contrastive and invariance components of the loss function. Future work could also explore integrating the policy extraction objective more directly into the distance learning to enable desirable properties (stitching through architecture, horizon generalization) at the level of the policy. While we used the MRN [18] architecture in our experiments, alternative architectures such as IQE [19] that enforce the triangle inequality could be more expressive. While the size of models studied in our experiments make them unlikely to pose any real-world risks, methods which implicitly enable long-horizon decision making could have unintended consequences or poor interpretability. Future work should consider these implications.

References

[1] Eduardo D. Sontag. A 'Universal' Construction of Artstein's Theorem on Nonlinear Stabilization. *Systems & Control Letters*, 13(2):117–123, 1989.

[2] Aaron D. Ames, Samuel Coogan, Magnus Egerstedt, Gennaro Notomista, Koushil Sreenath, and Paulo Tabuada. Control Barrier Functions: Theory and Applications. *European Control Conference*, pp. 3420–3431, 2019.

[3] Matthias Althoff. Reachability Analysis and Its Application to the Safety Assessment of Autonomous Cars. 2010.

[4] Kai-Chieh Hsu, Haimin Hu, and Jaime F Fisac. The Safety Filter: A Unified View of Safety-Critical Control in Autonomous Systems. *Annual Review of Control*, 7, 2023.

[5] Shangding Gu, Long Yang, Yali Du, Guang Chen, Florian Walter, Jun Wang, and Alois Knoll. A Review of Safe Reinforcement Learning: Methods, Theory and Applications. arXiv:2205.10330, 2022.

[6] Dhruv Shah, Blazej Osinski, Brian Ichter, and Sergey Levine. LM-Nav: Robotic Navigation With Large Pre-Trained Models of Language, Vision, and Action. *Conference on Robot Learning*, 2022.

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

[8] 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. *International Conference on Learning Representations*, 2023.

[9] Suraj Nair, Aravind Rajeswaran, Vikash Kumar, Chelsea Finn, and Abhinav Gupta. R3m: A Universal Visual Representation for Robot Manipulation. *Conference on Robot Learning*, pp. 892–909, 2022.

[10] Long-Ji Lin. Self-Improving Reactive Agents Based on Reinforcement Learning, Planning and Teaching. 1992.

[11] 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. *Neural Information Processing Systems*, volume 30, 2017.

[12] Vitchyr Pong, Shixiang Gu, Murtaza Dalal, and Sergey Levine. Temporal Difference Models: Model-Free Deep RL for Model-Based Control. *International Conference on Learning Representations*, 2018.

[13] Aviral Kumar, Justin Fu, Matthew Soh, George Tucker, and Sergey Levine. Stabilizing Off-Policy Q-Learning via Bootstrapping Error Reduction. *Neural Information Processing Systems*, 32, 2019.

[14] Alexey Dosovitskiy and Vladlen Koltun. Learning to Act by Predicting the Future. arXiv:1611.01779, 2016.

[15] Tongzhou Wang, Antonio Torralba, Phillip Isola, and Amy Zhang. Optimal Goal-Reaching Reinforcement Learning via Quasimetric Learning. *International Conference on Machine Learning*, pp. 36411–36430, 2023.

[16] 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. *International Conference on Machine Learning*, 2024.

[17] Vivek Myers, Catherine Ji, and Benjamin Eysenbach. Horizon Generalization in Reinforcement Learning. *International Conference on Learning Representations*, 2025.

[18] Bo Liu, Yihao Feng, Qiang Liu, and Peter Stone. Metric Residual Network for Sample Efficient Goal-Conditioned Reinforcement Learning. *AAAI Conference on Artificial Intelligence*, volume 37, pp. 8799–8806, 2023.

[19] Tongzhou Wang and Phillip Isola. Improved Representation of Asymmetrical Distances With Interval Quasimetric Embeddings. *NeurIPS 2022 NeurReps Workshop Proceedings Track*, 2022.

[20] Junik Bae, Kwanyoung Park, and Youngwoon Lee. TLDR: Unsupervised Goal-Conditioned RL via Temporal Distance-Aware Representations. *Conference on Robot Learning*, 2024.

