GEOMETRY-INFORMED PLANNING

Anonymous authors
Paper under double-blind review

ABSTRACT

We introduce **G**eometry-**I**nformed **P**lanning (**GIP**), a novel planning framework for offline goal-conditioned reinforcement learning (GCRL). GIP views the state space as a smooth manifold, where temporal distance is defined by optimal goal-reaching time. Optimal trajectories then correspond to geodesics, which can be represented by initial tangent vectors through the exponential map. Guided by this geometric structure, GIP represents an entire path with a single latent path vector and learns to sample such vectors for arbitrary state–goal pairs to infer geodesics. Our geometric formulation enables theoretical recovery of optimal goal-reaching paths under suitable assumptions, while joint learning of temporal distances with generated paths discourages invalid shortcuts that leave the state manifold. GIP enables efficient planning with a fixed-dimensional latent vector independent of the planning horizon, while training without temporal difference learning supports its scalability. We empirically validate these properties on OGBench, where GIP outperforms model-free and model-based baselines, particularly on long-horizon tasks, while requiring substantially lower planning cost.

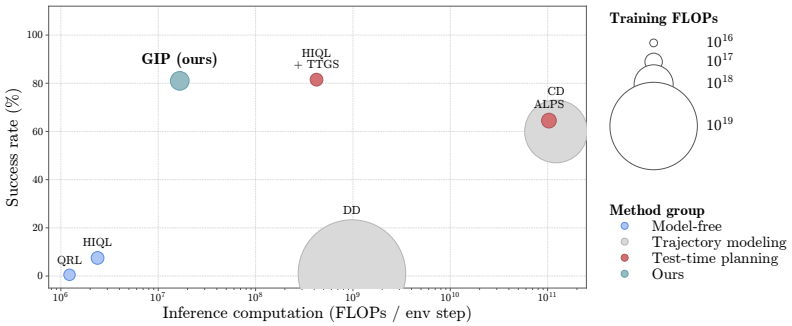


Figure 1: **GIP is efficient yet effective for goal-reaching planning and control.** On OGBench `humanoidmaze-giant` variants, we compare success rate, per-step inference FLOPs, and total training FLOPs, with inference FLOPs shown on a logarithmic scale. GIP achieves nearly $10.8\times$ higher success than the closest-compute baseline, HIQL, and $25.4\times$ lower inference cost than HIQL+TTGS at comparable performance.

1 INTRODUCTION

A growing body of work addresses large-scale and complex sequential decision-making problems that require coherent behavior over increasingly long horizons (Cheng et al., 2025; Black et al., 2025; Kim et al., 2026b). Such problems are particularly challenging in offline goal-conditioned reinforcement learning (GCRL), where an agent must reach arbitrary goals from limited, reward-free, and potentially suboptimal data. In the absence of dense reward supervision, recent approaches increasingly adopt explicit planning for long-horizon reasoning toward distant goals, either through test-time search over learned dynamics (Bagatella & Martius, 2023; Zhou et al., 2025; Chun et al., 2026; Yoon et al., 2025a;b) or direct trajectory optimization (Janner et al., 2021; Luo et al., 2025; Mishra et al., 2026; Sun et al., 2026).

Despite their effectiveness, existing planning approaches often incur substantial computational cost and lack theoretical guarantees of plan optimality. Prior methods commonly employ guidance-based generation (Janner et al., 2022; Ajay et al., 2023), search over learned dynamics susceptible to

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.

compounding model errors (Sobal et al., 2025; Zhou et al., 2025; Gao et al., 2025; Chung et al., 2026), or planning over heuristic structures (Koul et al., 2024; Baek et al., 2025; Shehmar et al., 2026), leaving optimal recovery from suboptimal data with limited theoretical support. Moreover, computational cost can grow substantially with problem size during training and inference (Luo et al., 2025; Mishra et al., 2026), while test-time planning requires considerable computation (Yoon et al., 2025b; Zhou et al., 2025; Opryshko et al., 2026).

Our core contribution is **G**eometry-**I**nformed **P**lanning **(GIP)**, a framework for efficient and optimal planning in GCRL. Under the assumption that the state space lies on a smooth manifold with suitable geometric structure, optimal goal-reaching trajectories correspond to the shortest smooth curves, or temporal geodesics, each represented by a single tangent vector through the exponential map. Motivated by this structure, GIP learns to encode an optimal path into a single latent path vector by directly exploiting the defining properties of temporal geodesics. To learn this representation reliably from offline data, we further introduce a joint temporal distance learning objective with out-of-support penalization, which discourages distance underestimation in unsupported regions and prevents generating invalid shortcuts.

These design choices give GIP several desirable properties. Under the stated assumptions, GIP recovers optimal goal-reaching paths. At inference, long-horizon reasoning requires only a fixed-dimensional path vector whose dimension is independent of the planning horizon, rather than explicit construction of raw-state sequences, substantially reducing inference cost while retaining the benefits of planning. Moreover, our training objective supports scalability by avoiding temporal difference (TD) learning and enables goal stitching by learning distances and paths between arbitrary state pairs from the dataset. We evaluate GIP on 13 OGBench tasks (Park et al., 2025a), where it achieves the best average success rate among competitive baselines. On the two longest-horizon among the benchmark, `humanoidmaze-giant` variants, GIP matches the strongest baseline with $25.4\times$ lower inference computation and achieves $10.8\times$ higher performance than a baseline with comparable computational cost.

2 Related Work

Offline Goal-Conditioned Reinforcement Learning. Offline goal-conditioned reinforcement learning aims to learn a generalist goal-reaching agent solely from a fixed, reward-free, and potentially suboptimal dataset (Park et al., 2025a). Prior approaches realize general goal-reaching through advantage-based regression (Ma et al., 2022), transitive regression (Park et al., 2026b), hierarchical policies with subgoals (Park et al., 2023; Hong et al., 2023), contrastive learning (Eysenbach et al., 2022; Myers et al., 2025; Zheng et al., 2026), quasimetric learning and representations (Wang et al., 2023; Myers et al., 2024; 2025; Giammarino & Qureshi, 2026) and compositional generalization (Lawson et al., 2026; Kim et al., 2026a). Our work builds on this line of research by jointly learning temporal distances and shortest goal-reaching paths.

Geometry. A related line of work exploits geometry for planning and control in nonlinear spaces. Prior methods learn Riemannian metrics for shortest-path planning and geodesic motion skills (Arvanitidis et al., 2019; Beik-Mohammadi et al., 2021), or impose geometric structures such as Riemannian motion policies (Mukadam et al., 2020; Xie et al., 2023; Merva et al., 2025) and hyperbolic state spaces (Cetin et al., 2023). Generative approaches similarly build on prescribed manifold geometries, including Riemannian diffusion (De Bortoli et al., 2022), manifold optimal-transport flows (Huguet et al., 2022), and flow matching on general or learned metric geometries (Chen & Lipman, 2024; Kapuśniak et al., 2024). In contrast, we assume only a general manifold with suitable geometric structure and exploit the resulting geometric inductive bias for efficient goal-directed planning.

3 Preliminaries

3.1 Offline Goal-Conditioned Reinforcement Learning and Quasimetrics

We consider a deterministic continuous-time controlled Markov process (CMP) $\mathcal{M} = (\mathcal{S}, \mathcal{A}, f)$, with state space $\mathcal{S} \subseteq \mathbb{R}^{n_S}$, action space $\mathcal{A}$, and deterministic dynamics $\dot{s}(t) = f(s(t), a(t))$ that preserves $\mathcal{S}$, where $f : \mathcal{S} \times \mathcal{A} \to \mathbb{R}^{n_S}$. Given a goal $g \in \mathcal{S}$, goal-conditioned reinforcement learning (GCRL)

aims to learn a generalist policy $\pi(a \mid s, g)$ to reach the goal as soon as possible. Accordingly, we define the *temporal distance* as

$$d^*(s, g) \coloneqq \inf_{\pi \in \Pi} \mathbb{E}^\pi \left[\inf_t \{t \geq 0 : s(t) = g\} \,\middle|\, s(0) = s \right], \tag{1}$$

where Π denotes the set of feasible goal-conditioned policies under dynamics f. In other words, the temporal distance is the infimum of the expected goal-reaching time. We study the offline GCRL setting, where the policy must be learned from a fixed, reward-free, and potentially suboptimal trajectory dataset $\mathcal{D} = \{\tau_i\}_{i=1}^N$. We use $p_\mathcal{D}$ to denote the empirical state distribution induced by $\mathcal{D}$.

A *quasimetric* on a space $\mathcal{X}$ is a function $d : \mathcal{X} \times \mathcal{X} \to \mathbb{R}_{\geq 0}$ satisfying

Identity: $d(x_1, x_2) = 0 \iff x_1 = x_2$,

Triangle inequality: $d(x_1, x_3) \leq d(x_1, x_2) + d(x_2, x_3)$.

for all $x_1, x_2, x_3 \in \mathcal{X}$, without requiring symmetry. Wang et al. (2023) established an exact correspondence between temporal distances and quasimetrics over the state space, and further proposed Quasimetric Reinforcement Learning (QRL), which learns the temporal distance using a parametric quasimetric architecture d_ϕ (*e.g.*, MRN (Liu et al., 2023), IQE (Wang & Isola, 2022)). QRL is built on two core principles: (1) penalizing violations of local transition cost upper bounds through local constraints and (2) recovering the global structure by maximally spreading states apart:

$$\max_\phi \quad \mathbb{E}_{s,g \sim p_\mathcal{D}}[d_\phi(s, g)] \quad \text{s.t.} \quad \mathbb{E}_{(s,s') \sim p_\mathcal{D}}[\max(d_\phi(s, s') - 1, 0)^2] \leq \epsilon^2. \tag{2}$$

Following QRL and its variants (Wang et al., 2023; Steccanella et al., 2026), our objective combines global distance spreading with local transition constraints to learn the quasimetric structure, while additionally preventing distance underestimation in out-of-support regions.

3.2 Geodesic Quasimetric Spaces

For a space $\mathcal{X}$ equipped with a quasimetric d, the length of a path $c : [0, 1] \to \mathcal{X}$ is defined as

$$L_d(c) \coloneqq \sup_{0 = t_0 < t_1 < \cdots < t_K = 1} \sum_{k=0}^{K-1} d(c(t_k), c(t_{k+1})),$$

where the supremum is taken over all finite partitions of $[0, 1]$. Let $C(x, y) = \{c : [0, 1] \to \mathcal{X} : c \text{ is continuous}, c(0) = x, c(1) = y\}$ denote the set of continuous curves connecting $x \in \mathcal{X}$ to $y \in \mathcal{X}$. By definition, for any $c \in C(x, y)$, its length $L_d(c)$ is at least the distance $d(x, y)$ between its endpoints.

A quasimetric d is called a *geodesic quasimetric* if, for every $x, y \in \mathcal{X}$, there exists a curve $c^* \in C(x, y)$, called a *geodesic*, such that

$$d(c^*(t_1), c^*(t_2)) = (t_2 - t_1) d(x, y), \tag{3}$$

for all $t_1, t_2 \in [0, 1]$ with $t_1 \leq t_2$. Accordingly, a geodesic $c^* \in C(x, y)$ satisfies $L_d(c^*) = \inf_{c \in C(x,y)} L_d(c) = d(x, y)$, so no path from x to y has smaller length than c^*.

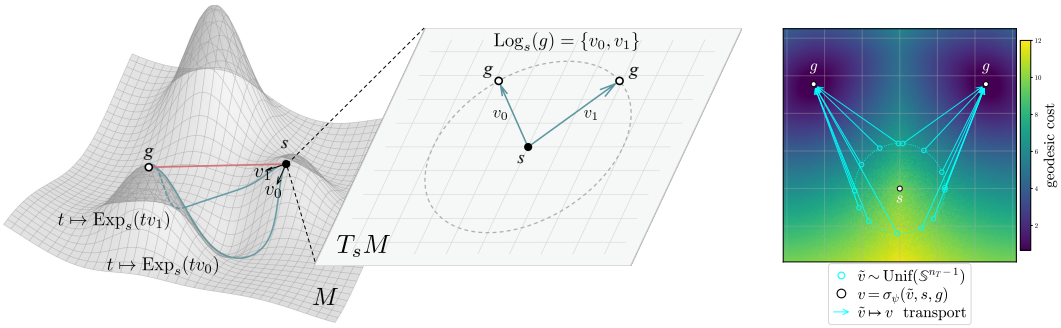
3.3 Manifolds and Exponential Maps

A smooth manifold M is a space that is locally diffeomorphic to a Euclidean space, while its global geometry may be non-Euclidean. At each $x \in M$, the tangent space $T_x M$ is the vector space of tangent vectors represented by the velocities of smooth curves through x. Accordingly, for a smooth curve $\gamma : [0, 1] \to M$, its velocity $\dot{\gamma}(t)$ lies in $T_{\gamma(t)} M$.

We consider a smooth manifold M equipped with a geodesic quasimetric d, admitting exponential maps $\mathrm{Exp}_x : T_x M \to M$ for each $x \in M$, compatible with the geodesics. Specifically, every geodesic γ^* is smooth and satisfies

$$\mathrm{Exp}_{\gamma^*(0)}(t \dot{\gamma}^*(0)) = \gamma^*(t), \qquad t \in [0, 1]. \tag{4}$$

In this paper, we leverage this property to model and optimize shortest paths for offline GCRL. We refer readers to Appendix D for formal definitions and to Kobayashi & Nomizu (1996) for a comprehensive treatment.



(a) State manifold and temporal geodesics. (b) Latent path vector selector.

Figure 2: (a) Multiple temporal geodesics from s to g are parameterized by distinct tangent vectors $v_i \in T_sM$ and recovered as $t \mapsto \mathrm{Exp}_s(tv_i)$. The learned distance assigns a larger length to the off-manifold shortcut (red) than to the temporal geodesics (teal). (b) The conditional selector σ_ψ transports reference vectors $\tilde{v}$ sampled uniformly from a unit sphere toward the possibly non-unique zero-geodesic cost set.

4 Geometry-Informed Planning

In this section, we introduce **G**eometry-**I**nformed **P**lanning (**GIP**), a novel approach to learning optimal goal-reaching paths and a policy grounded in the geometric properties of temporal geodesics.

4.1 Temporal Geodesics

Interpreting offline GCRL from the perspective of continuous-time path optimization, we first introduce a structural assumption that provides the geometric foundation of GIP.

Assumption 4.1. We assume that the state space $\mathcal{S}$ can be identified with a smooth manifold M, called the *state manifold*, equipped with the temporal distance d^* and an exponential map Exp. We assume that the temporal distance d^* is a continuous geodesic quasimetric and every geodesic $\gamma^*(t)$, which we call a *temporal geodesic*, is smooth and satisfies $\mathrm{Exp}_{\gamma^*(0)}(t\dot{\gamma}^*(0)) = \gamma^*(t)$ for $t \in [0, 1]$.

Assumption 4.1 implies that an optimal goal-reaching path with respect to the temporal distance d^* can be represented as a smooth temporal geodesic through the exponential map. In fact, these assumptions encompass many standard geometric settings, including Euclidean spaces and connected forward-complete Finsler manifolds. Note that we assumed a general manifold with suitable geometric structure rather than imposing specific metric structures on M (*e.g.*, a Riemannian metric) like prior approaches (Cetin et al., 2023; Park et al., 2024; Chen & Lipman, 2024).

Our desideratum is to recover temporal geodesics for arbitrary state–goal pairs and to learn a goal-conditioned policy that follows the recovered paths using only the offline dataset $\mathcal{D}$. We now establish a proposition that underpins our approach to modeling and learning temporal geodesics.

Proposition 4.2. *Under Assumption 4.1, for any $s, g \in M$, there exists $v^* \in T_sM$ satisfying the following conditions:*

Initial boundary: $\mathrm{Exp}_s(0) = s$,

Endpoint boundary: $\mathrm{Exp}_s(v^*) = g$,

Constant speed: $d^*(\mathrm{Exp}_s(t_i v^*), \mathrm{Exp}_s(t_j v^*)) = (t_j - t_i) d^*(s, g), \quad \forall t_i, t_j \in [0, 1],\ t_i \le t_j$.

Proposition 4.2 implies that the existence of an appropriate tangent vector $v^* \in T_sM$ ensures that its exponential image $t \mapsto \mathrm{Exp}_s(tv^*)$ forms a temporal geodesic from s to g. Note that such v^* may not be unique for each (s, g). Therefore, we define a set-valued function

$$\mathrm{Log}_s : g \mapsto \{v \in T_sM : v \text{ satisfies the three conditions in Proposition 4.2}\}. \tag{5}$$

For a selection σ_s of Log_s, one of the shortest paths from s to g can be represented as $t \mapsto \mathrm{Exp}_s(t\sigma_s(g))$ (see Figure 2 (a)). In the following section, we describe how temporal geodesics can be learned using parameterized function approximators inspired by the Exp_s and Log_s maps introduced above.

4.2 JOINTLY LEARNING DISTANCE AND GEODESICS

The conditions in Proposition 4.2 depend on the temporal distance d^*. We first develop a method for learning temporal geodesics assuming that d^* is given, and then extend the framework to jointly learn the temporal distance.

Learning Temporal Geodesics. Inspired by Exp_s and the selection σ_s introduced in Section 4.1, we model paths with a path map E_ξ and a latent path vector selector σ_ψ. Assuming that the temporal distance d^* is given, we jointly train them by imposing the conditions in Proposition 4.2 on the induced paths to recover temporal geodesics. In practice, since neither the tangent spaces nor the intrinsic dimension of M are known, we represent the latent path vectors in $\mathbb{R}^{n_T}$ that correspond to the tangent vectors, with $n_T < n_S$ treated as a hyperparameter.

Motivated by the exponential map Exp_s, we first introduce a path map $E_\xi : \mathbb{R}^{n_S} \times \mathbb{R}^{n_T} \to \mathbb{R}^{n_S}$, where a latent path vector v induces the path $t \mapsto E_\xi(s, tv)$, and parameterize it as

$$E_\xi(s, tv) = s + r_\xi(s, tv) - r_\xi(s, 0), \tag{6}$$

where $r_\xi : \mathbb{R}^{n_S} \times \mathbb{R}^{n_T} \to \mathbb{R}^{n_S}$ is a learnable function approximator. Although E_ξ is defined in the ambient space $\mathbb{R}^{n_S}$, in this paragraph, we consider an idealized setting in which all induced paths remain on M, the domain on which d^* is defined.

