QUASIMETRIC GOAL REPRESENTATION FOR OFFLINE GOAL-CONDITIONED RL

Anonymous authors
Paper under double-blind review

ABSTRACT

Goal-conditioned reinforcement learning (GCRL) requires goal representations that preserve information for optimal control while discarding exogenous noise. We introduce Quasimetric Goal Representation (Q-Rep), motivated by the asymmetry of optimal temporal distances. Under an exact quasimetric embedding, the resulting distance-profile representation satisfies sufficiency and, in the Exogenous Block CMP setting, noise invariance. A weighted asymmetric ℓ^1 parameterization provides an efficient practical instantiation. Across 17 state- and pixel-based OGBench tasks, Q-Rep achieves the highest or tied-highest mean score on 13 tasks compared with raw-goal and prior representation baselines. Code: https://anonymous.4open.science/r/Q-Rep-316F.

1 INTRODUCTION

Goal-conditioned reinforcement learning (GCRL) (Kaelbling, 1993; Park et al., 2024) learns policies for tasks specified by goal configurations. A goal-dependent success signal supports training across tasks with little manual reward design. Applications include navigation (Andrychowicz et al., 2017), manipulation (Plappert et al., 2018), and games (Schaul et al., 2015).

Raw goal observations can mix control-relevant information with visual distractors or sensor noise. A useful representation should retain the information needed for an optimal goal-conditioned policy (*sufficiency*) while discarding factors irrelevant to reaching the goal (*noise invariance*) (Park et al., 2025). Constructing a representation with both properties remains challenging.

Goal-reaching dynamics are often directional: climbing to a target can take more time or effort than descending. This asymmetry suggests that symmetric metrics may fail to capture the geometry relevant to control.

The optimal temporal distance, which equals shortest-path travel time in deterministic environments, induces a quasimetric. Because this distance is defined through the optimal goal-conditioned value function, an exact embedding preserves information for control. In an Exogenous Block CMP, goals corresponding to the same latent state have identical distance profiles, giving noise invariance.

We use this connection to construct Q-Rep: an encoder maps observations to a latent space equipped with a parameterized quasimetric. Our analysis establishes sufficiency and noise invariance for exact embeddings. Practical training approximates the embedding condition using offline value learning.

We instantiate the latent geometry with a weighted asymmetric ℓ^1 function, assigning separate weights to positive and negative displacements. This requires only element-wise operations and improves performance on most evaluated state- and pixel-based OGBench tasks.

1.1 CONTRIBUTIONS

Our contributions are twofold:

- We establish a geometric foundation for goal representation: exact quasimetric embeddings preserve information for optimal goal-conditioned control and yield noise-invariant distance profiles under the Ex-BCMP model.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

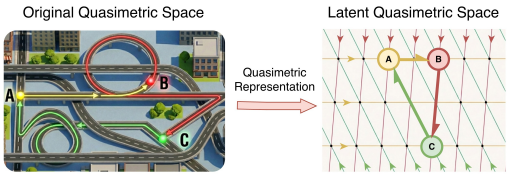


Figure 1: A quasimetric embedding $\phi(s)$ compresses a high-dimensional real-world quasimetric into a lower-dimensional latent space while preserving the quasimetric, yielding an effective goal-state representation.

- We instantiate Q-Rep with an efficient weighted asymmetric ℓ^1 parameterization. The learned representation improves on raw-goal and prior representation baselines on most evaluated state- and pixel-based OGBench tasks.

Relation to prior work. Dual Goal Representation (Park et al., 2025) uses bilinear representations, while TRA (Myers et al., 2025) aligns temporal representations through contrastive learning. Q-Rep instead parameterizes the asymmetric geometry of goal-reaching. Appendix E discusses these and related offline GCRL methods.

2 Preliminaries

Goal-Conditioned Reinforcement Learning. We consider a controlled Markov process (CMP) with goal-conditioned rewards, specified by $\mathcal{M} = (\mathcal{S}, \mathcal{A}, p, r, \gamma)$. Here, $\mathcal{S}$ and $\mathcal{A}$ are the state and action spaces, and $p(s' \mid s, a) : \mathcal{S} \times \mathcal{A} \to \Delta(\mathcal{S})$ is the transition kernel. In this work, we present our theoretical analysis under the assumption that both $\mathcal{S}$ and $\mathcal{A}$ are discrete. This restriction is adopted solely to streamline notation and avoid measure-theoretic technicalities; the core arguments do not rely on discreteness and can be extended to continuous spaces with standard modifications.

Given a goal $g \in \mathcal{S}$, the objective of GCRL is to learn a policy $\pi(a \mid s, g)$ that maximizes the cumulative discounted return. We define the value function V^π and the action-value function Q^π as:

$$V^\pi(s, g) = \mathbb{E}_{\tau \sim p^\pi(\cdot \mid s_0 = s, g)} \left[\sum_{t=0}^{\infty} \gamma^t r(s_t, g) \right],$$

$$Q^\pi(s, a, g) = \mathbb{E}_{\tau \sim p^\pi(\cdot \mid s_0 = s, a_0 = a, g)} \left[\sum_{t=0}^{\infty} \gamma^t r(s_t, g) \right],$$

where $\gamma \in (0, 1)$ is the discount factor and p^π denotes the trajectory distribution induced by the policy and transition dynamics. The optimal value functions are denoted by $V^*(s, g) = \max_\pi V^\pi(s, g)$ and $Q^*(s, a, g) = \max_\pi Q^\pi(s, a, g)$.

We consider a standard sparse-reward formulation in which a unit reward is granted upon first reaching the goal state, i.e., $r(s, g) = \mathbb{I}(s = g)$, where $\mathbb{I}$ is the indicator function. After this event, the process transitions to an absorbing terminal state, ensuring that the reward is received at most once along any trajectory. Define the first hitting time from state s to goal g as

$$T^\pi(s, g) \triangleq \inf\{t \geq 0 \mid s_t = g\}.$$

Under the above definition, we have

$$V^\pi(s, g) = \mathbb{E}\left[\gamma^{T^\pi(s,g)}\right].$$

A central quantity in this work is the *optimal temporal distance*, defined as

$$d^*(s, g) \triangleq \log_\gamma V^*(s, g). \tag{1}$$

If $V^*(s, g) = 0$ (i.e., the goal is unreachable), we set $d^*(s, g) = +\infty$. In deterministic environments, $V^*(s, g) = \gamma^{T^*(s,g)}$ and thus $d^*(s, g) = T^*(s, g)$, which coincides with the shortest-path (minimum-time) distance from s to g.

We focus on the offline setting, where learning is based on a static dataset $\mathcal{D} = \{\tau^{(i)}\}_{i=1}^{M}$ of trajectories $\tau = (s_0, a_0, s_1, \ldots, s_T)$ collected by an (unknown) behavior policy, without further environment interaction.

Exogenous Block CMPs. To analyze environments with high-dimensional observations driven by lower-dimensional dynamics, we utilize the Exogenous Block CMP (Ex-BCMP) framework (Efroni et al., 2021). An Ex-BCMP is characterized by the tuple $(\mathcal{S}, \mathcal{Z}, \mathcal{A}, p^{\ell}, p^{e}, \psi)$. In this model, $\mathcal{S}$ is an observation space, $\mathcal{A}$ is an action space, and $\mathcal{Z}$ represents the underlying latent state space governed by the transition kernel $p^{\ell}(z' \mid z, a)$. Observations $s \in \mathcal{S}$ are generated via an emission distribution $p^{e}(s \mid z)$. A key property of Ex-BCMPs is that observations provide sufficient information to recover the latent state; specifically, there exists a mapping function $\psi : \mathcal{S} \to \mathcal{Z}$ such that $\psi(s) = z$ holds for all $z \in \mathcal{Z}$ and $s \sim p^{e}(\cdot \mid z)$.

Intuitively, this structure describes a latent CMP $(\mathcal{Z}, \mathcal{A}, p^{\ell})$ augmented with exogenous noise, where the supports of emission distributions for distinct latent states are disjoint (i.e., every observation maps to a unique latent state).

Consistent with this structure, the reward function is defined on the latent space as $r^{\ell}(s, g) = \mathbb{I}(\psi(s) = \psi(g))$. The value functions and temporal distance definitions follow the same formulation as the standard CMP but utilize this latent-based reward r^{ℓ}. This framework is particularly relevant for tasks such as visual robotic control (Zhang et al., 2020), where the agent observes high-dimensional pixel inputs while the underlying physics (e.g., object poses) operate in a compact latent space.

3 Quasimetric Goal Representation

In this section, we introduce the main idea of Quasimetric Goal Representation. We aim to learn an effective goal representation $\phi : \mathcal{S} \to \mathcal{Z}$ for a goal-conditioned policy $\pi(a \mid s, \phi(g))$ parameterized by ϕ, where $\mathcal{Z}$ denotes a latent representation space.

