Physics-informed Value Learner for Offline Goal-Conditioned Reinforcement Learning

Vittorio Giammarino, Ruiqi Ni and Ahmed H. Qureshi
Department of Computer Science
Purdue University
{vgiammar,ni117,ahqureshi}@purdue.edu

Abstract

Offline Goal-Conditioned Reinforcement Learning (GCRL) holds great promise for domains such as autonomous navigation and locomotion, where collecting interactive data is costly and unsafe. However, it remains challenging in practice due to the need to learn from datasets with limited coverage of the state-action space and to generalize across long-horizon tasks. To improve on these challenges, we propose a *Physics-informed (Pi)* regularized loss for value learning, derived from the Eikonal Partial Differential Equation (PDE) and which induces a geometric inductive bias in the learned value function. Unlike generic gradient penalties that are primarily used to stabilize training, our formulation is grounded in continuous-time optimal control and encourages value functions to align with cost-to-go structures. The proposed regularizer is broadly compatible with temporal-difference-based value learning and can be integrated into existing Offline GCRL algorithms. When combined with Hierarchical Implicit Q-Learning (HIQL), the resulting method, Eikonal-regularized HIQL (Eik-HIQL), yields significant improvements in both performance and generalization, with pronounced gains in stitching regimes and large-scale navigation tasks. Code is available at link[1].

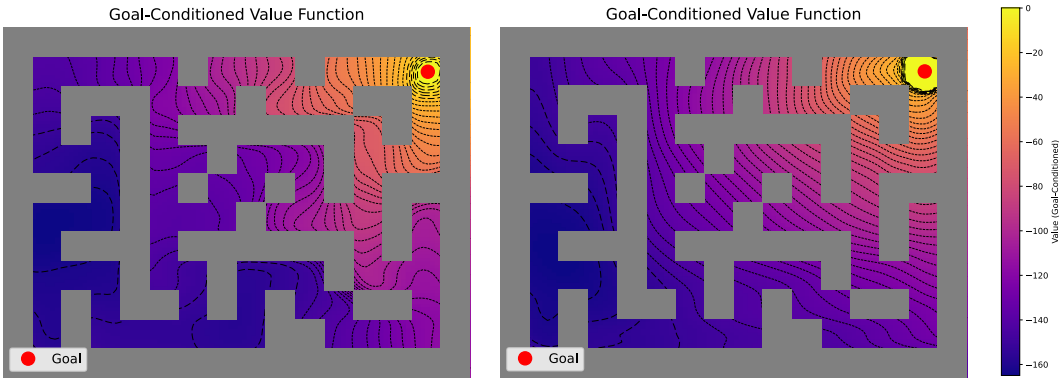
1 Introduction

In recent years, many of the most effective machine learning paradigms have capitalized on vast amounts of unlabeled or weakly labeled data. Similarly, in dynamic systems learning, Offline Goal-Conditioned Reinforcement Learning (GCRL) has emerged as a pivotal framework, enabling the use of large-scale, multitask datasets without requiring explicit reward annotations. Specifically, Offline RL [1, 2] leverages passively collected trajectories to learn control policies, offering great promise for applications such as autonomous navigation, locomotion, and manipulation, where interactive training is usually costly and unsafe. GCRL [3, 4] extends this capability by enabling learning across diverse datasets without explicit rewards. Despite its potential, Offline GCRL faces significant challenges, including accurate Goal-Conditioned Value Function (GCVF) estimation from limited data, policy extraction from imperfect value functions, and generalization to unseen state-goal pairs [5]. Among these issues, GCVF estimation remains the most fundamental, as improvements in this area can enhance both policy extraction and generalization, ultimately advancing the entire field of GCRL.

Physics-informed (Pi) inductive biases, defined as structural constraints grounded in physical laws such as symmetry, conservation principles, or consistency with Partial Differential Equations (PDEs), provide a promising direction for enhancing GCVF estimation in the offline setting. As demonstrated in prior work [6], Pi methods can introduce physically or geometrically meaningful structure into the learned value function, enhancing both sample efficiency and generalization. In Fig. 1, we illustrate a

[1] https://github.com/VittorioGiammarino/Eik-HIQL

39th Conference on Neural Information Processing Systems (NeurIPS 2025).



(a) Eik-HIQL (ours) (b) HIQL [10]

Figure 1: Contour plots of the GCVF for `antmaze-giant-navigate-v0` in [11], learned after 100,000 training steps by our Physics-informed algorithm **Eik-HIQL**, and the standard **HIQL**. The plots are generated by varying the agent's center of mass x-y coordinates while keeping all other states fixed. Recall that the policy π is trained to move the agent in the direction that maximizes the GCVF. The effects of the Eikonal regularizer are evident in Fig. 1a, where the contour plot closely follows the maze structure, in contrast to Fig. 1b, where the learned GCVF ignores the maze structure.

representative GCRL task in which an agent must navigate from various starting positions in a maze to a specified goal. Fig. 1b shows the contour plot of a GCVF learned by a non-Pi state-of-the-art (SOTA) algorithm. The resulting value function fails to robustly encode obstacle constraints, leading to suboptimal policies that often fail to reach the goal. These limitations motivate the use of Pi regularizers as a principled means to incorporate structural priors into value learning, and thus improve GCVF estimation in complex environments.

The primary contribution of this work is the introduction of an Eikonal regularizer for GCVF estimation in Offline GCRL tasks. Inspired by the Eikonal PDE [7], this regularizer imposes a distance-like cost-to-go structure on the learned GCVF, serving as an effective inductive bias during training. By enforcing this structure, the regularizer improves value estimation accuracy and promotes generalization to unseen states, while also reducing the number of required training steps compared to non-Pi approaches (see Fig. 1a). In contrast to Hamilton-Jacobi-Bellman (HJB) PDE-based methods [6], which require explicit system dynamics and often suffer from numerical instability [8, 9], our method is model-free and easy to implement. Empirically, it outperforms both HJB-regularized and unregularized baselines while adding only minimal computational overhead.

To validate the effectiveness of our Eikonal regularizer, we integrate it into the Hierarchical Implicit Q-Learning (HIQL) framework [10], a SOTA algorithm for Offline GCRL. We refer to this variant as Eik-HIQL. This choice is motivated by HIQL's strong baseline performance, making it an ideal candidate to highlight the benefits of our approach. Importantly, the Eikonal regularizer is broadly compatible with other temporal-difference-based algorithms, as we further demonstrate empirically in Appendix F. Our evaluation, conducted on the challenging OGbench benchmark [11], compares Eik-HIQL against Quasimetric RL (QRL) [12], Contrastive RL (CRL) [13], and the standard HIQL baseline. Eik-HIQL consistently outperforms or matches the baselines, achieving SOTA results in large-scale navigation and trajectory stitching scenarios. These gains underscore the utility of the Eikonal regularizer in enhancing GCVF estimation and overall Offline GCRL performance, with limited exceptions in tasks involving complex object interactions.

2 Related work

To the best of our knowledge, Pi regularization for value estimation has only recently been explored. Notably, Lien et al. [6] propose an Offline RL objective derived from the HJB equation in continuous-time optimal control [8, 9], aiming to enforce first-order derivative consistency within the critic network. In contrast, we introduce a simpler, model-free Pi regularizer for GCVF learning, based on the residual of the Eikonal PDE. We show that this regularizer induces a distance-like structure in the value function and integrates naturally into standard temporal-difference-based GCRL pipelines.

While gradient norm penalties, closely related to the Eikonal PDE residual, have been employed in generative modeling [14] and, more recently, in model-based RL to regularize Q-functions and mitigate overfitting [15], their application in GCRL remains, to our knowledge, unexplored. In contrast to these prior methods, which primarily aim to stabilize training, our regularizer is designed to inject a structural inductive bias into the GCVF, thereby improving sample efficiency and generalization. To the best of our knowledge, this work presents the first use of the Eikonal PDE as a regularization objective in value-based RL, and its first practical deployment in the Offline GCRL setting. More broadly, there has been a growing interest in incorporating physical priors and geometric, distance-like structures into RL algorithms, especially in model-based or Koopman-inspired frameworks [16, 17]. These approaches typically leverage the Koopman operator, which assumes access to (or approximations of) the underlying system dynamics, and may incorporate structural information such as reversibility or symmetry of the dynamics. In contrast, our method operates directly on the GCVF in a fully model-free setting.

In parallel, the use of non-Pi constraints for value learning has been extensively studied in Offline RL. Many approaches constrain learned policies to remain close to the behavior policy, either through explicit density modeling [18–20] or implicit divergence constraints [21, 22]. Others directly regularize the Q-function to assign low values to out-of-distribution actions and improve robustness [23, 24].

Our work is also closely related to the literature on GCRL [3, 4]. Eik-HIQL extends HIQL [10] by integrating the Eikonal regularizer into the GCVF estimation loss. HIQL itself combines hierarchical actors [25, 26] with Implicit Q-Learning [23]. Other GCRL methods include hindsight relabeling [27], contrastive representation learning [13], state-occupancy matching [28], and quasimetric RL [12]. Offline GCRL has also been studied through the lens of goal-conditioned supervised learning (GCSL) [29, 30], where goal-reaching policies are trained via conditional imitation or regression. Recent work has analyzed the out-of-distribution goal generalization problem [31] and addressed GCSL via self-supervised reward shaping [32] and occupancy-based score modeling [33].

Beyond RL, Pi losses and neural networks (NNs) have been widely applied to learn parameterizations of PDEs such as Burgers, Schrödinger, and Navier–Stokes equations [34, 35]. These methods leverage automatic differentiation to estimate derivatives with respect to NN inputs and solve high-dimensional PDEs. The Eikonal PDE, in particular, has been used in seismology [36] and motion planning [37, 38], where distance fields provide essential geometric structure. In this work, we extend the use of the Eikonal PDE to the Offline GCRL domain, demonstrating how its distance-preserving properties enhance GCVF estimation, especially in large environments and when data stitching is required.