As discussed in Section 4.1, a path $t \mapsto E_\xi(s, tv)$ represents a temporal geodesic from s to g only if it satisfies the three conditions in Proposition 4.2. Since the parameterization in (6) guarantees $E_\xi(s, 0) = s$ and thus satisfies the initial boundary condition for any v, we encode the remaining endpoint boundary and constant-speed conditions in the following temporal geodesic cost:

$$\mathcal{C}_\xi(v; s, g) := \underbrace{\rho(d^*(x_1, g), 0)}_{\text{Endpoint boundary}} + \underbrace{\mathbb{E}_{(t_i,t_j)\sim\mathrm{Unif}(\mathcal{T})}\left[\rho\big(d^*\big(x_{t_i}, x_{t_j}\big), (t_j - t_i)d^*(s, g)\big)\right]}_{\text{Constant speed}}, \tag{7}$$

where $x_t = E_\xi(s, tv)$, $\mathcal{T} = \{(t_i, t_j) \in [0, 1]^2 : t_i < t_j\}$, $\mathrm{Unif}(\mathcal{T})$ denotes the uniform distribution over $\mathcal{T}$, and $\rho : \mathbb{R} \times \mathbb{R} \to \mathbb{R}_{\geq 0}$ is a regression loss function satisfying $\rho(x, y) = 0$ if and only if $x = y$. In analogy with Log_s, we define the set-valued map L_ξ as the zero level set of the temporal geodesic cost:

$$L_\xi(s, g) := \{v \in \mathbb{R}^{n_T} : \mathcal{C}_\xi(v; s, g) = 0\}. \tag{8}$$

To approximate a selection of Log_s, we design a parameterized latent path vector selector σ_ψ. Specifically, we condition the parametric selector $\sigma_\psi : \mathbb{S}^{n_T-1} \times \mathbb{R}^{n_S} \times \mathbb{R}^{n_S} \to \mathbb{R}^{n_T}$ on a random reference vector $\tilde{v} \sim q_0(\cdot|s, g) = \mathrm{Unif}(\mathbb{S}^{n_T-1})$, and obtain a latent path vector $v = \sigma_\psi(\tilde{v}, s, g)$, where conditioning on $\tilde{v}$ allows the selector to capture the non-uniqueness of $L_\xi(s, g)$ for the same state–goal pair (see Figure 2 (b)).

Since $L_\xi(s, g)$ is defined by the zero level set of $\mathcal{C}_\xi(\cdot; s, g)$, the desired parameterization is obtained by minimizing the temporal geodesic cost of the selected vectors, yielding the following objective:

$$\min_{\xi,\psi} \quad \mathbb{E}_{s,g\sim p_\mathcal{D}}\mathbb{E}_{\tilde{v}\sim q_0(\cdot|s,g)}\left[\mathcal{C}_\xi\big(\sigma_\psi(\tilde{v}, s, g); s, g\big)\right]. \tag{9}$$

Minimizing (9), however, may cause distinct reference samples to be mapped to the same solution, leading to mode collapse in the induced selector and preventing it from fully representing the potentially multimodal set of valid latent path vectors (Florence et al., 2021). To encourage diversity among low-cost path vectors, we introduce a quadratic proximity regularizer (Moreau, 1965; Amos, 2023), which associates each reference sample with a nearby low-cost solution, and minimize the following regularized objective:

$$\mathcal{L}_{\mathrm{geo}}(\xi, \psi) := \mathbb{E}_{s,g\sim p_\mathcal{D}}\mathbb{E}_{\tilde{v}\sim q_0(\cdot|s,g)}\left[\mathcal{C}_\xi\big(\sigma_\psi(\tilde{v}, s, g); s, g\big) + \alpha \left\|\hat{\sigma}_\psi(\tilde{v}, s, g) - \tilde{v}\right\|_2^2\right], \tag{10}$$

where $\hat{\sigma}_\psi$ denotes ℓ_2-normalized latent path vector and $\alpha > 0$ denotes the regularization coefficient, which is annealed to zero over training. Intuitively, the proximity term encourages σ_ψ to transport each sample of q_0 to a nearby low-cost solution, allowing different regions of the unit sphere to represent distinct temporal geodesics when multiple solutions exist (see Figure 2 (b)).

After training, temporal geodesic planning requires only sampling $\tilde{v} \sim q_0(\cdot|s, g)$, computing $v = \sigma_\psi(\tilde{v}, s, g)$, and evaluating the path $t \mapsto E_\xi(s, tv)$ for $t \in [0, 1]$. In practice, however, E_ξ may generate intermediate path points outside M, since its outputs lie in the ambient space $\mathbb{R}^{n_S}$ and are not constrained to remain on M. This motivates jointly learning the temporal distance together with the temporal geodesic.

Learning Temporal Distance. We learn the temporal distance using a quasimetric architecture d_ϕ (*e.g.*, MRN (Liu et al., 2023), IQE (Wang & Isola, 2022)) and optimize it jointly with the path model under two requirements. First, d_ϕ should recover the temporal distance within the support of $\mathcal{D}$ to provide supervision for temporal geodesic learning. Second, d_ϕ should avoid underestimating distances through regions outside the state manifold or data support, since E_ξ is defined in an ambient space containing M and may otherwise generate spurious shortcuts through such regions.

The key idea for learning a distance function that satisfies both properties is to minimize the following objective, which expresses the global maximization in (2) through two complementary terms:

$$\mathcal{L}_{\text{dist}}(\phi;\bar{\xi},\bar{\psi}) \coloneqq \mathbb{E}_{s,g\sim p_{\mathcal{D}}}\mathbb{E}_{\tilde{v}\sim q_0(\cdot|s,g)}\Big[\underbrace{\rho\Big(d_\phi(s,g),\sum_{k=0}^{K-1} d_{\bar{\phi}}(x_k,x_{k+1})\Big)}_{\text{(i) Transitive regression}} - \underbrace{h\Big(\sum_{k=0}^{K-1} d_\phi(x_k,x_{k+1})\Big)}_{\text{(ii) Path length maximization}}\Big], \quad (11)$$

where $v=\sigma_{\bar{\psi}}(\tilde{v},s,g)$, $x_k=E_{\bar{\xi}}(s,t_k v)$ with $k=0,\dots,K-1$, $x_K=g$, $0=t_0<\cdots<t_K=1$ are the discretization times, K is a hyperparameter, $\rho:\mathbb{R}\times\mathbb{R}\to\mathbb{R}_{\geq 0}$ is a regression loss function , $\bar{\cdot}$ is a stop-gradient, and $h:\mathbb{R}_{\geq 0}\to\mathbb{R}$ is a strictly increasing function controlling global distance maximization. Together with (11), we impose the local constraint from (2):

$$\mathcal{L}_{\text{local}}(\phi) := \mathbb{E}_{(s,s')\sim p_{\mathcal{D}}}[\max(d_\phi(s,s')-1,0)^2] \leq \epsilon^2. \quad (12)$$

Since d_ϕ is parameterized as a quasimetric, the triangle inequality ensures $d_\phi(s,g) \leq \sum_{k=0}^{K-1} d_{\bar{\phi}}(x_k,x_{k+1})$. Thus, (i) in (11) increases $d_\phi(s,g)$ toward the path length estimated from the intermediate points.

Meanwhile, (ii) in (11) maximizes the path lengths of the generated paths. Under exact local constraints, supported transition chains upper-bound the learned distances by their elapsed times. However, for segments where at least one of the consecutive path points x_k and x_{k+1} is unsupported, they remain subject to path length maximization pressure, making them incompatible with zero temporal geodesic cost. Consequently, since temporal geodesic learning aims to achieve zero temporal geodesic cost, unsupported shortcuts are discouraged during path optimization (see Figure 2 (a)), while the in-support temporal distance recovery is preserved; these properties are theoretically established in Section 5 and Appendix C, respectively.

Total Objective. For joint optimization, we replace d^* in (7) with the concurrently learned distance d_ϕ and denote the resulting temporal geodesic cost under d_ϕ by $\widehat{\mathcal{C}}_{\xi,\phi}(v;s,g)$. And then we express the objective in (10) as $\mathcal{L}_{\text{geo}}(\xi,\psi;\bar{\phi})$, where $\bar{\cdot}$ denotes the stop-gradient. Finally, we jointly optimize the temporal geodesic and distance by optimizing

$$\min_{\phi,\xi,\psi} \mathcal{L}_{\text{total}}(\phi,\xi,\psi) \coloneqq \lambda_{\text{geo}}\,\mathcal{L}_{\text{geo}}(\xi,\psi;\bar{\phi}) + \lambda_{\text{dist}}\,\mathcal{L}_{\text{dist}}(\phi;\bar{\xi},\bar{\psi}) \quad \text{s.t.} \quad \mathcal{L}_{\text{local}}(\phi)\leq\epsilon^2, \quad (13)$$

where $\lambda_{\text{geo}},\lambda_{\text{dist}}>0$ are loss weights and $\bar{\cdot}$ denotes the stop-gradient. Further implementation details of training GIP are provided in Appendix E.

4.3 Extracting a Local Control Policy

Once the distance and geodesic models are trained, we extract a policy for following generated temporal geodesics. The policy is trained to translate generated path structure and path directions into actions, separating global planning from local control.

For a goal g, the reduction in remaining path length within one transition naturally defines the local progress surrogate

$$A_{\bar{\phi}}(s_t,a_t,s_{t+1},g) \coloneqq d_{\bar{\phi}}(s_t,g) - d_{\bar{\phi}}(s_{t+1},g) - \min\{d_{\bar{\phi}}(s_t,g),1\}. \quad (14)$$

For distant goals, however, the difference between two long-horizon distance estimates can be noisy. We thus sample $(s_t,a_t,s_{t+1},s_{t+H})$ from $\mathcal{D}$ and evaluate a local progress surrogate using the fixed-horizon future state s_{t+H}, and optimize a variant of advantage-weighted regression objective (Peters & Schaal, 2007; Peng et al., 2019; Park et al., 2023):

$$\mathcal{L}_{\text{policy}}(\theta) \coloneqq -\mathbb{E}_{(s_t,a_t,s_{t+1},s_{t+H})\sim p_{\mathcal{D}}}\mathbb{E}_{\tilde{v}_t\sim q_0(\cdot|s_t,s_{t+H})}\Big[e^{A_{\bar{\phi}}(s_t,a_t,s_{t+1},s_{t+H})/\beta}\log\pi_\theta(a_t\mid s_t,\hat{v}_t)\Big], \quad (15)$$

where $v_t = \sigma_{\bar{\psi}}(\tilde{v}_t, s_t, s_{t+H})$, $\hat{v}_t = v_t/\|v_t\|_2$, and $\beta > 0$ is the policy-weight temperature. Since a latent path vector v represents an entire path, all points along the path share the same normalized representation $\hat{v}$. Therefore, conditioning π_θ on $(s_t, \hat{v}_t)$ provides a direction-based interface between planning and control, without directly supplying the goal, ideally producing a purely local controller that can maximize progress along the remaining path.

At test time, global planning constructs $\tilde{g}_t = E_\xi(s_t, \min\{H/d_\phi(s_t, g), 1\}v_t^g)$ by sampling $\tilde{v}_t^g \sim q_0(\cdot \mid s_t, g)$ and setting $v_t^g = \sigma_\psi(\tilde{v}_t^g, s_t, g)$. To match the training distribution of latent path vectors, local control resamples $\tilde{v}_t \sim q_0(\cdot \mid s_t, \tilde{g}_t)$, sets $v_t = \sigma_\psi(\tilde{v}_t, s_t, \tilde{g}_t)$, and executes $a_t \sim \pi_\theta(\cdot \mid s_t, v_t/\|v_t\|_2)$, with replanning at every step ; see Appendix E.6 for further design details.

5 Properties of GIP

In this section, we present several attractive properties of GIP and explain how each property arises from the proposed formulation. All proofs and detailed analyses are provided in Appendix C.

Optimality. We show that, under ideal d^* supervision, proximity regularization's influence on the optimality of our geodesic objective vanishes as $\alpha \to 0$. We also provide, under additional assumptions, that our distance objective recovers d^* in the supported domain (see Theorem C.4).

Theorem 5.1 (Asymptotic Temporal Geodesic Recovery)**.** *Under Assumption 4.1, suppose E_ξ and ρ are continuous and the induced paths remain on M throughout optimization. Suppose the conditions of Theorem C.2 hold. For each $\alpha > 0$, let $(\xi_\alpha, \psi_\alpha)$ be a global minimizer of (10). If $(\xi_\alpha, \psi_\alpha) \to (\xi_0, \psi_0)$ as $\alpha \to 0$ and the population geodesic cost is continuous at the limit, then $t \mapsto E_{\xi_0}(s, t\sigma_{\psi_0}(\tilde{v}, s, g))$ is a temporal geodesic almost surely.*

This theorem establishes path-level optimality, and an accurate local controller can realize the recovered path as an optimal goal-reaching trajectory.

Out-of-Support Penalization. We also analyze how joint learning of distance and geodesic suppresses shortcuts through regions unsupported by the data. The key observation is that the path length maximization pressure ideally leaves all in-support distances unchanged.

Theorem 5.2 (Out-of-Support Penalization)**.** *Let $\mathcal{S}_\mathcal{D} \subseteq \mathcal{S}$ be a nonempty closed data-supported state domain, and suppose it and all generated paths lie in a common compact ambient domain. With generated paths and regression targets fixed, if $\Pr[\exists k : x_k \notin \mathcal{S}_\mathcal{D}] > 0$, there exists a continuous quasimetric $\widetilde{d}$ such that $\widetilde{d} = d_\phi$ on $\mathcal{S}_\mathcal{D} \times \mathcal{S}_\mathcal{D}$,, $\mathcal{L}_{\text{dist}}$ strictly decreases, and $L_{\widetilde{d}}(\gamma) > \widetilde{d}(s, g)$ for every out-of-support path $\gamma \in C(s, g)$ with $s, g \in \mathcal{S}_\mathcal{D}$.*

The theorem has a useful implication. By Theorem C.1, a zero-cost path must satisfy $L_d(\gamma) = d(s, g)$, hence the strict inequality eliminates the possibility of zero-cost out-of-support paths. Consequently, minimizing $\mathcal{L}_{\text{geo}}$ encourages σ_ψ and E_ξ to move toward supported alternatives when such alternatives are representable.

Scalability. GIP avoids TD learning by enforcing constant temporal-speed consistency directly between sampled path points. This Monte Carlo formulation is less subject to the accumulation of bootstrapping and model-estimation errors compared to TD updates and autoregressive rollout (Park et al., 2025b; Li et al., 2026).

Superior Efficiency. GIP represents an entire continuous path with a single fixed dimensional latent path vector. By amortizing path optimization and generation into learning the selector model, and sampling a random reference vector, GIP avoids computation that scales with the planning horizon.

6 Experiments

In this section, we empirically validate GIP from three perspectives. In Section 6.1, we compare its performance and computational efficiency with strong model-based planning and model-free baselines. In Section 6.2, we quantitatively evaluate whether GIP captures complete goal-reaching paths between arbitrary states. In Section 6.3, we empirically examine the scalability of GIP.

Table 1: **Results on OGBench tasks (8 seeds).** Top-3 methods are rank-colored, with scores within 95% of a higher-ranked method sharing its color. Averages use available entries only, with the best shown in **bold**.

			Model-Free					Test-Time Planning		Trajectory Modeling			Ours
Environment	**Dataset**	**Size**	GCIVL	GCIQL	CRL	QRL	HIQL	TTGS	ALPS	DD	HD	DV	GIP
antmaze	navigate	large	$16_{\pm 5}$	$34_{\pm 4}$	$83_{\pm 4}$	$75_{\pm 6}$	$91_{\pm 2}$	$92_{\pm 2}$	$93_{\pm 5}$	$14_{\pm 4}$	$31_{\pm 5}$	$51_{\pm 5}$	$91_{\pm 3}$
		giant	$0_{\pm 0}$	$0_{\pm 0}$	$16_{\pm 3}$	$14_{\pm 3}$	$65_{\pm 5}$	$66_{\pm 4}$	$69_{\pm 9}$	$5_{\pm 2}$	$19_{\pm 4}$	$35_{\pm 4}$	$74_{\pm 4}$
	stitch	large	$18_{\pm 2}$	$7_{\pm 2}$	$11_{\pm 2}$	$18_{\pm 2}$	$67_{\pm 5}$	$91_{\pm 2}$	$95_{\pm 2}$	–	–	–	$78_{\pm 3}$
		giant	$0_{\pm 0}$	$0_{\pm 0}$	$0_{\pm 0}$	$0_{\pm 0}$	$2_{\pm 2}$	$79_{\pm 13}$	$92_{\pm 3}$	–	–	–	$77_{\pm 6}$
	explore	large	$10_{\pm 3}$	$0_{\pm 0}$	$0_{\pm 0}$	$0_{\pm 0}$	$4_{\pm 5}$	$27_{\pm 34}$	$90_{\pm 15}$	–	–	–	$71_{\pm 5}$
humanoidmaze	navigate	medium	$24_{\pm 2}$	$27_{\pm 2}$	$60_{\pm 4}$	$21_{\pm 8}$	$89_{\pm 2}$	$93_{\pm 3}$	$89_{\pm 5}$	$9_{\pm 3}$	$24_{\pm 4}$	$38_{\pm 5}$	$92_{\pm 1}$
		large	$2_{\pm 1}$	$2_{\pm 1}$	$24_{\pm 4}$	$5_{\pm 1}$	$49_{\pm 4}$	$79_{\pm 6}$	$56_{\pm 5}$	$5_{\pm 2}$	$16_{\pm 3}$	$29_{\pm 3}$	$76_{\pm 2}$
		giant	$0_{\pm 0}$	$0_{\pm 0}$	$3_{\pm 2}$	$1_{\pm 0}$	$12_{\pm 4}$	$85_{\pm 6}$	$67_{\pm 11}$	$1_{\pm 1}$	$7_{\pm 3}$	$16_{\pm 3}$	$81_{\pm 1}$
	stitch	medium	$12_{\pm 2}$	$12_{\pm 3}$	$36_{\pm 2}$	$18_{\pm 2}$	$88_{\pm 2}$	$85_{\pm 3}$	$68_{\pm 5}$	–	–	–	$91_{\pm 2}$
		large	$1_{\pm 1}$	$0_{\pm 0}$	$4_{\pm 1}$	$3_{\pm 1}$	$28_{\pm 3}$	$65_{\pm 11}$	$39_{\pm 6}$	–	–	–	$67_{\pm 2}$
		giant	$0_{\pm 0}$	$0_{\pm 0}$	$0_{\pm 0}$	$0_{\pm 0}$	$3_{\pm 2}$	$78_{\pm 5}$	$62_{\pm 6}$	–	–	–	$81_{\pm 5}$
scene	play		$42_{\pm 4}$	$51_{\pm 4}$	$19_{\pm 2}$	$5_{\pm 1}$	$38_{\pm 3}$	$52_{\pm 4}$	$26_{\pm 3}$	$16_{\pm 4}$	$37_{\pm 6}$	$56_{\pm 3}$	$65_{\pm 12}$
cube	play	single	$53_{\pm 4}$	$68_{\pm 6}$	$19_{\pm 2}$	$5_{\pm 1}$	$15_{\pm 3}$	$34_{\pm 3}$	$68_{\pm 6}$	$8_{\pm 3}$	$20_{\pm 4}$	$32_{\pm 5}$	$32_{\pm 5}$
Average			13.7	15.5	21.2	12.7	42.4	71.2	70.3	8.3	22.0	36.7	**75.1**

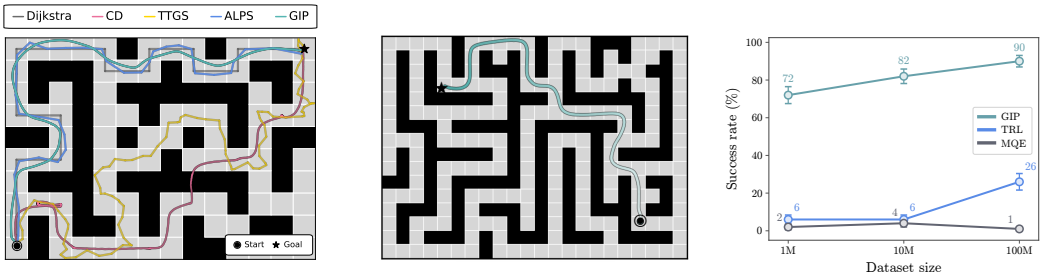
6.1 GIP IS EFFECTIVE YET EFFICIENT

Tasks and Baselines. We evaluate GIP on state-based OGBench tasks (Park et al., 2025a) across locomotion and manipulation domains. In locomotion, an Ant with 29-dimensional states and 8-dimensional actions and a Humanoid with 69-dimensional states and 21-dimensional actions navigate mazes toward target goals in `antmaze` and `humanoidmaze`, respectively. The `medium`, `large`, and `giant` variants denote maze size, while `navigate`, `stitch`, and `explore` correspond to noisy expert, trajectory-stitching, and exploratory datasets, respectively. In manipulation, `cube` and `scene` require interacting with objects to reach target states. Further details are provided in Appendix F.