The learned geometry is intended to expose task reachability to the policy and support generalization across goals.

3.1 Main Idea

Our idea originates from the inherent irreversibility of many physical environments—transitioning from state s to g may be easy, while returning from g to s could be difficult or impossible. Standard metrics (e.g., Euclidean distance) assume symmetry ($d(s, g) = d(g, s)$) and thus fail to capture these directed dynamics. To address this, we turn to the concept of quasimetric (Cobzas, 2012):

Definition 1 (Quasimetric)**.** A *quasimetric* on $\mathcal{S}$ is a function $d : \mathcal{S} \times \mathcal{S} \to \mathbb{R}$ satisfying the following properties for all $x, y, z \in \mathcal{S}$:
- **Positivity:** $d(x, y) \geq 0$.
- **Identity:** $d(x, y) = 0 \iff x = y$.
- **Triangle inequality:** $d(x, z) \leq d(x, y) + d(y, z)$.

While the utility of quasimetrics has been explored in metric learning (Wang & Isola, 2022) and recent reinforcement learning works (Liu et al., 2023; Myers et al., 2024), these approaches typically rely on constructing auxiliary distance functions (Liu et al., 2023) or learning successor features (Myers et al., 2024).

In contrast, we identify a more fundamental connection: the optimal temporal distance d^*, as derived directly from the optimal value function in Eq. (1), naturally constitutes a quasimetric structure under our GCRL formulation. This insight implies that learning the geometry of the representation space is equivalent to learning the optimal value function.

Theorem 3.1. *The optimal temporal distance $d^*(s, g) \triangleq \log_{\gamma} V^*(s, g)$ is a quasimetric over $\mathcal{S}$, satisfying all properties in Definition 1.*

It is important to note that the optimal value function $V^*(s, g)$ (Wang et al., 2023) does not qualify as a quasimetric under the rigorous mathematical definition (Cobzas, 2012), as it fails to satisfy the identity property. Similarly, recent work by Park et al. (2025) attempts to approximate $d^*(s, g)$ using a bilinear form $\phi_1(s)^\top \phi_2(g)$. However, this formulation is not a quasimetric, thereby inevitably missing the critical geometric structure inherent to d^*. In contrast, our proposed quasimetric goal representation is explicitly designed to preserve this fundamental quasimetric structure.

We then introduce the notions of asymmetric norms and quasimetric embeddings, which form the foundation of our goal representation design.

Definition 2 (Asymmetric norm)**.** An *asymmetric norm* on a real vector space $\mathcal{X}$ is a function $F : \mathcal{X} \to \mathbb{R}_{\geq 0}$ satisfying the following properties for all $x, y \in \mathcal{X}$ and all $\lambda \geq 0$:
- **Nonnegativity:** $F(x) \geq 0$, $F(x) = 0 \iff x = 0$,
- **Subadditivity:** $F(x + y) \leq F(x) + F(y)$,
- **Positive homogeneity:** $F(\lambda x) = \lambda F(x)$.

In general, an asymmetric norm does not satisfy the symmetry condition $F(x) = F(-x)$. Given such a function F, it naturally induces a quasimetric d_F defined by

$$d_F(x, y) \triangleq F(y - x).$$

We refer to d_F as the quasimetric induced by the asymmetric norm F.

Definition 3 (Quasimetric Embedding)**.** Let $(\mathcal{S}, d^*)$ be the state space equipped with the optimal temporal distance. A *quasimetric embedding* of $(\mathcal{S}, d^*)$ is a pair (ϕ, F), where $\phi : \mathcal{S} \to \mathcal{Z}$ maps $\mathcal{S}$ into a latent vector space $\mathcal{Z}$ and $F : \mathcal{Z} \to \mathbb{R}_{\geq 0}$ is an asymmetric norm inducing the latent quasimetric $d_F(z_1, z_2) \triangleq F(z_2 - z_1)$, such that for all $s, g \in \mathcal{S}$,

$$d_F\big(\phi(s), \phi(g)\big) = d^*(s, g). \tag{2}$$

According to Theorem 3.1, the optimal temporal distance d^* constitutes a quasimetric on the state space $\mathcal{S}$. Our definition of a quasimetric embedding formally establishes an exact isometry between the task quasimetric space $(\mathcal{S}, d^*)$ and the latent quasimetric space $(\mathcal{Z}, d_F)$.

Recall that $d^*(s, g) = \log_\gamma V^*(s, g)$, such a realization preserves the full optimal value landscape and therefore retains all information required for optimal goal-conditioned control under the standard dynamic-programming principle. Therefore, a valid quasimetric embedding acts as a lossless representation of the control problem: the induced latent quasimetric d_F naturally recovers the optimal goal-conditioned policy without approximation errors.

In summary, our approach seeks a latent mapping ϕ and an induced quasimetric d_F that faithfully reconstruct the task's intrinsic quasimetric structure.

3.2 Theoretical Foundations

We now establish sufficiency and noise invariance for the exact distance-profile representation.

Definition 4 (Quasimetric Goal Representation)**.** Let $\mathcal{M} = (\mathcal{S}, \mathcal{A}, p, r, \gamma)$ be a CMP, and let (ϕ, F) be a quasimetric embedding of $\mathcal{S}$. Let $\mathbb{R}^{\mathcal{S}}$ be the space of functions from $\mathcal{S}$ to $\mathbb{R}$. The *quasimetric goal representation* is a map $f : \mathcal{S} \to \mathbb{R}^{\mathcal{S}}$ such that, for a goal $g \in \mathcal{S}$, the representation of g is the function f_g defined by

$$f_g(s) \triangleq F\big(\phi(g) - \phi(s)\big) \quad \text{for all } s \in \mathcal{S}.$$

The *quasimetric profile space* is the image of $\mathcal{S}$ under the quasimetric goal representation, namely

$$\widehat{\mathcal{S}} \triangleq \{f_g \mid g \in \mathcal{S}\}.$$

Intuitively, for every goal g, the quasimetric goal representation yields a function f_g that characterizes the goal by its geometry relative to the entire state space. Specifically, it maps any start state s to its estimated distance to g, satisfying $f_g(s) \triangleq F(\phi(g) - \phi(s))$. This effectively defines a goal-conditioned potential landscape over $\mathcal{S}$, characterizing the geometry of the state space relative to the goal g.

We then show the sufficiency property of such quasimetric goal representation.

Theorem 3.2 (Sufficiency). *Let $\mathcal{M} = (\mathcal{S}, \mathcal{A}, p)$ be a CMP and $\widehat{S}$ be the quasimetric profile space. Then, there exists a deterministic policy $\hat{\pi} : \mathcal{S} \times \widehat{\mathcal{S}} \to \mathcal{A}$ that takes a quasimetric goal representation as input, such that its induced policy $\tilde{\pi} : \mathcal{S} \times \mathcal{S} \to \mathcal{A}$ defined as $\tilde{\pi}(s, g) := \hat{\pi}(s, f_g)$ satisfies $V^{\tilde{\pi}}(s, g) = V^*(s, g)$ for all $s, g \in \mathcal{S}$.*

Theorem 3.2 establishes the *sufficiency* of quasimetric goal representations: conditioning a policy on f_g is, in principle, as expressive as conditioning on the raw goal g for optimal goal-reaching. Intuitively, since $f_g(s)$ encodes the asymmetric distance to the goal, the optimal policy $\hat{\pi}$ essentially operates as a greedy navigator on the potential landscape defined by f_g. This implies that preserving the underlying quasimetric structure is functionally equivalent to preserving the optimal policy itself, and f_g can be viewed as a lossless control-relevant description of the goal.

Importantly, the representation f_g depends on a goal only through its *relative geometry* with respect to all states, quantified by the latent quasimetric distance profile: $s \mapsto F(\phi(g) - \phi(s))$. This structural dependence suggests that nuisance variations in the observation of a goal that do not change the underlying latent state are naturally filtered out. We formalize this intuition in the next theorem, which shows that under the Ex-BCMP model, two goal observations corresponding to the same latent state induce *identical* quasimetric goal representations, yielding the desired *noise invariance*.

Theorem 3.3 (Noise Invariance). *Let $\mathcal{M} = (\mathcal{S}, \mathcal{Z}, \mathcal{A}, p^\ell, p^e, \psi)$ be an Ex-BCMP and f be its quasimetric goal representation function. Let $g_1, g_2 \in \mathcal{S}$ be two goal observations from the same latent state; i.e., $\psi(g_1) = \psi(g_2)$. Then, they have the same quasimetric goal representation; i.e.,*

$$f_{g_1} = f_{g_2}.$$

Detailed proofs appear in Appendix A. The profile construction is a directed analogue of the Kuratowski embedding (Heinonen, 2003): a latent goal z determines $h_z(\cdot) = F(z - \cdot)$, and $f_g = h_{\phi(g)} \circ \phi$. Thus $\phi(g)$ determines f_g. Conversely, if $f_{g_1} = f_{g_2}$, evaluation at $s = g_1$ gives $F(\phi(g_2) - \phi(g_1)) = 0$, so definiteness of F implies $\phi(g_1) = \phi(g_2)$. The vector and profile therefore carry the same information on the realized embedding image. We use the finite-dimensional vector $\phi(g)$ in the policy. These guarantees require exact distance preservation; the learned approximation is assessed empirically.