[21] Stephen Tian, Suraj Nair, Frederik Ebert, Sudeep Dasari, Benjamin Eysenbach, Chelsea Finn, and Sergey Levine. Model-Based Visual Planning With Self-Supervised Functional Distances. *International Conference on Learning Representations*, 2021.

[22] Scott Fujimoto, Herke van Hoof, and David Meger. Addressing Function Approximation Error in Actor-Critic Methods. *International Conference on Machine Learning*, 2018.

[23] Leslie Pack Kaelbling. Learning to Achieve Goals. *International Joint Conference on Artificial Intelligence*, volume 2, pp. 1094–1098, 1993.

[24] Chongyi Zheng, Ruslan Salakhutdinov, and Benjamin Eysenbach. Contrastive Difference Predictive Coding. *International Conference on Learning Representations*, 2023.

[25] Alekh Agarwal, Nan Jiang, Sham M Kakade, and Wen Sun. Reinforcement Learning: Theory and Algorithms. *CS Dept*, 32:96, 2019.

[26] Kristian Hartikainen, Xinyang Geng, Tuomas Haarnoja, and Sergey Levine. Dynamical Distance Learning for Semi-Supervised and Unsupervised Skill Discovery. *International Conference on Learning Representations*, 2020.

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

[28] Dibya Ghosh, Abhishek Gupta, Ashwin Reddy, Justin Fu, Coline Devin, Benjamin Eysenbach, and Sergey Levine. Learning to Reach Goals via Iterated Supervised Learning. *International Conference on Learning Representations*, 2021.

[29] Vivek Myers, Andre He, Kuan Fang, Homer Walke, Phillipe 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. *Conference on Robot Learning*, 2023.

[30] Benjamin Eysenbach, Matthieu Geist, Sergey Levine, and Ruslan Salakhutdinov. A Connection Between One-Step RL and Critic Regularization in Reinforcement Learning. *Th International Conference on Machine Learning*, pp. 9485–9507, 2023.

[31] John Schulman, Philipp Moritz, Sergey Levine, Michael Jordan, and Pieter Abbeel. High-Dimensional Continuous Control Using Generalized Advantage Estimation. arXiv:1506.02438, 2018.

[32] Cassidy Laidlaw, Banghua Zhu, Stuart Russell, and Anca Dragan. The Effective Horizon Explains Deep RL Performance in Stochastic Environments. *International Conference on Learning Representations*, 2024.

[33] Zhuang Ma and Michael Collins. Noise Contrastive Estimation and Negative Sampling for Conditional Models: Consistency and Statistical Efficiency. *Empirical Methods in Natural Language Processing*, pp. 3698–3707, 2018.

[34] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A. Rusu, Joel Veness, Marc G. Bellemare, Alex Graves, Martin Riedmiller, et al. Human-Level Control Through Deep Reinforcement Learning. *Nature*, volume 518, pp. 529–533, 2015.

[35] Dmitry Kalashnikov, Alex Irpan, Peter Pastor, Julian Ibarz, Alexander Herzog, Eric Jang, Deirdre Quillen, Ethan Holly, Mrinal Kalakrishnan, Vincent Vanhoucke, and Sergey Levine. Scalable Deep Reinforcement Learning for Vision-Based Robotic Manipulation. *Conference on Robot Learning*, pp. 651–673, 2018.

[36] L. M. Bregman. The Relaxation Method of Finding the Common Point of Convex Sets and Its Application to the Solution of Problems in Convex Programming. *USSR Computational Mathematics and Mathematical Physics*, 7(3):200–217, 1967.

[37] Fumitada Itakura. Analysis Synthesis Telephony Based on the Maximum Likelihood Method. *Reports of the 6th Int. Cong. Acoust*, 1968.

[38] Aviral Kumar, Aurick Zhou, George Tucker, and Sergey Levine. Conservative Q-Learning for Offline Reinforcement Learning. *Neural Information Processing Systems*, volume 33, pp. 1179–1191, 2020.