We compare GIP against model-free GCRL methods, including GCIVL (Park et al., 2023) and GCIQL (Kostrikov et al., 2022; Park et al., 2025a) based on value learning, CRL (Eysenbach et al., 2022) based on contrastive learning, QRL (Wang et al., 2023) based on quasimetric learning, and HIQL (Park et al., 2023) based on hierarchical goal-conditioned learning. We further compare with test-time graph-search planners TTGS (Opryshko et al., 2026) and ALPS (Shehmar et al., 2026), and diffusion-based trajectory models Decision Diffuser (DD) (Ajay et al., 2023), Hierarchical Diffuser (HD) (Chen et al., 2024), and DiffusionVeteran (DV) (Lu et al., 2025). For TTGS, we use HIQL for locomotion and GCIQL for manipulation as the base policy to provide a strong baseline in each domain. For a fair comparison, we exclude diffusion-based trajectory models from `stitch` and `explore`, for which their whole-trajectory modeling formulation is not well suited.

Results. We report success rates for 13 OGBench environment–dataset configurations in Table 1. Overall, GIP achieves the highest average success rate of 75.1, outperforming model-free, test-time planning, and trajectory-modeling baselines. Notably, GIP achieves the best or near-best performance on the high-dimensional, long-horizon `humanoidmaze` tasks and remains competitive on the challenging `stitch` and `explore` datasets, demonstrating strong goal-reaching performance even with highly suboptimal offline data.

Beyond performance, GIP benefits from its superior efficiency, as established in Section 5. In Figure 1, we compare average performance over 16 seeds, inference FLOPs per environment step, and total training FLOPs on `humanoidmaze-giant-navigate` and `-stitch`, which are among the longest-horizon configurations in the benchmark comparison. GIP requires $25.4\times$ fewer inference FLOPs per step than the closest-performing HIQL+TTGS, while achieving $10.8\times$ higher performance than HIQL, which has comparable inference cost. These results highlight the effectiveness of implicitly encoding a complete plan into a single sampled path vector and acting along its induced direction. Moreover, whereas the test-time search can grow exponentially with the planning horizon, GIP samples a fixed-dimensional path vector for a given agent, making its per-step inference cost independent of the horizon. Further details on how each metric is measured are provided in Appendix F.



(a) (b) (c)

Figure 3: (a) Comparison of plans generated by algorithms in `humanoidmaze-giant-stitch`. (b) A path generated by GIP on `humanoidmaze-colossal-navigate`. Each decoded state is 69-dimensional. (c) Performance comparison on `humanoidmaze-colossal-navigate` with `1M`, `10M`, `100M` transition datasets (4 seeds).

6.2 Does GIP Really Learn the Entire Goal-Reaching Path?

To assess whether GIP truly learns the entire path structure, we visualize paths in Figure 3 (a). Specifically, on `humanoidmaze-giant-stitch`, we visualize GIP's planned path by evaluating $E_\xi(s, t\,\sigma_\psi(\tilde{v}, s, g))$ at equally spaced values of $t \in [0, 1]$, with $\tilde{v}$ fixed. Note that each decoded state is 69-dimensional; only its xy position is plotted in the figure. We compare the resulting path with a grid-based Dijkstra path and paths generated by ALPS, TTGS, and CompDiffuser (CD) (Luo et al., 2025), a diffusion-based trajectory modeling approach for long-horizon trajectory stitching.

We find that GIP and ALPS match Dijkstra's shortest route by tile count, while GIP requires 6,193× less inference computation per environment step than ALPS. In contrast, TTGS cannot guarantee optimality due to its heuristic graph construction from randomly sampled dataset states. Further analysis of the limitations of test-time search based on heuristic structures, as employed by ALPS and TTGS, is provided in Appendix G. A similar limitation applies to CompDiffuser (CD), which generates plans by sampling from a learned trajectory distribution. Both GIP and CD model whole trajectories, but GIP structurally penalizes out-of-support paths such as wall crossings to discourage infeasible plans. The result highlights why GIP is geometry-informed *planning*: its learned geometry encodes an entire goal-reaching path as a temporal geodesic. Additional visualizations of plans generated by GIP from start states across diverse tasks are provided in Appendix G.

6.3 Is GIP Scalable?

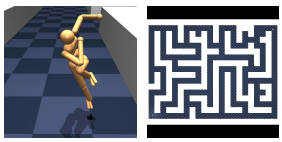


Figure 4: `hmaze-colossal`

To evaluate scalability over longer horizons and larger datasets, we construct the `humanoidmaze-colossal` environment (see Figure 4) using the `colossal` layout of MQE (Zheng et al., 2026), which is 50% larger than `giant` in each dimension and requires up to approximately 4,000 steps. With a 6,000-step episode limit, we evaluate 1M-, 10M-, and 100M-transition `navigate` datasets over five tasks with 20 episodes each. GIP outperforms scalability-oriented baselines TRL (Park et al., 2026b) and MQE (Zheng et al., 2026) at all dataset scales, while showing the most consistent, approximately linear improvement with increasing data (see Figure 3 (b) and Figure 3 (c)). These results demonstrate GIP's scalability to longer-horizon problems and larger datasets, consistent with its avoidance of recursive error propagation across the planning horizon.

7 Conclusion

We introduce Geometry-Informed Planning (GIP), an effective yet efficient planning approach with attractive properties for offline GCRL. Motivated by the relationship between tangent vectors, geodesics, and exponential maps, GIP learns to represent the optimal path through a single latent path vector, along with the temporal distance. Despite the limitations discussed in Appendix A, GIP provides both theoretical advantages and empirical evidence of effectiveness and efficiency, demonstrating the potential of geometric inductive biases for planning and control. We view this as a first step toward extending such geometric formulations to richer settings, including visual and language goals, which we leave for future work. We conclude with the hope that GIP and its underlying geometric principles will inspire broader applications beyond offline GCRL.

AI USE STATEMENT

In this work, we used generative AI tools to formulate mathematical claims, provide critical ingredients for proving mathematical claims, assist in the writing of proofs, refine hypotheses, provide feedback on research methodology or experiments, and assist with translation. We have not used generative AI tools to help develop theoretical models or conceptual frameworks, implement methods, or interpret results. The following required-disclosure tasks are not applicable to this work: generating synthetic data sets, cleaning and reformatting datasets, and supporting qualitative and thematic data analysis. Additionally, we used generative AI tools to create or modify figures or images, suggest experimental parameters, create or edit software code, identify relevant literature, format references, support brainstorming, source or search for information, and edit the research paper to improve readability. We have reviewed all AI-assisted work. In particular, AI-assisted mathematical claims and proof steps were checked by the authors for correctness, and AI-assisted code was reviewed and tested by the authors. We take responsibility for the final content of this work, including text, claims, code, and other artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

We provide the assumptions and complete proofs of our theoretical results in Appendices C and D. Detailed implementation choices, model parameterizations, and training and inference pseudocode are provided in Appendix E. We describe the environments, datasets, evaluation protocol, baselines, and computation accounting in Appendix F, and report the hyperparameters used for GIP in Table 4. Additional experiments, ablations, and complete benchmark results are provided in Appendix G. We additionally provide an anonymized implementation as supplementary material with the submission and intend to release the code publicly. Together, these materials are intended to support independent reproduction of both the theoretical and empirical results.

REFERENCES

Anurag Ajay, Yilun Du, Abhi Gupta, Joshua Tenenbaum, Tommi Jaakkola, and Pulkit Agrawal. Is conditional generative modeling all you need for decision-making? In *Proc. of the International Conference on Learning Representations (ICLR)*, Kigali, Rwanda, May 2023.

Brandon Amos. Tutorial on amortized optimization. *Foundations and Trends in Machine Learning*, 16(5):592–732, 2023. doi: 10.1561/2200000102.

Georgios Arvanitidis, Søren Hauberg, Philipp Hennig, and Michael Schober. Fast and robust shortest paths on manifolds learned from data. In *Proc. of the International Conference on Artificial Intelligence and Statistics (AISTATS)*, Naha, Okinawa, Japan, Apr. 2019.

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.

Seungho Baek, Taegeon Park, Jongchan Park, Seungjun Oh, and Yusung Kim. Graph-assisted stitching for offline hierarchical reinforcement learning. In *Proc. of the International Conference on Machine Learning (ICML)*, Vancouver, Canada, Jul. 2025.

Marco Bagatella and Georg Martius. Goal-conditioned offline planning from curious exploration. In *Proc. of the Advances in Neural Information Processing Systems (NeurIPS)*, New Orleans, LA, USA, Dec. 2023.

Hadi Beik-Mohammadi, Søren Hauberg, Georgios Arvanitidis, Gerhard Neumann, and Leonel Rozo. Learning Riemannian manifolds for geodesic motion skills. In *Proc. of Robotics: Science and Systems (RSS)*, Virtual conference, Jul. 2021.

Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Robert Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, Szymon Jakubczak, Tim Jones, Liyiming Ke, Sergey Levine, Adrian Li-Bell, Mohith Mothukuri, Suraj Nair, Karl Pertsch, Lucy Xiaoyang Shi, Laura Smith, James Tanner, Quan Vuong, Anna Walling, Haohuan Wang, and Ury Zhilinsky. π_0: A vision-language-action flow model for general robot control. In *Proc. of Robotics: Science and Systems (RSS)*, Los Angeles, CA, USA, Jun. 2025.

Edoardo Cetin, Benjamin Paul Chamberlain, Michael M. Bronstein, and Jonathan J. Hunt. Hyperbolic deep reinforcement learning. In *Proc. of the International Conference on Learning Representations (ICLR)*, Kigali, Rwanda, May 2023.

Chang Chen, Fei Deng, Kenji Kawaguchi, Caglar Gulcehre, and Sungjin Ahn. Simple hierarchical planning with diffusion. In *Proc. of the International Conference on Learning Representations (ICLR)*, Vienna, Austria, May 2024.

Ricky T. Q. Chen and Yaron Lipman. Flow matching on general geometries. In *Proc. of the International Conference on Learning Representations (ICLR)*, Vienna, Austria, May 2024.

Jie Cheng, Ruixi Qiao, Yingwei Ma, Binhua Li, Gang Xiong, Qinghai Miao, Yongbin Li, and Yisheng Lv. Scaling offline model-based RL via jointly-optimized world-action model pretraining. In *Proc. of the International Conference on Learning Representations (ICLR)*, Singapore, Apr. 2025.

Junha Chun, Youngjoon Jeong, and Taesup Kim. Sparse imagination for efficient visual world model planning. In *Proc. of the International Conference on Learning Representations (ICLR)*, Rio de Janeiro, Brazil, Apr. 2026.

Hojun Chung, Junseo Lee, and Songhwai Oh. Offline reinforcement learning with universal horizon models. In *Proc. of the International Conference on Machine Learning (ICML)*, Seoul, South Korea, Jul. 2026.

Stephen Chung, Ivan Anokhin, and David Krueger. Thinker: Learning to plan and act. In *Proc. of the Advances in Neural Information Processing Systems (NeurIPS)*, New Orleans, LA, USA, Dec. 2023.

Valentin De Bortoli, Emile Mathieu, Michael Hutchinson, James Thornton, Yee Whye Teh, and Arnaud Doucet. Riemannian score-based generative modelling. In *Proc. of the Advances in Neural Information Processing Systems (NeurIPS)*, New Orleans, LA, USA, Dec. 2022.

Benjamin Eysenbach, Tianjun Zhang, Sergey Levine, and Russ R Salakhutdinov. Contrastive learning as goal-conditioned reinforcement learning. In *Proc. of the Advances in Neural Information Processing Systems (NeurIPS)*, New Orleans, LA, USA, Nov. 2022.

Pete Florence, Corey Lynch, Andy Zeng, Oscar A. Ramirez, Ayzaan Wahid, Laura Downs, Adrian Wong, Johnny Lee, Igor Mordatch, and Jonathan Tompson. Implicit behavioral cloning. In *Proc. of the Conference on Robot Learning (CoRL)*, London, UK, Nov. 2021.

Chongkai Gao, Haozhuo Zhang, Zhixuan Xu, Zhehao Cai, and Lin Shao. FLIP: Flow-centric generative planning as general-purpose manipulation world model. In *Proc. of the International Conference on Learning Representations (ICLR)*, Singapore, Apr. 2025.

Vittorio Giammarino and Ahmed Hussain Qureshi. Goal reaching with eikonal-constrained hierarchical quasimetric reinforcement learning. In *Proc. of the International Conference on Learning Representations (ICLR)*, Rio de Janeiro, Brazil, Apr. 2026.

Nandiraju Gireesh, Yuanliang Ju, Chaoyi Xu, Weiheng Liu, Yuxuan Wan, and He Wang. HDFlow: Hierarchical diffusion-flow planning for long-horizon tasks. In *Proc. of the International Conference on Machine Learning (ICML)*, Seoul, South Korea, Jul. 2026.

Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Proc. of Advances in Neural Information Processing Systems (NIPS)*, Montreal, Canada, Dec. 2014.

Danijar Hafner, Jurgis Pasukonis, Jimmy Ba, and Timothy Lillicrap. Mastering diverse control tasks through world models. *Nature*, 640(8059):647–653, 2025.

Nicklas Hansen, Xiaolong Wang, and Hao Su. Temporal difference learning for model predictive control. In *Proc. of the International Conference on Machine Learning (ICML)*, Baltimore, MD, USA, Jul. 2022.

Dan Hendrycks and Kevin Gimpel. Gaussian error linear units (GELUs). *arXiv preprint arXiv:1606.08415*, 2016.

Mineui Hong, Minjae Kang, and Songhwai Oh. Diffused task-agnostic milestone planner. In *Proc. of the Advances in Neural Information Processing Systems (NeurIPS)*, New Orleans, LA, USA, Dec. 2023.

Peter J. Huber. Robust estimation of a location parameter. *The Annals of Mathematical Statistics*, 35 (1):73–101, Mar. 1964.

Guillaume Huguet, Daniel Sumner Magruder, Alexander Tong, Oluwadamilola Fasina, Manik Kuchroo, Guy Wolf, and Smita Krishnaswamy. Manifold interpolating optimal-transport flows for trajectory inference. In *Proc. of the Advances in Neural Information Processing Systems (NeurIPS)*, New Orleans, LA, USA, Dec. 2022.

Michael Janner, Qiyang Li, and Sergey Levine. Offline reinforcement learning as one big sequence modeling problem. In *Proc. of the Advances in Neural Information Processing Systems (NeurIPS)*, Virtual conference, Dec. 2021.

Michael Janner, Yilun Du, Joshua B Tenenbaum, and Sergey Levine. Planning with diffusion for flexible behavior synthesis. In *Proc. of the International Conference on Machine Learning (ICML)*, Baltimore, MD, USA, Jul. 2022.

Hyungkyu Kang, Byeongchan Kim, and Min-hwan Oh. Latent representation alignment for offline goal-conditioned reinforcement learning. In *Proc. of the International Conference on Machine Learning (ICML)*, Seoul, South Korea, Jul. 2026.

Kacper Kapuśniak, Peter Potaptchik, Teodora Reu, Leo Zhang, Alexander Tong, Michael Bronstein, Avishek Joey Bose, and Francesco Di Giovanni. Metric flow matching for smooth interpolations on the data manifold. In *Proc. of the Advances in Neural Information Processing Systems (NeurIPS)*, Vancouver, Canada, Dec. 2024.

Junseok Kim, Dohyeong Kim, Mineui Hong, and Songhwai Oh. Compositional transduction with latent analogies for offline goal-conditioned reinforcement learning. In *Proc. of the International Conference on Machine Learning (ICML)*, Seoul, South Korea, July 2026a.

Moo Jin Kim, Yihuai Gao, Tsung-Yi Lin, Yen-Chen Lin, Yunhao Ge, Grace Lam, Percy Liang, Shuran Song, Ming-Yu Liu, Chelsea Finn, and Jinwei Gu. Cosmos policy: Fine-tuning video models for visuomotor control and planning. In *Proc. of the International Conference on Learning Representations (ICLR)*, Rio de Janeiro, Brazil, Apr. 2026b.

Shoshichi Kobayashi and Katsumi Nomizu. *Foundations of differential geometry*, volume 2. John Wiley & Sons, 1996.

Ilya Kostrikov, Ashvin Nair, and Sergey Levine. Offline reinforcement learning with implicit Q-learning. In *Proc. of the International Conference on Learning Representations (ICLR)*, Virtual conference, Apr. 2022.