4 Practical Implementation

In this section, we introduce our practical method: **Q**uasimetric Goal **Rep**resentation (Q-Rep) for Offline Goal-Conditioned RL.

Parameterizing the Quasimetric Embedding. Recall that the core objective of Q-Rep is to construct a quasimetric embedding (ϕ, F) that satisfies the isometry condition:

$$d_F\big(\phi(s), \phi(g)\big) \triangleq F\big(\phi(g) - \phi(s)\big) = d^*(s, g). \tag{3}$$

To approximate this isometry in practice, we parameterize $\phi : \mathcal{S} \to \mathbb{R}^n$ as a flexible trainable encoder (e.g., a neural network) without imposing additional architectural constraints. The key design choice is therefore the asymmetric norm F, which specifies a structured quasimetric geometry on the latent space and thereby constrains the learned representation in a principled manner.

To demonstrate the universality of the Q-Rep framework, we employ a simple yet canonical instance of asymmetric norms: the weighted asymmetric ℓ_1 norm. This choice obviates the need for intricate,

task-specific geometric designs, allowing us to rely instead on the framework's inherent capacity to learn the underlying topology. Formally, we parameterize $F : \mathbb{R}^n \to \mathbb{R}_{\geq 0}$ as:

$$F(x) = \sum_{i=1}^{n} \left(\alpha_i \max(x_i, 0) + \beta_i \max(-x_i, 0) \right), \tag{4}$$

where $\alpha, \beta \in \mathbb{R}^n_{\geq 0}$ are trainable weight vectors. This formulation effectively decouples the asymmetry across latent dimensions, assigning distinct unit costs (α_i versus β_i) to positive and negative deviations along each axis. Note that in the special case where $\alpha_i = \beta_i$ for all i, F reduces to a standard weighted ℓ_1 norm, rendering d_F a symmetric metric instead of a quasimetric.

While we adopt this formulation for its efficiency and interpretability, we note that the framework is compatible with a broader class of asymmetric norms (Cobzas, 2012), which may offer further performance gains in specific scenarios.

Approximating the optimal temporal distance. To supervise the learning of Q-Rep, we require an estimate of the optimal temporal distance $d^*(s, g)$ from offline data. Instead of learning d^* directly, we exploit the relationship $d^*(s, g) = \log_\gamma V^*(s, g)$ in Eq. (1) and first learn the optimal goal-conditioned value function V^* using an existing offline value-learning algorithm. In particular, we adopt goal-conditioned IQL (GCIQL) (Kostrikov et al., 2021) for its simplicity and strong empirical performance. This supervision step depends only on value learning and can be readily instantiated with other goal-conditioned offline value-learning methods; in our experiments, GCIQL provides a convenient and effective choice.

Algorithm Implementation. Algorithm 1 outlines the complete training procedure, which proceeds in two coordinated phases: representation learning and policy learning.

Representation Learning. In the first phase (Lines 5–9), we optimize the encoder ϕ and metric parameters F to align the latent geometry with the task's temporal dynamics. Specifically, for each batch, we compute the latent distance $d_F\big(\phi(s), \phi(g)\big)$ and the corresponding induced value $V_F(s, g) = \gamma^{d_F}$. Crucially, we treat this geometry-induced value V_F as the parameterized value function to be learned via the GCIQL objective (Kostrikov et al., 2021) (Lines 7–9 in Algorithm 1). Through this optimization, we effectively distill the optimal value landscape directly into the quasimetric embedding, yielding our final representation ϕ.

Policy Learning. In the second phase (Line 10), we train a goal-conditioned policy $\pi(a \mid s, \phi(g))$ and value function $V(s, \phi(g))$ utilizing the learned Q-Rep $\phi(g)$ as the goal representation. While our framework is compatible with various offline GCRL algorithms (e.g., Contrastive RL (Eysenbach et al., 2022)), we instantiate this step using GCIVL (Park et al., 2024). This choice ensures a fair comparison with the baselines in Table 1, which standardly adopt GCIVL for policy optimization.

5 Experiments

We evaluate Q-Rep on OGBench, compare asymmetric and symmetric representations, examine distance approximation, and measure robustness to noisy goals.

Dataset. We evaluate our method on OGBench (Park et al., 2024), a benchmark suite designed for offline goal-conditioned RL. The suite encompasses a diverse array of goal-reaching tasks, ranging from robotic navigation (e.g., `pointmaze`, `antmaze`, `humanoidmaze`) to complex manipulation (e.g., `cube`, `scene`, `puzzle`). For pixel-based variants, agents operate solely from $64 \times 64 \times 3$ image observations. We utilize the standard Maps and play datasets, which are collected via task-agnostic, "play"-style demonstrations (Lynch et al., 2020) featuring diverse atomic behaviors. Please refer to Appendix B for comprehensive details regarding the tasks and dataset specifications.

5.1 OGBench Results

We evaluate Q-Rep (Algorithm 1) on the OGBench suite across 17 robotic navigation and manipulation tasks, comprising 12 state-based and 5 pixel-based (visual) environments. We benchmark our approach against several goal representation learning methods, including VIB (Shah et al., 2021), VIP (Ma et al., 2022), TRA (Myers et al., 2025), BYOL-γ (Lawson et al., 2025), and Dual (Park

Algorithm 1 Offline Goal-Conditioned RL with Quasimetric Goal Representations.

Initialize $\phi, F, Q(s, a, g)$ for representation learning
Initialize $\pi(a \mid s, \phi(g)), V(s, \phi(g))$ for policy training
while not converged **do**
  Sample batch $\{(s, a, s', g)^{(i)}\}_i$ from $\mathcal{D}$
  ▷ Representation Learning from GCIQL
  Compute $d_F(\phi(s), \phi(g))$ via Eq. (3)
  Set $V_F(s, g) = \gamma^{d_F}$ according to Eq. (1)
  Update ϕ, F by minimizing

$$\mathbb{E}[\ell_\kappa^2(V_F(s, g) - \bar{Q}(s, a, g))]$$

  Update Q by minimizing

$$\mathbb{E}[(Q(s, a, g) - r(s, g) - \gamma V_F(s', g))^2]$$

  Update $\bar{Q}$ using exponential moving averaging
  ▷ Policy Learning from GCIVL
  Update $\pi(a \mid s, \phi(g)), V(s, \phi(g))$ with GCIVL
end while
return $\pi(a \mid s, \phi(g))$

Table 1: Normalized average scores on OGBench tasks, averaged over 5 random seeds.