3 Preliminaries

Offline GCRL We model the decision process as a finite-horizon discounted Markov Decision Process (MDP) described by the tuple $(\mathcal{S}, \mathcal{G}, \mathcal{A}, \mathcal{T}, \mathcal{R}, \mathcal{P}_g, \rho_0, \gamma)$ where $\mathcal{S}$ is the set of states, $\mathcal{G}$ is the set of goals, $\mathcal{A}$ is the set of actions, $\mathcal{T} : \mathcal{S} \times \mathcal{A} \to \mathcal{P}(\mathcal{S})$ is the transition probability function where $\mathcal{P}(\mathcal{S})$ denotes the space of probability distributions over $\mathcal{S}$, $\mathcal{R} : \mathcal{S} \times \mathcal{G} \to \mathbb{R}$ is a goal-conditioned reward function defined $\mathcal{R}(s, g) = -1$ when $s \neq g$ and $\mathcal{R}(s, g) = 0$ otherwise, $\mathcal{P}_g \in \mathcal{P}(\mathcal{G})$ is the goal distribution, $\rho_0 \in \mathcal{P}(\mathcal{S})$ is the initial state distribution and $\gamma \in (0, 1]$ is the discount factor. In this work, we assume the goal space $\mathcal{G}$ to be equivalent to the state space, i.e., $\mathcal{G} = \mathcal{S}$, and the goal g is sampled according to $\mathcal{P}_g$ ($g \sim \mathcal{P}_g$) at the beginning of each episode. The learning agent's objective is to maximize the expected sum of discounted rewards $\mathcal{R}(s_t, g)$ to successfully reach g. Formally, this objective is expressed as $J(\pi) = \mathbb{E}_{\tau_\pi(g)}[\sum_{t=0}^{T} \gamma^t \mathcal{R}(s_t, g)]$ where $\tau_\pi(g) = (g, s_0, a_0, s_1, a_1, \ldots, s_T)$ and $\pi : \mathcal{S} \times \mathcal{G} \to \mathcal{P}(\mathcal{A})$. Additionally, we define the GCVF induced by the policy π as $V^\pi(s, g) = \mathbb{E}_{\tau_\pi(g)}[\sum_{t=0}^{T} \gamma^t \mathcal{R}(s_t, a_t, g) | S_0 = s, G = g]$ and the goal-conditioned state-action value function as $Q^\pi(s, a, g) = \mathbb{E}_{\tau_\pi(g)}[\sum_{t=0}^{T} \gamma^t \mathcal{R}(s_t, a_t, g) | S_0 = s, A_0 = a, G = g]$. In the offline setting, the learning agent must optimize $J(\pi)$ using only a static, offline dataset $\mathcal{D}$, which comprises trajectories of the form $\tau = (s_0, a_0, s_1, s_2, \ldots, s_T)$. Finally, note that we write $\pi_{\boldsymbol{\theta}}$ when a function is parameterized with parameters $\boldsymbol{\theta} \in \Theta \subset \mathbb{R}^k$.

Hierarchical IQL Our algorithm, Eik-HIQL, extends the Offline GCRL algorithm HIQL, as introduced by Park et al. [10]. HIQL incorporates two key components: a GCVF estimation process that is robust to out-of-distribution actions and a hierarchical actor. The hierarchical actor comprises a high-level policy, $\pi^{hi}_{\boldsymbol{\theta}_{hi}} : \mathcal{S} \times \mathcal{G} \to \mathcal{S}$, which predicts subgoals, and a low-level policy, $\pi^{lo}_{\boldsymbol{\theta}_{lo}} : \mathcal{S} \times \mathcal{S} \to$

$\mathcal{P}(\mathcal{A})$, which generates actions to achieve those subgoals. Robustness in the GCVF estimation step is enabled by an action-free variant of implicit Q-learning [23], inspired by [39, 40]:

$$\mathcal{L}_V(\boldsymbol{\theta}_V) = \mathbb{E}_{(s,s')\sim\mathcal{D},g\sim\mathcal{P}_g}\Big[L_2^\iota\big(\mathcal{R}(s,g) + \gamma V_{\bar{\boldsymbol{\theta}}_V}(s',g) - V_{\boldsymbol{\theta}_V}(s,g)\big)\Big], \tag{1}$$

where $\bar{\boldsymbol{\theta}}_V$ denotes the parameters of the target GCVF network [41] and $L_2^\iota(\cdot)$ represents the expectile loss function with $\iota \in [0.5, 1]$: $L_2^\iota(x) = |\iota - \mathbb{1}(x < 0)|x^2$. In Eq. (1), the expectile regression induced by $L_2^\iota(\cdot)$ replaces the $\max$ operator in the Bellman equation [42] with the goal of avoiding queries of out-of-distributions actions. The ability to properly handle overestimated values for out-of-distribution actions is a critical challenge in Offline RL, since, unlike in online RL, erroneous estimates cannot be corrected through environment interactions. The estimated GCVF is subsequently used to train the hierarchical actor where $\pi^{hi}_{\boldsymbol{\theta}_{hi}}(s_{t+k}|s_t, g)$ and $\pi^{lo}_{\boldsymbol{\theta}_{lo}}(a|s_t, s_{t+k})$ aim to maximize $V_{\boldsymbol{\theta}_V}(s_{t+k}, g)$ and $V_{\boldsymbol{\theta}_V}(s_{t+1}, s_{t+k})$, respectively. It is shown in Park et al. [10] that this hierarchy, compared to the flat formulation using a single policy $\pi_{\boldsymbol{\theta}}(a|s_t, g)$, can better address low signal-to-noise ratios in the estimated GCVF.

The Eikonal equation The Eikonal equation is a non-linear first-order PDE that describes wave propagation in heterogeneous media [7]. It is expressed in its general form as:

$$||\nabla_s T(s,g)||^2 = \frac{1}{S(s)^2}, \tag{2}$$

where $||\cdot||$ denotes the Euclidean norm, $T : \mathcal{S} \times \mathcal{G} \to \mathbb{R}$ represents the travel-time through the medium from the state s to a goal location g and $\nabla_s T(s,g)$ is the partial derivative of the travel-time T with respect to s. The function $S : \mathcal{S} \to \mathbb{R}$ defines the speed profile of the medium in the state location s. As described in [37], higher values for $S(s)$ lead to a low travel-time $T(s,g)$ from s to g and therefore to preferable paths $(s_0, s_1 \ldots, s_T)$ compared to those with lower $S(s)$. Consequently, the solution to the Eikonal PDE in (2), represented by the travel-time $T(s,g)$, reflects a cost-to-go structure. Minimizing $T(s,g)$ yields the shortest travel time from s to g, as determined by the speed profile $S(s)$. In other words, $T(s,g)$ encodes a GCVF for a specific class of optimal control problems. We formally establish this connection in the next section. In our method, we propose a regularizer, inspired by the Eikonal PDE in (2), with the goal of providing an additional distance-like structure to the learned GCVF. Under smooth dynamics assumptions, we demonstrate that our regularizer significantly improves performance over the SOTA baselines while adding minimal complexity.

4 Physics-informed Eikonal regularizer

In this section, we first relate the Eikonal PDE in (2) to the HJB equation [8, 9] for continuous-time, undiscounted optimal control. This connection draws parallels to prior studies such as Lien et al. [6], offering additional motivation for our regularizer and insights into its empirical effectiveness. We then introduce the Eikonal-regularized loss for GCVF learning, which, when combined with a hierarchical actor, forms the core of our Eik-HIQL algorithm. Finally, we summarize the main components of Eik-HIQL.

Optimal control perspective on the Eikonal PDE We start our analysis by considering the following continuous-time dynamical system:

$$\dot{s}(t) = f(s(t), a(t)), \quad t \geq 0,$$

where $s(t)$ denotes the state of the system at time t, $\dot{s}(t)$ its derivative and $a(t)$ the control action. The function $f(\cdot)$ represents the system dynamics, determining how the state evolves in response to the control action. As common in the literature, we assume $f(\cdot)$ to be a Lipschitz continuous function [43]. Given the initial conditions $s(0) = s_0$, $a(0) = a_0$ and the goal $s(T) = g$, the undiscounted optimal control problem seeks to minimize

$$J = \int_0^T c(s(t), a(t))dt, \tag{3}$$

where $c(s(t), a(t))$ is the instantaneous cost function. The optimal value function $V(s,g)$ associated with (3) is defined as

$$V(s,g) = \inf_{a(\cdot)} \int_0^T c(s(t), a(t))dt,$$

and satisfies the principle of optimality

$$V(s,g) = \inf_{a\in\mathcal{A}}[c(s,a)\Delta t + V(s(t+\Delta t),g)], \tag{4}$$

where Δt is a small time step. Note that, unless otherwise specified, throughout this section we keep the standard continuous-time optimal control theory notation, where $V(s,g)$ is used in place of $V^*(s,g)$ to denote the optimal value function [44].

Approximating $V(s(t+\Delta t),g)$ in (4) with its first-order Taylor expansion, $V(s(t+\Delta t),g) = V(s,g) + \nabla_s V(s,g)^\intercal f(s,a)\Delta t + O(\Delta t^2)$, and taking the limit $\Delta t \to 0$, we derive the following HJB equation:

$$\inf_{a\in\mathcal{A}}[c(s,a) + \nabla_s V(s,g)^\intercal f(s,a)] = 0, \tag{5}$$

which holds true at optimality. Refer to Appendix C for the step-by-step derivations. The HJB equation in (5) encodes the relationship between the system dynamics, the cost function, and the value function; where the left-hand side $H(s,g,\nabla_s V(s,g)) \equiv \inf_{a\in\mathcal{A}}[c(s,a) + \nabla_s V(s,g)^\intercal f(s,a)]$ is referred to as the Hamiltonian. The following proposition establishes the connection between the Hamiltonian and the Eikonal PDE in (2), highlighting their equivalence under specific conditions.

Proposition 4.1. *Given the Hamiltonian $H(s,g,\nabla_s V(s,g))$, the following inequality holds*

$$H(s,g,\nabla_s V(s,g)) \equiv \inf_{a\in\mathcal{A}}[c(s,a) + \nabla_s V(s,g)^\intercal f(s,a)] \leq c^*(s) + ||\nabla_s V(s,g)||F^*(s), \tag{6}$$

where $c^*(s) = \inf_{a\in\mathcal{A}} c(s,a)$ and $F^*(s) = \sup_{a\in\mathcal{A}} ||f(s,a)||$. In the special case in which $f(s,a) = a$, $||a|| = 1$ and $c(s,a)$ is constant over $||a|| = 1$ the Hamiltonian simplifies to

$$H(s,g,\nabla_s V(s,g)) = c^*(s) - ||\nabla_s V(s,g)||. \tag{7}$$

Proof. The inequality in (6) follows from using the Cauchy-Schwarz inequality on the inner product $\nabla_s V(s,g)^\intercal f(s,a)$ in (5). Using the definitions $F^*(s) = \sup_{a\in\mathcal{A}} ||f(s,a)||$ and $c^*(s) = \inf_{a\in\mathcal{A}} c(s,a)$, we obtain the upper bound in (6). For the equality in (7), when $c(s,a)$ is constant over $||a|| = 1$, the inner product $\nabla_s V(s,g)^\intercal f(s,a)$ attains its minimal value when a points in the direction opposite to $\nabla_s V(s,g)$. Specifically, this occurs when $a^* = \arg\inf_{||a||=1} \nabla_s V(s,g)^\intercal f(s,a) = -\nabla_s V(s,g)/||\nabla_s V(s,g)||$. Substituting $f(s,a) = a^*$ into the Hamiltonian in (6) and simplifying yields the result in (7). Refer to Appendix C for the full proof. □

Remark 4.2 (Connection between HJB and Eikonal residuals). Proposition 4.1 shows that, even without assumptions on the dynamics, the Hamiltonian $H(s,g,\nabla_s V(s,g))$ is upper-bounded by an Eikonal-like residual, where the ratio $F^*(s)/c^*(s)$ defines a local speed profile $S(s)$ as in (2). In the special case of isotropic dynamics with $f(s,a) = a$, $\|a\| = 1$, and constant cost, the Hamiltonian reduces exactly to the Eikonal PDE with $S(s) = 1/c^*(s)$. Thus, while the HJB PDE formalizes cost-to-go under known dynamics, the Eikonal PDE captures a related spatial structure through $S(s)$, making it a natural approximation when dynamics are unknown.