Anurag Koul, Shivakanth Sujit, Shaoru Chen, Ben Evans, Lili Wu, Byron Xu, Rajan Chari, Riashat Islam, Raihan Seraj, Yonathan Efroni, Lekan P Molu, Miroslav Dudík, John Langford, and Alex Lamb. PcLast: Discovering plannable continuous latent states. In *Proc. of the International Conference on Machine Learning (ICML)*, Vienna, Austria, Jul. 2024.

Daniel Lawson, Adriana Hugessen, Charlotte Cloutier, Glen Berseth, and Khimya Khetarpal. Self-predictive representations for combinatorial generalization in behavioral cloning. In *Proc. of the International Conference on Learning Representations (ICLR)*, Rio de Janeiro, Brazil, Apr. 2026.

Qiyang Li, Seohong Park, and Sergey Levine. Decoupled q-chunking. In *Proc. of the International Conference on Learning Representations (ICLR)*, Rio de Janeiro, Brazil, Apr. 2026.

Wenhao Li. Efficient planning with latent diffusion. In *Proc. of the International Conference on Learning Representations (ICLR)*, Vienna, Austria, May 2024.

Bo Liu, Yihao Feng, Qiang Liu, and Peter Stone. Metric residual networks for sample efficient goal-conditioned reinforcement learning. In *Proc. of the AAAI Conference on Artificial Intelligence*, Washington, DC, USA, Feb. 2023.

Haofei Lu, Dongqi Han, Yifei Shen, and Dongsheng Li. What makes a good diffusion planner for decision making? In *Proc. of the International Conference on Learning Representations (ICLR)*, Singapore, Apr. 2025.

Yunhao Luo, Utkarsh A Mishra, Yilun Du, and Danfei Xu. Generative trajectory stitching through diffusion composition. In *Proc. of the Advances in Neural Information Processing Systems (NeurIPS)*, San Diego, CA, USA, Dec. 2025.

Yecheng Jason Ma, Jason Yan, Dinesh Jayaraman, and Osbert Bastani. How far I'll go: Offline goal-conditioned reinforcement learning via f-advantage regression. In *Proc. of the Advances in Neural Information Processing Systems (NeurIPS)*, New Orleans, LA, USA, Nov. 2022.

Tomas Merva, Saray Bakker, Max Spahn, Danning Zhao, Ivan Virgala, and Javier Alonso-Mora. Globally-guided geometric fabrics for reactive mobile manipulation in dynamic environments. *IEEE Robotics and Automation Letters*, 10:5553–5560, Jun. 2025.

Lars Mescheder, Andreas Geiger, and Sebastian Nowozin. Which training methods for GANs do actually converge? In *Proc. of the International Conference on Machine Learning (ICML)*, Stockholm, Sweden, Jul. 2018.

Utkarsh A. Mishra, David He, Yongxin Chen, and Danfei Xu. Compositional diffusion with guided search for long-horizon planning. In *Proc. of the International Conference on Learning Representations (ICLR)*, Rio de Janeiro, Brazil, Apr. 2026.

Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. Spectral normalization for generative adversarial networks. In *Proc. of the International Conference on Learning Representations (ICLR)*, Vancouver, Canada, Apr. 2018.

Jean-Jacques Moreau. Proximité et dualité dans un espace hilbertien. *Bulletin de la Société Mathématique de France*, 93:273–299, 1965.

Mustafa Mukadam, Ching-An Cheng, Dieter Fox, Byron Boots, and Nathan Ratliff. Riemannian motion policy fusion through learnable lyapunov function reshaping. In *Proc. of the Conference on Robot Learning (CoRL)*, Osaka, Japan, Oct. 2020.

Vivek Myers, Chongyi Zheng, Anca Dragan, Sergey Levine, and Benjamin Eysenbach. Learning temporal distances: Contrastive successor features can provide a metric structure for decision-making. In *Proc. of the International Conference on Machine Learning (ICML)*, Vienna, Austria, Jul. 2024.

Vivek Myers, Bill Zheng, Benjamin Eysenbach, and Sergey Levine. Offline goal-conditioned reinforcement learning with quasimetric representations. In *Proc. of the Advances in Neural Information Processing Systems (NeurIPS)*, San Diego, CA, USA, Dec. 2025.

Evgenii Opryshko, Junwei Quan, Claas Voelcker, Yilun Du, and Igor Gilitschenski. Test-time graph search for goal-conditioned reinforcement learning. In *Proc. of the International Conference on Machine Learning (ICML)*, Seoul, South Korea, Jul. 2026.

Seohong Park, Dibya Ghosh, Benjamin Eysenbach, and Sergey Levine. HIQL: Offline goal-conditioned RL with latent states as actions. In *Proc. of the Advances in Neural Information Processing Systems (NeurIPS)*, New Orleans, LA, USA, Dec. 2023.

Seohong Park, Tobias Kreiman, and Sergey Levine. Foundation policies with Hilbert representations. In *Proc. of the International Conference on Machine Learning (ICML)*, Vienna, Austria, Jul. 2024.

Seohong Park, Kevin Frans, Benjamin Eysenbach, and Sergey Levine. OGBench: Benchmarking offline goal-conditioned RL. In *Proc. of the International Conference on Learning Representations (ICLR)*, Singapore, Apr. 2025a.

Seohong Park, Kevin Frans, Deepinder Mann, Benjamin Eysenbach, Aviral Kumar, and Sergey Levine. Horizon reduction makes RL scalable. In *Proc. of the Advances in Neural Information Processing Systems (NeurIPS)*, San Diego, CA, USA, Dec. 2025b.

Seohong Park, Deepinder Mann, and Sergey Levine. Dual goal representations. In *Proc. of the International Conference on Learning Representations (ICLR)*, Rio de Janeiro, Brazil, Apr. 2026a.

Seohong Park, Aditya Oberai, Pranav Atreya, and Sergey Levine. Transitive RL: Value learning via divide and conquer. In *Proc. of the International Conference on Learning Representations (ICLR)*, Rio de Janeiro, Brazil, Apr. 2026b.

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*, Oct. 2019.

Jan Peters and Stefan Schaal. Reinforcement learning by reward-weighted regression for operational space control. In *Proc. of the International Conference on Machine Learning (ICML)*, Corvallis, OR, USA, Jun. 2007.

Dikshant Shehmar, Matthew Schlegel, Matthew E. Taylor, and Marlos C. Machado. Laplacian representations for decision-time planning. In *Proc. of the International Conference on Machine Learning (ICML)*, Seoul, South Korea, Jul. 2026.

Zhongmin Shen. *Differential Geometry of Spray and Finsler Spaces*. Springer, Dordrecht, 2001.

Uladzislau Sobal, Wancong Zhang, Kyunghyun Cho, Randall Balestriero, Tim G. J. Rudner, and Yann LeCun. Learning from reward-free offline data: A case for planning with latent dynamics models. In *Proc. of the Advances in Neural Information Processing Systems (NeurIPS)*, San Diego, CA, USA, Dec. 2025.

Lorenzo Steccanella, Joshua B. Evans, Özgür Şimşek, and Anders Jonsson. Learning the minimum action distance. In *Proc. of the International Conference on Machine Learning (ICML)*, Seoul, South Korea, Jul. 2026.

Tao Sun, Utkarsh A. Mishra, Jiaxin Lu, Danfei Xu, and Iro Armeni. Energy-based compositional diffusion planning. In *Proc. of the International Conference on Machine Learning (ICML)*, Seoul, South Korea, Jul. 2026.

Saket Tiwari, Omer Gottesman, and George Konidaris. Geometry of neural reinforcement learning in continuous state and action spaces. In *Proc. of the International Conference on Learning Representations (ICLR)*, Singapore, Apr. 2025.

Tongzhou Wang and Phillip Isola. Improved representation of asymmetrical distances with interval quasimetric embeddings. *arXiv preprint arXiv:2211.15120*, Nov. 2022.

Tongzhou Wang, Antonio Torralba, Phillip Isola, and Amy Zhang. Optimal goal-reaching reinforcement learning via quasimetric learning. In *Proc. of the International Conference on Machine Learning (ICML)*, Honolulu, HI, USA, Jul. 2023.

Mandy Xie, Ankur Handa, Stephen Tyree, Dieter Fox, Harish Ravichandar, Nathan D. Ratliff, and Karl Van Wyk. Neural geometric fabrics: Efficiently learning high-dimensional policies from demonstration. In *Proc. of the Conference on Robot Learning (CoRL)*, Auckland, New Zealand, Dec. 2023.

Jaesik Yoon, Hyeonseo Cho, and Sungjin Ahn. Compositional Monte Carlo tree diffusion for extendable planning. In *Proc. of the Advances in Neural Information Processing Systems (NeurIPS)*, San Diego, CA, USA, Dec. 2025a.

Jaesik Yoon, Hyeonseo Cho, Doojin Baek, Yoshua Bengio, and Sungjin Ahn. Monte Carlo tree diffusion for System 2 planning. In *Proc. of the International Conference on Machine Learning (ICML)*, Vancouver, Canada, Jul. 2025b.

Jaesik Yoon, Hyeonseo Cho, Yoshua Bengio, and Sungjin Ahn. Fast Monte Carlo tree diffusion: 100× speedup via parallel and sparse planning. In *Proc. of the Advances in Neural Information Processing Systems (NeurIPS)*, San Diego, CA, USA, Dec. 2025c.

Bill Zheng, Vivek Myers, Benjamin Eysenbach, and Sergey Levine. Multistep quasimetric learning for scalable goal-conditioned reinforcement learning. In *Proc. of the International Conference on Learning Representations (ICLR)*, Rio de Janeiro, Brazil, Apr. 2026.

Gaoyue Zhou, Hengkai Pan, Yann LeCun, and Lerrel Pinto. DINO-WM: World models on pre-trained visual features enable zero-shot planning. In *Proc. of the International Conference on Machine Learning (ICML)*, Vancouver, Canada, Jul. 2025.

A Limitations

GIP achieves efficient implicit planning by representing an entire goal-reaching path with a single latent path vector, but it has several limitations.

Geometric assumptions. GIP relies on Assumption 4.1, which assumes that the state space can be modeled as a smooth manifold equipped with a geodesic temporal quasimetric and a compatible exponential map. If this geometric structure does not adequately describe the underlying state space, the path parameterization inspired by the relationship between exponential map and geodesic may fail to represent optimal goal-reaching paths accurately. However, this assumption remains relatively general because GIP does not require a prescribed Riemannian metric tensor or a specific fixed metric structure. Moreover, related geometric assumptions have also been widely adopted in continuous control (Tiwari et al., 2025), supporting their use as a reasonable abstraction for the domains considered in this work.

Dependence on transition coverage. GIP builds its distance learning on the QRL principle of combining global distance spreading with local constraints on observed transitions. This can be less effective in manipulation domains, where demonstrations often occupy weakly overlapping regions of the state space and therefore provide insufficient transition coverage to constrain the global temporal geometry. Consistent with this limitation, GIP performs comparatively worse on some manipulation tasks. Our additional experiments in Appendix G show that performance improves substantially when using the higher-coverage `noisy` dataset instead of the lower-coverage `play` dataset, supporting transition coverage as an important factor in temporal distance learning.

Coupled distance–path optimization. GIP jointly learns the distance field and generated paths through an adversarial game-like interaction in which the distance model increases the costs of undesirable paths while the path model adapts to reduce its geodesic cost. Although this interaction enables out-of-support penalization, it can introduce nonstationarity and optimization instability, including undesirable changes in the global scale of the learned distance. We mitigate this issue through the scale-separated quasimetric parameterization described in Appendix E.3.

B Extended Related Work

Planning with Learned Models. Planning with learned models has become a prominent paradigm for long-horizon sequential decision-making (Hansen et al., 2022; Chung et al., 2023; Hafner et al., 2025). Existing methods either perform explicit test-time optimization or search over learned models (Zhou et al., 2025; Sobal et al., 2025; Baek et al., 2025; Yoon et al., 2025b; Opryshko et al., 2026; Shehmar et al., 2026), or directly generate plans from learned autoregressive or generative trajectory models (Janner et al., 2021; 2022; Hong et al., 2023; Li, 2024; Chen et al., 2024; Luo et al., 2025). Both paradigms can suffer from compounding errors (Chung et al., 2026), lack plan-optimality guarantees, and incur growing training or inference costs with horizon and task complexity (Yoon et al., 2025c; Chun et al., 2026). GIP instead represents an entire path by a single latent path vector, requiring only one vector sample per plan and enabling efficient long-horizon planning.

C Theoretical Analysis

We provide the proofs and supporting results for Section 5. We first characterize zero temporal geodesic cost and show that it exactly corresponds to a constant-speed shortest path. We then use this characterization to prove asymptotic temporal geodesic recovery and to connect the distance objective to out-of-support penalization. Finally, we give additional conditions under which the learned quasimetric recovers the temporal distance on the data-supported domain.

Throughout this section, we retain the notation $x_t = E_\xi(s, tv)$ and write $\gamma(t) = x_t$ for the corresponding continuous path. Unless stated otherwise, expectations are taken over $s, g \sim p_\mathcal{D}$ and $\tilde{v} \sim q_0(\cdot \mid s, g)$. We begin by proving Proposition 4.2, which provides the geometric basis for representing an entire temporal geodesic by a single vector.

Proposition 4.2. *Under Assumption 4.1, for any $s, g \in M$, there exists $v^* \in T_sM$ satisfying the following conditions:*

Initial boundary: $\mathrm{Exp}_s(0) = s$,

Endpoint boundary: $\mathrm{Exp}_s(v^*) = g$,

Constant speed: $d^*(\mathrm{Exp}_s(t_i v^*), \mathrm{Exp}_s(t_j v^*)) = (t_j - t_i) d^*(s, g), \quad \forall t_i, t_j \in [0, 1],\ t_i \le t_j$.

Proof. Under Assumption 4.1, choose a temporal geodesic $\gamma^* \in C(s, g)$ and set $v^* = \dot{\gamma}^*(0) \in T_sM$. Compatibility of the exponential map gives $\mathrm{Exp}_s(tv^*) = \gamma^*(t)$ for $t \in [0, 1]$. Evaluating at $t = 0$ and $t = 1$ yields the two boundary conditions. Moreover, for $0 \le t_i \le t_j \le 1$,

$$d^*(\mathrm{Exp}_s(t_i v^*), \mathrm{Exp}_s(t_j v^*)) = d^*(\gamma^*(t_i), \gamma^*(t_j)) = (t_j - t_i) d^*(s, g).$$

This proves the constant-speed condition. □

This proposition establishes the geometric representation used by GIP. We next characterize when the temporal geodesic cost introduced in (7) exactly recovers a constant-speed shortest path structure that follows conditions in the proposition.

C.1 Temporal Geodesic Cost

We first characterize the zero level set of the temporal geodesic cost. This result provides the bridge between the regression objective in (7) and the geometric notion of a shortest path, which will be used in both the recovery and out-of-support (OOS) analyses below. The characterization applies equally to the ideal distance d^* and to a learned continuous quasimetric such as d_ϕ.

Theorem C.1 (Zero Temporal Geodesic Cost and Path Length)**.** *Let d be a continuous quasimetric on a domain containing s, g, and $\gamma(t) = E_\xi(s, tv)$. Suppose γ, ρ are continuous and the time-pair distribution has full support on $\mathcal{T}$. The temporal geodesic cost in (7), evaluated with d in place of d^*, is zero if and only if $\gamma \in C(s, g)$ and*

$$d(x_{t_i}, x_{t_j}) = (t_j - t_i) d(s, g), \qquad 0 \le t_i \le t_j \le 1, \tag{16}$$

In this case,

$$L_d(\gamma) = d(s, g). \tag{17}$$

In particular, if $d = d^*$ and $\gamma([0, 1]) \subseteq M$, then γ is a temporal geodesic.

Proof. The parameterization in (6) gives $x_0 = s$. Since both cost terms are nonnegative, zero cost implies $\rho(d(x_1, g), 0) = 0$, hence $d(x_1, g) = 0$ and $x_1 = g$. For the constant-speed term, define

$$F(t_i, t_j) := \rho\big(d(x_{t_i}, x_{t_j}), (t_j - t_i) d(s, g)\big).$$

This function is continuous and nonnegative, with zero expectation. If it were positive at any point of $\mathcal{T}$, continuity would make it positive on a relatively open neighborhood with positive sampling probability, which is a contradiction. Thus $F = 0$ throughout $\mathcal{T}$ for $t_i < t_j$. The case $t_i = t_j$ follows from $d(x, x) = 0$. Conversely, the endpoint and constant-speed conditions obviously make both terms in (7) zero, proving (16). In addition, by the definition of temporal geodesic in Assumption 4.1, satisfying (16) on $d = d^*$ makes γ a temporal geodesic. Therefore, when $d = d^*$, zero temporal geodesic cost is equivalent to γ being a temporal geodesic.

To establish the property of path length explicitly, consider an arbitrary finite partition $0 = t_0 < t_1 < \cdots < t_m = 1$. Under zero temporal geodesic cost, by (16),

$$\sum_{j=0}^{m-1} d(x_{t_j}, x_{t_{j+1}}) = \sum_{j=0}^{m-1} (t_{j+1} - t_j) d(s, g) = d(s, g).$$

Taking the supremum over all such partitions, as required by the definition of L_d, gives $L_d(\gamma) = d(s, g)$, proving (17). □

Thus, zero temporal geodesic cost evaluated on any quasimetric d enforces endpoint matching and imposes the constant-speed structure. In particular, when $d = d^*$, $L_\xi(s, g)$ in (8) consists exactly of latent path vectors whose decoded paths are temporal geodesics. This characterization will be the key step in translating vanishing population cost into path recovery.

C.2 Asymptotic Temporal Geodesic Recovery

We now lift the zero-cost characterization in Theorem C.1 to the regularized population objective in (10). We first quantify how much geodesic cost can remain at a global optimum for finite α, and then show that this residual vanishes as $\alpha \to 0$.

Define the population geodesic cost and proximity regularization by

$$F(\xi, \psi) := \mathbb{E}_{s,g\sim p_{\mathcal{D}}} \mathbb{E}_{\tilde{v}\sim q_0(\cdot|s,g)} \left[\mathcal{C}_\xi\big(\sigma_\psi(\tilde{v}, s, g); s, g\big) \right], \tag{18}$$

$$R(\psi) := \mathbb{E}_{s,g\sim p_{\mathcal{D}}} \mathbb{E}_{\tilde{v}\sim q_0(\cdot|s,g)} \left[\|\hat{\sigma}_\psi(\tilde{v}, s, g) - \tilde{v}\|_2^2 \right], \tag{19}$$

so that $\mathcal{L}_{\text{geo}}(\xi, \psi) = F(\xi, \psi) + \alpha R(\psi)$.