Environment	Orig	VIB	VIP	TRA	BYOL-γ	Dual	Q-Rep
pointmaze-medium-navigate-v0	$78_{\pm 8}$	$69_{\pm 13}$	$0_{\pm 1}$	$3_{\pm 6}$	$37_{\pm 7}$	$76_{\pm 7}$	$\mathbf{95}_{\pm 1}$
pointmaze-large-navigate-v0	$52_{\pm 6}$	$50_{\pm 7}$	$0_{\pm 0}$	$1_{\pm 2}$	$22_{\pm 12}$	$46_{\pm 6}$	$\mathbf{72}_{\pm 5}$
antmaze-medium-navigate-v0	$71_{\pm 4}$	$68_{\pm 4}$	$31_{\pm 5}$	$22_{\pm 15}$	$39_{\pm 5}$	$75_{\pm 4}$	$\mathbf{88}_{\pm 4}$
antmaze-large-navigate-v0	$16_{\pm 3}$	$9_{\pm 3}$	$9_{\pm 2}$	$22_{\pm 12}$	$11_{\pm 5}$	$28_{\pm 11}$	$\mathbf{61}_{\pm 4}$
humanoidmaze-medium-navigate-v0	$27_{\pm 3}$	$24_{\pm 2}$	$7_{\pm 3}$	$21_{\pm 3}$	$18_{\pm 5}$	$29_{\pm 3}$	$\mathbf{48}_{\pm 2}$
humanoidmaze-large-navigate-v0	$3_{\pm 0}$	$3_{\pm 1}$	$1_{\pm 0}$	$2_{\pm 1}$	$2_{\pm 1}$	$3_{\pm 2}$	$\mathbf{6}_{\pm 1}$
antsoccer-arena-navigate-v0	$\mathbf{47}_{\pm 4}$	$34_{\pm 4}$	$2_{\pm 1}$	$8_{\pm 2}$	$11_{\pm 4}$	$31_{\pm 3}$	$46_{\pm 5}$
cube-single-play-v0	$52_{\pm 3}$	$\mathbf{90}_{\pm 3}$	$40_{\pm 7}$	$40_{\pm 5}$	$51_{\pm 11}$	$89_{\pm 3}$	$64_{\pm 1}$
cube-double-play-v0	$35_{\pm 5}$	$33_{\pm 3}$	$3_{\pm 2}$	$7_{\pm 2}$	$6_{\pm 4}$	$\mathbf{60}_{\pm 4}$	$35_{\pm 3}$
scene-play-v0	$46_{\pm 3}$	$58_{\pm 1}$	$23_{\pm 6}$	$46_{\pm 6}$	$44_{\pm 9}$	$\mathbf{72}_{\pm 6}$	$\mathbf{72}_{\pm 5}$
puzzle-3x3-play-v0	$5_{\pm 1}$	$\mathbf{14}_{\pm 3}$	$3_{\pm 1}$	$5_{\pm 1}$	$0_{\pm 0}$	$5_{\pm 1}$	$9_{\pm 2}$
puzzle-4x4-play-v0	$14_{\pm 1}$	$6_{\pm 3}$	$1_{\pm 1}$	$10_{\pm 3}$	$1_{\pm 2}$	$23_{\pm 3}$	$\mathbf{29}_{\pm 3}$
Average (state-based)	$37_{\pm 3}$	$38_{\pm 4}$	$10_{\pm 2}$	$16_{\pm 5}$	$20_{\pm 5}$	$45_{\pm 4}$	$\mathbf{52}_{\pm 3}$
visual-antmaze-medium-navigate-v0	$66_{\pm 4}$	$18_{\pm 9}$	$30_{\pm 7}$	$48_{\pm 4}$	$32_{\pm 5}$	$78_{\pm 4}$	$\mathbf{92}_{\pm 2}$
visual-antmaze-large-navigate-v0	$26_{\pm 5}$	$5_{\pm 2}$	$9_{\pm 1}$	$13_{\pm 3}$	$9_{\pm 4}$	$40_{\pm 4}$	$\mathbf{53}_{\pm 4}$
visual-cube-single-play-v0	$53_{\pm 4}$	$18_{\pm 19}$	$39_{\pm 6}$	$31_{\pm 24}$	$35_{\pm 8}$	$58_{\pm 5}$	$\mathbf{65}_{\pm 3}$
visual-cube-double-play-v0	$9_{\pm 2}$	$0_{\pm 0}$	$0_{\pm 0}$	$3_{\pm 2}$	$2_{\pm 1}$	$9_{\pm 2}$	$\mathbf{13}_{\pm 2}$
visual-scene-play-v0	$25_{\pm 2}$	$6_{\pm 3}$	$4_{\pm 1}$	$15_{\pm 6}$	$10_{\pm 8}$	$26_{\pm 5}$	$\mathbf{34}_{\pm 4}$
Average (pixel-based)	$36_{\pm 3}$	$9_{\pm 7}$	$16_{\pm 3}$	$22_{\pm 8}$	$18_{\pm 7}$	$42_{\pm 4}$	$\mathbf{51}_{\pm 3}$

et al., 2025), as well as a baseline using raw states ("Original"). We adopt baseline results from Park et al. (2025).

As summarized in Table 1, Q-Rep achieves the highest or tied-highest mean score on 13 of 17 tasks. In state-based domains, Q-Rep outperforms the strongest baseline by an average of **16%**. These gains are even more pronounced in pixel-based settings, where Q-Rep widens the margin to **21%**.

The larger gain on pixel-based tasks is consistent with the benefit of a structured representation when observations include visual distractors. This comparison alone does not isolate the mechanism responsible for the gain.

Table 2: Comparison of Q-Rep against a symmetric baseline (where $\alpha_i = \beta_i$) on state-based OGBench tasks. Results show normalized returns averaged over 5 seeds.

Environment	Symmetric	Asymmetric
pointmaze-medium-navigate-v0	$\mathbf{95}_{\pm 2}$	$\mathbf{95}_{\pm 1}$
pointmaze-large-navigate-v0	$60_{\pm 3}$	$\mathbf{72}_{\pm 5}$
antmaze-medium-navigate-v0	$\mathbf{90}_{\pm 2}$	$88_{\pm 4}$
antmaze-large-navigate-v0	$58_{\pm 4}$	$\mathbf{61}_{\pm 4}$
humanoidmaze-medium-navigate-v0	$32_{\pm 3}$	$\mathbf{48}_{\pm 2}$
humanoidmaze-large-navigate-v0	$\mathbf{6}_{\pm 2}$	$\mathbf{6}_{\pm 1}$
antsoccer-arena-navigate-v0	$39_{\pm 6}$	$\mathbf{46}_{\pm 5}$
cube-single-play-v0	$43_{\pm 1}$	$\mathbf{64}_{\pm 1}$
cube-double-play-v0	$28_{\pm 6}$	$\mathbf{35}_{\pm 3}$
scene-play-v0	$66_{\pm 8}$	$\mathbf{72}_{\pm 5}$
puzzle-3x3-play-v0	$8_{\pm 2}$	$\mathbf{9}_{\pm 2}$
puzzle-4x4-play-v0	$17_{\pm 4}$	$\mathbf{29}_{\pm 3}$
Average (all tasks)	$45_{\pm 4}$	$\mathbf{52}_{\pm 3}$

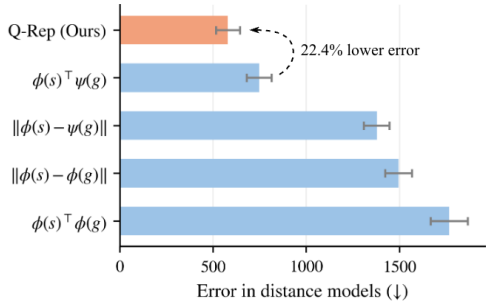


Figure 2: **Approximation error of different distance architectures.** Results are averaged over 4 random seeds across the four tasks: *antmaze-large-navigate, humanoidmaze-medium-navigate, antsoccer-arena-navigate-v0, and cube-double-play-v0.*

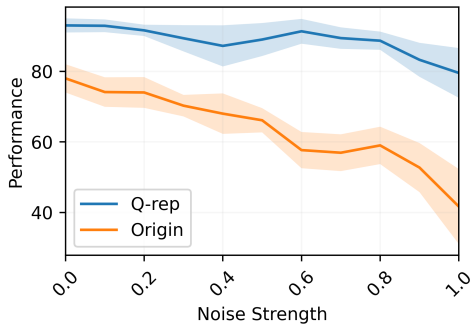


Figure 3: **Robustness to goal observation noise.** We evaluate performance on *pointmaze-medium-navigate-v0* under varying magnitudes of Gaussian noise injected into goal observations. Results are averaged over 4 random seeds.

5.2 Effect of Asymmetry

A distinctive property of our quasimetric approach is its asymmetry, which is pivotal for modeling the inherent irreversibility of optimal goal-reaching strategies. To isolate the role of this asymmetry, we compare Q-Rep against a symmetric baseline obtained by enforcing $\alpha_i = \beta_i$ while keeping all other architectural components unchanged.

Table 2 shows higher mean scores for the asymmetric model on 9 of 12 tasks, ties on 2, and a lower score on antmaze-medium. Its reported mean is 52 versus 45 for the symmetric model, an approximately 16% relative gain. The effect is therefore positive on average but task-dependent.

5.3 Approximation Capability of Distance Architectures

In this work, we posit that the optimal temporal distance $d^*(s, g)$ follows a quasimetric structure, reflecting the irreversibility of state transitions. We propose approximating this geometry using a weighted ℓ_1 quasimetric. A fundamental question arises: *does this specific geometric constraint yield a more accurate approximation of optimal temporal distance compared to alternative parameterizations?*

To investigate this, we benchmark the approximation accuracy of our method against several baseline parameterizations, including asymmetric bilinear, asymmetric Euclidean, symmetric Euclidean, and symmetric inner product forms. We construct a learned reference by training a 'monolithic' value function, $V_{\text{mono}}(s, g)$, via GCIQL using a standard MLP architecture devoid of explicit geometric constraints. For a fair comparison, all distance architectures are trained using the representation learning phase of Algorithm 1, where we substitute the distance formula (Line 5) with the corresponding metric parameterization while maintaining the same GCIQL objective. We then evaluate the discrepancy between the value estimates learned by each architecture and the learned reference $V_{\text{mono}}(s, g)$. Given the bijective mapping $d^*(s, g) = \log_\gamma V^*(s, g)$, this value approximation error serves as a comparison of agreement with the learned reference. Figure 2 summarizes the comparative results. The total error is calculated as the sum of squared errors over 1,024 sampled (s, g) pairs, averaged across four diverse tasks.

