Dual Advantage Fields

Alexey Zemtsov[1,2], **Maxim Bobrin**[3], **Alexander Nikulin**[2,5], **Dmitry V. Dylov**[3], **Fakhri Karray**[4], **Vladislav Kurenkov**[5,6], **Martin Takáč**[4], **Arip Asadulaev**[4]
[1]NUST MISIS [2]MSU [3]Computational Imaging Lab
[4]MBZUAI [5]dunnolab [6]Innopolis University

Abstract

Offline goal-conditioned reinforcement learning requires both long-horizon reachability estimates and local action comparisons. Dual goal representations provide value fields that capture global goal reachability, but they do not directly specify which action should be preferred at a given state. We propose Dual Advantage Fields, a policy-extraction method that turns a bilinear dual value model into a local advantage signal. Under bilinear dual parameterization, the goal embedding is the gradient of the value field with respect to the state representation. DAF learns an action-effect model that predicts the discounted feature displacement induced by an action and scores actions by the alignment between this displacement and the goal direction. In the realizable case, this score equals the goal-conditioned Bellman advantage, yielding a standard local policy-improvement guarantee. On OGBench locomotion, manipulation, and puzzle tasks, DAF improves aggregate RLiable metrics and performs strongly in settings where locally correct actions differ from direct movement toward the final goal.

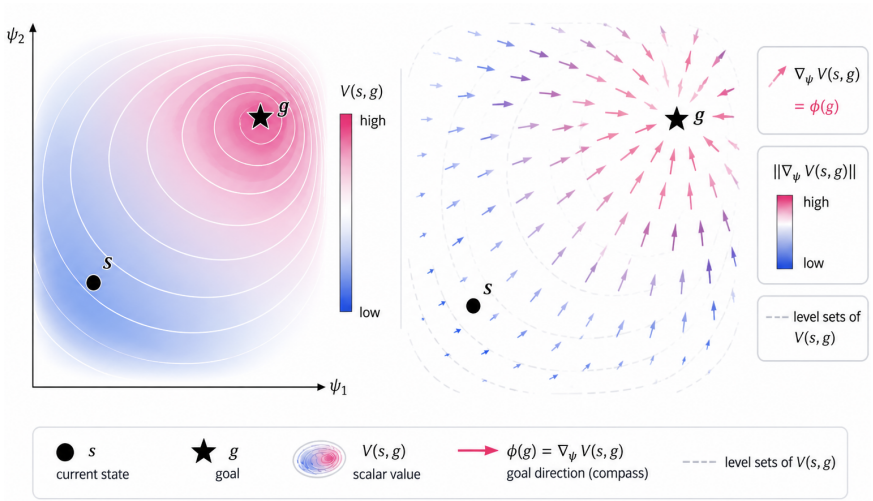

Figure 1: A dual goal-conditioned value model defines a global value surface over state representations. DAF converts this global surface into a local action-comparative signal by predicting how each action moves the state representation and measuring whether this movement aligns with the goal direction.

1 Introduction

Goal-conditioned reinforcement learning (GCRL) aims to learn policies that reach arbitrary goals from a fixed dataset of prior experience. This requires solving two different problems at once. First,

Accepted at ICML 2026 Workshop on Decision Making from Offline Datasets to Online Adaptation, Black-Box Optimization to Reinforcement Learning.

the agent must reason globally: it must infer how states are connected over long horizons so that behavior observed in one part of the dataset can be stitched together with behavior observed elsewhere. Second, the agent must act locally: at the current state, it must decide which available action makes the most progress toward the requested goal. A good goal-conditioned agent therefore needs both a global map of reachability and a local compass for action selection.

Recent dual goal representations provide a strong answer to the first problem. They parameterize a goal-conditioned value function as a bilinear interaction between a state embedding and a goal embedding $V_\theta(s, g) = \psi_\theta(s)^\top \phi_\theta(g)$. This structure induces a value surface for each goal, where states that are more reachable or desirable for the goal receive higher values. Such value surfaces are well suited for long-horizon reasoning: they encode temporal structure, support stitching across offline trajectories, and generalize across state-goal pairs. However, a value surface alone does not directly answer the local control question. It says how good the current state is for a goal, but not which action should be preferred among the actions available at that state.

This distinction is central in offline GCRL. Policy extraction requires an action-comparative signal. Two actions can start from the same state and therefore share the same value $V_\theta(s, g)$, while only one of them may move the agent toward the goal. What is missing is not another global estimate of reachability, but a local advantage-like quantity: a way to score whether an action changes the state in a direction that improves goal-conditioned value.

Our key observation is that this local signal is already implicit in the geometry of dual representations. Under the bilinear parameterization above, the goal embedding $\phi_\theta(g)$ is the direction in state-representation space along which the goal-conditioned value increases:

$$\nabla_\psi V_\theta(s, g) = \phi_\theta(g).$$

Thus, if an action induces a displacement in the state representation, its usefulness for the goal can be evaluated by a simple geometric test: does the predicted displacement align with the goal direction? This turns goal-conditioned policy improvement into a local alignment problem in the dual representation space.

We introduce *Dual Advantage Fields* (DAF), a policy-extraction method that makes this geometry explicit; see Figure 1. DAF learns an action-effect model that predicts the discounted change in the state representation caused by an action. It then scores actions by the inner product between this predicted action effect and the goal embedding. The resulting score is local, goal-conditioned, and action-comparative: it prefers actions whose predicted latent effect points in the direction of increasing value for the goal.

This perspective leads to a simple principle for offline GCRL: *global value fields should be paired with local advantage fields*. Dual representations provide the global map; DAF extracts from the same representation space the local compass needed for policy improvement. This yields an efficient actor-free mechanism for policy extraction: rather than learning a separate goal-conditioned action-value function, DAF reuses the geometry of the dual critic to obtain an advantage-like score for weighting offline actions.

Our contributions are:

- We show that, under the standard dual goal representation parameterization, the goal embedding can be interpreted as the gradient direction of the goal-conditioned value field with respect to the learned state representation.
- We introduce *Dual Advantage Fields*, which learn action-effect vectors and score actions by their alignment with this goal direction, producing a local advantage-like signal for goal-conditioned policy extraction.
- We use this signal to extract policies from offline data without training a separate goal-conditioned action-value function, and evaluate the resulting method across challenging offline GCRL benchmarks.

2 Preliminaries

Goal-conditioned Reinforcement Learning. We study *offline* goal-conditioned reinforcement learning (GCRL) [5, 14, 15, 18]: the learner has access to a fixed offline dataset of transitions

but cannot collect new experience in the environment [20]. The objective is to infer a an optimal goal-conditioned policy even for unseen during training combinations of state-goal pairs.