Theorem C.2 (Geodesic Cost Bound)**.** *Suppose there exists ξ^* such that $L_{\xi^*}(s, g)$ is nonempty for $p_{\mathcal{D}}$-almost every (s, g), and a selector σ_{ψ^*} satisfying*

$$\sigma_{\psi^*}(\tilde{v}, s, g) \in \arg\min_{v\in L_{\xi^*}(s,g)} \|\tilde{v} - \hat{v}\|_2^2$$

almost surely, where $\hat{v} = \frac{v}{\|v\|_2}$. If

$$D^* := \mathbb{E}_{s,g\sim p_{\mathcal{D}}} \mathbb{E}_{\tilde{v}\sim q_0(\cdot|s,g)} \left[\inf_{v\in L_{\xi^*}(s,g)} \|\tilde{v} - \hat{v}\|_2^2 \right] < \infty, \tag{20}$$

then every global minimizer $(\xi_\alpha, \psi_\alpha)$ of (10) satisfies

$$F(\xi_\alpha, \psi_\alpha) \le \alpha D^*. \tag{21}$$

Proof. Since $\sigma_{\psi^*}(\tilde{v}, s, g) \in L_{\xi^*}(s, g)$, the definition of L_{ξ^*} in (8) gives

$$F(\xi^*, \psi^*) = 0, \qquad R(\psi^*) = D^*.$$

By global optimality of $(\xi_\alpha, \psi_\alpha)$,

$$\begin{aligned} F(\xi_\alpha, \psi_\alpha) + \alpha R(\psi_\alpha) &\le F(\xi^*, \psi^*) + \alpha R(\psi^*) \\ &= \alpha D^*. \end{aligned} \tag{22}$$

Since $R(\psi_\alpha) \ge 0$, dropping the regularization term yields (21). □

Theorem C.2 shows that upper bound of the residual population geodesic cost is linear in α and finite D^*. In our implementation, the proximity regularizer is applied on a path direction vector $\hat{v}$ and $\tilde{v}$ as described in Appendix E.5. Since both lie on the unit sphere, their squared distance is uniformly bounded by 4, and hence $\|\hat{v} - \tilde{v}\|_2^2 \le 4$ without requiring an additional boundedness assumption. The proximity regularizer can therefore influence which low-cost path direction is selected for finite α, while its effect on geodesic optimality vanishes as $\alpha \to 0$.

We can now prove the asymptotic recovery result stated in Section 5.

Theorem 5.1 (Asymptotic Temporal Geodesic Recovery)**.** *Under Assumption 4.1, suppose E_ξ and ρ are continuous and the induced paths remain on M throughout optimization. Suppose the conditions of Theorem C.2 hold. For each $\alpha > 0$, let $(\xi_\alpha, \psi_\alpha)$ be a global minimizer of (10). If $(\xi_\alpha, \psi_\alpha) \to (\xi_0, \psi_0)$ as $\alpha \to 0$ and the population geodesic cost is continuous at the limit, then $t \mapsto E_{\xi_0}(s, t\sigma_{\psi_0}(\tilde{v}, s, g))$ is a temporal geodesic almost surely.*

Proof. By Theorem C.2,

$$0 \le F(\xi_\alpha, \psi_\alpha) \le \alpha D^*,$$

and hence $F(\xi_\alpha, \psi_\alpha) \to 0$ as $\alpha \to 0$. By the assumed convergence $(\xi_\alpha, \psi_\alpha) \to (\xi_0, \psi_0)$ and continuity of the population geodesic cost at the limit,

$$F(\xi_0, \psi_0) = 0.$$

Since the temporal geodesic cost is nonnegative,

$$\mathcal{C}_{\xi_0}\big(\sigma_{\psi_0}(\tilde{v}, s, g); s, g\big) = 0$$

for $p_{\mathcal{D}}$-almost every (s, g) and $q_0(\cdot \mid s, g)$-almost every $\tilde{v}$. Applying Theorem C.1 with $d = d^*$ then shows that

$$t \mapsto E_{\xi_0}\left(s, t\sigma_{\psi_0}(\tilde{v}, s, g)\right)$$

is a temporal geodesic. □

The result separates the roles of the two terms in (10). The temporal geodesic cost determines the limiting solution set, while the proximity regularizer biases the selection among low cost path vectors at finite α. The guarantee concerns almost-sure recovery under the training distribution and does not require the selector to represent every element of $L_\xi(s, g)$.

C.3 Out-of-Support Penalization

We next analyze the mechanism behind Theorem 5.2. The key idea is to construct a quasimetric perturbation that is identically zero between data-supported states but becomes positive when a path moves away from the data support. Such a perturbation preserves the learned geometry on supported states while increasing the length of unsupported path segments.

Let $\mathcal{X} \subseteq \mathbb{R}^{n_S}$ denote the compact ambient domain in Theorem 5.2, and let $\mathcal{S}_\mathcal{D} \subseteq M \cap \mathcal{X}$ denote the closed data-supported state domain. Two notions of path length appear in the analysis. The distance objective in (11) uses a finite chain of generated nodes, whereas Theorem 5.2 concerns the continuous path length L_d. We therefore write

$$\ell_d := \sum_{k=0}^{K-1} d(x_k, x_{k+1})$$

for the finite chain length, with $x_K = g$ as in (11). Using such a symbol, we write

$$\mathcal{L}_{\text{dist}}(d; \bar{d}) := \mathbb{E}\big[\rho\big(d(s, g), \ell_{\bar{d}}\big) - h(\ell_d)\big]\,. \tag{23}$$

This is exactly (11) evaluated with d and $\bar{d}$ in place of d_ϕ and $d_{\bar{\phi}}$, respectively. The generated nodes and $\bar{d}$ are fixed when varying d. Additionally, as stated in the main text, we require h to be strictly increasing.

Theorem 5.2 (Out-of-Support Penalization)**.** *Let $\mathcal{S}_\mathcal{D} \subseteq \mathcal{S}$ be a nonempty closed data-supported state domain, and suppose it and all generated paths lie in a common compact ambient domain. With generated paths and regression targets fixed, if $\Pr[\exists k : x_k \notin \mathcal{S}_\mathcal{D}] > 0$, there exists a continuous quasimetric $\widetilde{d}$ such that $\widetilde{d} = d_\phi$ on $\mathcal{S}_\mathcal{D} \times \mathcal{S}_\mathcal{D}$,, $\mathcal{L}_{\text{dist}}$ strictly decreases, and $L_{\widetilde{d}}(\gamma) > \widetilde{d}(s, g)$ for every out-of-support path $\gamma \in C(s, g)$ with $s, g \in \mathcal{S}_\mathcal{D}$.*

Proof. Define

$$b(x) := \inf_{z \in \mathcal{S}_\mathcal{D}} \|x - z\|_2, \qquad \widetilde{d}(x, y) := d_\phi(x, y) + \lambda |b(x) - b(y)|, \quad \lambda > 0. \tag{24}$$

Closedness of the support makes b continuous and zero exactly on $\mathcal{S}_\mathcal{D}$. The added term is nonnegative, zero on the diagonal, and satisfies the triangle inequality because

$$|b(x) - b(z)| \le |b(x) - b(y)| + |b(y) - b(z)|.$$

Since d_ϕ already separates distinct points, $\widetilde{d}$ is a continuous quasimetric. Moreover,

$$\widetilde{d}(s, g) = d_\phi(s, g), \qquad s, g \in \mathcal{S}_\mathcal{D}, \tag{25}$$

so every supported distance and every local constraint value remain unchanged, including when $\epsilon > 0$.

For a sampled chain, set $V := \sum_{k=0}^{K-1} |b(x_{k+1}) - b(x_k)|$. Because $x_0 = s$ and $x_K = g$ are supported, $b(x_0) = b(x_K) = 0$. Splitting the sum at a node maximizing $b(x_k)$ gives

$$V \ge 2 \max_{0 \le k \le K} b(x_k).$$

Thus $V > 0$ whenever a sampled node is unsupported, and this event has positive probability by assumption. Since $\ell_{\widetilde{d}} = \ell_{d_\phi} + \lambda V$, while the endpoint prediction and regression target are unchanged,

$$\begin{aligned} &\mathcal{L}_{\text{dist}}(\widetilde{d}; d_{\bar{\phi}}) - \mathcal{L}_{\text{dist}}(d_\phi; d_{\bar{\phi}}) \\ &\quad = -\mathbb{E}\big[h(\ell_{d_\phi} + \lambda V) - h(\ell_{d_\phi})\big] < 0. \end{aligned} \tag{26}$$

Continuity, compactness, and fixed K ensure finiteness of these expectations for each fixed λ.

Finally, consider any continuous OOS path $\gamma \in C(s, g)$ in $\mathcal{X}$ with supported endpoints. Choose $t_* \in (0, 1)$ such that $b(\gamma(t_*)) > 0$. By the partition $\{0, t_*, 1\}$,

$$\begin{aligned} L_{\widetilde{d}}(\gamma) &\geq \widetilde{d}(s, \gamma(t_*)) + \widetilde{d}(\gamma(t_*), g) \\ &= d_\phi(s, \gamma(t_*)) + d_\phi(\gamma(t_*), g) + 2\lambda b(\gamma(t_*)) \\ &\geq \widetilde{d}(s, g) + 2\lambda b(\gamma(t_*)) > \widetilde{d}(s, g). \end{aligned} \tag{27}$$

The same $\widetilde{d}$ therefore satisfies the required inequality for every such OOS path. □

The construction has two complementary effects. First, it strictly improves the frozen regression target distance objective whenever the sampled path contains an unsupported node. Second, it gives every continuous OOS path a length strictly larger than its endpoint distance.

Combining the second property with Theorem C.1 shows that an OOS path cannot achieve zero temporal geodesic cost under the modified quasimetric. At the same time, because all pairwise distances between supported states are unchanged, the geodesic costs of fully supported paths are preserved.

C.4 Temporal Distance Recovery

The OOS result above preserves distances between supported states but does not by itself identify them with d^*. We now show that the distance objective recovers d^* under additional coverage and stationarity assumptions. The argument is simple: the local constraints prevent overestimation of d^*, while the distance objective rules out strict underestimation.

For this subsection, let each supported transition $e = (u, w, c_e)$ have elapsed time $c_e > 0$, and let $\mathcal{Q}_\mathcal{D}$ denote the continuous quasimetrics on the ambient domain satisfying

$$d(u, w) \leq c_e$$

for every supported transition. The unit-duration setting in the main text corresponds to $c_e = 1$.

Assumption C.3 (Transition Coverage)**.** For every $s, g \in \mathcal{S}_\mathcal{D}$ and $\delta > 0$, there exists a finite chain of supported transitions from s to g whose total elapsed time is at most $d^*(s, g) + \delta$.

Under this assumption, every $d \in \mathcal{Q}_\mathcal{D}$ satisfies

$$d(s, g) \leq d^*(s, g), \qquad s, g \in \mathcal{S}_\mathcal{D}. \tag{28}$$

Indeed, for any supported chain $z_0 = s, \ldots, z_m = g$, the triangle inequality and local constraints give

$$d(s, g) \leq \sum_{j=0}^{m-1} d(z_j, z_{j+1}) \leq \sum_{j=0}^{m-1} c_{e_j}.$$

Applying Assumption C.3 and letting $\delta \to 0$ yields (28).

To rule out strict underestimation, assume ρ and h are continuously differentiable and satisfy

$$\partial_1 \rho(a, b) < 0 \quad \text{for } 0 \leq a < b, \qquad \partial_1 \rho(a, a) = 0, \qquad h'(r) > 0. \tag{29}$$

These conditions ensure that increasing an underestimated endpoint distance lowers the transitive regression term, while increasing the generated path length lowers the objective through the path length maximization term.

Because (11) uses stopped-gradient targets, we call $\widehat{d} \in \mathcal{Q}_\mathcal{D}$ a *self-consistent stationary point* if

$$\left.\frac{d}{d\eta}\right|_{\eta=0^+} \mathcal{L}_{\text{dist}}\big((1-\eta)\widehat{d} + \eta q; \widehat{d}\big) \geq 0, \qquad \forall q \in \mathcal{Q}_\mathcal{D}, \tag{30}$$

where the generated nodes and regression target are held fixed.

Theorem C.4 (Exact Temporal Distance Recovery)**.** *Suppose Assumption C.3 and (29) hold, and suppose $\mathcal{Q}_\mathcal{D}$ contains a quasimetric d^{ext} that agrees with d^* on $\mathcal{S}_\mathcal{D} \times \mathcal{S}_\mathcal{D}$. Then every self-consistent stationary point $\widehat{d} \in \mathcal{Q}_\mathcal{D}$ satisfies*

$$\widehat{d}(s, g) = d^*(s, g) \qquad p_\mathcal{D}\text{-almost surely}.$$

Proof. By (28), $\widehat{d} \le d^*$ on the supported domain. Define

$$q(x,y) := \max\{\widehat{d}(x,y), d^{\text{ext}}(x,y)\}.$$

The maximum of two quasimetrics is again a quasimetric, since

$$q(x,z) \le \max\{\widehat{d}(x,y) + \widehat{d}(y,z), d^{\text{ext}}(x,y) + d^{\text{ext}}(y,z)\} \le q(x,y) + q(y,z).$$

It also preserves all local constraints, so $q \in \mathcal{Q}_{\mathcal{D}}$. Moreover, $q \ge \widehat{d}$ everywhere and $q = d^*$ on $\mathcal{S}_{\mathcal{D}} \times \mathcal{S}_{\mathcal{D}}$.

Suppose, toward a contradiction, that $\widehat{d}(s,g) < d^*(s,g)$ on a set of positive $p_{\mathcal{D}}$-probability. Consider

$$d_\eta := (1-\eta)\widehat{d} + \eta q.$$

For each sampled path, define

$$\Delta := q(s,g) - \widehat{d}(s,g) \ge 0, \qquad B := \ell_q - \ell_{\widehat{d}} \ge 0.$$

With the regression target fixed at $\ell_{\widehat{d}}$,

$$\left.\frac{d}{d\eta}\right|_{\eta=0^+} \mathcal{L}_{\text{dist}}(d_\eta; \widehat{d}) = \mathbb{E}\Big[\partial_1 \rho(\widehat{d}(s,g), \ell_{\widehat{d}})\Delta - h'(\ell_{\widehat{d}})B\Big]. \tag{31}$$

Compactness and smoothness justify differentiation under the expectation.

Both terms are nonpositive. Whenever $\Delta > 0$, if $B > 0$, the spreading term is strictly negative. If $B = 0$, then

$$\ell_{\widehat{d}} = \ell_q \ge q(s,g) > \widehat{d}(s,g)$$

by the triangle inequality for q, so the regression term is strictly negative by (29). Thus (31) is strictly negative on a set of positive probability, contradicting (30). Therefore $\widehat{d} = d^*$ almost surely. □

The theorem does not require the generated paths used by $\mathcal{L}_{\text{dist}}$ to be shortest or dynamically feasible. Transition coverage is the key assumption that identifies the temporal scale from local constraints, while the objective eliminates remaining underestimation. If $p_{\mathcal{D}}$ has full support on $\mathcal{S}_{\mathcal{D}} \times \mathcal{S}_{\mathcal{D}}$, continuity upgrades the almost-sure equality to pointwise recovery under distance objective on the supported domain.

Combined with Theorem C.1, pointwise distance recovery implies that a fully supported decoded path with zero temporal geodesic cost is a true temporal geodesic. The result remains conditional on transition coverage, function-class realizability, and function-space stationarity, and does not by itself establish convergence of the parameterized joint optimizer.

D Extended Preliminaries

We review smooth manifolds and the spray-based definitions of geodesics, exponential maps, and logarithmic maps. We then relate these constructions to the geodesic quasimetrics introduced in the main text. Throughout this section, curve continuity and smoothness refer to the manifold topology and smooth structure, respectively.

D.1 Smooth Manifolds and Tangent Spaces

A smooth manifold $M \subseteq \mathbb{R}^{n_M}$ is a finite-dimensional, Hausdorff, second-countable topological space with dimension n_M, equipped with a smooth atlas. Each chart $\chi : U \to \chi(U) \subseteq \mathbb{R}^{n_M}$ is a homeomorphism from an open subset of M onto an open subset of $\mathbb{R}^{n_M}$, and the transition maps between overlapping charts are smooth. A curve is smooth when its coordinate representations are smooth (Kobayashi & Nomizu, 1996).

A tangent vector at $x \in M$ is an equivalence class of smooth curves through x. Two curves γ_1, γ_2, defined near 0 with $\gamma_1(0) = \gamma_2(0) = x$, represent the same tangent vector if $(\chi \circ \gamma_1)'(0) = (\chi \circ \gamma_2)'(0)$ in a chart χ around x. The chain rule makes this definition independent of the chart. These equivalence classes form the vector space T_xM, and the disjoint union of the tangent spaces forms the tangent bundle $TM = \bigsqcup_{x \in M} T_xM$.

For an arbitrary smooth curve γ on M, its velocity $\dot\gamma(t)$ is a tangent vector in $T_{\gamma(t)}M$, represented in a chart χ by $(\chi \circ \gamma)'(t)$. In particular, when $\gamma(0) = x$, the vector $\dot\gamma(0) \in T_sM$ specifies its initial velocity.

D.2 Spray Geodesics

A smooth manifold alone does not specify a geodesic equation. We introduce a *geodesic spray* to define geodesics from an initial position and velocity (Shen, 2001). A spray is a smooth vector field on $TM \setminus 0_M$, where $0_M = \{(x, 0) : x \in M\}$ is the zero section. In coordinates (x^i, v^i) induced by a chart on M, where x^i are position coordinates and v^i are tangent vector components, it has the form

$$S = v^i \frac{\partial}{\partial x^i} - 2G^i(x, v)\frac{\partial}{\partial v^i}, \qquad G^i(x, \lambda v) = \lambda^2 G^i(x, v), \quad \lambda > 0,$$

where repeated indices are summed. To include zero initial velocities, we set $G^i(x, 0) = 0$. The smoothness and degree-two homogeneity of G^i ensure that this extends S across the zero section to a C^1 vector field on all of TM. A curve γ^* is a *spray geodesic* if its coordinates satisfy

$$\ddot\gamma^{*i}(t) + 2G^i\big(\gamma^*(t), \dot\gamma^*(t)\big) = 0. \tag{32}$$

Equivalently, its tangent lift $t \mapsto (\gamma^*(t), \dot\gamma^*(t))$ is an integral curve of the extended vector field on TM.

Since the extended spray is C^1, an initial position $x \in M$ and an initial velocity $v \in T_xM$ uniquely determine a spray geodesic denoted by $\gamma^*_{x;v}$, satisfying

$$\gamma^*_{x;v}(0) = x, \qquad \dot\gamma^*_{x;v}(0) = v.$$

We consider this curve on the largest time interval containing 0 to which it can be extended while satisfying the geodesic equation.