Asymmetric bilinear and asymmetric Euclidean models match the monolithic reference more closely than their symmetric counterparts. Q-Rep has the lowest discrepancy, approximately **22%** below the strongest asymmetric baseline. This measures agreement with a learned value reference, not error against a known optimal temporal distance.

5.4 Robustness to Goal Noise

We evaluate robustness to Gaussian perturbations of goal observations on `pointmaze-medium-navigate-v0`. This tests sensitivity to input noise; it does not directly test the same-latent-state condition in Theorem 3.3.

Figure 3 shows that returns decline for all methods as noise increases, but Q-Rep degrades more gradually than the raw-goal baseline. The experiment supports empirical robustness over the tested noise levels.

6 Conclusion

Q-Rep uses the asymmetric geometry of goal-reaching to structure goal representations. Exact quasimetric embeddings provide sufficiency and noise invariance under the stated theoretical setting. A learned weighted asymmetric ℓ^1 instantiation improves performance on most evaluated OGBench tasks and shows robustness to goal perturbations.

Impact Statement

This paper presents work whose goal is to advance the field of Machine Learning. There are many potential societal consequences of our work, none of which we feel must be specifically highlighted here.

AI Use Statement

An AI assistant helped edit the language and structure, format the manuscript, and check consistency between the text, equations, and reported results. It also assisted with interpreting reported comparisons and clarifying mathematical exposition. Experimental results were retained from the source manuscript.

References

Alexander A Alemi, Ian Fischer, Joshua V Dillon, and Kevin Murphy. Deep variational information bottleneck. *arXiv preprint arXiv:1612.00410*, 2016.

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. *Advances in neural information processing systems*, 30, 2017.

Pablo Samuel Castro. Scalable methods for computing state similarity in deterministic markov decision processes. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp. 10069–10076, 2020.

Stefan Cobzas. *Functional analysis in asymmetric normed spaces*. Springer Science & Business Media, 2012.

Ayoub Echchahed and Pablo Samuel Castro. A survey of state representation learning for deep reinforcement learning. *arXiv preprint arXiv:2506.17518*, 2025.

Yonathan Efroni, Dipendra Misra, Akshay Krishnamurthy, Alekh Agarwal, and John Langford. Provably filtering exogenous distractors using multistep inverse dynamics. In *International Conference on Learning Representations*, 2021.

Lasse Espeholt, Hubert Soyer, Remi Munos, Karen Simonyan, Vlad Mnih, Tom Ward, Yotam Doron, Vlad Firoiu, Tim Harley, Iain Dunning, et al. Impala: Scalable distributed deep-rl with importance weighted actor-learner architectures. In *International conference on machine learning*, pp. 1407–1416. PMLR, 2018.

Ben Eysenbach, Russ R Salakhutdinov, and Sergey Levine. Search on the replay buffer: Bridging planning and reinforcement learning. *Advances in neural information processing systems*, 32, 2019.

Benjamin Eysenbach, Tianjun Zhang, Sergey Levine, and Russ R Salakhutdinov. Contrastive learning as goal-conditioned reinforcement learning. *Advances in Neural Information Processing Systems*, 35:35603–35620, 2022.

Dibya Ghosh, Chethan Anand Bhateja, and Sergey Levine. Reinforcement learning from passive data via latent intentions. In *International Conference on Machine Learning*, pp. 11321–11339. PMLR, 2023.

Juha Heinonen. *Geometric embeddings of metric spaces*. Number 90. University of Jyväskylä, 2003.

D Hendrycks. Gaussian error linear units (gelus). *arXiv preprint arXiv:1606.08415*, 2016.

Leslie Pack Kaelbling. Learning to achieve goals. In *IJCAI*, volume 2, pp. 1094–8, 1993.

Diederik P Kingma. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Ilya Kostrikov, Ashvin Nair, and Sergey Levine. Offline reinforcement learning with implicit q-learning. *arXiv preprint arXiv:2110.06169*, 2021.

Sascha Lange, Thomas Gabel, and Martin Riedmiller. Batch reinforcement learning. In *Reinforcement learning: State-of-the-art*, pp. 45–73. Springer, 2012.

Michael Laskin, Aravind Srinivas, and Pieter Abbeel. Curl: Contrastive unsupervised representations for reinforcement learning. In *International conference on machine learning*, pp. 5639–5650. PMLR, 2020.

Daniel Lawson, Adriana Hugessen, Charlotte Cloutier, Glen Berseth, and Khimya Khetarpal. Self-predictive representations for combinatorial generalization in behavioral cloning. *arXiv preprint arXiv:2506.10137*, 2025.

Sergey Levine, Aviral Kumar, George Tucker, and Justin Fu. Offline reinforcement learning: Tutorial, review, and perspectives on open problems. *arXiv preprint arXiv:2005.01643*, 2020.

Bo Liu, Yihao Feng, Qiang Liu, and Peter Stone. Metric residual network for sample efficient goal-conditioned reinforcement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pp. 8799–8806, 2023.

Corey Lynch, Mohi Khansari, Ted Xiao, Vikash Kumar, Jonathan Tompson, Sergey Levine, and Pierre Sermanet. Learning latent plans from play. In *Conference on robot learning*, pp. 1113–1132. Pmlr, 2020.

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. *arXiv preprint arXiv:2210.00030*, 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. *arXiv preprint arXiv:2406.17098*, 2024.

Vivek Myers, Bill Chunyuan Zheng, Anca Dragan, Kuan Fang, and Sergey Levine. Temporal representation alignment: Successor features enable emergent compositionality in robot instruction following. *arXiv preprint arXiv:2502.05454*, 2025.

Seohong Park, Dibya Ghosh, Benjamin Eysenbach, and Sergey Levine. Hiql: Offline goal-conditioned rl with latent states as actions. *Advances in Neural Information Processing Systems*, 36:34866–34891, 2023.

Seohong Park, Kevin Frans, Benjamin Eysenbach, and Sergey Levine. Ogbench: Benchmarking offline goal-conditioned rl. *arXiv preprint arXiv:2410.20092*, 2024.

Seohong Park, Deepinder Mann, and Sergey Levine. Dual goal representations. *arXiv preprint arXiv:2510.06714*, 2025.

Xue Bin Peng, Aviral Kumar, Grace Zhang, and Sergey Levine. Advantage-weighted regression: Simple and scalable off-policy reinforcement learning. *arXiv preprint arXiv:1910.00177*, 2019.

Matthias Plappert, Marcin Andrychowicz, Alex Ray, Bob McGrew, Bowen Baker, Glenn Powell, Jonas Schneider, Josh Tobin, Maciek Chociej, Peter Welinder, et al. Multi-goal reinforcement learning: Challenging robotics environments and request for research. *arXiv preprint arXiv:1802.09464*, 2018.

Nikolay Savinov, Alexey Dosovitskiy, and Vladlen Koltun. Semi-parametric topological memory for navigation. *arXiv preprint arXiv:1803.00653*, 2018.

Tom Schaul, Daniel Horgan, Karol Gregor, and David Silver. Universal value function approximators. In *International conference on machine learning*, pp. 1312–1320. PMLR, 2015.

Max Schwarzer, Ankesh Anand, Rishab Goel, R Devon Hjelm, Aaron Courville, and Philip Bachman. Data-efficient reinforcement learning with self-predictive representations. *arXiv preprint arXiv:2007.05929*, 2020.

Pierre Sermanet, Corey Lynch, Yevgen Chebotar, Jasmine Hsu, Eric Jang, Stefan Schaal, Sergey Levine, and Google Brain. Time-contrastive networks: Self-supervised learning from video. In *2018 IEEE international conference on robotics and automation (ICRA)*, pp. 1134–1141. IEEE, 2018.

Dhruv Shah, Benjamin Eysenbach, Gregory Kahn, Nicholas Rhinehart, and Sergey Levine. Rapid exploration for open-world navigation with latent goal models. *arXiv preprint arXiv:2104.05859*, 2021.

Harshit Sikchi, Qinqing Zheng, Amy Zhang, and Scott Niekum. Dual rl: Unification and new methods for reinforcement and imitation learning. *arXiv preprint arXiv:2302.08560*, 2023.

Tongzhou Wang and Phillip Isola. On the learning and learnability of quasimetrics. *arXiv preprint arXiv:2206.15478*, 2022.

Tongzhou Wang, Antonio Torralba, Phillip Isola, and Amy Zhang. Optimal goal-reaching reinforcement learning via quasimetric learning. In *International Conference on Machine Learning*, pp. 36411–36430. PMLR, 2023.

Yifan Wu, George Tucker, and Ofir Nachum. The laplacian in rl: Learning representations with efficient approximations. *arXiv preprint arXiv:1810.04586*, 2018.

Amy Zhang, Rowan McAllister, Roberto Calandra, Yarin Gal, and Sergey Levine. Learning invariant representations for reinforcement learning without reconstruction. *arXiv preprint arXiv:2006.10742*, 2020.