[39] Scott Fujimoto and Shixiang Shane Gu. A Minimalist Approach to Offline Reinforcement Learning. arXiv:2106.06860, 2021.

[40] Michał Bortkiewicz, Władek Pałucki, Vivek Myers, Tadeusz Dziarmaga, Tomasz Arczewski, Łukasz Kuciński, and Benjamin Eysenbach. Accelerating Goal-Conditioned RL Algorithms and Research. *International Conference on Learning Representations*, 2025.

[41] Seohong Park, Kevin Frans, Benjamin Eysenbach, and Sergey Levine. OGBench: Benchmarking Offline Goal-Conditioned RL. *International Conference on Learning Representations*, 2025.

[42] Yiming Ding, Carlos Florensa, Pieter Abbeel, and Mariano Phielipp. Goal-Conditioned Imitation Learning. *Neural Information Processing Systems*, volume 32, 2019.

[43] Ilya Kostrikov, Ashvin Nair, and Sergey Levine. Offline Reinforcement Learning With Implicit Q-Learning. arXiv:2110.06169, 2021.

[44] Yim-Ming Wong and Kung-Fu Ng. On a Theorem of Dini. *Journal of the London Mathematical Society*, s2-11(1):46–48, 1975.

[45] James Bradbury, Roy Frostig, Peter Hawkins, Matthew James Johnson, Chris Leary, Dougal Maclaurin, George Necula, Adam Paszke, Jake VanderPlas, Skye Wanderman-Milne, and Qiao Zhang. JAX: Composable Transformations of Python+NumPy Programs. 2018.

[46] Arindam Banerjee, Srujana Merugu, Inderjit Dhillon, and Joydeep Ghosh. Clustering With Bregman Divergences. *SIAM International Conference on Data Mining*, pp. 234–245, 2004.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: Yes, we show with theoretical justification (Section 3.4) and experiments (Section 5) that our method outperforms the baselines and enables the capabilities discussed in the introduction.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Yes, limitations are discussed in Section 6.1.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

Justification: Yes, full proofs are in Appendices B and C.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: Yes, we provide an implementation of our method and experimental details in the code

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: Yes, code is referenced in Appendix A and datasets used from Park et al. [41] are openly available.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [Yes]

Justification: Yes, experimental details are discussed in Appendix D.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [Yes]

Justification: Yes, we report error bars across seeds in our results.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.

- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Yes, we provide compute resources in Appendix D.3.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: All components of the NeurIPS Code of Ethics are respected.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: Yes, we discuss both positive and negative societal impacts in Section 6.1.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.

- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: The models and datasets used in this paper do not pose a high risk for misuse.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: Yes, we credit the source of the datasets and baseline implementations used in our experiments [41].

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    - The answer NA means that the paper does not release new assets.

- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: No crowdsourcing or human subjects were used in this research.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: No crowdsourcing or human subjects were used in this research.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

    Answer: [NA]

    Justification: No LLMs were used in this research.

    Guidelines:

    - The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.

A Code

Code and videos are referenced here: https://anonymous.4open.science/w/tmd-website-6E33/. The evaluation and base agent structure follows the OGBench codebase [41]. The TMD agent is implemented in agents/tmd.py.

B Analysis of TMD

This section provides the proofs of the results in Section 3.5. The main result is Theorem 3, which shows that enforcing the TMD constraints on a learned quasimetric distance recovers the optimal distance d^*_{SD}.

Theorem 1. *Take $d \in \mathcal{D}$ and consider the sequence*

$$d_n = \mathcal{P}^n(d).$$

Then, d_n converges uniformly to a fixed point $d_\infty \in \mathcal{Q}$.

Proof. From Lemma 7, we have that $d_{n+1}(s, g) \leq d_n(s, g)$ for all $s, g \in \mathcal{S}$. Thus, the sequence $\{d_n\}$ is monotonically decreasing (and positive). By the monotone convergence theorem, the sequence converges pointwise to a limit d_∞. Since $\mathcal{S}$ is compact, by Dini's theorem [44], the convergence is uniform, i.e., $d_n \to d_\infty$ under the L^∞ topology over $\mathcal{D}$.