Remark 4.3 (Why the Eikonal residual helps in Offline GCRL). Temporal-difference learning (e.g., (1)) is known to converge to the optimal GCVF V^* under ideal conditions: namely, on-policy data and an infinite sample budget [42]. However, these assumptions are often violated in the offline setting, where biased datasets and long-horizon tasks exacerbate extrapolation error and limit generalization [10]. In this context, the Eikonal residual in Eq. (7) can play a crucial role by introducing a geometric inductive bias that encourages the learned value function to behave like a distance field, through the constraint $\|\nabla_s V_{\boldsymbol{\theta}}(s,g)\| \approx c(s)$. This regularization is particularly effective when the true V^* is Lipschitz continuous, i.e., in environments where the dynamics do not induce sharp discontinuities [45]. Under such conditions, the Eikonal residual shapes the gradient norm of the learned GCVF $V_{\boldsymbol{\theta}}$ to match the local structure of V^*, up to a scaling factor. This effect is supported by Rademacher's theorem [46], which guarantees that Lipschitz continuous functions are differentiable almost everywhere, with $\|\nabla_s V^*(s,g)\| \leq L$, where L is the minimal Lipschitz constant. As a result, when V^* is Lipschitz continuous, the Eikonal residual in Eq. (7) provides a principled inductive bias that, as we empirically demonstrate, improves both sample efficiency and generalization, particularly in long-horizon tasks with smooth dynamics.

Furthermore, in practice, the Eikonal residual offers a tractable alternative to HJB regularization in model-free settings where the dynamics $f(s,a)$ are unknown. Prior work [6] approximates the HJB

term using finite differences, i.e., $f(s,a) \approx s' - s$, but our experiments show no clear advantage over the simpler Eikonal residual regularization. Moreover, note that, while the HJB equation in Eq. (5) holds only at optimality, the inequality in Proposition 4.1 remains valid throughout training, providing consistent structural guidance even when V is suboptimal.

Eikonal-regularized implicit V-learning Based on the upper-bound in Proposition 4.1 and the discussion in Remark 4.3, we propose the following Eikonal-regularized implicit V-learning loss for GCVF estimation:

$$\mathcal{L}_V(\boldsymbol{\theta}_V) = \mathbb{E}_{(s,s')\sim\mathcal{D},g\sim\mathcal{P}_g}\Big[L_2^\iota\big(\mathcal{R}(s,g) + \gamma V_{\bar{\boldsymbol{\theta}}_V}(s',g) - V_{\boldsymbol{\theta}_V}(s,g)\big) \tag{8}$$

$$+ \big(||\nabla_s V_{\boldsymbol{\theta}_V}(s,g)|| \cdot S(s) - 1\big)^2\Big], \tag{9}$$

where the term in (8) corresponds to the expectile regression in (1) (cf. [10]) and (9) is our Eikonal regularizer. In (9), $||\nabla_s V_{\boldsymbol{\theta}_V}(s,g)||$ is the Euclidean norm of the GCVF gradient with respect to its input s and $S(s)$ is a pre-specified speed profile that maps states to scalar values (see Preliminaries in Section 3). The term $\nabla_s V_{\boldsymbol{\theta}_V}(s,g)$ is computed via automatic differentiation, following standard approaches in PiNNs (see Algorithm 2 in Appendix D). The speed profile $S(s)$ is designed such that it encapsulates additional biases or task-specific information into the Eikonal regularizer [37]. We demonstrate in our experiments that the simple choice of $S(s) = 1$ works best in practice; however, we believe that more interesting designs might further improve collision avoidance in cluttered environments and consequently enhance both safety and performance. We defer this direction on Pi Safe GCRL to future work. Finally, note that in (9), with respect to the upper bound in (6), we set $c^*(s) = -1$ and opt to multiply $S(s)$ with $||\nabla_s V_{\boldsymbol{\theta}_V}(s,g)||$ rather than using $c^*(s) = -1/S(s)$ as in (2). This in order to ensure numerical stability.

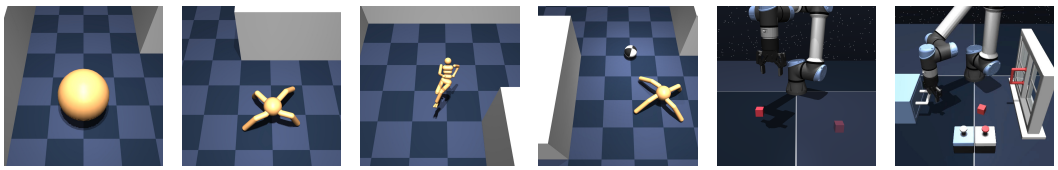
As discussed in Remark 4.3, the effectiveness of the regularization in Eq. (9) stems from the geometric structure of the Eikonal PDE, whose solution defines a signed distance field [36, 37]. In our final formulation, we uniformly set the speed profile to $S(s) = 1$, corresponding to the constraint $\|\nabla_s V_{\boldsymbol{\theta}_V}(s,g)\| \approx 1$ across the feasible state space. By Rademacher's theorem [46], this enforces 1-Lipschitz continuity almost everywhere, encouraging smoother and more generalizable value estimates even under limited offline data coverage. Although the true optimal value function V^* may not be exactly 1-Lipschitz over the entire feasible space, we show in our experiments (Section 5) that this regularization remains effective in practice. Note that using a different constant $S(s) > 0$ everywhere in the feasible space would simply uniformly rescale the value function without affecting its shape. Consequently, since policy gradients are invariant to such rescaling [42, 47, 48], this design choice has no impact on the induced policy.

Eikonal-regularized HIQL In the following, we provide a brief summary of our algorithm, Eik-HIQL. During training, Eik-HIQL performs an Eikonal-regularized value estimation step, where the loss in (8)-(9) is minimized to learn a GCVF. This is followed by a policy extraction step, in which the hierarchical actor introduced in Park et al. [10] (see Preliminaries in Section 3) is trained based on the estimated GCVF. The full pseudo-code for Eik-HIQL as well as a JAX [49] implementation showing how to compute the gradient $\nabla_s V_{\boldsymbol{\theta}_V}$ in (9) are provided in Appendix D, respectively Algorithm 1 and Algorithm 2. For the full implementation of Eik-HIQL refer to our GitHub repository.

5 Experiments

In this section, we analyze the effects of the Eikonal regularizer in (9) on the GCVF estimation problem. Specifically, we will perform an ablation over different designs of speed profiles $S(s)$, compare the Eikonal regularizer with an HJB regularizer, and analyze value functions learned with and without our Eikonal term. Then, we compare the performance obtained by our Eikonal-regularized algorithm, Eik-HIQL, against the SOTA algorithms for Offline GCRL. Finally, we will also present the limitations of our approach. The experiments in this section are conducted on the environments in Fig. 2. A table summarizing the most relevant hyperparameter values is provided in Appendix D.

Speed profiles and HJB regularizer comparison Recall that our algorithm, *Eik-HIQL*, estimates the GCVF using the loss defined in Eqs. (8)-(9), where the speed profile $S(s)$ encodes task-specific structure into the Eikonal regularizer. We perform an ablation over different choices of $S(s)$ and



(a) `pointmaze` (b) `antmaze` (c) `humanoid-maze` (d) `antsoccer` (e) `cube` (f) `scene`

Figure 2: Environments from OGbench [11] used in our experiments. These include a variety of goal-conditioned tasks spanning navigation and locomotion (e.g., `pointmaze`, `antmaze`, `humanoidmaze`), contact-rich locomotion (`antsoccer`), and contact-rich manipulation (`cube`, `scene`). The environments differ significantly in dynamics complexity, dimensionality, and task structure, providing a comprehensive testbed for evaluating Offline GCRL algorithms.

Table 1: Summary of the speed profiles ablation. All agents are trained for **100,000 training steps** using 10 seeds. We report the mean and standard deviation across seeds for the best evaluation achieved during training. For each seed, evaluations are conducted over 5 different random goals, as designed in Park et al. [11], with the learned policy tested for 50 episodes per goal. Results within 95% of the best value are written in **bold**.

Environment	Dataset Type	Maze Dimension	Eik-HIQL	Eik-HIQL Exp (10)	Eik-HIQL Lin (11)	HJB-HIQL (12)
`pointmaze`	`navigate`	`medium`	$\mathbf{94 \pm 4}$	$\mathbf{95 \pm 2}$	$\mathbf{93 \pm 4}$	90 ± 6
		`large`	$\mathbf{83 \pm 9}$	61 ± 8	60 ± 5	53 ± 9
		`giant`	$\mathbf{79 \pm 13}$	38 ± 15	42 ± 12	9 ± 8
		`teleport`	$\mathbf{47 \pm 10}$	39 ± 7	40 ± 6	18 ± 5
	`stitch`	`medium`	$\mathbf{97 \pm 2}$	90 ± 9	85 ± 10	64 ± 13
		`large`	$\mathbf{73 \pm 6}$	33 ± 17	41 ± 7	6 ± 7
		`giant`	$\mathbf{22 \pm 10}$	5 ± 8	2 ± 3	0 ± 0
		`teleport`	$\mathbf{44 \pm 9}$	$\mathbf{43 \pm 5}$	38 ± 7	18 ± 8

find the simple constant setting $S(s) = 1$ to be particularly effective. It outperforms more complex alternatives that require explicit knowledge of obstacle coordinates, which may not be available in practice. Specifically, we compare $S(s) = 1$ against the following two speed profiles:

$$S_{\text{exp}}(s) = S_{\text{min}} + (1 - S_{\text{min}}) \exp\left(-\lambda \frac{d_{\text{max}} - d(s)}{d_{\text{max}} - d_{\text{min}}}\right), \tag{10}$$

$$S_{\text{lin}}(s) = \text{clip}\left(\frac{d(s)}{d_{\text{max}}}, \frac{d_{\text{min}}}{d_{\text{max}}}, 1.0\right), \tag{11}$$

where λ is a decay rate, S_{min} represents the minimum tolerable speed, d_{min} and d_{max} are respectively the minimum and maximum tolerable distances of a state s from its nearest obstacle and $d(s)$ is a function $d : \mathcal{S} \to \mathbb{R}$ describing the Euclidean distance of the state s from its nearest obstacle (see Appendix B for more information). We refer to $S_{\text{exp}}(\cdot)$ in (10) as the exponential speed profile (*Eik-HIQL Exp* in Table 1) and to $S_{\text{lin}}(\cdot)$ in (11), originally introduced in [37], as the linear speed profile (*Eik-HIQL Lin* in Table 1). Both (10) and (11) assign high speed values to states s far from obstacles, and low speed values to states near obstacles. This encourages the agent to avoid obstacles, as proximity to them results in longer travel-time from s to g. Moreover, due to the use of the $\exp(\cdot)$ function, when compared to $S_{\text{lin}}(\cdot)$; $S_{\text{exp}}(\cdot)$ ensures a smoother decay of the speed value as the distance from the obstacles decreases [50]. Finally, note that, in both these speed profiles, computing $d(s)$ requires knowledge of the obstacles' coordinates which represents a strong assumption in some settings. In addition to this ablation, we also compare our Eikonal regularization term in (9) with an HJB regularizer derived from the HJB PDE in (5):

$$\mathcal{L}_V^{\text{HJB}}(\boldsymbol{\theta}_V) = \mathbb{E}_{(s,s')\sim\mathcal{D}, g\sim\mathcal{P}_g}\left[\left(\nabla_s V_{\boldsymbol{\theta}_V}(s,g)^\intercal (s' - s) - 1\right)^2\right]. \tag{12}$$