The degree-two homogeneity of G^i ensures that, for $\lambda > 0$, the rescaled spray geodesic $t \mapsto \gamma^*_{x;v}(\lambda t)$ also satisfies (32). This curve starts at x and has initial velocity λv by the chain rule. Uniqueness therefore yields

$$\gamma^*_{x;\lambda v}(t) = \gamma^*_{x;v}(\lambda t), \tag{33}$$

whenever both sides are defined.

D.3 Exponential and Logarithmic Maps

For a given $x \in M$, the *exponential map* sends an initial velocity to the point reached by its spray geodesic at $t = 1$ (Shen, 2001). Its domain and value are defined by

$$\begin{aligned} \mathcal{U}_x &:= \{v \in T_xM : \gamma^*_{x;v} \text{ is defined for all } t \in [0, 1]\}, \\ \operatorname{Exp}_x(v) &:= \gamma^*_{x;v}(1), \qquad v \in \mathcal{U}_x. \end{aligned} \tag{34}$$

The domain $\mathcal{U}_x$ is an open neighborhood of 0 and is star-shaped, meaning that $tv \in \mathcal{U}_x$ for every $v \in \mathcal{U}_x$ and $t \in [0,1]$. By definition and (33),

$$\mathrm{Exp}_x(tv) = \gamma^*_{x;v}(t), \qquad \mathrm{Exp}_x(0) = x, \qquad \left.\frac{d}{dt}\right|_{t=0^+} \mathrm{Exp}_x(tv) = v, \tag{35}$$

for $v \in \mathcal{U}_x$ and $t \in [0,1]$. Thus, the input of the exponential map is the initial velocity of the curve it generates. The globally defined maps $\mathrm{Exp}_x : T_xM \to M$ used in the main text correspond to $\mathcal{U}_x = T_xM$, so every initial velocity must admit a solution on $\forall t \in [0,1]$.

The C^1 dependence of the geodesic solution on its initial data makes Exp_x a C^1 map. Moreover, (35) gives $\mathrm{D}\,\mathrm{Exp}_x(0) = \mathrm{Id}_{T_xM}$ under the canonical identification $T_0(T_xM) \simeq T_xM$. The inverse function theorem therefore provides neighborhoods $U_x \subseteq \mathcal{U}_x$ of 0 and $V_x \subseteq M$ of x such that $\mathrm{Exp}_x|_{U_x} : U_x \to V_x$ is a C^1 diffeomorphism. Its inverse defines the *logarithmic map on* V_x

$$\mathrm{Log}^{\mathrm{loc}}_x := (\mathrm{Exp}_x|_{U_x})^{-1} : V_x \to U_x. \tag{36}$$

For each $y \in V_x$, this map returns the unique vector $v \in U_x$ satisfying $\mathrm{Exp}_x(v) = y$. The corresponding spray geodesic reaches y, but need not minimize a given quasimetric. The superscript distinguishes it from the set-valued minimizing logarithm in (5), whose relationship to this local inverse is described below.

D.4 Compatibility with Geodesic Quasimetrics

Let d be a geodesic quasimetric on M, and let $C(x,y)$ denote the set of continuous curves from $x \in M$ to $y \in M$ on $[0,1]$. We write $\mathcal{G}(x,y) \subseteq C(x,y)$ for the set of constant-speed minimizing geodesics characterized by (3). Unlike spray geodesics, these minimizing geodesics are defined by d rather than a differential equation.

The exponential map of the given spray is compatible with d precisely when, for every $x, y \in M$, each $\gamma^* \in \mathcal{G}(x,y)$ is smooth and satisfies (32) in its constant speed parameterization. Since γ^* solves the spray equation on $t \in [0,1]$, its initial velocity $v^* = \dot\gamma^*(0)$ belongs to $\mathcal{U}_x$. Uniqueness for these initial conditions then gives

$$\gamma^*(t) = \gamma_{x;v^*}(t) = \mathrm{Exp}_x(tv^*), \qquad t \in [0,1], \tag{37}$$

which is the relation in (4). Conversely, a minimizing geodesic with this exponential representation satisfies the spray equation.

D.5 Compatible Logarithmic Map

Under compatibility with geodesic quasimetrics, the initial velocities of minimizing geodesics define the set-valued logarithmic map

$$\mathrm{Log}_x : y \mapsto \{\dot\gamma^*(0) : \gamma^* \in \mathcal{G}(x,y)\}. \tag{38}$$

Since d is a geodesic quasimetric and Exp_x is compatible with d, $\mathrm{Log}_x(y)$ is nonempty for every $x, y \in M$. This set may contain multiple vectors corresponding to different minimizing geodesics from x to y. Unlike the local inverse $\mathrm{Log}^{\mathrm{loc}}_x$, this map collects all minimizing initial velocities rather than selecting a single inverse branch.

For $v \in \mathrm{Log}_x(y)$, we define its *d-speed* by the initial forward speed of its exponential curve,

$$|v|_d := \lim_{h \to 0^+} \frac{d(x, \mathrm{Exp}_x(hv))}{h}. \tag{39}$$

The constant speed condition gives $d(x, \mathrm{Exp}_x(hv)) = h\, d(x,y)$ for $0 < h \le 1$. Hence, the limit exists and satisfies

$$|v|_d = d(x,y) = L_d(\gamma^*). \tag{40}$$

Thus, all vectors in $\mathrm{Log}_x(y)$ have the same d-speed, even when they represent different minimizing paths. Note that this speed measures forward speed along minimizing exponential curves, without presupposing a norm on the entire tangent space. Such a property of the logarithmic map motivates our design of the latent path vector described in Appendix E.5.

E Implementation Details

We provide pseudocode for GIP training in Algorithm 1 and inference in Algorithm 2, and explain implementation details below.

E.1 Path Length Maximization.

In practice, we implement the path length maximization term using a loss that enforces lower bound rather than directly maximizing an unbounded function h. Let

$$\ell_\phi = \sum_{k=0}^{K-1} d_\phi(x_k, x_{k+1})$$

denote the generated path length. We minimize a clipped surrogate

$$\frac{[\tau_{\min} d_{\bar{\phi}}(s,g) - \ell_\phi]_+}{1 + \tau_{\min} d_{\bar{\phi}}(s,g)},$$

where $\tau_{\min} > 1$ denotes a endpoint distance multiplier of lower bound and the lower-bound target is stop-gradient. Thus, the distance model is encouraged to increase the generated path length until it reaches $\tau_{\min} d_{\bar{\phi}}(s,g)$, after which no additional spreading pressure is applied.

E.2 Local Constraint.

Following QRL (Wang et al., 2023), we implement the local constraint in (12) using one-step transitions from the offline dataset. For each sampled transition (s, s'), we compute

$$\ell_{\text{loc}}(s,s') = [d_\phi(s,s') - 1]_+^2$$

and use its batch average as $\mathcal{L}_{\text{local}}$. The unit upper bound follows from measuring temporal distance in units of one dataset transition.

As in QRL, we enforce the constraint with a learnable positive Lagrange multiplier λ_{loc}. In particular, we optimize the primal-dual term

$$\mathcal{L}_{\text{primal-dual}} = \bar{\lambda}_{\text{loc}} \mathcal{L}_{\text{local}} + \lambda_{\text{loc}} \left(\epsilon^2 - \bar{\mathcal{L}}_{\text{local}}\right), \tag{41}$$

where $\bar{\cdot}$ denotes the stop-gradient. The first term updates the distance model to reduce local violations, while the second updates λ_{loc} according to whether the constraint exceeds its prescribed budget. We parameterize λ_{loc} to remain positive throughout training.

E.3 Practical Quasimetric Architecture.

In practice, we use a variant of Metric Residual Networks (MRN) (Liu et al., 2023) to represent the quasimetric distance function. Let MRN latents $x, y \in \mathbb{R}^{2K_{\text{MRN}}A}$ be divided into K_{MRN} components $x = [x^{(1)}; \cdots; x^{(K_{\text{MRN}})}]$ and $y = [y^{(1)}; \cdots; y^{(K_{\text{MRN}})}]$, where each component is further divided into $x^{(k)} = (x_{\text{a}}^{(k)}, x_{\text{s}}^{(k)})$ and $y^{(k)} = (y_{\text{a}}^{(k)}, y_{\text{s}}^{(k)})$ with $x_{\text{a}}^{(k)}, x_{\text{s}}^{(k)}, y_{\text{a}}^{(k)}, y_{\text{s}}^{(k)} \in \mathbb{R}^A$. We define

$$d_{\text{MRN}}(x,y) = \frac{1}{K} \sum_{k=1}^{K_{\text{MRN}}} \left(\max_{1 \le j \le A} \left[x_{\text{a},j}^{(k)} - y_{\text{a},j}^{(k)} \right]_+ + \left\| x_{\text{s}}^{(k)} - y_{\text{s}}^{(k)} \right\|_2 \right),$$

where $[z]_+ = \max(z, 0)$. The state space quasimetric between a state pair (s, g) is then given by

$$d_{\phi,\lambda_d}(s,g) = \lambda_d \, d_{\text{MRN}}\left(\varphi_\phi(s), \varphi_\phi(g)\right),$$

where $\varphi_\phi : \mathbb{R}^{n_S} \to \mathbb{R}^{2K_{\text{MRN}}A}$, is a latent state representation and $\lambda_d > 0$ is an adaptive distance scale. For notational simplicity, we absorb the adaptive scale λ_d into the distance parameters and write d_ϕ for d_{ϕ,λ_d}. Accordingly, a stopped distance $d_{\bar{\phi}}$ also stops gradients through λ_d.

φ_ϕ is implemented using a scale-separated MLP to decouple the geometry of the learned representation from the global magnitude of the quasimetric. Let z_ϕ denote the final hidden representation

before the output projection. We apply a non-affine layer normalization to z_ϕ and use a bias-free output projection $\widetilde{W}_\phi$ as

$$\varphi_\phi(s) = \widetilde{W}_\phi^\top \, \mathrm{LN}_0(z_\phi(s)) \,,$$

where LN_0 denotes layer normalization without learnable scale or bias. To obtain the special projection matrix $\widetilde{W}_\phi$, we first remove the mean of each output column of an output weight matrix $W_\phi \in \mathbb{R}^{p \times 2K_{\mathrm{MRN}}A}$ as

$$W_{\phi,ij}^{\mathrm{ctr}} = W_{\phi,ij} - \frac{1}{p}\sum_{r=1}^{p} W_{\phi,rj},$$

so that each column of W_ϕ^{ctr} has zero mean across its input dimensions. We then rescale W_ϕ^{ctr} as

$$\widetilde{W}_\phi = \frac{\sigma_0}{K_W \|W_\phi^{\mathrm{ctr}}\|_F} W_\phi^{\mathrm{ctr}},$$

where K_W is a normalization constant that converts the Frobenius norm into the corresponding RMS scale, and σ_0 denotes the RMS of W_ϕ^{ctr} at initialization. This rescaling keeps the global RMS of the output projection fixed at its initial value while preserving the relative structure of the learned representation. Thus, the final representation layer cannot increase the distance merely by increasing its global weight magnitude.

This scale separation is particularly important in our coupled distance–geodesic optimization. The distance model increases the lengths of generated paths through (ii) in (11), whereas the geodesic model is trained to minimize its temporal geodesic cost under the resulting distance. Although the overall objective in (13) is neither formulated as a generative adversarial network (GAN) (Goodfellow et al., 2014) objective nor as a strictly zero-sum game, the distance and geodesic models exhibit an adversarial game-like interaction: d_ϕ assigns higher costs to the undesirable generated paths, while the geodesic model adapts to reduce those costs.

Without explicit scale control, the distance encoder can respond to the spreading pressure by globally rescaling its representation, rather than by refining the relative quasimetric geometry that distinguishes supported from unsupported paths. We therefore constrain the global scale of φ_ϕ and isolate overall temporal distance calibration in the adaptive scalar λ_d. This design is motivated by the broader principle that controlling the scale and regularity of the discriminator or critic can stabilize coupled adversarial optimization (Miyato et al., 2018; Mescheder et al., 2018), while our method does not impose spectral normalization or an explicit Lipschitz constraint.

E.4 Augmented State Representation.

The learned latent state representation φ_ϕ is also used as a geometry-aware feature for E_ξ, σ_ψ, and π_θ, which are all defined on the same temporal geometry. Since the representation is directly shaped by the temporal distance learning objective, it therefore encodes state information relevant to the learned temporal geometry. Thus, we provide φ_ϕ as an additional conditioning feature to these networks.

Concretely, for each state s, we use the augmented state $[s; \varphi_\phi(s)]$ in place of s whenever a state is provided as network conditioning information. For E_ξ in (6), we retain the ambient state s as the residual anchor and implement

$$E_\xi(s, tv) = s + r_\xi([s; \varphi_\phi(s)], tv) - r_\xi([s; \varphi_\phi(s)], 0) \,,$$

which preserves $E_\xi(s, 0) = s$ by construction. Likewise, the deterministic transport defining the latent path vector distribution is conditioned on the augmented representations of both the current state and goal,

$$v = \sigma_\psi(\tilde{v}, [s; \varphi_\phi(s)], [g; \varphi_\phi(g)]) \,, \qquad \tilde{v} \sim q_0(\cdot|s, g),$$

while the local control policy is parameterized as

$$\pi_\theta(a_t \,|\, [s_t; \varphi_\phi(s_t)], \hat{v}_t) \,.$$

Although the actual models condition on augmented states, we write them as functions of the raw states only for notational convenience. This design is primarily motivated by prior work showing that

representations induced by temporal relations can be less sensitive to exogenous observation variations that do not affect the underlying goal-reaching structure (Park et al., 2026a). Accordingly, augmenting s with $\varphi_\phi(s)$ provides the downstream networks with an additional task-oriented geometric feature that may offer complementary information under nuisance perturbations. At the same time, retaining s alongside $\varphi_\phi(s)$ ensures direct access to the original ambient-state information required for path generation and local control, without imposing the stronger assumption that $\varphi_\phi(s)$ alone constitutes a sufficient or invariant representation of the state. We provide an ablation study of the choice of an input state representation in Appendix G.

E.5 Sampling Latent Path Vectors.

We implement σ_ψ by separating the direction and magnitude of the latent path vector, following the geometric interpretation of Log_s in Section 4.1 and Appendix D. For each state–goal pair (s, g), we sample $\tilde{v} \sim q_0(\cdot|s, g) = \mathrm{Unif}(\mathbb{S}^{n_T-1})$ and obtain a unit path direction

$$\hat{v} = u_\psi(\tilde{v}, s, g)\,, \qquad \|\hat{v}\|_2 = 1,$$

using the augmented state representation introduced above. We then construct

$$v = \sigma_\psi(\tilde{v}, s, g) := d_{\mathrm{MRN}}\big(\varphi_{\bar{\phi}}(s), \varphi_{\bar{\phi}}(g)\big)\,\hat{v}. \tag{42}$$

This factorization lets different reference samples select different paths between the same state and goal, while using the same magnitude for all of them. This matches the geometric view that several temporal geodesics can connect the same endpoints even though they share the same endpoint distance.

We use the raw MRN distance before applying the adaptive scale λ_d as the radius of v. Since λ_d calibrates the global magnitude of d_ϕ, excluding it from the path vector radius prevents global distance rescaling from directly changing the latent path parameterization. The distance features used in (42) are treated with stop-gradient during geodesic learning. Correspondingly, because $\tilde{v}$ and $\hat{v}$ encode directions, we apply the proximity regularizer in (10) as $\alpha\|\hat{v} - \tilde{v}\|_2^2$, so that it regularizes the choice of path direction rather than its magnitude.

E.6 Policy Training and Inference.

For policy training, we construct the local goal directly from the trajectory of the dataset rather than from the learned path model. Let T denote the final index of the trajectory containing s_t. We use

$$g_t^{\mathrm{loc}} = s_{t+\min\{H, T-t\}}, \tag{43}$$

so the target is exactly H transitions ahead whenever enough trajectory remains, which aligns with the main text. When fewer than H transitions remain, we instead use the terminal state of that trajectory. This clipping provides policy supervision for goals whose remaining horizon is shorter than the nominal subgoal horizon and avoids constructing targets beyond trajectory boundaries. The path direction used for policy conditioning is sampled for the resulting pair $(s_t, g_t^{\mathrm{loc}})$, with gradients stopped through the geometric quantities.

We additionally use a clipped version of the local progress surrogate defined in the main text. Specifically, we set

$$A_t^{\mathrm{clip}} = \min\left\{A_{\bar{\phi}}(s_t, a_t, s_{t+1}, g_t^{\mathrm{loc}}), 0\right\}.$$

Hence, transitions that achieve the desired one-step progress all receive the maximal score of zero, while only insufficient progress is penalized. This prevents unusually large estimated progress from dominating policy regression.

At inference, we make two additional implementation choices beyond the procedure described in the main text. When the learned remaining distance satisfies $d_\phi(s_t, g) < H$, we use the final goal g itself as the local target rather than decoding $E_\xi(s_t, v_t^g)$. This avoids unnecessary endpoint reconstruction error when the goal already lies within the local control horizon and mirrors the shorter-horizon examples produced by (43). Otherwise, we decode the H-step lookahead point along the sampled global path and resample a local path direction from the current state to this lookahead target. Both the global lookahead and the local direction are recomputed at every environment step.