To see that d_∞ is a fixed point of $\mathcal{P}$, we note that if $\mathcal{P}d_\infty = d' \neq d_\infty$, we can construct disjoint neighborhoods N of d_∞ and N' of d' (since $L^\infty(\mathcal{D})$ is normed vector space and thus Hausdorff). By construction, the preimage $\mathcal{P}^{-1}(N')$ contains d_∞ and is open by Lemma 6. Thus, we can define another, smaller open neighborhood $N'' = N \cap \mathcal{P}^{-1}(N')$ of d_∞. Now, since $d_n \to d_\infty$, there exists some k so $d_k, d_{k+1} \in N'' \subset N$. But then since $d_k \in \mathcal{P}^{-1}(N')$, we have that $d_{k+1} \in N'$. This is a contradiction as N and N' were disjoint by construction.

Thus, we have that d_∞ is a fixed point of $\mathcal{P}$. That $d_\infty \in \mathcal{Q}$ follows from Lemma 5. □

Theorem 3. *The TMD algorithm converges pointwise to the optimal successor distance d^*_{SD} for any policy π with full state and action coverage, i.e.,*

$$\lim_{n\to\infty} (\mathcal{P}_* \circ \mathcal{T} \circ \mathcal{I})^n \, \mathcal{C}(\pi) = d^*_{\text{SD}}. \tag{16}$$

Proof of Theorem 3. The initial distance $\mathcal{C}(\pi) = d^\pi_{\text{SD}} \geq d^*_{\text{SD}}$ for any policy π. So, $\mathcal{C}(\pi) \in \mathcal{D}^*_+$. Define the sequence of distances $d_n = (\mathcal{P}_* \circ \mathcal{T} \circ \mathcal{I})^n \mathcal{C}(\pi)$ in $\mathcal{D}^*_+$. Note that $\mathcal{P}_*$ and $\mathcal{I}$ are monotone decreasing. So, the restriction $(d_n)_{\mathcal{X}}$ is monotonically decreasing on the domain $\mathcal{X} = \mathcal{S} \times (\mathcal{S} \cup \mathcal{S} \times \mathcal{A})$, and thus converges pointwise on $\mathcal{X}$ as $n \to \infty$.

Since $\mathcal{T}$ and $\mathcal{P}_*$ are continuous operators (Lemma 6 and Eq. (4)), and $\mathcal{T}$ is fully-determined by the restriction to $\mathcal{X}$, the sequence $(\mathcal{P} \circ \mathcal{T})d_n = d_{n+1}$ converges pointwise on its full domain. The pointwise limit of d_n is a fixed point of $(\mathcal{P}_* \circ \mathcal{T} \circ \mathcal{I})$, which must be the unique fixed point d^*_{SD} on $\mathcal{D}^*_+$ by Lemma 4. □

Lemma 4. *The optimal successor distance d^*_{SD} is the unique fixed point of $\mathcal{P}_* \circ \mathcal{T} \circ \mathcal{I}$ on $\mathcal{D}^*_+$.*

Proof of Lemma 4. For existence, we note

$$\begin{aligned}
(\mathcal{P}_* \circ \mathcal{T} \circ \mathcal{I}) d^*_{\text{SD}} &= (\mathcal{P}_* \circ \mathcal{T})(\mathcal{I} d^*_{\text{SD}}) && \text{(Remark 2)} \\
&= (\mathcal{P}_* \circ \mathcal{T}) d^*_{\text{SD}} && \text{(Bellman optimality of } Q^*_g\text{)} \\
&= \mathcal{P}_* d^*_{\text{SD}}. && \text{(Lemma 5)} \\
&= d^*_{\text{SD}}. && (30)
\end{aligned}$$