In (12), the dynamics $f(s, a)$, originally required by the HJB PDE in (5), is replaced by a finite difference term $(s' - s)$ as proposed by Lien et al. [6]. The term in (12) is then used in place of the Eikonal regularizer in (9) for GCVF estimation in Eik-HIQL. We refer to this approach as *HJB-HIQL* in Table 1. The results for these experiments are summarized in Table 1 and all the learning curves are available in Appendix E.

Two main observations explain the superior performance of the simple choice $S(s) = 1$. First, in Offline GCRL, sampled trajectories are already constrained to the feasible space, and there is no explicit penalty for colliding with obstacles during learning. As such, complex speed profiles incorporating obstacle proximity provide limited additional benefit over the uniform case. Second, the simplicity of $S(s) = 1$ enables more efficient and stable learning of the GCVF, whereas profiles requiring privileged information (e.g., obstacle maps) may introduce unnecessary complexity. Our choice of $S(s) = 1$ also aligns with standard practice in the literature, which commonly adopts uniform speed profiles to model wavefront propagation in the feasible state space [37, 51]. This also ensures a fair comparison with the Offline GCRL baselines, as none of them leverage privileged information. Furthermore, we provide visualizations for the learned GCVF in Appendix B, where contour plots for `pointmaze-giant-stitch-v0` illustrate that *Eik-HIQL* with $S(s) = 1$ learns a more structured and accurate value function, closely aligning with the maze layout. In contrast, *Eik-HIQL Exp* and *Eik-HIQL Lin* display artifacts even near the goal, while *Eik-HIQL HJB* fails to recover the maze geometry altogether. Based on this empirical evidence, and to ensure a fair comparison with baselines, we adopt the constant speed profile $S(s) = 1$ in all subsequent experiments. Nonetheless, we highlight that in the context of Safe GCRL, where value functions must encode both safety and task performance, investigating how different choices of $S(s)$ influence the shape and behavior of the GCVF represents an interesting direction for future work.

Eik-HIQL vs HIQL We compare Eik-HIQL with its non-regularized counterpart, HIQL [10], to isolate the effect of the Eikonal regularizer. This comparison is conducted under tightly controlled conditions: both methods use identical network architectures, hyperparameters, and training pipelines, ensuring that the only difference lies in the presence of the regularizer. As shown in Table 2, Eik-HIQL consistently outperforms HIQL on navigation tasks, where the value function is typically Lipschitz continuous. The benefits are particularly pronounced in large mazes and stitching regimes, with improvements exceeding 100% in 7 out of 31 evaluated tasks.

Given the magnitude of these gains relative to the standard deviations across 10 random seeds, the improvements are both practically meaningful and statistically significant. Full learning curves for these experiments are provided in Appendix E. These results highlight the robustness of the Eikonal regularizer in smooth environments and demonstrate its ability to enhance generalization in settings where HIQL struggles to scale. As illustrated in Fig. 1 for the `antmaze-navigate-giant-v0` task, Eik-HIQL produces a GCVF that reflects the underlying maze geometry, whereas HIQL fails to capture this structure, leading to poor goal-directed performance.

Table 2: Complete comparison between Eik-HiQRL and the Offline GCRL baselines. Agents are trained for **100,000 steps** on `pointmaze` tasks and **1 million steps** on the remaining tasks, each using 10 seeds. The evaluation follows the methodology described in Table 1. We report the mean and standard deviation across seeds for the best evaluation achieved during training. Results within 95% of the best value are written in **bold**, and rows are highlighted when the Eikonal regularizer improves performance by 100% or more compared to the non-regularized HIQL performance.

Environment	Dataset	Dim	Eik-HIQL	HIQL	QRL	CRL
pointmaze	navigate	medium	**93 ± 5**	**92 ± 2**	83 ± 3	54 ± 19
		large	83 ± 9	49 ± 13	**90 ± 5**	56 ± 9
		giant	**79 ± 13**	7 ± 8	72 ± 7	37 ± 17
		teleport	**47 ± 10**	29 ± 7	34 ± 7	**50 ± 5**
	stitch	medium	**96 ± 3**	76 ± 8	80 ± 10	3 ± 5
		large	73 ± 6	19 ± 7	**85 ± 11**	4 ± 6
		giant	22 ± 10	1 ± 4	**56 ± 9**	0 ± 0
		teleport	**43 ± 9**	38 ± 5	**42 ± 6**	12 ± 6
antmaze	navigate	medium	**95 ± 1**	**96 ± 1**	87 ± 5	**94 ± 2**
		large	**86 ± 2**	**90 ± 6**	80 ± 5	**86 ± 3**
		giant	**67 ± 5**	**69 ± 3**	14 ± 6	18 ± 4
		teleport	**52 ± 4**	43 ± 3	39 ± 4	**55 ± 4**
	stitch	medium	**94 ± 2**	**95 ± 14**	68 ± 6	54 ± 8
		large	**84 ± 3**	74 ± 6	24 ± 5	13 ± 4
		giant	**48 ± 11**	3 ± 3	2 ± 2	0 ± 0
		teleport	**47 ± 2**	35 ± 3	29 ± 6	34 ± 4
	explore	medium	**43 ± 15**	33 ± 15	5 ± 4	4 ± 2
		large	**13 ± 1**	6 ± 7	0 ± 0	0 ± 0
		teleport	15 ± 10	**45 ± 5**	2 ± 2	22 ± 5
humanoidmaze	navigate	medium	**86 ± 2**	**90 ± 3**	22 ± 2	61 ± 4
		large	**64 ± 7**	50 ± 4	7 ± 3	22 ± 9
		giant	**68 ± 5**	18 ± 5	1 ± 1	4 ± 2
	stitch	medium	79 ± 2	**88 ± 3**	22 ± 4	40 ± 7
		large	**29 ± 7**	**28 ± 2**	3 ± 1	4 ± 2
		giant	**19 ± 5**	3 ± 1	0 ± 0	0 ± 0
antsoccer	navigate	arena	19 ± 2	**60 ± 4**	10 ± 3	24 ± 2
		medium	3 ± 2	**13 ± 3**	2 ± 2	4 ± 2
	stitch	arena	2 ± 0	**17 ± 3**	2 ± 1	1 ± 1
		medium	1 ± 0	**5 ± 1**	0 ± 0	0 ± 0
manipulation	cube-single-play		25 ± 1	**31 ± 4**	11 ± 3	**32 ± 2**
	scene-play		**52 ± 7**	**52 ± 3**	8 ± 2	35 ± 6

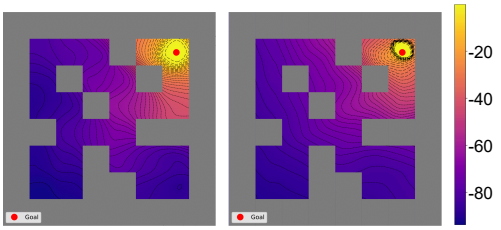
By contrast, in interactive, contact-rich domains such as `antsoccer` and `manipulation`, where the dynamics (and consequently the value function) exhibit discontinuities, Eik-HIQL offers limited advantage. We defer a detailed discussion of these cases to the Limitations section below.

Offline GCRL We extend our comparison to state-of-the-art Offline GCRL algorithms, including QRL [12] and CRL [13], alongside HIQL. Results across all 31 tasks are summarized in Table 2, with full learning curves in Appendix E. Eik-HIQL consistently outperforms all baselines in the most challenging settings, most notably in the `antmaze-stitch` tasks, which combine complex locomotion with, composite datasets, and in `humanoidmaze`, where high-dimensional states and unstable control amplify learning difficulty.

QRL performs well in simpler domains such as `pointmaze`, where its quasimetric structure aids goal-conditioned estimation. CRL is competitive on several `navigate` variants, but struggles in high-dimensional and stitched tasks such as `humanoidmaze`. In contrast, Eik-HIQL demonstrates strong generalization and planning performance in both long-horizon and large-scale environments.

In `antsoccer` and `manipulation`, however, Eik-HIQL performs comparably to the baselines. As mentioned, these domains involve discontinuous or contact-rich dynamics, and we do not observe consistent benefits from the Eikonal regularizer due to the fact that the imposed gradient condition does not hold globally at optimality throughout the entire state space.

Limitations We conclude by discussing some limitations of our approach, particularly in contact-rich tasks that involve interactions with external objects. In the `antsoccer` domain, for instance, an ant agent must not only navigate but also interact with a soccer ball to reach a specified goal. Similarly, in manipulation tasks such as `cube-single-play`, the agent must coordinate precise contact interactions with objects in the scene. These tasks introduce hybrid, non-Lipschitz dynamics due to discontinuities in contact states, often modeled as categorical variables (e.g., in-contact vs. free motion), which pose a challenge for regularizers grounded in smoothness and Lipschitz continuity assumptions [43], such as our Eikonal term. As visualized in Fig. 3 for the `antsoccer` experiments, Eik-HIQL learns a GCVF that aligns well with the environment geometry, particularly for the navigational components. However, this structured value function does not consistently yield better performance (cf. Table 2) as these tasks also require complex interactions with objects in the environment. Similar trends are observed in manipulation tasks, where Eik-HIQL performs comparably to the baselines but does not show marked gains. These experiments are included for completeness and highlight that, while Eik-HIQL excels in navigation-dominated domains, additional mechanisms, such as task-adaptive speed profiles or representation learning tailored to contact dynamics, may be necessary to extend its benefits to interactive, contact-rich environments. We leave this direction for future work.