Algorithm 1 Training GIP

Require: Offline dataset $\mathcal{D}$, reference distribution q_0, local horizon H
Ensure: Temporal distance d_ϕ, path map E_ξ, selector σ_ψ, policy π_θ
Initialize parameters $(\phi, \xi, \psi, \theta)$ and local-constraint multiplier λ_{loc}
for each gradient step **do**
  # Distance and geodesic update
  Sample $(s, s', g) \sim \mathcal{D}$, with g sampled according to the distribution described in Table 4
  Sample $\tilde{v} \sim q_0(\cdot \mid s, g)$ and compute $v = \sigma_\psi(\tilde{v}, s, g)$
  Sample temporal anchors and compute (7)
  Compute $\mathcal{L}_{\mathrm{geo}}$, $\mathcal{L}_{\mathrm{dist}}$, and $\mathcal{L}_{\mathrm{local}}$ using (10), (11), and (12)
  Update (ϕ, ξ, ψ) and λ_{loc} using the joint objective in (13) with (41)
  # Policy update
  Sample $(s_t, a_t, s_{t+1}, g_t^{\mathrm{loc}}) \sim \mathcal{D}$, where $g_t^{\mathrm{loc}} = s_{t+\min\{H, T-t\}}$
  Sample the path direction $\hat{v}_t$ for $(s_t, g_t^{\mathrm{loc}})$
  Compute the local progress score $A_{\bar{\phi}}(s_t, a_t, s_{t+1}, g_t^{\mathrm{loc}})$
  Update θ minimizing $\mathcal{L}_{\mathrm{policy}}$
end for
return $(d_\phi, E_\xi, \sigma_\psi, \pi_\theta)$

Algorithm 2 Inference with GIP

Require: Trained $(d_\phi, E_\xi, \sigma_\psi, \pi_\theta)$, goal g, subgoal horizon H
for each environment step t **do**
  Observe current state s_t
  Sample $\tilde{v}_t^g \sim q_0(\cdot \mid s_t, g)$ and compute $v_t^g = \sigma_\psi(\tilde{v}_t^g, s_t, g)$
  Compute $d_t = d_\phi(s_t, g)$
  if $d_t < H$ **then**
    Set $\tilde{g}_t = g$
  else
    Set $\tilde{g}_t = E_\xi\left(s_t, \dfrac{H}{d_t} v_t^g\right)$
  end if
  Sample $\tilde{v}_t \sim q_0(\cdot \mid s_t, \tilde{g}_t)$ and compute $v_t = \sigma_\psi(\tilde{v}_t, s_t, \tilde{g}_t)$
  Execute $a_t \sim \pi_\theta(\cdot \mid s_t, v_t / \|v_t\|_2)$
end for

F Experimental Details

F.1 OGBench Benchmark

Environments. Our main experiments are conducted on the state-based OGBench benchmark (Park et al., 2025a), covering both locomotion and manipulation. For locomotion, we evaluate on `pointmaze`, `antmaze`, and `humanoidmaze`. `pointmaze` controls a 2-D point mass and isolates long-horizon navigation from complex low-level locomotion, while `antmaze` and `humanoidmaze` additionally require controlling high-dimensional agents. The Ant has a 29-dimensional state space and an 8-dimensional action space, while the Humanoid has a 69-dimensional state space and a 21-dimensional action space. We consider the `medium`, `large`, and `giant` maze variants reported in Table 1, which require increasingly long-horizon navigation while simultaneously learning low-level locomotion from offline trajectories.

For manipulation, we evaluate on `scene` and `cube-single`. In `cube-single`, the agent manipulates a cube to reach a target configuration. The `scene` environment requires interaction with multiple objects such as a drawer, window, and buttons, and goal states can require a sequence of distinct manipulation behaviors. Together, these domains evaluate GIP across long-horizon navigation and object-centric control.

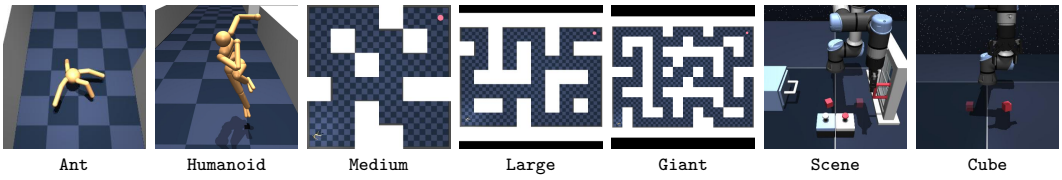


Figure 5: Visualization of the OGBench environments.

Datasets. We use the standard offline datasets provided by OGBench. For maze navigation, `navigate` contains noisy expert trajectories, while `stitch` is designed so that solving evaluation tasks requires combining information across different trajectories. The `explore` dataset instead contains exploratory behavior and provides substantially weaker goal-directed coverage. For manipulation tasks, we use the `play` and `noisy` datasets provided by OGBench. The `play` datasets are collected using open-loop, non-Markovian expert policies with temporally correlated noise, resulting in relatively natural interaction trajectories. In contrast, the `noisy` datasets are collected using closed-loop, Markovian expert policies with stronger uncorrelated Gaussian noise, which generally provides broader state coverage.

Evaluation. We evaluate goal-reaching success following the standard OGBench benchmark evaluation protocol and report the corresponding success rate. The main results in Table 1 are reported over eight random seeds. For each method, we evaluate the same environment, dataset, and goal-reaching tasks and report the mean success rate across seeds.

F.2 Baselines

We compare GIP with representative model-free, test-time planning, and trajectory-modeling approaches. The model-free baselines include GCIVL (Park et al., 2023), GCIQL (Kostrikov et al., 2022; Park et al., 2025a), CRL (Eysenbach et al., 2022), QRL (Wang et al., 2023), and HIQL (Park et al., 2023). We refer readers to the original papers for detailed descriptions of these methods.

For explicit test-time planning, we compare with TTGS (Opryshko et al., 2026) and ALPS (Shehmar et al., 2026). TTGS requires a goal-conditioned base policy, for which we use HIQL on the locomotion domains and GCIQL on the manipulation domains. We further compare with trajectory-modeling approaches Decision Diffuser (DD) (Ajay et al., 2023), Hierarchical Diffuser (HD) (Chen et al., 2024), and DiffusionVeteran (DV) (Lu et al., 2025). We omit these diffusion-based trajectory models on `stitch` and `explore`, where their whole-trajectory modeling formulation is not directly suited to the dataset structure.

For the scalability experiment on `humanoidmaze-colossal`, we additionally compare with TRL (Park et al., 2026b) and MQE (Zheng et al., 2026), which are designed to improve learning over long temporal horizons.

F.3 Per step inference FLOPs

We report the amortized inference cost in floating-point operations (FLOPs) per executed environment step under the OGBench evaluation protocol of five tasks and 50 episodes per task in Figure 1. For the model-free baselines and GIP, inference requires only feedforward network evaluations, so we omit further details. Meanwhile, TTGS and ALPS construct graphs that are reused across all 250 evaluation episodes. For a fair comparison, we divide these shared costs by the total number of executed environment steps.

Let T_i be the number of environment steps in episode i, and define $S = \sum_{i=1}^{K} T_i$, for $K = 250$, as the total number of executed environment steps. For ALPS, let C_{CEM}, C_{warm}, and C_{state} denote the per-step costs of CEM planning, the GCBC warm start, and state encoding, respectively. Since ALPS executes only the first action of each planned trajectory, the full CEM cost is incurred at every environment step. Let $C_{\mathrm{graph}}^{\mathrm{ALPS}}$ denote the cost of constructing the Laplacian embeddings, clusters, and transition graph, which are shared across all episodes, and let $C_{\mathrm{episode},i}^{\mathrm{ALPS}}$ denote the additional cost incurred once in episode i. We compute

$$C_{\mathrm{ALPS}} = C_{\mathrm{CEM}} + C_{\mathrm{warm}} + C_{\mathrm{state}} + \frac{C_{\mathrm{graph}}^{\mathrm{ALPS}} + \sum_{i=1}^{K} C_{\mathrm{episode},i}^{\mathrm{ALPS}}}{S}.$$

For TTGS, let $C_{\mathrm{graph}}^{\mathrm{TTGS}}$ denote the graph construction cost, $C_{\mathrm{connect},i}$ the cost of connecting the initial state and goal to the graph, and $C_{\mathrm{search},i}$ the shortest-path search cost in episode i. Let $C_{\mathrm{path}}(L_i)$ denote the per-step cost of processing a path of length L_i, and let C_{actor} denote the cost of one actor evaluation. We compute

$$C_{\mathrm{TTGS}} = \frac{C_{\mathrm{graph}}^{\mathrm{TTGS}} + \sum_{i=1}^{K} \left(C_{\mathrm{connect},i} + C_{\mathrm{search},i} + T_i C_{\mathrm{path}}(L_i)\right)}{S} + C_{\mathrm{actor}}.$$

For Dijkstra search, we count one floating-point addition per directed edge, such that $C_{\mathrm{search},i} = |\mathcal{E}_i|$.

F.4 Hyperparameters

For the hyperparameters we used for GIP, please refer to Table 4.

G Additional Experiments

G.1 A toy experiment: Cylinder

To visually demonstrate that the GIP objective indeed learns geodesics on the underlying manifold, we conduct a controlled experiment on a 3D cylinder. Since the geodesic distance between any two points on a cylinder can be computed in closed form, this experiment isolates geodesic learning from distance learning. We parameterize each point on the cylinder using its intrinsic 2D coordinates (θ, z), ensuring that the learned paths remain on the manifold and do not produce invalid off-manifold trajectories.

For a target lying at the cut locus of a source point—i.e., diametrically opposite to the source in the angular direction—there exist two distinct minimizing geodesics. Figure 6 visualizes the learned geodesic cost and the corresponding geodesics for cases with a unique geodesic as well as cases at the cut locus, where two minimizing geodesics exist. The learned geodesic cost correctly captures the non-uniqueness of Log_s at the cut locus. At the same time, our deterministic sampler, conditioned on a given $\tilde{v}$, selects and generates one of the valid geodesics.

G.2 Optimality Analysis

As discussed in Section 6.1, two of the strongest baselines, ALPS (Shehmar et al., 2026) and TTGS (Opryshko et al., 2026), both construct a graph from the offline dataset and perform test-time search over the resulting graph. ALPS builds this graph by applying k-means clustering in the learned representation space, whereas TTGS constructs the graph from a randomly sampled subset of states from the dataset.

To highlight the distinction between GIP and test-time planning methods that search over heuristically constructed graphs, we design a toy experiment in which the optimal path can be identified from temporal distance despite severe dataset imbalance. We construct a simple point-maze environment with two distinct routes from a start state to a goal state, and collect a navigation dataset with a 1:99 visitation ratio between the two routes. The upper route is shorter, and hence optimal, but is substantially underrepresented in the dataset relative to the rightward route, which detours (see Figure 7).

We visualize the plans produced by each method in the maze in Figure 7. Both graph-based baselines produce suboptimal detours, as their heuristically constructed graphs are dominated by the data-dense route. In contrast, GIP directly learns shortest paths with respect to the learned temporal distance, allowing it to recover the upper, optimal route despite its much lower frequency in the offline dataset.

G.3 Transition coverage analysis

Following prior work (Wang et al., 2023; Steccanella et al., 2026), GIP adopts a quasimetric architecture and learns temporal distance by combining global distance maximization with a local constraint that bounds distances across observed transitions. However, optimization schemes of this form have been reported to fail severely in manipulation domains (Kang et al., 2026).

GIP exhibits a similar limitation on `Cube`, motivating an analysis of the underlying cause (see Table 1). Both global maximization and the local constraint rely on randomly sampled state–goal pairs and observed local transitions, making sufficient transition coverage important for constraining the global distance structure. In manipulation environments such as `Cube`, demonstrations can occupy weakly overlapping regions of the state space, reducing the effectiveness of the local constraint and degrading distance learning.

To study this effect, GIP is evaluated on `cube-single-play`, which contains higher quality but lower coverage demonstrations, and `cube-single-noisy`, which provides broader transition coverage. GIP performs substantially better on the higher coverage `noisy` dataset (see Table 2), consistent with the need to learn global geometry from diverse state and goal pairs. Notably, QRL (Wang et al., 2023), which uses a related distance learning objective, exhibits consistently low performance on both datasets, whereas GIP remains strong on `noisy`. This improvement is consistent with GIP using generated paths within the distance learning objective to penalize distances in regions outside the data support, together with implicit planning for more stable goal-reaching.

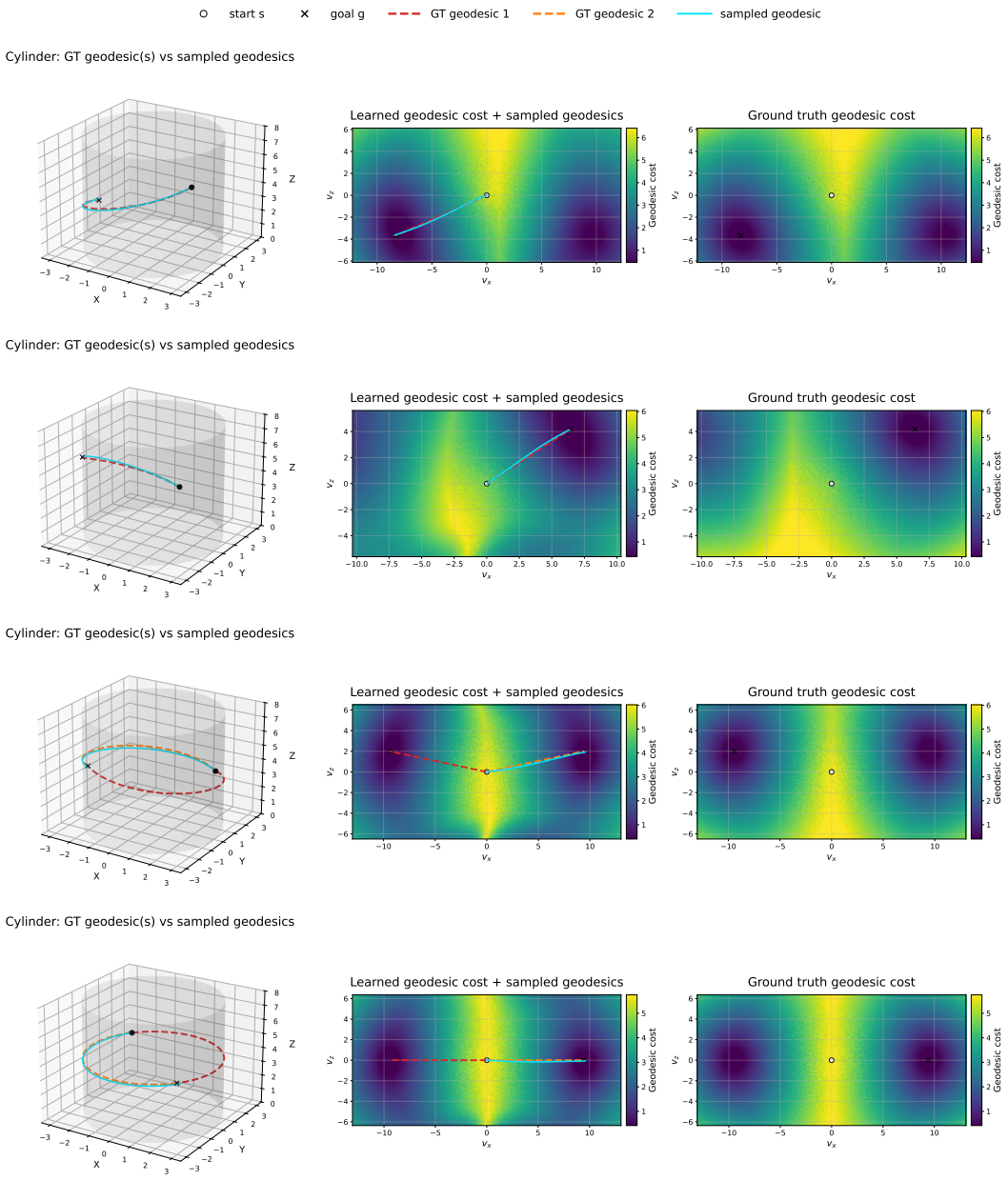


Figure 6: Qualitative results on the cylinder toy experiment. (Left) Geodesic samples reconstructed as $t \mapsto E_\xi(s, t\sigma_\psi(\tilde{v}, s, g))$. (Middle) Learned geodesic cost together with the sampled direction $v = \sigma_\psi(\tilde{v}, s, g)$. (Right) Ground-truth geodesic cost computed in closed form.

G.4 Ablations

Latent path vector dimension. In Section 4, the path map E_ξ and selector σ_ψ are defined over the ambient space $\mathbb{R}^{n_T}$, where n_T is treated as a hyperparameter corresponding to the intrinsic dimension of the underlying manifold. Accordingly, n_T determines the dimension of the latent path vector representing each path. We ablate n_T on four environments: `antmaze-large-navigate-v0`, `antmaze-giant-navigate-v0`, `humanoidmaze-large-navigate-v0`, and `humanoidmaze-giant-navigate-v0`. As shown in Figure 8, performance remains largely insensitive to n_T as long as the latent dimension is not excessively small, indicating that GIP is robust to the choice of latent path dimension.

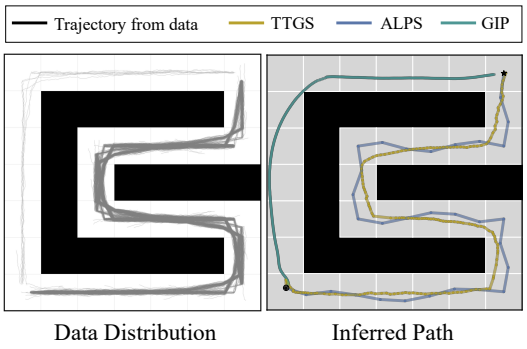


Figure 7: Qualitative results for the optimality analysis. GIP recovers optimal plans based on the temporal geodesic distance, whereas ALPS and TTGS are affected by heuristically constructed graphs, which can bias planning toward data-dense routes rather than optimal ones.

Table 2: **Results on `cube-single` datasets for the transition coverage analysis (8 seeds).** Numbers at or above 95% of the best value in each row are shown in **bold**.

			Baselines						Ours
Environment	**Dataset**	**Size**	GCBC	GCIVL	GCIQL	QRL	CRL	HIQL	GIP
`cube`	`play`	`single`	$6_{\pm 2}$	$53_{\pm 4}$	$\mathbf{68}_{\pm 6}$	$5_{\pm 1}$	$19_{\pm 2}$	$15_{\pm 3}$	$32_{\pm 5}$
	`noisy`	`single`	$8_{\pm 3}$	$71_{\pm 9}$	$\mathbf{99}_{\pm 1}$	$25_{\pm 6}$	$38_{\pm 2}$	$41_{\pm 6}$	$\mathbf{94}_{\pm 5}$

This design is motivated by the observation that, under suitable assumptions in continuous control, the intrinsic dimension of the state manifold admits an upper bound related to the action dimension (Tiwari et al., 2025). Increasing n_T enlarges the ambient space and could in principle permit paths that leave the data manifold. However, the out-of-support penalization in the distance learning objective discourages such paths, making GIP relatively insensitive to overestimating the intrinsic dimension.

State representation. In Appendix E.4, we introduce an augmented state representation for GIP. Since the tangent space is defined locally at each state and its coordinates depend on the reference state, even small perturbations to s can substantially change the path represented by a latent path vector. To improve robustness to such perturbations, we additionally anchor the path representation using $\varphi_\phi(s)$ and $\varphi_\phi(g)$, which are motivated by the noise invariant state representations from Park et al. (2026a). Figure 9 presents an ablation on `humanoidmaze-giant-navigate`, varying the state representation provided to the path map E_ξ, selector $\sigma_\psi(\cdot; s, g)$, and policy $\pi(\cdot \mid s, \hat{v})$.

The results show that the dual state representation provides a stable anchor for the coordinates of the latent path space, allowing a latent path vector to represent a consistent plan under state perturbations. However, the invariant representation can discard information required for low level control. Consequently, replacing the raw policy state with $\varphi_\phi(s)$ alone degrades performance, even though $\varphi_\phi(s)$ is beneficial for stabilizing the latent path representation.