Chongyi Zheng, Ruslan Salakhutdinov, and Benjamin Eysenbach. Contrastive difference predictive coding. *arXiv preprint arXiv:2310.20141*, 2023.

A PROOF OF THE THEOREMS

Proof of Theorem 3.1.

Proof. Recall that the optimal goal-conditioned value function is defined as $V^*(s,g) = \max_\pi \mathbb{E}\left[\gamma^{T^\pi(s,g)}\right]$, where $T^\pi(s,g)$ denotes the first hitting time from state s to goal g. We assume a standard sparse reward formulation where $r(s,g) = \mathbb{I}(s=g)$ and the process terminates upon reaching the goal. Consequently, $V^*(s,g) \in (0,1]$ for reachable goals (with $V^* = 0$ implies $d^* = \infty$ for unreachable goals). The discount factor satisfies $\gamma \in (0,1)$. To show that $d^*(s,g)$ is a quasimetric, we verify the three properties specified in Definition 1:

(1) Positivity: For any pair $s, g \in \mathcal{S}$, we have $0 \le V^*(s,g) \le 1$. Since the base of the logarithm is $\gamma \in (0,1)$, the function $\log_\gamma(\cdot)$ is monotonically decreasing. Therefore:

$$\begin{aligned} d^*(s,g) &= \log_\gamma V^*(s,g) \\ &\ge \log_\gamma(1) \\ &= 0. \end{aligned}$$

Thus, $d^*(s,g) \ge 0$ holds.

(2) Identity: We will show that $d^*(s,g) = 0 \iff s = g$.

($\Longleftarrow$). If $s = g$, the agent is already at the goal. The hitting time is $T = 0$, yielding $V^*(s,s) = 1$. Thus,

$$d^*(s,s) = \log_\gamma(1) = 0.$$

($\Longrightarrow$). Conversely, if $d^*(s,g) = 0$, then $\log_\gamma V^*(s,g) = 0$, which implies $V^*(s,g) = 1$. Given $\gamma < 1$, the expected discounted return can only equal 1 if the discount is applied with an exponent of 0, i.e., γ^0. This implies the minimum time to reach the goal is 0, which is only possible if $s = g$.

(3) Triangle Inequality: Fix $s, y, g \in \mathcal{S}$. Consider a policy that follows an optimal policy for reaching y and then an optimal policy for reaching g. Define the waypoint hitting time T_1 and the subsequent time T_2 from y to g on the underlying CMP, continued past an earlier visit to g solely to define these auxiliary times. The actual first hitting time T_g satisfies $T_g \le T_1 + T_2$. With $\gamma^\infty = 0$, the strong Markov property gives

$$\begin{aligned} V^{\pi_{s\to y\to g}}(s,g) = \mathbb{E}[\gamma^{T_g}] &\ge \mathbb{E}[\gamma^{T_1+T_2}] \\ &= \mathbb{E}[\gamma^{T_1}] V^*(y,g) \\ &= V^*(s,y) V^*(y,g). \end{aligned} \tag{5}$$

The inequality accounts for paths that reach g before y. Optimality therefore implies

$$V^*(s,g) \ge V^*(s,y) V^*(y,g).$$

If either factor on the right is zero, the desired distance inequality has an infinite right-hand side and is immediate. Otherwise both factors are positive. Since $\gamma \in (0,1)$, the logarithm $\log_\gamma$ is a strictly decreasing function, which reverses the inequality direction when applied to both sides:

$$\begin{aligned} \log_\gamma V^*(s,g) &\le \log_\gamma \left(V^*(s,y) \cdot V^*(y,g)\right) \\ &= \log_\gamma V^*(s,y) + \log_\gamma V^*(y,g). \end{aligned}$$

Substituting the definition of the optimal temporal distance d^* from Equation (1), we obtain:

$$d^*(s,g) \le d^*(s,y) + d^*(y,g).$$

This confirms the triangle inequality.

Overall, $d^*(s,g)$ satisfies positivity, identity, and the triangle inequality. Therefore, we conclude that d^* is a quasimetric. □

Proof of Theorem 3.2.

Proof. We define $\hat{\pi} : \mathcal{S} \times \widehat{\mathcal{S}} \to \mathcal{A}$ as

$$\hat{\pi}(s, f) := \arg\max_{a \in \mathcal{A}} \mathbb{E}_{s' \sim p(s'|s,a)} \left[\gamma^{f(s')} \right].$$

For $s \in \mathcal{S}$ and $f \in \widehat{\mathcal{S}}$, we then have, for any $s, g \in \mathcal{S}$,

$$\begin{aligned}
\tilde{\pi}(s, g) &= \hat{\pi}(s, f_g) \\
&= \arg\max_{a \in \mathcal{A}} \mathbb{E}_{s' \sim p(s'|s,a)} \left[\gamma^{f_g(s')} \right] \\
&= \arg\max_{a \in \mathcal{A}} \mathbb{E}_{s' \sim p(s'|s,a)} \left[\gamma^{F(\phi(g) - \phi(s'))} \right] \\
&= \arg\max_{a \in \mathcal{A}} \mathbb{E}_{s' \sim p(s'|s,a)} \left[\gamma^{d^*(s', g)} \right] \\
&= \arg\max_{a \in \mathcal{A}} \mathbb{E}_{s' \sim p(s'|s,a)} \left[V^*(s', g) \right] \\
&= \arg\max_{a \in \mathcal{A}} Q^*(s, a, g).
\end{aligned}$$

□

Proof of Theorem 3.3.

Proof. We need to show $f_{g_1}(s) = f_{g_2}(s)$ for all $s \in \mathcal{S}$. This follows from

$$\begin{aligned}
f_{g_1}(s) &= \log_\gamma V^*(s, g_1) \\
&= \log_\gamma \left(\max_\pi V^\pi(s, g_1) \right) \\
&= \log_\gamma \left(\max_\pi \mathbb{E}_{\tau \sim p^\pi(\tau | s_0 = s)} \left[\sum_{t=0}^{\infty} \gamma^t r^\ell(s_t, g_1) \right] \right) \\
&= \log_\gamma \left(\max_\pi \mathbb{E}_{\tau \sim p^\pi(\tau | s_0 = s)} \left[\sum_{t=0}^{\infty} \gamma^t \mathbb{I}(\psi(s_t) = \psi(g_1)) \right] \right) \\
&= \log_\gamma \left(\max_\pi \mathbb{E}_{\tau \sim p^\pi(\tau | s_0 = s)} \left[\sum_{t=0}^{\infty} \gamma^t \mathbb{I}\left(\psi(s_t) = \psi(g_2)\right) \right] \right) \\
&= \log_\gamma \left(\max_\pi \mathbb{E}_{\tau \sim p^\pi(\tau | s_0 = s)} \left[\sum_{t=0}^{\infty} \gamma^t r^\ell(s_t, g_2) \right] \right) \\
&= \log_\gamma \left(\max_\pi V^\pi(s, g_2) \right) \\
&= \log_\gamma V^*(s, g_2) \\
&= f_{g_2}(s),
\end{aligned}$$

where the max is taken over the set of all non-goal-conditioned policies $\pi : \mathcal{S} \to \Delta(\mathcal{A})$. □

B Datasets and Tasks

OGBench overview. We follow OGBench (Park et al., 2024) and evaluate goal-conditioned offline RL tasks using both state-based and pixel-based observations. Each environment provides five evaluation goals per task for multi-goal testing. We organize the benchmark into seven task families (instead of OGBench "domains"). This yields a cleaner and more reproducible description of the 12 state-based tasks and 5 pixel-based tasks reported in Table 1, with representative screenshots shown in Figure 4.

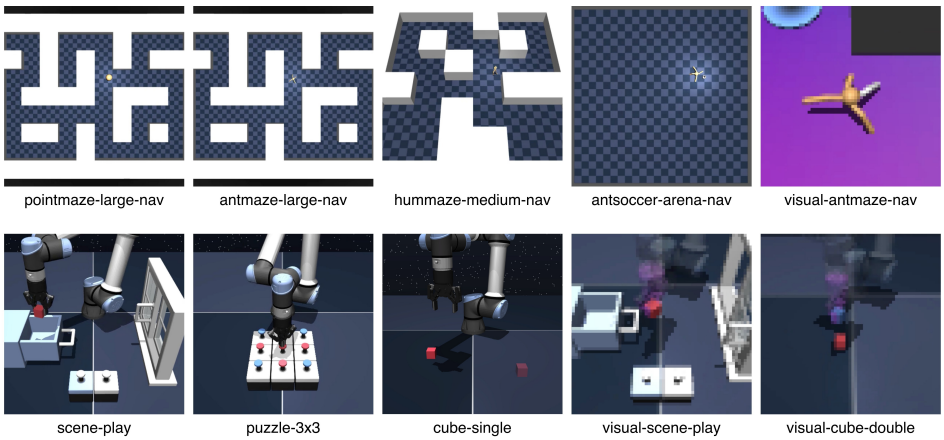


Figure 4: **OGBench tasks.** Representative screenshots for the seven task families used in our experiments, covering locomotion and navigation (PointMaze, AntMaze, HumanoidMaze, AntSoccer) as well as manipulation (Cube, Scene, Puzzle), in both state-based and pixel-based observation settings where applicable (Park et al., 2024).