(a) Eik-HIQL (b) HIQL

Figure 3: Countour plots of the GCVF on `antsoccer-medium-navigate-v0` [11], learned after 1 million steps by Eik-HIQL and HIQL respectively. These plots are generated following the same methodology in Fig. 1.

6 Conclusion

We introduced Eik-HIQL, a novel approach to Offline GCRL that integrates an Eikonal PDE-based regularizer for GCVF estimation. Our analysis demonstrated that the Eikonal regularizer effectively introduces a useful distance-like inductive bias, which promotes consistent gradient magnitudes and improves value estimation in high-dimensional spaces (cf. Fig. 1). This property mitigates irregularities from the limited coverage of offline datasets, resulting in robust, globally consistent GCVFs that accurately capture the underlying structure of the environment. Consequently, Eik-HIQL outperformed SOTA baselines across diverse tasks, excelling in complex scenarios such as large-scale mazes and trajectory stitching, where traditional methods often fail to generalize.

However, our experiments also highlighted limitations in interactive tasks. We discuss how our Eikonal regularizer induces excessive smoothness in the learned GCVFs which does hinder perfor-

mance in interactive tasks where non-smoothness, or at least non-global smoothness, is required. These findings suggest that, while the Eikonal regularizer significantly enhances navigation tasks, future work should incorporate mechanisms that better capture task-specific dynamics, such as object interaction, to improve applicability across diverse domains.

Overall, this work underscores the potential of Pi methods to address fundamental challenges in Offline GCRL. By enhancing scalability and generalization of value estimation, the Eikonal regularizer provides a foundation for leveraging domain knowledge in RL. Future research could expand on this foundation by exploring multi-agents settings and integrating task-specific biases for interactive environments.

References

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

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

[3] Leslie Pack Kaelbling. Learning to achieve goals. In *International Joint Conference on Artificial Intelligence*, volume 2, pages 1094–8. Citeseer, 1993.

[4] Tom Schaul, Daniel Horgan, Karol Gregor, and David Silver. Universal value function approximators. In *International Conference on Machine Learning*, pages 1312–1320. PMLR, 2015.

[5] Seohong Park, Kevin Frans, Sergey Levine, and Aviral Kumar. Is value learning really the main bottleneck in offline rl? *arXiv preprint arXiv:2406.09329*, 2024.

[6] Yun-Hsuan Lien, Ping-Chun Hsieh, Tzu-Mao Li, and Yu-Shuen Wang. Enhancing value function estimation through first-order state-action dynamics in offline reinforcement learning. In *International Conference on Machine Learning*, 2024.

[7] Marcus M Noack and Stuart Clark. Acoustic wave and eikonal equations in a transformed metric space for various types of anisotropy. *Heliyon*, 3(3), 2017.

[8] Rémi Munos. A convergent reinforcement learning algorithm in the continuous case based on a finite difference method. In *International Joint Conference on Artificial Intelligence*, pages 826–831, 1997.

[9] Rémi Munos and Paul Bourgine. Reinforcement learning for continuous stochastic control problems. *Advances in Neural Information Processing Systems*, 10, 1997.

[10] Seohong Park, Dibya Ghosh, Benjamin Eysenbach, and Sergey Levine. Hiql: Offline goal-conditioned rl with latent states as actions. *Advances in Neural Information Processing Systems*, 36, 2024.

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

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

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

[14] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C Courville. Improved training of wasserstein gans. *Advances in Neural Information Processing Systems*, 30, 2017.

[15] Ruijie Zheng, Xiyao Wang, Huazhe Xu, and Furong Huang. Is model ensemble necessary? model-based rl via a single model with lipschitz regularized value function. *arXiv preprint arXiv:2302.01244*, 2023.

[16] Matthias Weissenbacher, Samarth Sinha, Animesh Garg, and Kawahara Yoshinobu. Koopman q-learning: Offline reinforcement learning via symmetries of dynamics. In *International Conference on Machine Learning*, pages 23645–23667. PMLR, 2022.

[17] Peng Cheng, Xianyuan Zhan, Wenjia Zhang, Youfang Lin, Han Wang, Li Jiang, et al. Look beneath the surface: Exploiting fundamental symmetry for sample-efficient offline rl. *Advances in Neural Information Processing Systems*, 36:7612–7631, 2023.

[18] Yifan Wu, George Tucker, and Ofir Nachum. Behavior regularized offline reinforcement learning. *arXiv preprint arXiv:1911.11361*, 2019.

[19] Scott Fujimoto, David Meger, and Doina Precup. Off-policy deep reinforcement learning without exploration. In *International Conference on Machine Learning*, pages 2052–2062. PMLR, 2019.

[20] Aviral Kumar, Justin Fu, Matthew Soh, George Tucker, and Sergey Levine. Stabilizing off-policy q-learning via bootstrapping error reduction. *Advances in Neural Information Processing Systems*, 32, 2019.

[21] Ziyu Wang, Alexander Novikov, Konrad Zolna, Josh S Merel, Jost Tobias Springenberg, Scott E Reed, Bobak Shahriari, Noah Siegel, Caglar Gulcehre, Nicolas Heess, et al. Critic regularized regression. *Advances in Neural Information Processing Systems*, 33:7768–7778, 2020.

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

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

[24] Aviral Kumar, Aurick Zhou, George Tucker, and Sergey Levine. Conservative q-learning for offline reinforcement learning. *Advances in Neural Information Processing Systems*, 33: 1179–1191, 2020.

[25] Richard S Sutton, Doina Precup, and Satinder Singh. Between mdps and semi-mdps: A framework for temporal abstraction in reinforcement learning. *Artificial Intelligence*, 112(1-2): 181–211, 1999.

[26] Vittorio Giammarino and Ioannis Ch Paschalidis. Online baum-welch algorithm for hierarchical imitation learning. In *2021 60th IEEE Conference on Decision and Control (CDC)*, pages 3717–3722. IEEE, 2021.

[27] Marcin Andrychowicz, Filip Wolski, Alex Ray, Jonas Schneider, Rachel Fong, Peter Welinder, Bob McGrew, Josh Tobin, OpenAI Pieter Abbeel, and Wojciech Zaremba. Hindsight experience replay. *Advances in Neural Information Processing Systems*, 30, 2017.

[28] Jason Yecheng Ma, Jason Yan, Dinesh Jayaraman, and Osbert Bastani. Offline goal-conditioned reinforcement learning via f-advantage regression. *Advances in Neural Information Processing Systems*, 35:310–323, 2022.

[29] Yevgen Chebotar, Karol Hausman, Yao Lu, Ted Xiao, Dmitry Kalashnikov, Jake Varley, Alex Irpan, Benjamin Eysenbach, Ryan Julian, Chelsea Finn, et al. Actionable models: Unsupervised offline reinforcement learning of robotic skills. *arXiv preprint arXiv:2104.07749*, 2021.

[30] Rui Yang, Yiming Lu, Wenzhe Li, Hao Sun, Meng Fang, Yali Du, Xiu Li, Lei Han, and Chongjie Zhang. Rethinking goal-conditioned supervised learning and its connection to offline rl. *arXiv preprint arXiv:2202.04478*, 2022.

[31] Rui Yang, Lin Yong, Xiaoteng Ma, Hao Hu, Chongjie Zhang, and Tong Zhang. What is essential for unseen goal generalization of offline goal-conditioned rl? In *International Conference on Machine Learning*, pages 39543–39571. PMLR, 2023.

[32] Lina Mezghani, Sainbayar Sukhbaatar, Piotr Bojanowski, Alessandro Lazaric, and Karteek Alahari. Learning goal-conditioned policies offline with self-supervised reward shaping. In *Conference on Robot Learning*, pages 1401–1410. PMLR, 2023.

[33] Harshit Sikchi, Rohan Chitnis, Ahmed Touati, Alborz Geramifard, Amy Zhang, and Scott Niekum. Smore: Score models for offline goal-conditioned reinforcement learning. *arXiv preprint arXiv:2311.02013*, 2023.

[34] Samuel H Rudy, Steven L Brunton, Joshua L Proctor, and J Nathan Kutz. Data-driven discovery of partial differential equations. *Science Advances*, 3(4):e1602614, 2017.

[35] Maziar Raissi, Paris Perdikaris, and George E Karniadakis. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational Physics*, 378:686–707, 2019.

[36] Jonathan D Smith, Kamyar Azizzadenesheli, and Zachary E Ross. Eikonet: Solving the eikonal equation with deep neural networks. *IEEE Transactions on Geoscience and Remote Sensing*, 59 (12):10685–10696, 2020.

[37] Ruiqi Ni and Ahmed H Qureshi. Ntfields: Neural time fields for physics-informed robot motion planning. In *The Eleventh International Conference on Learning Representations*, 2023.

[38] Ruiqi Ni, Zherong Pan, and Ahmed H Qureshi. Physics-informed temporal difference metric learning for robot motion planning. In *The Thirteenth International Conference on Learning Representations*, 2025. URL `https://openreview.net/forum?id=TOiageVNru`.

[39] Haoran Xu, Li Jiang, Li Jianxiong, and Xianyuan Zhan. A policy-guided imitation approach for offline reinforcement learning. *Advances in Neural Information Processing Systems*, 35: 4085–4098, 2022.

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

[41] Volodymyr Mnih. Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*, 2013.

[42] Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 2018.

[43] Jeongho Kim, Jaeuk Shin, and Insoon Yang. Hamilton-jacobi deep q-learning for deterministic continuous-time systems with lipschitz continuous controls. *Journal of Machine Learning Research*, 22(206):1–34, 2021.

[44] Wendell H Fleming and Halil Mete Soner. *Controlled Markov processes and viscosity solutions*, volume 25. Springer Science & Business Media, 2006.

[45] Bora S Banjanin and Samuel A Burden. Nonsmooth optimal value and policy functions in mechanical systems subject to unilateral constraints. *IEEE Control Systems Letters*, 4(2): 506–511, 2019.

[46] Juha Heinonen. *Lectures on Lipschitz analysis*. Number 100. University of Jyväskylä, 2005.

[47] Richard S Sutton, David McAllester, Satinder Singh, and Yishay Mansour. Policy gradient methods for reinforcement learning with function approximation. *Advances in Neural Information Processing Systems*, 12, 1999.

[48] John Schulman, Sergey Levine, Pieter Abbeel, Michael Jordan, and Philipp Moritz. Trust region policy optimization. In *International Conference on Machine Learning*, pages 1889–1897. PMLR, 2015.