For uniqueness, we need to show that $(\mathcal{P}_* \circ \mathcal{T} \circ \mathcal{I})$ has no fixed points besides d^*_{SD} in $\mathcal{D}^*_+$. Suppose there exists some $d \in \mathcal{D}^*_+$ such that $(\mathcal{P}_* \circ \mathcal{T} \circ \mathcal{I})d = d$. Then, we have for $x \in \mathcal{S} \cup \mathcal{S} \times \mathcal{A}$, $s, g \in \mathcal{S}$, and $a \in \mathcal{A}$:

$$(\mathcal{P}_* \circ \mathcal{T} \circ \mathcal{I}) d\big(x, (g, a)\big) = d\big(x, (g, a)\big) = d(x, g). \tag{31}$$

Denote by $Q(s,a) = e^{-d((s,a),g)}$, and let $\mathcal{B}$ be the goal-conditioned Bellman operator defined as

$$\mathcal{B}Q(s,a) \triangleq \mathbb{E}_{\mathrm{P}(s'|s,a)}\big[\mathbb{1}\{s'=g\} + \gamma Q(s',g)\big] \tag{32}$$

At any fixed point $d \in \mathcal{D}_+^*$, we have

$$(\mathcal{P}_* \circ \mathcal{T} \circ \mathcal{I}) d\big((s,a),g\big) = d\big((s,a),g\big) \tag{33}$$

This last expression implies that

$$\begin{aligned} Q(s,a) &= \exp\big[-d\big((s,a),g\big)\big] \\ &= \exp\big[-(\mathcal{P}_* \circ \mathcal{T} \circ \mathcal{I}) d\big((s,a),g\big)\big] \\ &\leq \mathbb{E}_{\mathrm{P}(s'|s,a)}\big[\min_{a' \in \mathcal{A}} \exp d\big((s',a'),g\big)\big] - \log\gamma \\ &= \mathcal{B}Q(s,a). \end{aligned} \tag{34}$$

Since $\mathcal{B}$ is a contraction on the exponentiated distance space, and $d\big((s,a),g\big) \geq d^*_{\mathrm{SD}}\big((s,a),g\big)$, Eq. (34) is only consistent with $Q(s,a) = Q^*_g(s,a)$. This implies that

$$d\big((s,a),g\big) = d^*_{\mathrm{SD}}\big((s,a),g\big). \tag{35}$$

We also know that at this fixed point, $d(s,(s,a)) = 0$, and thus from Eq. (35) we have

$$d(s,g) = d^*_{\mathrm{SD}}(s,g). \tag{36}$$

So, $d = d^*_{\mathrm{SD}}$ must be the unique fixed point of $(\mathcal{P}_* \circ \mathcal{T} \circ \mathcal{I})$.

□

C Path Relaxation and Quasimetric Distances

We provide short proofs of the claims in Section 3.2

Lemma 5. *We have $\mathcal{P}(d) = d$ if and only if $d \in \mathcal{Q}$.*

Proof. $$\begin{aligned} \mathcal{P}(d)(s,g) &= \min_{w \in \mathcal{S}}\big[d(s,w) + d(w,g)\big] \\ &\leq d(s,s) + d(s,g) \\ &= d(s,g). \end{aligned}$$ □

Lemma 6. *The path relaxation operator $\mathcal{P}$ is continuous with respect to the L^∞ topology over $\mathcal{D}$.*

Proof. Let $d, d' \in \mathcal{D}$ and $\epsilon > 0$. We have

$$\begin{aligned} \big|\mathcal{P}(d)(s,g) - \mathcal{P}(d')(s,g)\big| &= \Big|\min_{w \in \mathcal{S}}\big[d(s,w) + d(w,g)\big] - \min_{w \in \mathcal{S}}\big[d'(s,w) + d'(w,g)\big]\Big| \\ &\leq \min_{w \in \mathcal{S}}\big|d(s,w) + d(w,g) - d'(s,w) - d'(w,g)\big| \\ &\leq \min_{w \in \mathcal{S}}\big|d(s,w) - d'(s,w)\big| + \min_{w \in \mathcal{S}}\big|d(w,g) - d'(w,g)\big| \\ &\leq \|d - d'\|_\infty + \|d - d'\|_\infty \\ &= 2\|d - d'\|_\infty. \end{aligned}$$