PointMaze. PointMaze is a 2-D goal-conditioned navigation task where a point mass must reach a target location in a maze with offline data. We use the medium and large maze variants (e.g., `pointmaze-medium-navigate-v0` and `pointmaze-large-navigate-v0`), which differ primarily in path length and planning horizon. This family isolates long-horizon navigation and goal-reaching with simple dynamics, serving as a low-dimensional baseline for the broader maze suite.

AntMaze. AntMaze is a goal-conditioned maze navigation task with a quadrupedal Ant agent, coupling high-level planning with nontrivial locomotion control. We use the medium and large navigate variants (e.g., `antmaze-medium-navigate-v0` and `antmaze-large-navigate-v0`), where the agent must reach randomly sampled goals in sparse-reward mazes using offline trajectories. We additionally include the pixel-based counterparts (e.g., `visual-antmaze-medium-navigate-v0` and `visual-antmaze-large-navigate-v0`), which keep the same task definition but replace low-dimensional observations with $64 \times 64 \times 3$ RGB inputs from a third-person camera.

HumanoidMaze. HumanoidMaze shares the same goal-reaching maze-navigation objective but controls a substantially higher-dimensional Humanoid agent, making policy learning and long-horizon reasoning harder under offline data. We use the medium and large navigate variants (e.g., `humanoidmaze-medium-navigate-v0` and `humanoidmaze-large-navigate-v0`), which stress robustness to distribution shift in both planning and high-DoF control. Compared to AntMaze, this family increases control complexity while preserving the core maze-navigation structure.

AntSoccer. AntSoccer extends locomotion beyond pure navigation: an Ant agent must dribble a soccer ball while moving toward a target location, requiring coordinated control of both the agent and the ball. We use the arena navigate variant (`antsoccer-arena-navigate-v0`), which tests goal reaching with additional interaction dynamics and longer effective horizons. This family emphasizes compositional control skills (locomotion + object interaction) within a goal-conditioned offline setting.

Cube. Cube is a goal-conditioned manipulation task with a 6-DoF UR5e robot arm, where the objective is to rearrange cube blocks to match a target configuration. We use the single- and double-cube play datasets (e.g., `cube-single-play-v0` and `cube-double-play-v0`), which are collected via scripted "play" interactions that randomly pick and place blocks. Evaluation goals require operations such as moving, stacking, or swapping blocks, and difficulty increases with the number

of objects. We also report pixel-based variants (e.g., `visual-cube-single-play-v0` and `visual-cube-double-play-v0`), which preserve task semantics but use RGB observations.

Scene. Scene targets long-horizon, sequential manipulation with multiple interactive objects and preconditions (e.g., locks, drawers, windows, and buttons), where earlier actions can enable or disable subsequent interactions. We use `scene-play-v0`, collected via scripted play that produces diverse multi-step interaction sequences, and its pixel-based counterpart `visual-scene-play-v0`. Compared to Cube, Scene emphasizes temporal ordering and multi-step dependency structure rather than only multi-object geometric rearrangement.

Puzzle. Puzzle instantiates a continuous-control version of the "Lights Out" combinatorial puzzle with a robot arm: pressing a button toggles its own state and neighboring buttons, and the goal is to reach a desired binary color pattern. We use the 3×3 and 4×4 play variants (e.g., `puzzle-3x3-play-v0` and `puzzle-4x4-play-v0`), collected by scripted button-pressing policies that generate diverse configurations. This family stresses combinatorial generalization and long-horizon planning due to the large discrete state space induced by local-toggle dynamics.

C Downstream Offline RL Algorithm

In this work, we employ **Goal-Conditioned Implicit V-Learning (GCIVL)**, a lightweight variant of Implicit Q-Learning (IQL) (Kostrikov et al., 2021), as our downstream offline reinforcement learning algorithm. Unlike standard IQL which maintains both Q and V functions, GCIVL simplifies the architecture by learning only a goal-conditioned value function $V_\psi(s, g)$ and extracting the policy purely from value estimates. This reduction improves training stability and reduces memory overhead, which is particularly beneficial when using large visual encoders.

C.1 Algorithm Overview

The GCIVL agent consists of a value network V_ψ and a policy network π_θ. The training process decouples value learning from policy extraction:

1. **Value Learning via Expectile Regression:** The value function approximates the optimal value of the behavior policy's state distribution. We use expectile regression to estimate the upper envelope of the return distribution, effectively performing "implicit" maximization without querying out-of-distribution actions.

2. **Policy Extraction via Advantage Weighted Regression:** The policy is trained to clone actions from the dataset, weighted by their estimated advantage. In the absence of a Q-network, we approximate the Q-value using the immediate reward and the next state's value: $Q(s, a, g) \approx r + \gamma V(s', g)$.

The complete training procedure is summarized in Algorithm 2.

Algorithm 2 Goal-Conditioned Implicit V-Learning (GCIVL)

Input: Offline dataset $\mathcal{D} = \{(s, a, r, s', g)\}$, Encoder ϕ, Discount γ, Expectile τ, Temperature α.
Initialize: Value network V_ψ, Policy network π_θ, Target value network $V_{\bar{\psi}} \leftarrow V_\psi$.
while training **do**
  Sample batch $B \sim \mathcal{D}$.
  # — Value Update (Expectile Regression) —
  Compute target Q-estimate: $\hat{Q} = r + \gamma \cdot V_{\bar{\psi}}(s', g)$
  Compute advantage for weighting: $A_{est} = \hat{Q} - V_\psi(s, g)$
  Update V_ψ by minimizing expectile loss:

$$\mathcal{L}_V(\psi) = \mathbb{E}_{(s,g)\sim B}\left[L_2^\tau\left(\hat{Q} - V_\psi(s, g)\right)\right]$$

  # — Policy Update (Advantage Weighted Regression) —
  Compute advantage weights: $w = \exp(\alpha \cdot (V_\psi(s', g) - V_\psi(s, g)))$
  Update π_θ by maximizing weighted log-likelihood:

$$\mathcal{L}_\pi(\theta) = \mathbb{E}_{(s,a,g)\sim B}\left[-w \cdot \log \pi_\theta(a|s, g)\right]$$

  # — Target Update —
  Update target network: $\bar{\psi} \leftarrow \rho\bar{\psi} + (1 - \rho)\psi$
end while

C.2 Implementation Details

Value Loss. We employ expectile regression to minimize the difference between the target return and the current value estimate. The expectile loss L_2^τ is defined as:

$$L_2^\tau(u) = |\tau - \mathbb{I}(u < 0)|u^2, \tag{6}$$

where $\mathbb{I}$ is the indicator function. Following standard practice to mitigate overestimation, we use a clipped double-value trick for the target calculation: $V_{\text{target}}(s', g) = \min(V_{\bar{\psi}_1}(s', g), V_{\bar{\psi}_2}(s', g))$.

Policy Loss. The policy is updated using Advantage Weighted Regression (AWR) (Peng et al., 2019). The weights are determined by the advantage estimate. In our V-only implementation, we estimate the advantage of a transition (s, a, s') as the temporal difference in values:

$$A(s, a, g) \approx V(s', g) - V(s, g). \tag{7}$$

We apply an exponential transformation to these advantages to compute the weights $w = \min(\exp(\alpha \cdot A), \text{MAX_W})$, where α is the inverse temperature hyperparameter controlling the distinctness of the high-value actions.

D Experiment Details

D.1 Baselines

The baseline results reported in Table 1 are sourced from Park et al. (2025). We refer readers to the original paper for comprehensive details on the experimental setup and implementation. In this section, we briefly review the goal representation learning strategies employed by these methods.

VIB (Shah et al., 2021). VIB trains a goal representation based on the information bottleneck principle (Alemi et al., 2016). A VIB representation φ is trained using the value loss in the downstream GCRL algorithm. Specifically, the downstream value function is modeled as $V^{\text{GCRL}}(s, \varphi(g))$, where $\varphi(g)$ denotes a stochastic latent vector sampled from $\mathcal{N}(\mu(g), \Sigma(g))$, with μ acting as a mean network and Σ as a diagonal covariance network. This stochastic latent-parameterized value network is optimized using the downstream GCRL loss augmented with a KL divergence term, $\beta D_{\text{KL}}(\mathcal{N}(\mu(g), \Sigma(g)) \,\|\, \mathcal{N}(0, I))$, weighted with a coefficient β.