[49] James Bradbury, Roy Frostig, Peter Hawkins, Matthew James Johnson, Chris Leary, Dougal Maclaurin, George Necula, Adam Paszke, Jake VanderPlas, Skye Wanderman-Milne, and Qiao Zhang. JAX: composable transformations of Python+NumPy programs, 2018. URL `http://github.com/jax-ml/jax`.

[50] Ruiqi Ni and Ahmed H Qureshi. Physics-informed neural motion planning on constraint manifolds. *arXiv preprint arXiv:2403.05765*, 2024.

[51] James A Sethian. A fast marching level set method for monotonically advancing fronts. *Proceedings of the National Academy of Sciences*, 93(4):1591–1595, 1996.

NeurIPS Paper Checklist

1. **Claims**

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

   Answer: [Yes]

   Justification:

   Guidelines:

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

2. **Limitations**

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

   Answer: [Yes]

   Justification: we have a limitations paragraph.

   Guidelines:

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

3. **Theory assumptions and proofs**

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

   Answer: [Yes]

Justification: the proof is included in the main text.

Guidelines:

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

4. **Experimental result reproducibility**

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

   Answer: [Yes]

   Justification: All the details are reported between main text and the appendix. Furthermore, we have released the code.

   Guidelines:

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

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

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

Answer: [Yes]

Justification: Code has been released.

Guidelines:

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

6. **Experimental setting/details**

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

   [Yes]

   Justification: The most important details are available in the main text and the rest is reported in the Appendix.

   Guidelines:

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

7. **Experiment statistical significance**

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

   Answer: [Yes]

   Justification: yes, we report the standard deviation.

   Guidelines:

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

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

8. **Experiments compute resources**

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

   Answer: [Yes]

   Justification: Appendix.

   Guidelines:

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

9. **Code of ethics**

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

   Answer: [Yes]

   Justification:

   Guidelines:

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

10. **Broader impacts**

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

    Answer: [Yes]

    Justification:

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

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

    Answer: [NA]

    Justification:

    Guidelines:

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

12. **Licenses for existing assets**

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

    Answer: [NA]

    Justification:

    Guidelines:

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

- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

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

    Answer: [NA]

    Justification:

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

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

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

    Answer: [NA]

    Justification:

    Guidelines:

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

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

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

    Answer: [NA]

    Justification:

    Guidelines:

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

16. **Declaration of LLM usage**

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

Answer: [NA]

Justification:

Guidelines:

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

A Impact Statement

This paper aims to advance the field of machine learning for autonomous decision-making and control in robotic systems. We achieved this goal through the development of physics-informed methods for offline goal-conditioned reinforcement learning. While our work has potential societal implications, we do not identify any that require specific emphasis here.

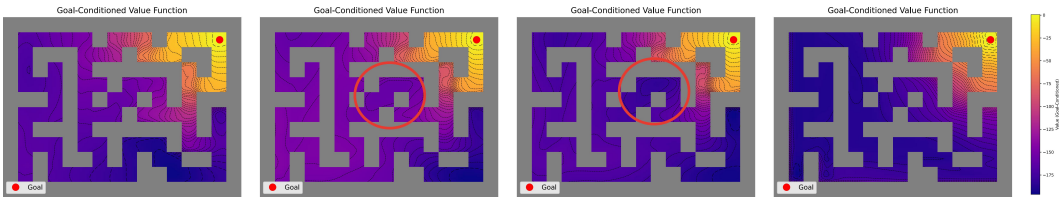
B Speed profiles and HJB comparison: additional details and contour plots

In this section, we provide additional details related to the *Speed Profiles and HJB Regularizer Comparison* paragraph in Section 5. Specifically, we illustrate how the distance function $d(s)$, used to define the speed profiles in (10) and (11), is computed in Fig. 4.

Additionally, Fig. 5 presents contour plots of the GCVFs learned by the algorithms evaluated in Table 1. These visualizations provide qualitative support for the quantitative results in Table 1, showing that higher returns tend to correlate with smoother, artifact-free value functions that more closely follow the structure of the underlying maze.

(a) $d(s)$. (b) $S_{\text{lin}}(s)$ in (11).

Figure 4: Fig. 4a illustrates the computation of the distance function $d(s)$ used in (10) and (11). Let the state be represented by its spatial coordinates $s = (x, y) \in \mathbb{R}^2$, and let $\mathcal{O} = \{o_1, \ldots, o_M\}$ denote the set of obstacle coordinates in the maze. We define $d(s) = \min_{o \in \mathcal{O}} \|s - o\|_2$, i.e., the Euclidean distance from s to the nearest obstacle. Fig. 4b reports the resulting speed profile obtained using $S_{\text{lin}}(s)$ in (11) for the `pointmaze-medium-navigate-v0` dataset.



(a) Eik-HIQL (b) Eik-HIQL Exp (10) (c) Eik-HIQL Lin (11) (d) HJB-HIQL (12)

Figure 5: Contour plots of the GCVF on `pointmaze-giant-stitch-v0` [11], learned after 100,000 training steps by the algorithms in Table 1. We observe that Eik-HIQL with constant speed profile $S(s) = 1$ provides the most accurate GCVF estimation, resulting in the highest score for this task. In contrast, the contour plot for HJB-HIQL in (d) fails to effectively capture the maze layout. Furthermore, in (b) and (c) we annotate, with red circles, examples of artifacts arising in Eik-HIQL Exp and Eik-HIQL Lin, respectively.

C HJB PDE step-by-step derivations and Proof of Proposition 4.1

In the following we provide the step-by-step derivations for the HJB PDE in (5) and the full proof for Proposition 4.1.

C.1 HJB PDE derivations

The optimal value function $V(s, g)$ associated with the undiscounted optimal control problem in (3), satisfies the following principle of optimality

$$V(s, g) = \inf_{a \in \mathcal{A}} [c(s, a)\Delta t + V(s(t + \Delta t), g)], \tag{13}$$

where Δt is a small time step. By substituting the Taylor expansion

$$V(s(t + \Delta t), g) = V(s, g) + \nabla_s V(s, g)^\top f(s, a)\Delta t + O(\Delta t^2)$$

into (13), we obtain

$$V(s, g) = \inf_{a \in \mathcal{A}} [c(s, a)\Delta t + V(s, g) + \nabla_s V(s, g)^\top f(s, a)\Delta t + O(\Delta t^2)].$$

Subtracting $V(s, g)$ on both sides and dividing by Δt gives:

$$0 = \inf_{a \in \mathcal{A}} [c(s, a) + \nabla_s V(s, g)^\top f(s, a) + O(\Delta t)].$$

Taking the limit $\Delta t \to 0$, we recover Eq. (5):

$$\inf_{a \in \mathcal{A}} [c(s, a) + \nabla_s V(s, g)^\top f(s, a)] = 0.$$

C.2 Proof of Proposition 4.1

Proof. Consider the HJB PDE in (5). By applying the Cauchy-Schwarz inequality to the argument of the minimization we obtain

$$c(s, a) + \nabla_s V(s, g)^\top f(s, a) \leq c(s, a) + \|\nabla_s V(s, g)\| \|f(s, a)\|.$$

Then, by defining

$$F^*(s) = \sup_{a \in \mathcal{A}} \|f(s, a)\|,$$

we can further upper bound the right-hand-side and obtain:

$$c(s, a) + \nabla_s V(s, g)^\top f(s, a) \leq c(s, a) + \|\nabla_s V(s, g)\| F^*(s).$$

Finally, applying the infimum over $a \in \mathcal{A}$ on both sides yields:

$$\inf_{a \in \mathcal{A}} [c(s, a) + \nabla_s V(s, g)^\top f(s, a)] \leq \inf_{a \in \mathcal{A}} [c(s, a) + \|\nabla_s V(s, g)\| F^*(s)],$$

where the result in Eq. (6) is obtained by defining $c^*(s) = \inf_{a \in \mathcal{A}} c(s, a)$.

For the equality in (7), note that for the isotropic dynamics $f(s, a) = a$ and given $c(s, a)$ constant over $||a|| = 1$, the inner product $\nabla_s V(s, g)^\intercal f(s, a)$ attains its minimal value when a points in the direction opposite to $\nabla_s V(s, g)$. Specifically, this occurs when

$$a^* = \arg \inf_{||a||=1} \nabla_s V(s, g)^\intercal a = -\frac{\nabla_s V(s, g)}{||\nabla_s V(s, g)||}.$$

Substituting $f(s, a) = a^*$ into

$$H(s, g, \nabla_s V(s, g)) = \inf_{a \in \mathcal{A}} [c(s, a) + \nabla_s V(s, g)^\top f(s, a)]$$

and simplifying yields

$$H(s, g, \nabla_s V(s, g)) = c^*(s) - ||\nabla_s V(s, g)||.$$

□

D Psuedocode and Hyperparameters

During training, Eik-HIQL performs an Eikonal-regularized value estimation step followed by a hierarchical policy extraction step. During the value estimation step, the following loss, as provided in (8)-(9), is minimized:

$$\mathcal{L}_V(\boldsymbol{\theta}_V) = \mathbb{E}_{(s_t,s_{t+1})\sim\mathcal{D},g\sim\mathcal{P}_g}\Big[L_2^\iota\big(\mathcal{R}(s_t,g) + \gamma V_{\bar{\boldsymbol{\theta}}_V}(s_{t+1},g) - V_{\boldsymbol{\theta}_V}(s_t,g)\big) + \big(||\nabla_s V_{\boldsymbol{\theta}_V}(s_t,g)|| \cdot S(s_t) - 1\big)^2\Big]. \tag{14}$$

The hierarchical policy extraction step follows Park et al. [10] and leverages, for both $\pi^{hi}_{\boldsymbol{\theta}_{hi}}$ and $\pi^{lo}_{\boldsymbol{\theta}_{lo}}$, an advantage-weighted regression-style objective:

$$J_{\pi^{hi}}(\boldsymbol{\theta}_{hi}) = \mathbb{E}_{(s_t,s_{t+k})\sim\mathcal{D},g\sim\mathcal{P}_g}\Big[\exp(\beta\cdot\tilde{A}^{hi}(s_t,s_{t+k},g))\log\pi^{hi}_{\boldsymbol{\theta}_{hi}}(s_{t+k}|s_t,g)\Big], \tag{15}$$

$$J_{\pi^{lo}}(\boldsymbol{\theta}_{lo}) = \mathbb{E}_{(s_t,a_t,s_{t+1},s_{t+k})\sim\mathcal{D}}\Big[\exp(\beta\cdot\tilde{A}^{lo}(s_t,a_t,s_{t+k}))\log\pi^{lo}_{\boldsymbol{\theta}_{lo}}(a_t|s_t,s_{t+k})\Big], \tag{16}$$

where β is an inverse temperature hyperparameter and $\tilde{A}^{hi}(s_t,s_{t+k},g)$ and $\tilde{A}^{lo}(s_t,a_t,s_{t+k})$ are respectively approximated as $V_{\boldsymbol{\theta}_V}(s_{t+k},g) - V_{\boldsymbol{\theta}_V}(s_t,g)$ and $V_{\boldsymbol{\theta}_V}(s_{t+1},s_{t+k}) - V_{\boldsymbol{\theta}_V}(s_t,s_{t+k})$. The full pseudocode for Eik-HIQL is provided in Algorithm 1. Furthermore, a function written in JAX on how to compute the gradient $\nabla_s V_{\boldsymbol{\theta}_V}$ in (14) is summarized in Algorithm 2. Finally, Table 3 reports the hyperparameter values most commonly used in our experiments. For more implementation details, refer to our GitHub repository[2].