Thus, if $\|d - d'\|_\infty < \epsilon/2$, we have $\big\|\mathcal{P}(d) - \mathcal{P}(d')\big\|_\infty < \epsilon$. □

Lemma 7. *For any $s, g \in \mathcal{S}$ and $d \in \mathcal{D}$ we have that $\mathcal{P}(d)(s,g) \leq d(s,g)$.*

Proof. Let $d, d' \in \mathcal{D}$ and $\epsilon > 0$. We have

$$\begin{aligned} \big|\mathcal{P}(d)(s,g) - \mathcal{P}(d')(s,g)\big| &= \Big|\min_{w \in \mathcal{S}}\big[d(s,w) + d(w,g)\big] - \min_{w \in \mathcal{S}}\big[d'(s,w) + d'(w,g)\big]\Big| \\ &\leq \min_{w \in \mathcal{S}}\big|d(s,w) + d(w,g) - d'(s,w) - d'(w,g)\big| \\ &\leq \min_{w \in \mathcal{S}}\big|d(s,w) - d'(s,w)\big| + \min_{w \in \mathcal{S}}\big|d(w,g) - d'(w,g)\big| \\ &\leq \|d - d'\|_\infty + \|d - d'\|_\infty \\ &= 2\|d - d'\|_\infty. \end{aligned}$$

Table 2: Hyperparameters for TMD

Hyperparameter	Value
batch size	256
learning rate	$3 \cdot 10^{-4}$
discount factor	0.995
invariance weight ζ	0.2

Table 3: Network configuration for TMD.

Configuration	Value
latent dimension size	512
encoder MLP dimensions	$(512, 512, 512)$
policy MLP dimensions	$(512, 512, 512)$
layer norm in encoder MLPs	`True`
visual encoder (`visual-` envs)	`impala-small`
MRN components	8

Thus, if $\|d - d'\|_\infty < \epsilon/2$, we have $\left\|\mathcal{P}(d) - \mathcal{P}(d')\right\|_\infty < \epsilon$. □

Lemma 5. *We have $\mathcal{P}(d) = d$ if and only if $d \in \mathcal{Q}$.*

Proof. ($\Rightarrow$) Suppose $\mathcal{P}(d) = d$. Then, for all $s, g, w \in \mathcal{S}$ we have

$$d(s, g) = \mathcal{P}(d)(s, g) = \min_{w \in \mathcal{S}} \big[d(s, w) + d(w, g)\big] \leq d(s, w) + d(w, g).$$

Thus, $d \in \mathcal{Q}$.

($\Leftarrow$) Suppose $d \in \mathcal{Q}$. Then, for all $s, g \in \mathcal{S}$ we have

$$d(s, g) \leq \min_{w \in \mathcal{S}} \big[d(s, w) + d(w, g)\big] = \mathcal{P}(d)(s, g).$$

We also have $\mathcal{P}(d)(s, g) \leq d(s, g)$ by Lemma 7. Thus, $\mathcal{P}(d) = d$.

□

D Experimental Details

General hyperparameters are provided in Table 2.

We implemented TMD using JAX [45] within the OGBench [41] framework. OGBench requires a per-environment hyperparameter α controlling the behavioral cloning weight to be tuned for each method based on the scale of its losses. We generally found TMD to work well with similar α values to those used by CRL. We used the same values of α as CRL in OGBench [41, Table 9], with the exception for `humanoidmaze-large-stitch`, in which we used 0.2 instead of 0.1.

To prevent gradients from overflowing, we clip the $\mathcal{T}$ invariance loss per component to be no more than 5. We also found using a slightly smaller batch size of 256 compared to 512 to be helpful for reducing memory usage.

D.1 Implementation Details

The network architecture for TMD is described in Table 3. The "MRN components" refers to the number of ensemble terms K in Eq. (20). We found 8 components enabled stable learning and expressive distances.

D.2 Ablations

The full ablation results for TMD in the `pointmaze-teleport-stitch` is presented in Table 4, with both success rates and standard errors.

Table 4: Ablation Success rate.