Let $\mathcal{S}$ and $\mathcal{A}$ denote state and action spaces, and let $\mathcal{G} \subseteq \mathcal{S}$ (or an abstract goal space) denote goals. At each step the environment emits a transition (s, a, s') according to an unknown Markov kernel $P(s' \,|\, s, a)$. A goal $g \in \mathcal{G}$ induces a reward signal $r(s, a, g)$: in sparse goal-reaching problems this is often zero until a success condition holds. A stochastic policy $\pi(a \,|\, s, g)$ induces the usual discounted return with discount $\gamma \in (0, 1)$. The goal-conditioned value and action-value functions are

$$Q^\pi(s, a, g) := \mathbb{E}_\pi\Big[\sum_{t=0}^{\infty} \gamma^t \, r(s_t, a_t, g) \;\Big|\; s_0 = s,\, a_0 = a\Big], \quad V^\pi(s, g) := \mathbb{E}_{a\sim\pi(\cdot|s,g)}\big[Q^\pi(s, a, g)\big].$$

$$Q^\pi(s, a, g) = \mathbb{E}_{s'\sim P(\cdot|s,a)}\big[r(s, a, g) + \gamma\, V^\pi(s', g)\big], \quad V^\pi(s, g) = \mathbb{E}_{a\sim\pi(\cdot|s,g)}\big[Q^\pi(s, a, g)\big]. \quad (1)$$

Recent GCRL methods combine several ideas, including representation learning, quasimetric objectives [11, 18, 25], and hierarchical horizon reduction [7, 17, 19] over value functions, Q-functions, and actors. These design choices are often complementary, but existing methods still show domain-specific strengths: hierarchical methods tend to excel in long-horizon locomotion, while quasimetric representations often work well for manipulation. In contrast, DAF emphasizes local policy improvement during training while retaining long-horizon reasoning, leading to more consistent performance across both domains.

Hierarchical Implicit Q-Learning (HIQL). In GCRL, accurately estimating the value function for distant goals is the main challenge in solving complex long-horizon tasks [19]. To address this issue, HIQL [19] proposed a hierarchical policy structure that utilizes a value function learned with IQL [13]. This hierarchical design enables the agent to produce effective actions even when value estimates for distant goals are noisy or unreliable. More specifically, HIQL trains a goal-conditioned state-value function V with the following loss:

$$\mathcal{L}(V) = \mathbb{E}_{(s,s')\sim\mathcal{D},\; g\sim p(g)}\left[L_2^\tau\left(r(s, g) + \gamma \bar{V}(s', g) - V(s, g)\right)\right], \quad (2)$$

where the expectile loss is defined as $L_2^\tau(u) = |\tau - \mathbf{1}(u < 0)|u^2$, with $\tau > 0.5$, and $\bar{V}$ denotes the target V network.[1] Following prior works [3, 19, 25], we adopt the sparse reward $r(s, g) = -\mathbf{1}\{s \neq g\}$. Under this reward, the optimal value $|V^\star(s, g)|$ corresponds to the *discounted temporal distance*, *i.e.*, a discounted measure of the minimum number of environment steps required to reach the goal g from state s. HIQL separates policy extraction[2] into two levels: a high-level policy $\pi^h(s_{t+k}|s_t, g)$ generates a k-step subgoal to guide progress toward the goal, while a low-level policy $\pi^\ell(a_t|s_t, s_{t+k})$ produces primitive actions to reach the subgoal. Both policies are extracted using advantage-weighted regression (AWR) [23, 26] with the following objective:

$$\mathcal{J}(\pi^h) = \mathbb{E}_{(s_t,s_{t+k},g)\sim\mathcal{D}}\left[\exp\left(\beta^h \cdot A^h(s_t, s_{t+k}, g)\right)\log \pi^h(s_{t+k}|s_t, g)\right], \quad (3)$$

$$\mathcal{J}(\pi^\ell) = \mathbb{E}_{(s_t,a_t,s_{t+1},s_{t+k})\sim\mathcal{D}}\left[\exp\left(\beta^\ell \cdot A^\ell(s_t, s_{t+1}, s_{t+k})\right)\log \pi^\ell(a_t|s_t, s_{t+k})\right], \quad (4)$$

where β^h and β^l are inverse temperature parameters, $A^h(s_t, s_{t+k}, g) = V^h(s_{t+k}, g) - V^h(s_t, g)$ denotes the high-level policy advantage, and $A^\ell(s_t, s_{t+1}, s_{t+k}) = V^\ell(s_{t+1}, s_{t+k}) - V^\ell(s_t, s_{t+k})$ denotes the low-level policy advantage. HIQL uses a single goal-conditioned value function V, which is shared between both π^h and π^ℓ (*i.e.*, $V^h = V^\ell = V$). However, despite this design, HIQL still struggles with long-horizon, complex tasks, as shown in the GCRL benchmark, OGBench [20].

Dual Goal Representations [22]. In goal-conditioned RL, the goal representation determines what information the policy and value function use about the target state. Rather than conditioning directly on the raw goal observation, which may contain irrelevant or exogenous factors, dual goal representations encode a goal by its reachability relation to other states. A goal g is represented by

$$\phi^\vee(g) : s \mapsto d^\star(s, g),$$

where $d^\star(s, g)$ denotes the optimal temporal distance from state s to goal g. In practice, we approximate this functional through a bilinear goal-conditioned potential [9]:

$$V_\theta(s, g) = \psi_\theta(s)^\top \phi_\theta(g), \quad (5)$$

where $\psi_\theta : \mathcal{S} \to \mathbb{R}^d$ and $\phi_\theta : \mathcal{G} \to \mathbb{R}^d$ are state and goal embeddings. The goal embedding $\phi_\theta(g)$ then serves as a finite-dimensional dual representation: when paired with $\psi_\theta(s)$, it predicts a value or distance-like quantity that reflects the environment's reachability structure.

[1] Since the inherent over-estimation problem of IQL, we assume that the environment dynamics is deterministic.

[2] Policy extraction refers to learning a policy from a learned value function, emphasizing the separation between value learning and policy learning.