Algorithm 1 Eikonal-regularized Hierarchical Implicit Q-Learning (Eik-HIQL)

Input: Offline dataset $\mathcal{D}$, value function $V_{\boldsymbol{\theta}_V}$, target value function $V_{\bar{\boldsymbol{\theta}}_V}$, high-level policy $\pi^{hi}_{\boldsymbol{\theta}_{hi}}$, low-level policy $\pi^{lo}_{\boldsymbol{\theta}_{lo}}$, speed profile S, expectile factor ι, discount factor γ, inverse temperature parameter β, learning rates α_V, α_{hi}, α_{lo}, target update rate τ
while not converged **do**
  $(s_t, s_{t+1}, g) \sim \mathcal{D}$
  Update $V_{\boldsymbol{\theta}_V}$ minimizing $\mathcal{L}_V(\boldsymbol{\theta}_V)$ in (14) with learning rate α_V
  $\bar{\boldsymbol{\theta}}_V \leftarrow (1-\tau)\bar{\boldsymbol{\theta}}_V + \tau\boldsymbol{\theta}_V$
end while
while not converged **do**
  $(s_t, s_{t+k}, g) \sim \mathcal{D}$
  Update $\pi^{hi}_{\boldsymbol{\theta}_{hi}}$ maximizing $J_{\pi^{hi}}(\boldsymbol{\theta}_{hi})$ in (15) with learning rate α_{hi}
end while
while not converged **do**
  $(s_t, a_t, s_{t+1}, s_{t+k}) \sim \mathcal{D}$
  Update $\pi^{lo}_{\boldsymbol{\theta}_{lo}}$ maximizing $J_{\pi^{lo}}(\boldsymbol{\theta}_{lo})$ in (16) with learning rate α_{lo}
end while

Algorithm 2 Compute $\nabla_s V_{\boldsymbol{\theta}_V}$

Input: states s, goals g, network parameters $\boldsymbol{\theta}_V$
Define FORWARD$(s, g, \boldsymbol{\theta}_V)$:
  return NETWORK.SELECT$(V_{\boldsymbol{\theta}_V})(s, g, \text{params} = \boldsymbol{\theta}_V)$
grad_s $\leftarrow$ JAX.VMAP(JAX.GRAD(FORWARD, argnums $= 0$), in_axes $= (0, 0, \textbf{None}))(s, g, \boldsymbol{\theta}_V)$
return grad_s

[2] https://github.com/VittorioGiammarino/Eik-HIQL

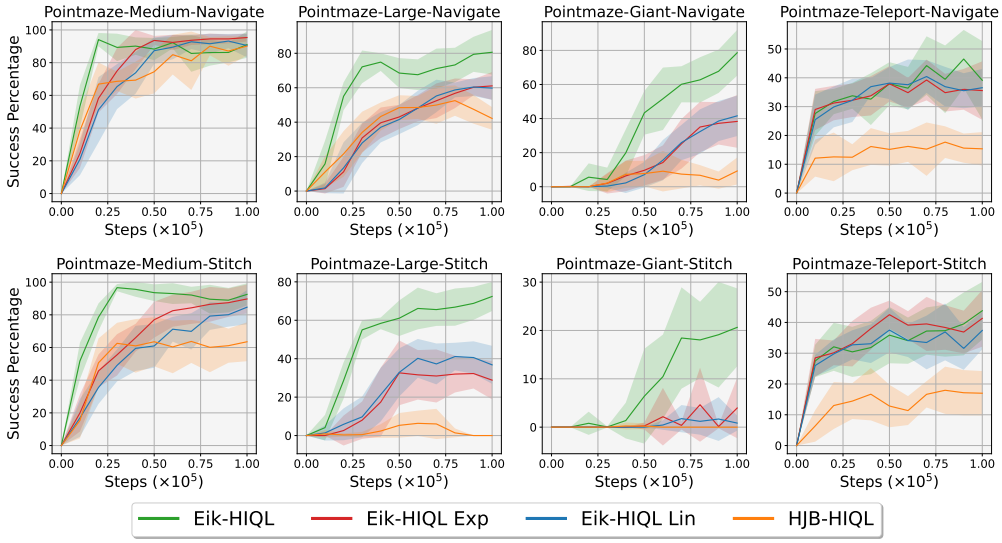


Figure 6: Learning curves for the speed profile ablation and the comparison with an HJB regularizer in Table 1. Plots show the average success percentage per evaluation across seeds as a function of training steps.

Table 3: Hyperparameter values for Eik-HIQL.

Hyperparameter Name	Value
Decay rate (λ)	1.0
Minimum speed (S_{min})	0.1
Discount factor (γ)	0.99
Batch size (B)	1024
Optimizer	Adam
Learning rates α_V, α_{hi}, α_{lo}	$3 \cdot 10^{-4}$
Target update rate (τ)	0.005
Expectile factor (ι)	0.7
Inverse temperature parameter (β)	3.0

E Learning Curves

Fig. 6 shows the complete learning curves, plotted as a function of training steps, for the experiments reported in Table 1. Figs. 7, 8, 9, 10, and 11 display the learning curves for the `pointmaze`, `antmaze`, `humanoidmaze`, `antsoccer`, and `manipulation` experiments in Table 2, respectively.

All experiments were conducted on a single NVIDIA RTX 3090 GPU (24 GB VRAM), using a local server equipped with a 12th Gen Intel i7-12700F CPU, 32 GB RAM. No cloud services or compute clusters were used. Each individual experimental run required approximately 4 hours of compute time on the GPU.

F Additional Experiments

In the following, we present additional experiments demonstrating that our Eikonal regularizer can be seamlessly integrated with a broad range of temporal-difference (TD)-based GCRL algorithms. In particular, we apply it to Goal-Conditioned variants of IQL [23] and IVL [39, 40], yielding Eik-GCIQL and Eik-GCIVL, respectively. The corresponding results are summarized in Table 4, with learning curves provided in Fig. 12 and 13. These experiments confirm the same conclusions drawn in the main paper from the comparison between Eik-HIQL and HIQL (Table 2), and further support our claim that the Eikonal regularizer can be successfully combined with diverse TD-based algorithms.

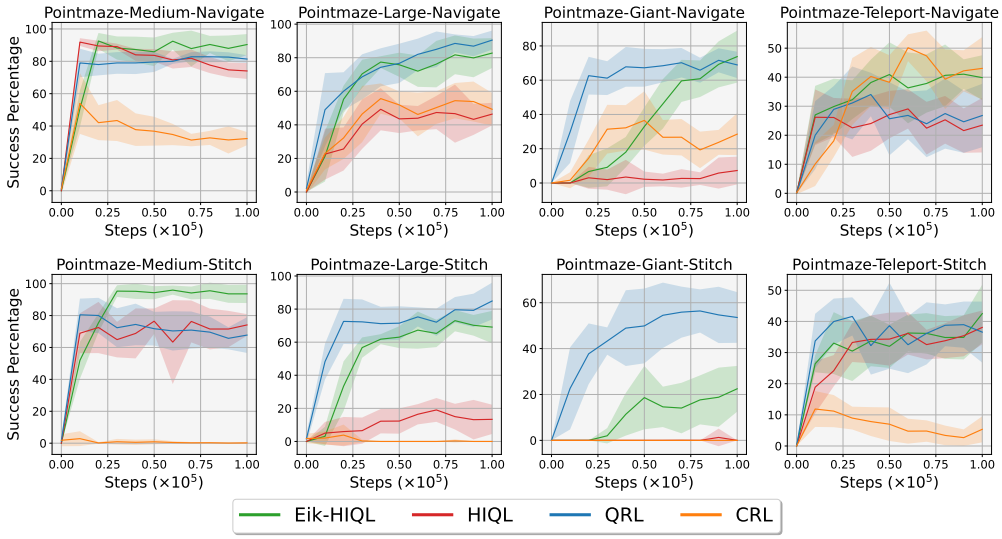


Figure 7: Learning curves for the `pointmaze` experiments in Table 2. Plots show the average success percentage per evaluation across seeds as a function of training steps.

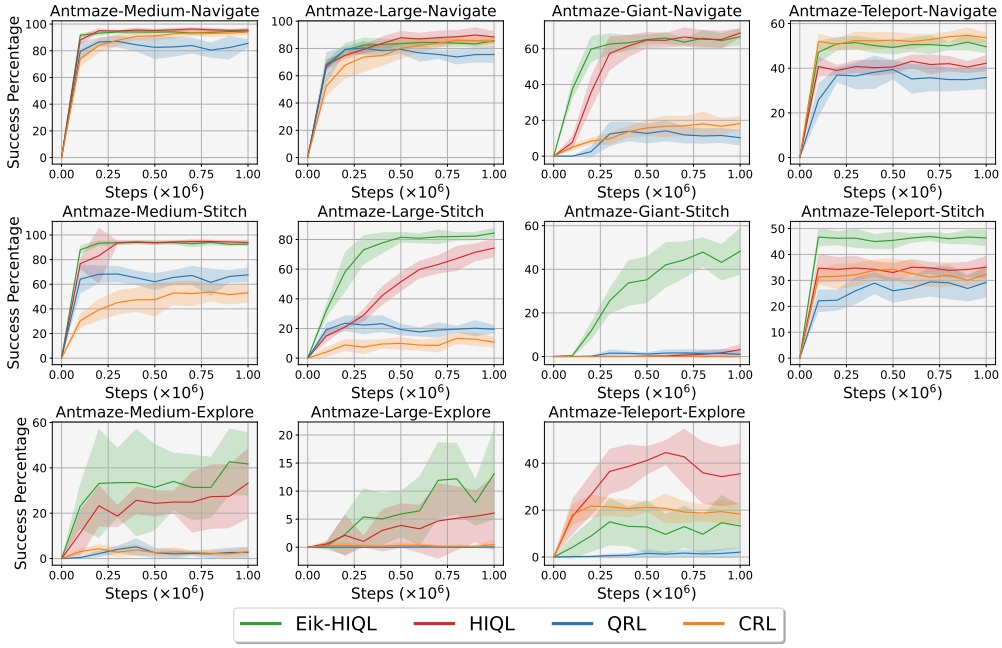


Figure 8: Learning curves for the `antmaze` experiments in Table 2. Plots show the average success percentage per evaluation across seeds as a function of training steps.