Ablation	Success Rate
None	**29.3**$^{(\pm 2.2)}$
No gradient stopping in $\mathcal{L}_{\mathcal{T}}$	18.7$^{(\pm 1.8)}$
No contrastive loss	9.8$^{(\pm 1.7)}$
No $\mathcal{I}$ loss	13.3$^{(\pm 2.9)}$
No $\mathcal{T}$ loss	18.5$^{(\pm 2.1)}$

D.3 Computational Resources

Experiments were run using NVIDIA A6000 GPUs with 48GB of memory, and 4 CPU cores and 1 GPU per experiment. Each experiment took around 4 hours to run with these resources.

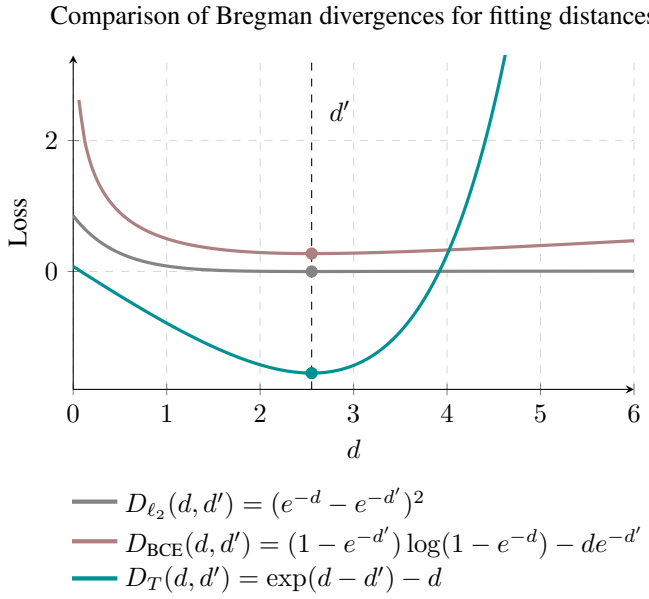

Figure 4: Comparison of Bregman divergences for e^{-d} onto $e^{-d'}$ in expectation. All losses are minimized at $d = d'$, and share the property that they will be minimized in expectation when $e^{-d} = \mathbb{E}[e^{-d'}]$. But only the $D_T(d, d')$ loss has non-vanishing gradients $d \gg d'$ for large d'.

E Bregman Divergence in $\mathcal{T}$-invariance

Recall the divergence used in Eq. (24):

$$D_T(d, d') \triangleq \exp(d - d') - d. \tag{24}$$

This divergence is proportional to the Bregman divergence [36] for the function $F(x) = -\log(x)$, similar to the Itakura-Saito divergence [37].

$$\begin{aligned} D_F(e^{-d'}, e^{-d}) &= F(e^{-d'}) - F(e^{-d}) - F'(e^{-d})(e^{-d'} - e^{-d}) \\ &= d' - d + \frac{1}{e^{-d}}(e^{-d'} - e^{-d}) \\ &= d' - d + \exp(d - d') - 1 \\ &= D_T(d, d') + d' - 1. \end{aligned} \tag{37}$$

The minimizer of Eq. (24) satisfies

$$\arg\min_{d \geq 0} \mathbb{E}_{d'}[D_T(d, d')] = -\log \mathbb{E}_{d'}[e^{-d'}] \tag{38}$$

when d' is a random "target" distance [46]. In other words, using Eq. (24) as a loss function regresses e^{-d} onto the expected value of $e^{-d'}$ (or onto the expected value of $e^{\log \gamma - d'}$ as used in Eq. (25)).

The key advantage of this divergence when backing up temporal distances is that the gradients do not vanish when either d, d', or the difference between them is small or large. This property is *not* shared by more standard loss functions like the squared loss or binary cross-entropy loss when applied to the probability space and the models (distances) are in log-probability space.