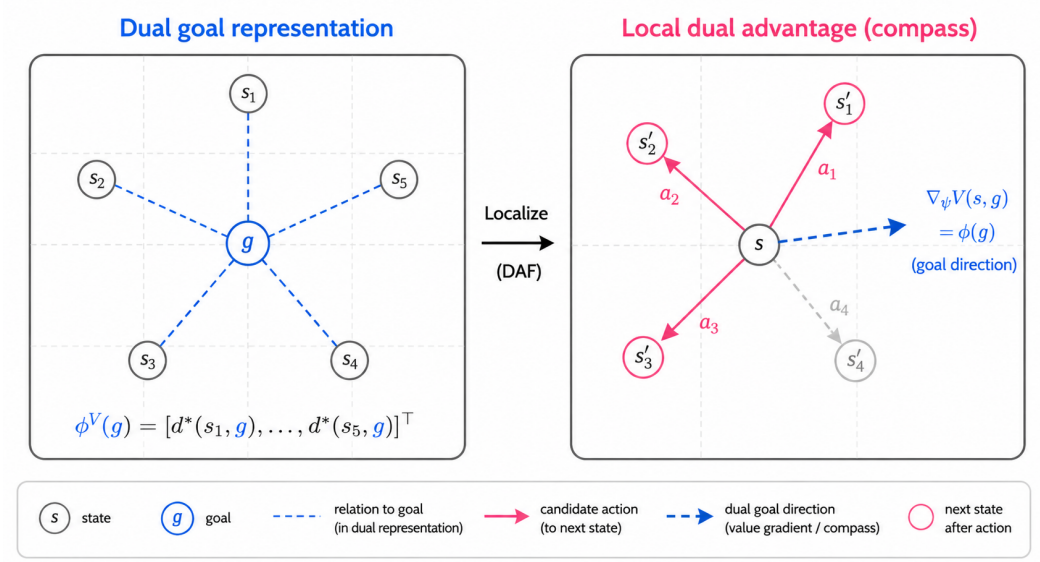

Figure 2: **Dual Advantage Fields.** Under a bilinear goal-conditioned value model, the goal embedding defines a direction in representation space. DAF scores an action by projecting its induced feature displacement onto this goal direction, yielding a local advantage-like signal for policy improvement.

3 Dual Advantage Fields

Our method is based on a simple insight from bilinear value decomposition in Eq. (5). Holding the goal fixed and viewing the value as a function of the state embedding ψ, we have

Proposition 3.1. *Under the bilinear goal-conditioned value model $V_\theta(s, g) = \psi_\theta(s)^\top \phi_\theta(g)$, the gradient of the value with respect to the state embedding is the goal embedding:*

$$\nabla_\psi V_\theta(s, g) = \nabla_\psi \big(\psi^\top \phi_\theta(g)\big) = \phi_\theta(g). \tag{6}$$

Thus, the goal embedding $\phi_\theta(g)$ is the value-gradient direction in representation space under the Euclidean geometry of the learned embedding. Please, see Figure 2 for intuition. For any transition from s to s', the change in the bilinear value is exactly

$$V_\theta(s', g) - V_\theta(s, g) = \phi_\theta(g)^\top \big(\psi_\theta(s') - \psi_\theta(s)\big). \tag{7}$$

We use this identity to construct an advantage-like local policy improvement signal. For a policy π, the standard goal-conditioned advantage is

$$A^\pi(s, a, g) = \mathbb{E}_{s' \sim p(\cdot|s,a)} \left[r(s, a, g) + \gamma V^\pi(s', g) - V^\pi(s, g) \right]. \tag{8}$$

Replacing V^π with the learned bilinear field V_θ gives the model-induced Bellman advantage

$$A_\theta(s, a, g) = \mathbb{E}_{s' \sim p(\cdot|s,a)} \left[r(s, a, g) + \phi_\theta(g)^\top \big(\gamma \psi_\theta(s') - \psi_\theta(s)\big) \right]. \tag{9}$$

In offline learning, each dataset transition (s, a, s') provides a sample estimate of this quantity:

Corollary 3.2. *The sample-level Dual Advantage Field score is*

$$\boxed{\widehat{A}_\theta(s, a, s', g) = r(s, a, g) + \phi_\theta(g)^\top \big(\gamma \psi_\theta(s') - \psi_\theta(s)\big).} \tag{10}$$

Local policy improvement. In the realizable case, the DAF score is exactly the goal-conditioned Bellman advantage. Specifically, if $V^\pi(s, g) = \psi(s)^\top \phi(g)$ and $u(s, a) = \mathbb{E}_{s' \sim P(\cdot|s,a)}[\gamma \psi(s') - \psi(s)]$, then

$$r(s, a, g) + u(s, a)^\top \phi(g) = A^\pi(s, a, g).$$

Thus, increasing the probability of actions (alignment) with positive DAF score is a standard policy-improvement step. Repeated exact DAF improvement therefore recovers an optimal primitive

goal-conditioned policy; in particular, its limiting policy is at least as good as any policy restricted to a fixed hierarchical class. We provide the formal statement and proof in Appendix F.1.

Equation (10) defines the DAF score. The term $\gamma\psi_\theta(s') - \psi_\theta(s)$ is the discounted feature displacement caused by action a, and $\phi_\theta(g)$ is the value-gradient direction toward goal g. Their inner product measures the one-step increase in the bilinear value field, with the reward term completing the Bellman advantage. Thus, $\widehat{A}_\theta$ provides a local, goal-conditioned action-ranking signal derived from the learned dual value geometry. This follows the comparative view of policy improvement, where actions are improved by relative advantages rather than absolute value estimates [4].

3.1 Motivational Example

We illustrate the local geometry captured by Dual Advantage Fields on the `cube-single-play-v0-task1` manipulation task from OGBench [20]. This task highlights a common failure mode in goal-conditioned control: before the cube can be placed at the final target, the agent must first move the gripper into a pre-grasp configuration. Thus, a direction that points directly toward the terminal object location may be globally plausible but locally unhelpful.

DAF addresses this by scoring actions according to their local improvement of the learned goal-conditioned potential. By Eq. (6), the goal embedding $\phi_\theta(g)$ is the representation-space gradient of the bilinear value field. We define an action-effect model $u_\xi(s, a)$ that estimates the discounted feature displacement induced by action a,

$$u_\xi(s, a) \approx \mathbb{E}_{s' \sim p(\cdot|s,a)} \left[\gamma\psi_\theta(s') - \psi_\theta(s)\right].$$

Ignoring reward terms that are constant across actions in the pre-grasp region, DAF scores actions by

$$z_\theta(s, a, g) = u_\xi(s, a)^\top \phi_\theta(g). \tag{11}$$

This score favors actions whose predicted feature displacement is aligned with the local direction of value increase toward the goal.

Figure 3 visualizes this effect. We sample query states $\{\tilde{s}_i\}_{i=1}^N$ by perturbing only the gripper position around the cube, while keeping the object state and final goal fixed. For each method $m \in \{\text{OTA}, \text{DAF}\}$, we decode its high-level subgoal prediction into an X-Y coordinates via probing,

$$\hat{x}_i^m = D_m(h_m(\tilde{s}_i, g)), \tag{12}$$

where h_m is the method-specific latent output and D_m is a linear probe fitted on demonstration states. The plotted direction is

$$d_i^m = \frac{\hat{x}_i^m - x_{\text{ee}}(\tilde{s}_i)}{\|\hat{x}_i^m - x_{\text{ee}}(\tilde{s}_i)\|_2}, \tag{13}$$

drawn from the gripper position $x_{\text{ee}}(\tilde{s}_i)$. Near the cube, DAF produces directions that point toward the object, matching the immediate pre-grasp behavior required before transport. OTA instead points toward the terminal placement region in this example, which is appropriate only after grasping. The example shows why local advantage fields can be more useful than a high-level subgoals alone: they select actions by whether they locally improve the goal-conditioned potential.