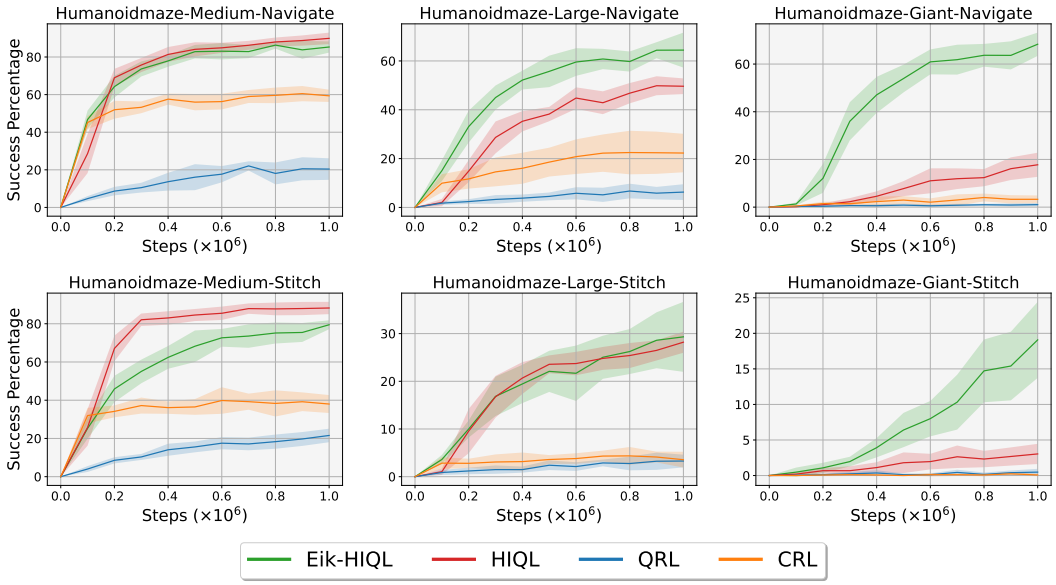


Figure 9: Learning curves for the `humanoidmaze` experiments in Table 2. Plots show the average success percentage per evaluation across seeds as a function of training steps.

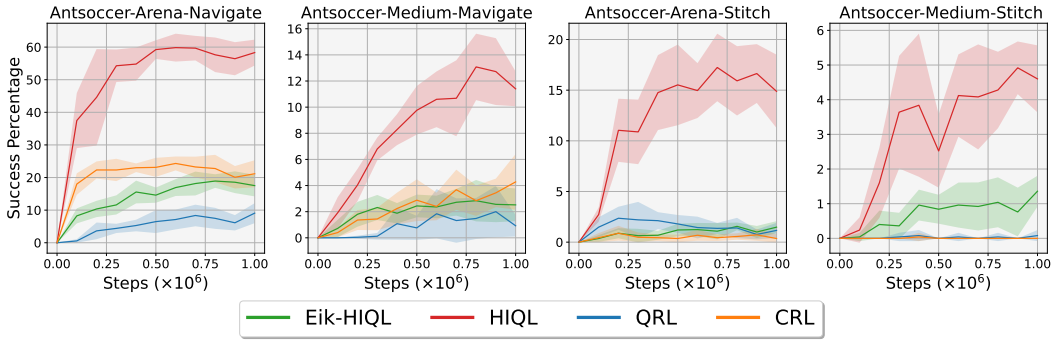


Figure 10: Learning curves for the `antsoccer` experiments in Table 2. Plots show the average success percentage per evaluation across seeds as a function of training steps.

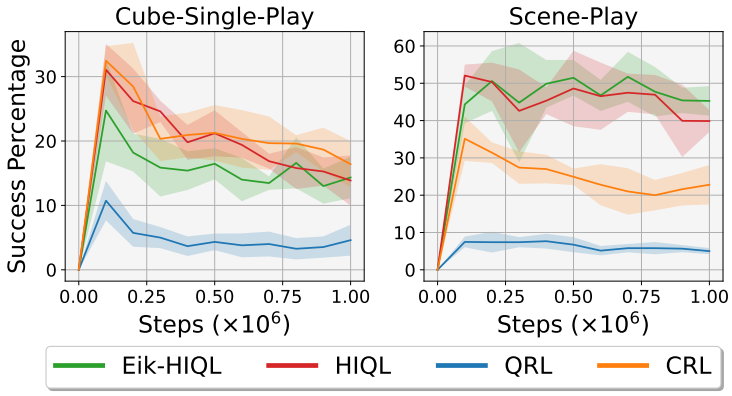


Figure 11: Learning curves for the `manipulation` experiments in Table 2. Plots show the average success percentage per evaluation across seeds as a function of training steps.

Table 4: Summary of the experiments with different TD-based GCRL algorithms. All agents are trained for **100,000 training steps** using 10 seeds. We report the mean and standard deviation across seeds for the best evaluation achieved during training. For each seed, evaluations are conducted over 5 different random goals, as designed in Park et al. [11], with the learned policy tested for 50 episodes per goal. Results within 95% of the best value are written in **bold**.

Environment	Dataset Type	Maze Dimension	GCIQL	Eik-GCIQL	GCIVL	Eik-GCIVL
pointmaze	navigate	medium	60 ± 1	59 ± 9	63 ± 6	$\mathbf{90 \pm 5}$
		large	39 ± 1	60 ± 9	38 ± 5	$\mathbf{82 \pm 39}$
		giant	0 ± 0	2 ± 4	0 ± 0	$\mathbf{86 \pm 11}$
		teleport	29 ± 5	25 ± 12	38 ± 5	$\mathbf{49 \pm 4}$
	stitch	medium	41 ± 11	56 ± 6	57 ± 9	$\mathbf{95 \pm 4}$
		large	25 ± 8	22 ± 3	11 ± 8	$\mathbf{67 \pm 9}$
		giant	0 ± 0	0 ± 0	0 ± 0	$\mathbf{23 \pm 10}$
		teleport	28 ± 5	25 ± 3	$\mathbf{41 \pm 5}$	$\mathbf{38 \pm 3}$
antmaze	navigate	medium	27 ± 4	25 ± 6	36 ± 5	$\mathbf{50 \pm 5}$
		large	9 ± 3	7 ± 2	$\mathbf{16 \pm 4}$	$\mathbf{15 \pm 3}$
		giant	0 ± 0	0 ± 0	0 ± 0	0 ± 0
		teleport	24 ± 3	23 ± 2	$\mathbf{32 \pm 5}$	$\mathbf{30 \pm 3}$
	stitch	medium	19 ± 4	21 ± 5	$\mathbf{25 \pm 4}$	$\mathbf{27 \pm 6}$
		large	6 ± 3	3 ± 3	$\mathbf{12 \pm 3}$	7 ± 2
		giant	0 ± 0	0 ± 0	0 ± 0	0 ± 0
		teleport	18 ± 5	23 ± 3	$\mathbf{30 \pm 3}$	$\mathbf{28 \pm 3}$

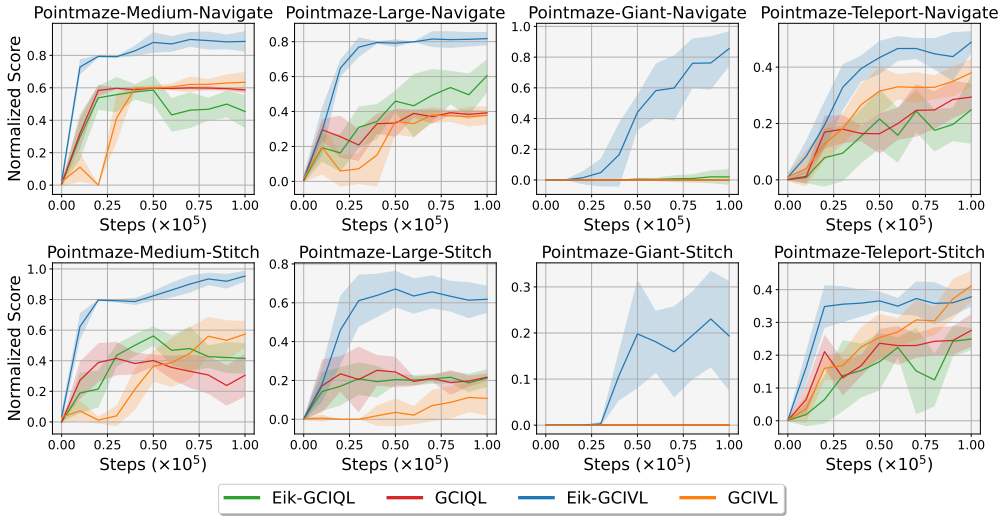


Figure 12: Learning curves for the `pointmaze` experiments in Table 4. Plots show the average success percentage per evaluation across seeds as a function of training steps.

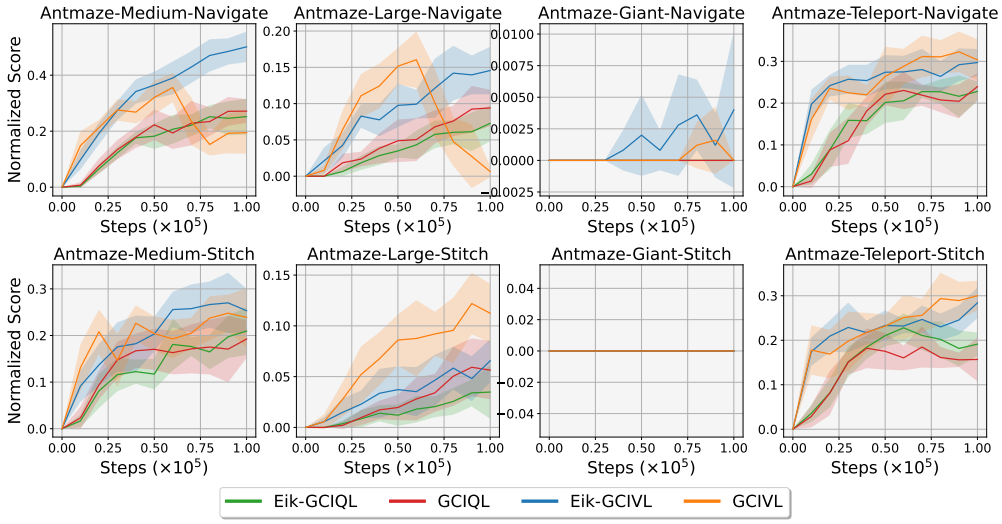


Figure 13: Learning curves for the `antmaze` experiments in Table 2. Plots show the average success percentage per evaluation across seeds as a function of training steps.