Table 3: **Hyperparameters for OGBench experiments.**

Hyperparameter	Value
Gradient steps	2×10^6 (state-based), 5×10^5 (pixel-based)
Optimizer	Adam (Kingma, 2014)
Learning rate	3e-4
Batch size	1024 (state-based), 256 (pixel-based)
MLP size	[512, 512, 512]
Nonlinearity	GELU (Hendrycks, 2016)
Target network update rate	0.005
Discount factor γ	0.99 (default), 0.995 (humanoidmaze)
Goal representation dimensionality N	256 (default), 64 (pointmaze)
GCIQL expectile κ for Q-Rep representation	0.7 (state-based), 0.85 (pixel-based)
Representation $(p^D_{\text{cur}}, p^D_{\text{geom}}, p^D_{\text{traj}}, p^D_{\text{rand}})$ ratio	(0.2, 0.5, 0, 0.3)
Downstream GCIVL expectile	0.9
Downstream policy extraction hyperparameters	Identical to OGBench (Park et al., 2024)
Downstream policy $(p^D_{\text{cur}}, p^D_{\text{geom}}, p^D_{\text{traj}}, p^D_{\text{rand}})$ ratio	(0, 0, 1, 0)
Downstream value $(p^D_{\text{cur}}, p^D_{\text{geom}}, p^D_{\text{traj}}, p^D_{\text{rand}})$ ratio	(0.2, 0.5, 0, 0.3)
Visual encoder	Impala_small (Espeholt et al., 2018)
Image augmentation probability	0.5

VIP (Ma et al., 2022). VIP acquires a representation via a metric embedding loss. Specifically, the value function is parameterized as $V(s, g) = -\|\varphi(s) - \varphi(g)\|_2$, and φ is trained using a dual RL loss function (Sikchi et al., 2023). In the downstream policy and value networks, φ represents only the goals.

TRA (Myers et al., 2025). TRA learns a goal representation via a symmetric InfoNCE variant of the CRL loss (Eysenbach et al., 2022). This representation is subsequently utilized to parameterize the downstream GCRL policy and value functions.

BYOL-γ (Lawson et al., 2025). BYOL-γ trains a representation based on a temporal self-predictive representation loss (Schwarzer et al., 2020). The bidirectional variant is adopted here, using the representation φ to parameterize goals in the downstream policy and value networks.

Dual (Park et al., 2025). Dual learns a representation using an asymmetric bilinear function. Specifically, the optimal temporal distance is parameterized as $d^*(s, g) = \psi(s)^\top \varphi(g)$, where the components ψ and φ are trained via GCIQL. The resulting representation φ is utilized to parameterize goals in the downstream policy and value networks.

D.2 Hyperparameters

Table 3 summarizes the hyperparameters used in our OGBench experiments. Unless otherwise specified, we use a single set of default settings across task families, with only a few task- or modality-specific adjustments (e.g., fewer gradient steps and smaller batch size for pixel-based training, γ=0.995 for humanoidmaze, and a reduced goal-representation dimensionality for pointmaze). For downstream policy extraction, we follow the OGBench protocol without modification.

D.3 Metrics

We follow the OGBench evaluation protocol (Park et al., 2024) and report the results in Table 1 and Table 2 as an average over the final three evaluation checkpoints to reduce variance. For state-based tasks, we use the checkpoints at 1.8M, 1.9M, and 2.0M gradient steps and report the mean performance across them. For pixel-based tasks, we follow the standard OGBench checkpoints and average the results at 300K, 400K, and 500K gradient steps.

At each checkpoint, we evaluate on five predefined goals. For state-based tasks, we run 50 episodes per goal, while for pixel-based tasks we run 15 episodes per goal, consistent with OGBench. Overall, this reporting protocol provides a stable summary of late-training performance while keeping the evaluation budget comparable across methods.

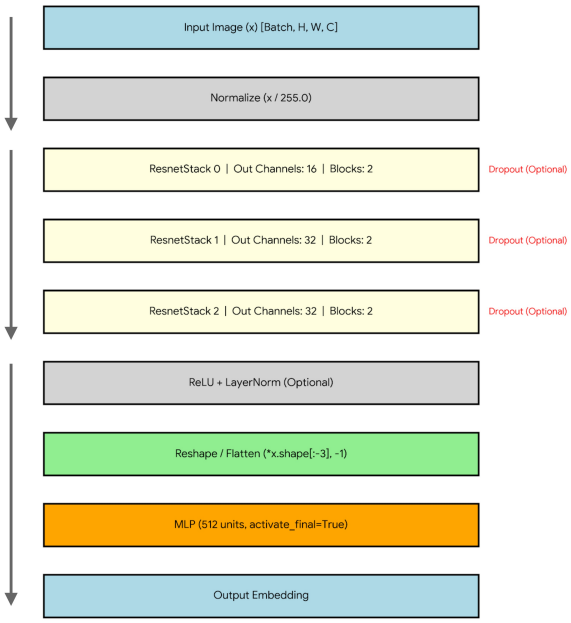


Figure 5: **Architecture of the IMPALA Visual Encoder.** The network processes raw pixel inputs through a sequence of three ResNet stacks with increasing channel depths (16, 32, 32), followed by a dense projection layer to produce the latent embedding z_t. We employ dual encoders of this architecture to separately process inputs for the policy and value functions.

D.4 Visual Encoder

For high-dimensional pixel-based tasks, we employ the convolutional ResNet architecture introduced in IMPALA (Espeholt et al., 2018) to extract compact spatial features. As illustrated in Figure 5, the encoder transforms raw RGB observations $o_t \in \mathbb{R}^{H \times W \times C}$ into a latent embedding vector $z_t \in \mathbb{R}^d$. Formally, the encoding process can be denoted as:

$$z_t = f_\theta(o_t), \tag{8}$$

where f_θ represents the parameterized encoder network. The architecture consists of a sequence of residual stacks, where each stack comprises a convolutional layer followed by multiple residual blocks. This design facilitates effective gradient flow through deep structures, ensuring robust feature learning essential for reinforcement learning stability. Following the residual stacks, a spatial flattening operation and a multi-layer perceptron (MLP) project the features into the final d-dimensional representation (typically $d = 512$).

In our implementation, we instantiate separate encoders with identical architectures for the policy and value networks. Decoupling the visual backbones prevents the high-variance gradients from the value estimation (common in offline RL) from distorting the policy's feature representation. Both encoders are trained end-to-end alongside their respective downstream heads without any pre-training, using standard pixel normalization techniques as a preprocessing step.

E Related Work

Offline Goal-Conditioned Reinforcement Learning. Offline GCRL lies at the confluence of offline RL (Lange et al., 2012; Levine et al., 2020) and goal-conditioned RL (Kaelbling, 1993), and seeks to learn goal-reaching policies solely from fixed, reward-free datasets. By framing goal-reaching as a self-supervised learning problem, offline GCRL enables the pretraining of policies, representations, and value functions that can subsequently be transferred or adapted to downstream tasks (Ma et al., 2022; Ghosh et al., 2023).

Existing offline GCRL methods span a broad range of techniques, including implicit value-based learning (Park et al., 2023), contrastive objectives (Eysenbach et al., 2022; Zheng et al., 2023), metric-based formulations (Wang et al., 2023; Myers et al., 2024; Liu et al., 2023), and planning-oriented approaches (Savinov et al., 2018; Eysenbach et al., 2019). In this work, we focus on enhancing both the effectiveness and generalization of goal-reaching agents through the learning of structured goal representations. Our approach is complementary to existing offline GCRL algorithms and can be seamlessly integrated with them.

Representation Learning. Representation learning has been a longstanding topic in reinforcement learning (Echchahed & Castro, 2025). Prior studies seek to improve learning efficiency and generalization by constructing informative representations of states, actions, or other elements of the environment. A variety of representation learning approaches have been proposed, including scalable state similarity measures (Castro, 2020), spectral methods based on Laplacian operators (Wu et al., 2018), self-supervised representation learning (Schwarzer et al., 2020), and contrastive learning techniques (Laskin et al., 2020).

In the context of GCRL, learned representations have been widely adopted for various purposes, such as constructing state representations (Ma et al., 2022), shaping auxiliary rewards (Sermanet et al., 2018), or supporting planning procedures (Sermanet et al., 2018). However, goal representation in GCRL remains largely underexplored. Prior work that explicitly studies goal representations in GCRL includes Dual Goal Representation (Park et al., 2025) and Temporal Representation Alignment (TRA) (Myers et al., 2025). Dual Goal Representation (Park et al., 2025) parameterizes goal-conditioned functions using generic vector representations combined with a bilinear aggregation, while TRA learns goal representations by enforcing behavioral temporal alignment across trajectories via contrastive objectives. In contrast, our approach is motivated by the intrinsic geometry of goal-reaching problems and designs goal representations that explicitly encode the asymmetric quasimetric structure induced by the underlying control dynamics.