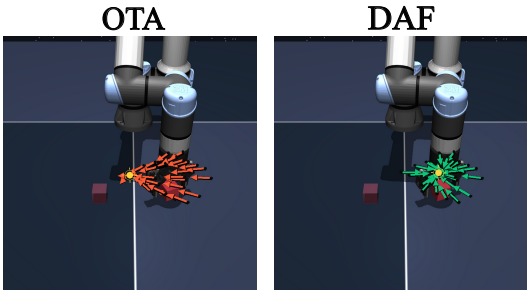

Figure 3: **Pre-grasp vector field in `cube-single`.** Arrows show decoded high-level directions from sampled gripper positions around the cube, with the cube and final goal fixed. DAF points locally toward the cube before grasping, while OTA points toward the terminal placement goal. The yellow marker denotes the mean decoded target.

4 Training and goal-conditioned policy extraction

Dayan and Singh [4] showed that policy improvement can be organized around *relative* measures of how actions compare at a state-*merits* that need not reduce to a fully trusted global value oracle. In the goal-conditioned setting, the Bellman advantage $A^\pi(s, a, g)$ in (8) is exactly such an object: it ranks a by the expected one-step gain in return, isolating the effect of the transition from the baseline

Algorithm 1 DAF training.

Input: offline dataset $\mathcal{D}$ of (s, a, s', g);
Initialize: ψ_θ, ϕ_θ, displacement map u_ξ, policy π_ω, target networks $(Q^{\mathrm{tgt}}, V^{\mathrm{tgt}})$.
while not converged **do**
  Sample a minibatch from $\mathcal{D}$.
  Critic: update ψ_θ, ϕ_θ so $V_\theta(s, g) = \psi_\theta(s)^\top \phi_\theta(g)$ (5) using target networks.
  AFU coupling: minimize the actor-free loss coupling V_θ to z_θ (14) {Appendix E} and minimize $\mathcal{L}_{\mathrm{ae}}$ (15) for u_ξ.
  Policy: $w_\theta \leftarrow \min\{\exp(\alpha z_\theta(s, a, g)), W_{\max}\}$; minimize $-\mathbb{E}_{\mathcal{D}}[w_\theta \log \pi_\omega(a \mid s, c)]$ over ω.
  Update target networks.
end while

$V^\pi(s, g)$. Our bilinear potential (5) turns this comparison into explicit geometry in ψ. Under the model V_θ, the backup contribution $\gamma V_\theta(s', g) - V_\theta(s, g)$ equals $\phi_\theta(g)^\top \big(\gamma\psi_\theta(s') - \psi_\theta(s)\big)$ by (7), so the analogue of the advantage (8) with V^π replaced by V_θ is the closed form (9)-(10). The goal embedding $\phi_\theta(g)$ acts as $\nabla_\psi V_\theta$ (Eq. (6)): the inner product in (10) measures whether the *local* feature displacement induced by a is aligned with steepest increase of the learned potential toward g. Thus Dayan's comparative view of improvement is instantiated here as projection of one-step ψ-dynamics onto the value-gradient direction.

In practice we estimate the discounted increment $\gamma\psi_\theta(s') - \psi_\theta(s)$ with a map $u_\xi(s, a)$ trained on offline transitions (Sec. 4.1), and absorb r in the critic stack where noted. The raw dual score is

$$z_\theta(s, a, g) \;:=\; u_\xi(s, a)^\top \phi_\theta(g)\,, \tag{14}$$

which agrees with (10) when $u_\xi(s, a) \approx \gamma\psi_\theta(s') - \psi_\theta(s)$ and rewards are handled by the value heads feeding the same Bellman targets.

4.1 Offline critic and feature dynamics

We learn $(\psi_\theta, \phi_\theta)$, and the displacement map u_ξ from offline tuples (s, a, s', g) [20]. For stability we used a common approach in offline RL [13] that learns twin critics $Q_\theta^{(1)}, Q_\theta^{(2)}$, and the bilinear head $V_\theta(s, g) = \psi_\theta(s)^\top \phi_\theta(g)$ is tied to pessimistic Q-estimates via expectile regression and to Bellman backups on $Q_\theta^{(j)}$. To avoid brittle $\max_a Q$ operators in continuous control [16], we add an *actor-free* coupling between V_θ and the scalar dual score z_θ from (14), following Perrin-Gilbert [24]; the explicit construction is deferred to Appendix E. Finally, we ground u_ξ with the auxiliary loss

$$\mathcal{L}_{\mathrm{ae}} \;=\; \mathbb{E}\Big[\big\|u_\xi(s, a) - \mathrm{sg}\big(\gamma\psi_\theta(s') - \psi_\theta(s)\big)\big\|_2^2\Big]\,, \tag{15}$$

with sg stopping gradients through the target, so u_ξ tracks one-step feature dynamics on $\mathcal{D}$.

4.2 Policy extraction

Let $\pi_\omega(a \mid s, c)$ denote the policy with conditioning c on g through $\phi_\theta(g)$ (and optionally s). Advantage-weighted regression [23] uses weights

$$w_\theta(s, a, g) \;=\; \min\Big\{\exp\big(\alpha\, z_\theta(s, a, g)\big),\, W_{\max}\Big\} \tag{16}$$

with temperature $\alpha > 0$ and cap $W_{\max}$, and minimizes $-\mathbb{E}_{\mathcal{D}}[w_\theta \log \pi_\omega(a \mid s, c)]$. Because z_θ does not depend on ω, this is weighted behavior cloning that up-weights actions whose local ψ-displacement aligns with the goal direction $\phi_\theta(g)$, i.e. actions that the bilinear model classifies as improving the goal-conditioned potential in the sense of (10).

Hierarchical goals. For long horizons, a high-level policy over subgoals can be trained alongside the low-level stack above, with value differences along options as in hierarchical offline GCRL [19]; option-aware temporally abstracted value learning offers a related hierarchical baseline [2].

Remark 4.1 (Variants). Concrete instantiations differ by which of the optional terms above are active; experimental details are summarized in Section 5 and Appendix A.

5 Experiments

In this section, we empirically validate the findings developed in the previous sections on the OGBench benchmark [20]. OGBench is designed to evaluate several core capabilities required by offline goal-conditioned reinforcement learning, including long-horizon reasoning, trajectory stitching, generalization to unseen goals, robustness to suboptimal data, and control under imperfect offline coverage. We focus on the state-based locomotion and manipulation tasks used in prior work, which allows us to test whether DAF provides consistent improvements across domains with substantially different control structure.