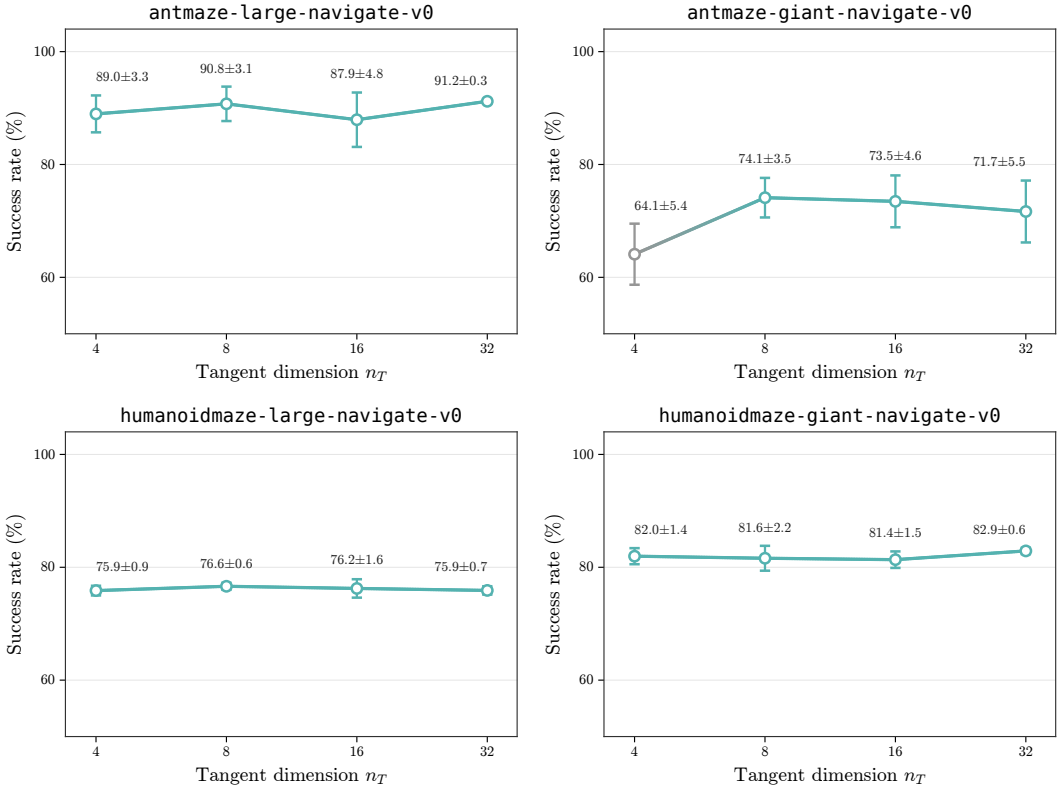


Figure 8: Ablation studies on the latent path vector dimension n_T (4 seeds).

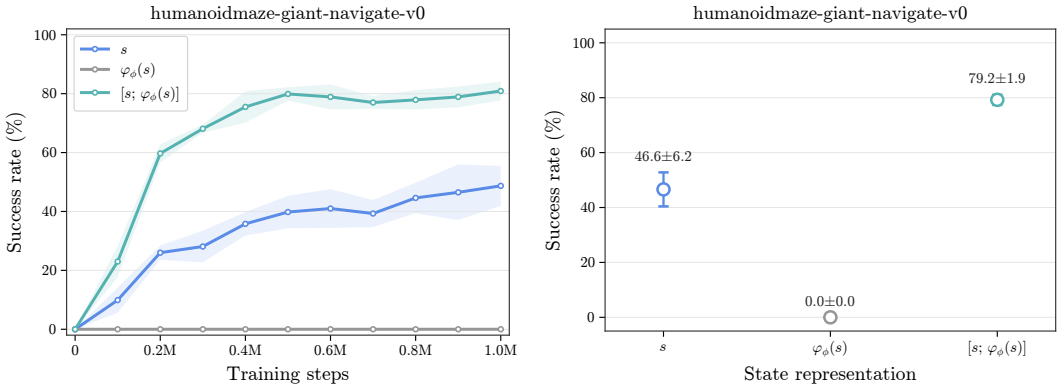


Figure 9: Ablation studies on the design choice of state representation (4 seeds).

Table 3: **Results on OGBench tasks (8 seeds).** Scores at or above 95% of the best value in each row are shown in **bold**. Averages use available entries only.

			Model-Free					Test-Time Planning		Trajectory Modeling			Ours
Environment	**Dataset**	**Size**	GCIVL	GCIQL	CRL	QRL	HIQL	TTGS	ALPS	DD	HD	DV	GIP
pointmaze	navigate	medium	$63_{\pm 6}$	$53_{\pm 8}$	$29_{\pm 7}$	$82_{\pm 5}$	$79_{\pm 5}$	$86_{\pm 5}$	$82_{\pm 10}$	–	–	–	$\mathbf{96_{\pm 3}}$
		large	$45_{\pm 5}$	$34_{\pm 3}$	$39_{\pm 7}$	$86_{\pm 9}$	$58_{\pm 5}$	$78_{\pm 10}$	$80_{\pm 8}$	–	–	–	$\mathbf{94_{\pm 2}}$
		giant	$0_{\pm 0}$	$0_{\pm 0}$	$27_{\pm 10}$	$68_{\pm 7}$	$46_{\pm 9}$	$71_{\pm 12}$	$67_{\pm 11}$	–	–	–	$\mathbf{91_{\pm 3}}$
	stitch	medium	$70_{\pm 14}$	$21_{\pm 9}$	$0_{\pm 1}$	$80_{\pm 12}$	$74_{\pm 6}$	$81_{\pm 20}$	$\mathbf{94_{\pm 6}}$	–	–	–	$\mathbf{96_{\pm 2}}$
		large	$12_{\pm 6}$	$31_{\pm 2}$	$0_{\pm 0}$	$84_{\pm 15}$	$13_{\pm 6}$	$\mathbf{92_{\pm 5}}$	$\mathbf{96_{\pm 2}}$	–	–	–	$\mathbf{95_{\pm 1}}$
		giant	$0_{\pm 0}$	$0_{\pm 0}$	$0_{\pm 0}$	$50_{\pm 8}$	$0_{\pm 0}$	$81_{\pm 9}$	$\mathbf{98_{\pm 1}}$	–	–	–	$90_{\pm 4}$
antmaze	navigate	medium	$72_{\pm 8}$	$71_{\pm 4}$	$\mathbf{95_{\pm 1}}$	$88_{\pm 3}$	$\mathbf{96_{\pm 1}}$	$\mathbf{95_{\pm 1}}$	$\mathbf{97_{\pm 2}}$	$22_{\pm 4}$	$42_{\pm 7}$	$60_{\pm 4}$	$\mathbf{96_{\pm 1}}$
		large	$16_{\pm 5}$	$34_{\pm 4}$	$83_{\pm 4}$	$75_{\pm 6}$	$\mathbf{91_{\pm 2}}$	$\mathbf{92_{\pm 2}}$	$\mathbf{93_{\pm 5}}$	$14_{\pm 4}$	$31_{\pm 5}$	$51_{\pm 5}$	$\mathbf{91_{\pm 3}}$
		giant	$0_{\pm 0}$	$0_{\pm 0}$	$16_{\pm 3}$	$14_{\pm 3}$	$65_{\pm 5}$	$66_{\pm 4}$	$69_{\pm 9}$	$5_{\pm 2}$	$19_{\pm 4}$	$35_{\pm 4}$	$\mathbf{74_{\pm 4}}$
	stitch	medium	$44_{\pm 6}$	$29_{\pm 6}$	$53_{\pm 6}$	$59_{\pm 7}$	$\mathbf{94_{\pm 1}}$	$\mathbf{95_{\pm 1}}$	$\mathbf{93_{\pm 7}}$	–	–	–	$86_{\pm 6}$
		large	$18_{\pm 2}$	$7_{\pm 2}$	$11_{\pm 2}$	$18_{\pm 2}$	$67_{\pm 5}$	$\mathbf{91_{\pm 2}}$	$\mathbf{95_{\pm 2}}$	–	–	–	$78_{\pm 3}$
		giant	$0_{\pm 0}$	$0_{\pm 0}$	$0_{\pm 0}$	$0_{\pm 0}$	$2_{\pm 2}$	$79_{\pm 13}$	$\mathbf{92_{\pm 3}}$	–	–	–	$77_{\pm 6}$
	explore	medium	$19_{\pm 3}$	$13_{\pm 2}$	$3_{\pm 2}$	$1_{\pm 1}$	$37_{\pm 10}$	$94_{\pm 2}$	$\mathbf{100_{\pm 0}}$	–	–	–	$87_{\pm 4}$
		large	$10_{\pm 3}$	$0_{\pm 0}$	$0_{\pm 0}$	$0_{\pm 0}$	$4_{\pm 5}$	$27_{\pm 34}$	$\mathbf{90_{\pm 15}}$	–	–	–	$71_{\pm 5}$
humanoidmaze	navigate	medium	$24_{\pm 2}$	$27_{\pm 2}$	$60_{\pm 4}$	$21_{\pm 8}$	$\mathbf{89_{\pm 2}}$	$\mathbf{93_{\pm 3}}$	$\mathbf{89_{\pm 5}}$	$9_{\pm 3}$	$24_{\pm 4}$	$38_{\pm 5}$	$\mathbf{92_{\pm 1}}$
		large	$2_{\pm 1}$	$2_{\pm 1}$	$24_{\pm 4}$	$5_{\pm 1}$	$49_{\pm 4}$	$\mathbf{79_{\pm 6}}$	$56_{\pm 5}$	$5_{\pm 2}$	$16_{\pm 3}$	$29_{\pm 3}$	$\mathbf{76_{\pm 2}}$
		giant	$0_{\pm 0}$	$0_{\pm 0}$	$3_{\pm 2}$	$1_{\pm 0}$	$12_{\pm 4}$	$\mathbf{85_{\pm 6}}$	$67_{\pm 11}$	$1_{\pm 1}$	$7_{\pm 3}$	$16_{\pm 3}$	$\mathbf{81_{\pm 1}}$
	stitch	medium	$12_{\pm 2}$	$12_{\pm 3}$	$36_{\pm 2}$	$18_{\pm 2}$	$\mathbf{88_{\pm 2}}$	$85_{\pm 3}$	$68_{\pm 5}$	–	–	–	$\mathbf{91_{\pm 2}}$
		large	$1_{\pm 1}$	$0_{\pm 0}$	$4_{\pm 1}$	$3_{\pm 1}$	$28_{\pm 3}$	$\mathbf{65_{\pm 11}}$	$39_{\pm 6}$	–	–	–	$\mathbf{67_{\pm 2}}$
		giant	$0_{\pm 0}$	$0_{\pm 0}$	$0_{\pm 0}$	$0_{\pm 0}$	$3_{\pm 2}$	$\mathbf{78_{\pm 5}}$	$62_{\pm 6}$	–	–	–	$\mathbf{81_{\pm 5}}$
scene	play		$42_{\pm 4}$	$51_{\pm 4}$	$19_{\pm 2}$	$5_{\pm 1}$	$38_{\pm 3}$	$52_{\pm 4}$	$26_{\pm 3}$	$16_{\pm 4}$	$37_{\pm 6}$	$56_{\pm 3}$	$\mathbf{65_{\pm 12}}$
cube	play	single	$53_{\pm 4}$	$\mathbf{68_{\pm 6}}$	$19_{\pm 2}$	$5_{\pm 1}$	$15_{\pm 3}$	$34_{\pm 3}$	$\mathbf{68_{\pm 6}}$	$8_{\pm 3}$	$20_{\pm 4}$	$32_{\pm 5}$	$32_{\pm 5}$
Average			22.9	20.6	23.7	34.7	47.6	77.2	**78.2**	10.0	24.5	39.6	**82.1**

G.5 Full Experiments

For conciseness, Table 1 focuses on the more challenging OGBench settings and omits the relatively easier `pointmaze` and `antmaze-medium` variants. Table 3 provides the complete results for all OGBench environments evaluated in this work (Park et al., 2025a). In both Table 1 and Table 3, the results for GCIVL, GCIQL, CRL, QRL, and HIQL are taken from the original OGBench paper (Park et al., 2025a). For TTGS, we evaluate `antmaze-medium-explore` and `cube-single-play` ourselves, while the results for the remaining environments are taken from the original TTGS paper (Opryshko et al., 2026). The ALPS results are taken from Shehmar et al. (2026), and the results for the trajectory modeling baselines are taken from Gireesh et al. (2026).

G.6 Additional visualization

Figure 10 visualizes the plans generated by GIP from the start state to the goal for five tasks in each OGBench `humanoidmaze-navigate` environment. Specifically, we visualize the planned trajectory $t \mapsto E_\xi(s,\, t\,\sigma_\psi(\tilde{v}, s, g))$. Note that GIP reconstructs the full 69-dimensional state along the trajectory, while we visualize only the xy-coordinates for clarity. We also illustrate some failure cases that traverse the out-of-support regions, due to the practical training errors and sampling budget.

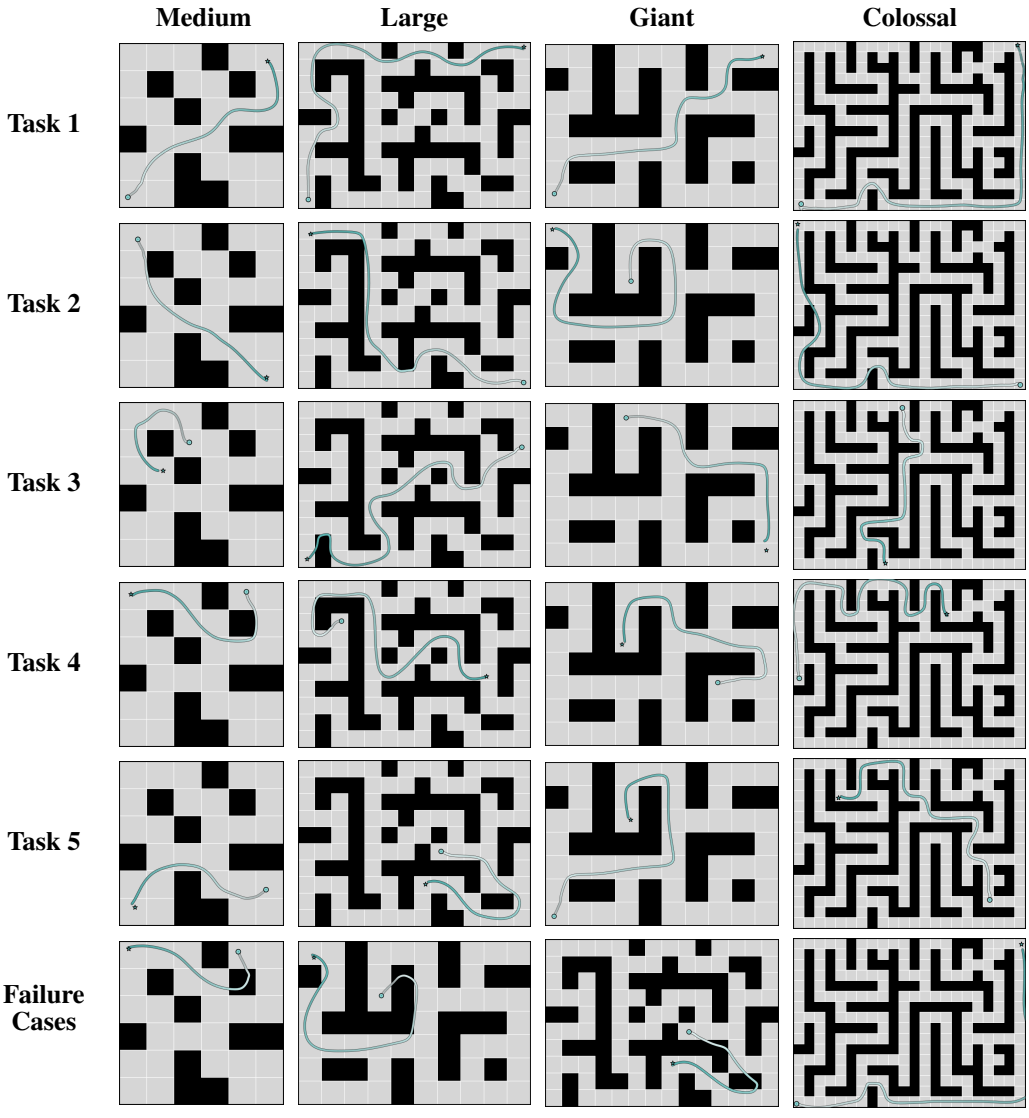


Figure 10: Additional qualitative visualizations across five evaluation tasks and failure cases. Columns correspond to `medium`, `large`, `giant`, and `colossal`.

Table 4: Hyperparameters used for GIP.

Hyperparameter	**Value**
Batch size	`1024`
Path-map and selector learning rate	`3e-4`
Distance learning rate	`1e-4`
Policy learning rate	`3e-4`
MLP hidden dimensions (φ_ϕ, r_ξ, u_ψ, and π_θ)	`(512, 512, 512)`
Nonlinearity	`GELU` (Hendrycks & Gimpel, 2016)
Layer normalization (Ba et al., 2016)	`True`
Observation normalization	`True`
MRN components	`8`
Number of sampled intermediate times K	`4`
Regression loss ρ	`huber loss` (Huber, 1964)
Huber threshold	`0.1`
Transitive regression coefficient	`1.0`
Path length maximization coefficient	`1.0`
Local-constraint budget ϵ^2	`0.05`
Proximity regularizer coefficient α	`0.2`
Proximity-regularizer annealing steps	`5e5`

Policy-weighting temperature β	`0.3 (scene)`
	`0.3 (cube-single)`
	`0.3 (antmaze-explore)`
	`1.0 (antmaze-navigate/stitch)`
	`1.0 (pointmaze, humanoidmaze)`

Subgoal steps H	`50  (scene)`
	`25  (pointmaze, antmaze, cube)`
	`100 (humanoidmaze)`

State representation dimension $\dim(\varphi_\phi)$	`128 (pointmaze, antmaze, cube)` `512 (humanoidmaze, scene)`
Latent path vector dimension n_T	`8   (pointmaze, antmaze, scene, cube)` `16  (humanoidmaze)`
Goal sampling: current-state probability p_{cur}	`0.0`
Goal sampling: trajectory-future probability p_{traj}	`0.0 (maze)` `0.625 (scene, cube)`
Goal sampling: random probability p_{rand}	`1.0 (maze)` `0.375 (scene, cube)`
Geometric goal sampling	`False`