Algorithm 1: Temporal Metric Distillation (TMD)

input: dataset $\mathcal{D}$, learning rate η
initialize representations ϕ, ψ, policy π
while training **do**
  sample $\mathcal{B} = \{s_i, a_i, s'_i, g_i\}_{i=1}^N \sim \mathcal{D}$
  $\overline{\psi} \leftarrow \psi$
  $(\phi, \psi) \leftarrow (\phi, \psi) - \eta \nabla_{\phi,\psi} \mathcal{L}_{\text{TMD}}(\phi, \psi; \overline{\psi}, \mathcal{B})$ ▷ *Eq.* (26)
  $\pi \leftarrow \pi - \eta \nabla_{\pi} \mathcal{L}_{\pi}(\phi, \psi, \pi; \mathcal{B})$ ▷ *Eq.* (27)
return π

E.1 Empirical Comparison

Table 5: Ablation of $\mathcal{T}$-invariance loss in `pointmaze-teleport-stitch`

Loss	Success Rate	Standard Error
D_T **(Ours)**	**29.3**$^{(\pm \mathbf{2.2})}$	**5.4**$^{(\pm \mathbf{0.5})}$
D_{ℓ_2}	16.1$^{(\pm 1.9)}$	2.0$^{(\pm 0.3)}$
D_{BCE}	15.1$^{(\pm 1.9)}$	1.9$^{(\pm 0.3)}$

In practice, we found it was important to use this divergence in TMD for stable learning (Table 5). Ths loss could also be applied to other goal-conditioned RL algorithms where learned value functions are probabilities but are predicted in log-space to improve gradients. Future work should explore this divergence in other goal-conditioned RL algorithms to improve training compared to the more commonly used squared loss or binary cross-entropy loss [35].

F Algorithm Pseudocode

Full pseudocode for TMD is provided in Algorithm 1. We provide the full TMD loss function in Eq. (26) and the policy extraction loss in Eq. (27) below for reference:

$$\mathcal{L}_{\text{TMD}}\big(\phi, \psi; \overline{\psi}, \mathcal{B}\big) = \mathcal{L}_{\text{NCE}}\left(\phi, \psi; \mathcal{B}\right) + \zeta\Big(\mathcal{L}_{\mathcal{I}}\left(\phi, \psi; \mathcal{B}\right) + \mathcal{L}_{\mathcal{T}}\left(\phi, \overline{\psi}; \mathcal{B}\right)\Big) \tag{26}$$

$$\mathcal{L}_{\pi}\big(\pi; \phi, \psi, \{s_i, a_i, s'_i, g_i\}_{i=1}^N\big) = \sum_{i,j=1}^{N} (1-\lambda) d_{\text{MRN}}\big(\phi(s_i, \hat{a}_{ij}), \psi(g_j)\big), g_j\big) + \lambda d_{\text{MRN}}\big(\phi(s_i, \hat{a}_{ii}), \psi(g_i)\big) + \alpha \big\|\hat{a}_{ii} - a_i\big\|_2^2 \tag{27}$$

where $\hat{a}_{ij} = \pi(s_i, g_j)$, batch $\mathcal{B} \sim p^{\pi_\beta} = \{s_i, a_i, s'_i, g_i\}_{i=1}^N$.

The components of Eq. (26) are (see Section 4):

$$\mathcal{L}_{\text{NCE}}\left(\phi,\psi;\{s_i,a_i,s'_i,g_i\}_{i=1}^N\right)=\sum_{i=1}^N\log\left(\frac{e^{f(s_i,a_i,g_i)}}{\sum_{j=1}^N e^{f(s_j,a_j,g_i)}}\right) \tag{18}$$

$$\mathcal{L}_{\mathcal{I}}\left(\phi,\psi;\{s_i,a_i,s'_i,g_i\}_{i=1}^N\right)=\sum_{i,j=1}^N d_{\text{MRN}}\big(\psi(s_i),\phi(s_i,a_j)\big) \tag{22}$$

$$\mathcal{L}_{\mathcal{T}}\left(\phi,\psi;\{s_i,a_i,s'_i,g_i\}_{i=1}^N\right)=\sum_{i,j=1}^N D_T\big(d_{\text{MRN}}(\phi(s_i,a_i),\psi(g_j)),d_{\text{MRN}}(\psi(s'_i),\psi(g_j))-\log\gamma\big). \tag{25}$$