All methods are trained purely offline on the provided datasets and are evaluated without additional environment interaction during training. We report success-based performance in $[0, 1]$, where higher values indicate better goal reaching. For each environment, we evaluate the corresponding OGBench dataset regimes. In maze-style locomotion, we use `navigate` and `stitch` datasets: `navigate` data is collected from noisy expert policies that traverse the environment, while `stitch` data contains shorter trajectory segments and therefore requires composing partial behaviors into longer goal-reaching solutions. In manipulation, we use `play` and `noisy` datasets: `play` data contains natural temporally correlated interactions generated by scripted policies, whereas `noisy` data increases state-action coverage through less structured exploration noise, making the offline data more suboptimal.

Baselines. We compare against a representative set of recent and relevant methods for offline GCRL, including HIQL [19], OTA [2], MQE [17], CRL [5], GCIQL [13], and GCIVL [10]. When applicable, we also include their corresponding variants that learn representations in the form of dual-goal representations [22]. These baselines cover the main families of methods used in offline GCRL, including horizon-reduction methods [21] and methods based on representation priors such as quasimetrics.

What DAF does in each dataset. Across all datasets, DAF uses the same policy-extraction principle: it scores offline actions by the alignment between their predicted local feature displacement and the goal direction induced by the dual value representation. Concretely, the action-effect model estimates $\gamma\psi_\theta(s') - \psi_\theta(s)$, and the dual score projects this displacement onto $\phi_\theta(g)$. Thus, DAF uses the learned value field not only as a global map of reachability, but also as a local compass for choosing among actions available in the offline dataset.

Maze locomotion: long-horizon navigation and stitching. We first evaluate on `humanoidmaze` and `antmaze`, shown in Table 1. These environments isolate the long-horizon navigation aspect of offline GCRL. The agent must reach target states from diverse initial states using only fixed offline data. The `antmaze` tasks require quadruped locomotion through maze layouts, while `humanoidmaze` is more challenging because it combines full-body humanoid control with long-horizon goal reaching. We include both `navigate` and `stitch` variants because they test complementary capabilities: `navigate` evaluates whether the method can exploit noisy expert trajectories, while `stitch` evaluates whether the method can compose shorter trajectory fragments into successful long-horizon behavior.

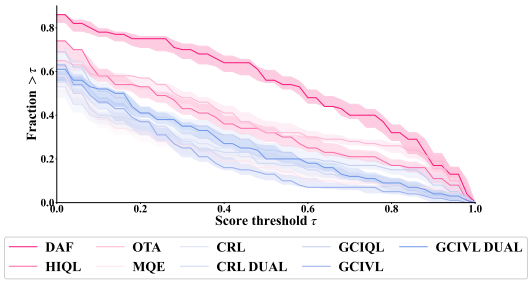

Figure 4: Performance profile across all tasks and environments. DAF achieves a better distribution of scores than the baselines across the OGBench evaluation suite.

These tasks are important because many prior offline GCRL methods are designed around horizon reduction or hierarchical subgoal prediction. DAF is not primarily a hierarchical method: instead, it extracts local action preferences from a dual value field. Strong performance on these mazes therefore tests whether local advantage-field extraction can preserve the long-horizon structure needed for navigation. DAF is competitive with the strongest horizon-reduction baselines on `navigate` datasets and obtains the best results on the harder `stitch` cases where composing partial trajectories is essential.

Table 1: Maze locomotion results on `humanoidmaze` and `antmaze`. These tasks test long-horizon goal reaching from fixed offline data. The `navigate` datasets evaluate learning from noisy expert trajectories, while the `stitch` datasets evaluate whether a method can compose shorter trajectory segments into successful goal-reaching behavior. Results within 95% of the best value are written in **bold**. Best values are highlighted in blue.

Env.	Dataset	Dimension	DAF	HIQL	OTA	MQE	CRL	CRL DUAL	GCIQL	GCIVL	GCIVL DUAL
HUMANOIDMAZE	NAVIGATE	MEDIUM	**0.93** ±0.03	**0.91** ±0.01	**0.95** ±0.01	0.49 ±0.09	0.59 ±0.03	0.62 ±0.03	0.31 ±0.04	0.31 ±0.03	0.32 ±0.03
		LARGE	0.66 ±0.03	0.45 ±0.04	**0.83** ±0.03	0.20 ±0.07	0.26 ±0.03	0.21 ±0.05	0.04 ±0.01	0.05 ±0.01	0.04 ±0.01
	STITCH	MEDIUM	**0.90** ±0.04	0.86 ±0.03	**0.92** ±0.01	0.62 ±0.09	0.53 ±0.03	0.57 ±0.01	0.15 ±0.03	0.14 ±0.01	0.20 ±0.02
		LARGE	**0.48** ±0.06	0.32 ±0.04	0.43 ±0.04	0.18 ±0.03	0.11 ±0.02	0.06 ±0.03	0.02 ±0.00	0.02 ±0.01	0.02 ±0.01
ANTMAZE	NAVIGATE	TELEPORT	0.51 ±0.08	0.46 ±0.03	0.53 ±0.03	0.49 ±0.04	**0.60** ±0.01	**0.57** ±0.04	0.37 ±0.02	0.44 ±0.02	0.41 ±0.02
		MEDIUM	**0.98** ±0.01	**0.96** ±0.01	**0.97** ±0.01	0.88 ±0.05	**0.96** ±0.01	**0.95** ±0.02	0.76 ±0.06	0.72 ±0.06	0.79 ±0.03
	STITCH	TELEPORT	**0.50** ±0.05	0.38 ±0.02	0.38 ±0.01	0.40 ±0.03	0.23 ±0.03	0.15 ±0.04	0.23 ±0.02	0.45 ±0.05	0.39 ±0.05
		MEDIUM	**0.97** ±0.02	**0.96** ±0.02	**0.95** ±0.02	**0.96** ±0.01	0.52 ±0.04	0.52 ±0.07	0.39 ±0.06	0.48 ±0.03	0.52 ±0.03

Table 2: Object-manipulation results on `cube` and `scene`. These datasets test whether an offline GCRL method can extract precise local skills from `play` data and remain robust under the less structured `noisy` regime. Results within 95% of the best value are written in **bold**. Best values are highlighted in blue.

Env.	Dataset	Dimension	DAF	HIQL	OTA	MQE	CRL	CRL DUAL	GCIQL	GCIVL	GCIVL DUAL
CUBE	PLAY	DOUBLE	0.41 ±0.04	0.13 ±0.01	0.05 ±0.01	0.03 ±0.00	0.16 ±0.01	0.38 ±0.06	0.35 ±0.06	0.33 ±0.05	**0.58** ±0.04
		TRIPLE	**0.17** ±0.03	0.05 ±0.02	0.02 ±0.00	0.01 ±0.00	0.06 ±0.02	0.05 ±0.05	0.02 ±0.01	0.01 ±0.01	0.01 ±0.00
		QUADRUPLE	**0.03** ±0.01	0.00 ±0.00	0.00 ±0.00	0.00 ±0.00	0.00 ±0.00	0.00 ±0.00	0.00 ±0.00	0.00 ±0.00	0.00 ±0.00
	NOISY	DOUBLE	**0.33** ±0.05	0.03 ±0.01	0.05 ±0.03	0.07 ±0.01	0.04 ±0.02	0.08 ±0.02	0.24 ±0.06	0.17 ±0.04	0.26 ±0.02
		TRIPLE	**0.23** ±0.01	0.04 ±0.01	0.01 ±0.00	0.04 ±0.02	0.03 ±0.01	0.06 ±0.02	0.05 ±0.01	0.11 ±0.02	0.09 ±0.03
		QUADRUPLE	**0.02** ±0.01	0.00 ±0.00	0.00 ±0.00	0.00 ±0.00	0.00 ±0.00	0.01 ±0.01	0.00 ±0.00	0.00 ±0.00	0.00 ±0.00
SCENE	PLAY		**0.81** ±0.04	0.55 ±0.09	0.34 ±0.04	0.20 ±0.03	0.29 ±0.02	0.56 ±0.06	0.53 ±0.02	0.51 ±0.05	**0.78** ±0.07
	NOISY		**0.43** ±0.03	0.27 ±0.02	0.10 ±0.02	0.07 ±0.02	0.02 ±0.01	0.06 ±0.01	0.29 ±0.02	0.31 ±0.05	**0.45** ±0.02

Object manipulation: local control from imperfect demonstrations. Next, we evaluate on `cube` and `scene`, shown in Table 2. Unlike maze navigation, these tasks require precise object-centric control. The `cube` tasks include pick-and-place, stacking, swapping, and multi-object rearrangement, while `scene` tasks require sequencing interactions with objects such as cubes, drawers, windows, and buttons.

These datasets test DAF's central motivation: globally plausible behavior can be locally wrong. For example, moving toward a final object placement may be inappropriate before reaching a pre-grasp state. DAF addresses this by ranking dataset actions according to whether their predicted feature displacement aligns with the goal direction. This is especially useful in `play` and `noisy` datasets, where demonstrations contain useful local skills but also incomplete or suboptimal trajectories.

Puzzle rearrangement: continuous control with combinatorial structure. Finally, we evaluate on `puzzle`, shown in Table 3. These environments are robotic versions of Lights Out: pressing one button changes the state of neighboring buttons. They therefore combine continuous control with combinatorial generalization over discrete configurations. The `3x3` and `4x4` variants further increase the configuration space, testing whether goal representations generalize beyond simple object reaching.

Puzzle tasks stress a failure mode not captured by maze navigation or standard manipulation. Here, each local action can affect a larger configuration, so policy extraction must compare actions by their downstream effect on the goal. DAF is suited to this setting because it scores actions by whether their predicted local transition improves the goal-conditioned value field.

Aggregate comparison. Tables 1, 2, and 3 show that DAF performs strongly across different kinds of offline coverage and control structure. The aggregate comparison in Figure 5 further summarizes performance across all tasks using RLiable metrics [1]. We report Median, interquartile mean (IQM), Mean, and Optimality Gap with stratified-bootstrap confidence intervals. The IQM reduces sensitivity to outlier tasks, while the optimality gap measures the average remaining shortfall from perfect success. Overall, DAF improves the aggregate metrics while also achieving strong per-task performance, indicating that the gains are not driven by a single environment family.

Table 3: Puzzle rearrangement results on `puzzle`. These tasks test structured spatial reasoning: each local button press changes neighboring button states, so the policy must combine continuous control with combinatorial goal generalization. Results within 95% of the best value are written in **bold**. Best values are highlighted in blue.

Env.	Dataset	Dimension	DAF	HIQL	OTA	MQE	CRL	CRL DUAL	GCIQL	GCIVL	GCIVL DUAL
PUZZLE	PLAY	3x3	0.74 ±0.04	0.17 ±0.03	0.64 ±0.06	0.11 ±0.00	0.07 ±0.01	0.09 ±0.03	**0.98** ±0.02	0.08 ±0.02	0.08 ±0.01
		4x4	0.40 ±0.05	0.17 ±0.03	**0.53** ±0.05	0.17 ±0.03	0.02 ±0.01	0.07 ±0.02	0.31 ±0.02	0.26 ±0.02	0.30 ±0.05
	NOISY	3x3	**0.98** ±0.04	0.70 ±0.10	0.64 ±0.13	0.03 ±0.01	0.37 ±0.05	0.42 ±0.05	**0.95** ±0.01	0.44 ±0.15	0.50 ±0.22
		4x4	**0.47** ±0.03	0.31 ±0.07	0.01 ±0.00	0.02 ±0.01	0.00 ±0.00	0.00 ±0.00	0.33 ±0.11	0.24 ±0.02	0.25 ±0.02

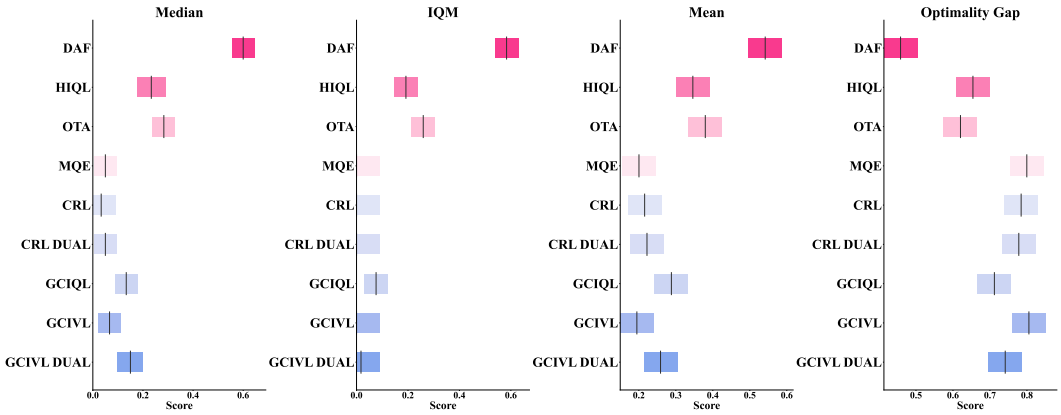

Figure 5: **Performance comparison.** Following the protocol proposed by Agarwal et al. [1], we report aggregate RLiable metrics, including Median, IQM, Mean, and Optimality Gap, with stratified-bootstrap confidence intervals across the offline GCRL environments. The colored horizontal segments denote confidence intervals, and the dark vertical markers denote point estimates.

6 Broader Impact and Limitations

DAF extracts goal-conditioned policies from offline data without additional environment interaction. Like other offline RL methods, it is reliable only when the dataset sufficiently covers the actions needed for improvement; poor coverage can produce incorrect action rankings. DAF also relies on the learned dual representation and action-effect model.

Although $\nabla_\psi V_\theta(s, g) = \phi_\theta(g)$ holds exactly for the bilinear head, this direction is useful only if the representation encodes reachability. In stochastic or poorly covered regions, $u_\xi(s, a)$ may predict inaccurate feature displacements. Future work should study uncertainty-aware or distributional action-effect models and extend DAF to image-based and more stochastic goal-reaching settings.

7 Conclusion

We introduced Dual Advantage Fields, a method that turns dual goal representations into local policy-improvement signals. Under the bilinear value parameterization, the goal embedding acts as the gradient of the goal-conditioned value field with respect to the state representation. DAF uses this observation to score actions by the alignment between their predicted feature displacement and the goal direction.

Empirically, DAF improves aggregate performance across offline GCRL benchmarks and is especially effective in manipulation tasks where local directional choices are important. Overall, the results suggest that dual representations should be used not only as global value maps, but also as local advantage fields for goal-conditioned policy extraction.

References

[1] Rishabh Agarwal, Max Schwarzer, Pablo Samuel Castro, Aaron C. Courville, and Marc G. Bellemare. Deep reinforcement learning at the edge of the statistical precipice. In *Advances in*

Neural Information Processing Systems (NeurIPS), 2021.

[2] Hongjoon Ahn, Heewoong Choi, Jisu Han, and Taesup Moon. Option-aware temporally abstracted value for offline goal-conditioned reinforcement learning, 2025. URL https://arxiv.org/abs/2505.12737.

[3] 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.

[4] Peter Dayan and Satinder P. Singh. Improving policies without measuring merits. In Gerald Tesauro, David Touretzky, and Todd Leen, editors, *Advances in Neural Information Processing Systems*, volume 8. MIT Press, 1995. URL https://proceedings.neurips.cc/paper/1995/hash/208e43f0e45c4c78cafadb83d2888cb6-Abstract.html.

[5] Benjamin Eysenbach, Ruslan Salakhutdinov, and Sergey Levine. Contrastive learning as goal-conditioned reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 35, 2022.

[6] Dibya Ghosh, Chethan Anand Bhateja, and Sergey Levine. Reinforcement learning from passive data via latent intentions. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 11321–11339. PMLR, 2023.

[7] Vittorio Giammarino and Ahmed H Qureshi. Goal reaching with eikonal-constrained hierarchical quasimetric reinforcement learning. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=5WhsCB0Vty.

[8] Dan Hendrycks and Kevin Gimpel. Gaussian error linear units (gelus). *arXiv preprint arXiv:1606.08415*, 2016.

[9] Zhang-Wei Hong, Ge Yang, and Pulkit Agrawal. Bilinear value networks. In *International Conference on Learning Representations*, 2022. arXiv:2204.13695.

[10] Kaiqiang Ke, Qian Lin, Zongkai Liu, Shenghong He, and Chao Yu. Conservative offline goal-conditioned implicit v-learning. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=5ryn8tYWHL.

[11] Kaiqiang Ke, Zhonghai Ruan, Shengwen Tan, and Weixia Wu. Hierarchical quasimetric reinforcement learning. In *Proceedings of the 2025 International Conference on Machine Learning and Neural Networks*, pages 34–41, 2025.

[12] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*, 2015.

[13] Ilya Kostrikov, Ashvin Nair, and Sergey Levine. Offline reinforcement learning with implicit Q-learning. In *International Conference on Learning Representations*, 2022.

[14] Minghuan Liu, Menghui Zhu, and Weinan Zhang. Goal-conditioned reinforcement learning: Problems and solutions. *arXiv preprint arXiv:2201.08299*, 2022.

[15] 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.

[16] Gabriel Matheron, Nicolas Perrin, and Olivier Sigaud. Understanding the impact of the max operation in value-based deep reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 33, 2020.

[17] Vivek Myers, Chongyi Zheng, Anca Dragan, Sergey Levine, and Benjamin Eysenbach. Learning temporal distances: Contrastive successor features can provide a metric structure for decision-making. *arXiv preprint arXiv:2406.17098*, 2024.

[18] Vivek Myers, Bill Chunyuan Zheng, Benjamin Eysenbach, and Sergey Levine. Offline goal-conditioned reinforcement learning with quasimetric representations. *arXiv preprint arXiv:2509.20478*, 2025.

[19] Seohong Park, Dibya Ghosh, Benjamin Eysenbach, and Sergey Levine. HIQL: Offline goal-conditioned RL with latent states as actions. In *Advances in Neural Information Processing Systems*, 2023. arXiv:2307.11949.

[20] Seohong Park, Kevin Frans, Benjamin Eysenbach, and Sergey Levine. OGBench: Benchmarking offline goal-conditioned RL. In *International Conference on Learning Representations*, 2025. arXiv:2410.20092.

[21] Seohong Park, Kevin Frans, Deepinder Mann, Benjamin Eysenbach, Aviral Kumar, and Sergey Levine. Horizon reduction makes rl scalable. *arXiv preprint arXiv:2506.04168*, 2025.

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

[23] Xue Bin Peng, Aviral Kumar, Grace Zhang, and Sergey Levine. Advantage-weighted regression: Simple and scalable off-policy reinforcement learning, 2019. URL https://arxiv.org/abs/1910.00177.

[24] Nicolas Perrin-Gilbert. AFU: Actor-free critic updates in off-policy RL for continuous control, 2024. URL https://arxiv.org/abs/2404.16159.

[25] 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.

[26] 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.

Table 4: Network configuration for DAF on OGBench.

Configuration	Value
Gradient steps	10^6
Optimizer	Adam [12]
Nonlinearity	GELU [8]
Target network update rate	0.005
Goal representation dimensionality	256
Batch size	1024
Action-effect MLP dimensions	$(512, 512, 512)$
Policy MLP dimensions	$(512, 512, 512)$
Layer norm in MLPs	`True`
Discount (γ)	0.99 (0.995 for `giant-` environments)
Learning rate	0.0003

Table 5: Coefficient α for each environment

Environment	α
`scene-*`	10.0
`antmaze`	10.0
`humanoidmaze`	10.0
`puzzle-3x3-*`	3.0
`puzzle-4x4-*`	0.1
`cube-double-*`	3.0
`cube-triple-*`	3.0
`cube-quadruple-*`	10.0

A Implementation and Reproducibility

Our method and baselines are implemented on top of the implementations given in OGBench [20] and Dual Goal Representations [22] codebases. Our method is employed upon hierachy of actors, with low actor being updated by dual score (Equation (14)) and high actor by AWR (Equation (3)).

Table 4 details the common hyperparameters for all methods on OGBench. Table 5 shows the α regularization hyperparameter that was found to be the best for performance of DAF. We also report the ablation studies on important architectural aspects of our proposed method: AFU Coupling [24], presence of action-effect module (Equation (15)), hierarchical actor [19] and integrating dual representations [22] upon the hierarchical backbone.

B Related Works

B.1 Goal-conditioned Implicit Q-Learning (GCIQL)

Implicit Q-Learning (IQL) [13] stabilizes offline RL by avoiding queries to out-of-distribution (OOD) actions through two key components: a state-value function $V_\psi(s)$ and an action-value function $Q_\theta(s, a)$. The value functions are trained via:

$$\mathcal{L}_Q(\theta) = \mathbb{E}_{(s,a,s')\sim\mathcal{D}}\left[\left(r(s,a) + \gamma V_\psi(s') - Q_\theta(s,a)\right)^2\right], \tag{17}$$

$$\mathcal{L}_V(\psi) = \mathbb{E}_{(s,a)\sim\mathcal{D}}\left[L_2^\tau\left(Q_{\bar{\theta}}(s,a) - V_\psi(s)\right)\right], \tag{18}$$

where $L_2^\tau(x) = |\tau - \mathbb{1}(x < 0)|x^2$ and $\tau \in [0.5, 1)$ controls conservatism (higher τ prioritizes optimistic returns), and $\bar{\theta}$ are the parameters of the target Q network. The policy $\pi_\phi(a|s)$ is then extracted via advantage-weighted regression (AWR) [23, 26]:

Table 6: Ablation on OGBench. We report full DAF and ablations: removing AFU coupling, removing the action-effect model (using direct one-step value-difference scoring), removing hierarchy, and using a dual-representation hierarchical baseline. Results within 95% of the best value are written in **bold**. Best values are highlighted in blue.

Env.	Dataset	Dimension	DAF	No AFU Coupling	No Action-Effect	DAF w/o Hierarchy	Dual-Rep Baseline + Hierarchy
HUMANOIDMAZE	NAVIGATE	MEDIUM	**0.93** ±0.03	0.18 ±0.03	0.35 ±0.03	0.38 ±0.04	0.06 ±0.01
		LARGE	**0.66** ±0.03	0.02 ±0.01	0.04 ±0.01	0.39 ±0.05	0.01 ±0.00
	STITCH	MEDIUM	**0.90** ±0.04	0.47 ±0.06	0.50 ±0.07	0.32 ±0.06	0.05 ±0.02
		LARGE	0.48 ±0.06	0.06 ±0.02	0.08 ±0.01	**0.65** ±0.07	0.01 ±0.01
ANTMAZE	NAVIGATE	TELEPORT	**0.51** ±0.08	0.33 ±0.05	0.35 ±0.05	0.39 ±0.05	0.18 ±0.07
		MEDIUM	**0.98** ±0.01	0.78 ±0.05	0.93 ±0.05	0.19 ±0.04	0.89 ±0.02
	STITCH	TELEPORT	**0.50** ±0.05	0.19 ±0.04	0.17 ±0.04	0.00 ±0.01	0.09 ±0.02
		MEDIUM	**0.97** ±0.02	0.42 ±0.05	0.42 ±0.10	0.00 ±0.01	0.21 ±0.06
CUBE	PLAY	DOUBLE	0.41 ±0.04	0.36 ±0.07	**0.51** ±0.05	0.39 ±0.05	0.02 ±0.01
		TRIPLE	**0.17** ±0.03	0.02 ±0.01	0.04 ±0.02	0.07 ±0.02	0.01 ±0.01
		QUADRUPLE	**0.03** ±0.01	0.00 ±0.00	0.00 ±0.00	0.01 ±0.01	0.00 ±0.00
	NOISY	DOUBLE	0.33 ±0.05	0.26 ±0.05	**0.39** ±0.03	0.35 ±0.04	0.03 ±0.01
		TRIPLE	**0.23** ±0.01	0.01 ±0.01	0.05 ±0.02	0.02 ±0.01	0.01 ±0.00
		QUADRUPLE	**0.02** ±0.01	0.00 ±0.00	0.00 ±0.00	0.01 ±0.00	0.00 ±0.00
SCENE	PLAY	-	**0.81** ±0.04	0.49 ±0.05	0.52 ±0.01	0.45 ±0.04	0.20 ±0.07
	NOISY	-	**0.43** ±0.03	0.29 ±0.03	0.40 ±0.04	0.37 ±0.05	0.05 ±0.02
PUZZLE	PLAY	3x3	**0.74** ±0.04	0.10 ±0.01	0.15 ±0.01	0.02 ±0.01	0.07 ±0.02
		4x4	0.40 ±0.05	0.18 ±0.03	0.17 ±0.03	**0.75** ±0.06	0.01 ±0.00
	NOISY	3x3	**0.98** ±0.04	0.15 ±0.03	0.17 ±0.02	0.37 ±0.06	0.05 ±0.01
		4x4	**0.47** ±0.03	0.07 ±0.01	0.09 ±0.01	0.03 ±0.01	0.00 ±0.01

$$J_\pi(\phi) = \mathbb{E}_{(s,a)\sim\mathcal{D}}\left[\exp\left(\beta \cdot A(s,a)\right)\log \pi_\phi(a|s)\right], \tag{19}$$

with $A(s,a) = Q_\theta(s,a) - V_\psi(s)$, and β is the inverse temperature parameter.

For goal-conditioned RL, IQL is extended to learn a goal-conditioned state-value function $V_\psi(s,g)$, preserving IQL's key advantage of stable value learning without requiring explicit Q-function evaluations on out-of-distribution actions [6].

B.2 Option-aware Temporally Abstracted Value (OTA)

HIQL [19] addresses long horizons by introducing a hierarchy over subgoals, but still relies on flat temporal-difference updates to a high-level value. OTA [2] instead bakes temporal abstraction directly into the Bellman operator by learning *option-aware* values: for an option o that lasts $k(o)$ steps, the high-level Bellman target becomes

$$V(s,g) \approx \mathbb{E}\big[r^{(o)}(s,g) + \gamma^{k(o)} V(s',g)\big], \tag{20}$$

where $r^{(o)}$ is the cumulative option reward and s' is the option-termination state. Each update contracts the effective horizon from $d^\star(s,g)$ to roughly $d^\star(s,g)/k(o)$, so value differences and the corresponding high-level advantages are computed over multi-step options rather than single primitive actions. This leads to more stable high-level signals and better long-horizon stitching on OGBench, at the cost of committing to a particular temporal abstraction schedule.

B.3 Quasimetric representations and MQE-style methods

Recent work views goal-conditioned value learning as estimating an asymmetric "distance" $d(s,g)$ between states and goals. Quasimetric approaches [18, 25] directly fit such distances with multistep returns: instead of bootstrapping only from immediate successors, they regress

$$d_\theta(s,g) \approx \mathbb{E}\Big[\sum_{t=0}^{K-1} c(s_t,g) + d_\theta(s_K,g) \;\Big|\; s_0 = s\Big] \tag{21}$$

for random horizons K, while encouraging triangle-like inequalities $d_\theta(s,g) \le d_\theta(s,\tilde{g}) + d_\theta(\tilde{g},g)$ for sampled pivots $\tilde{g}$. This multistep quasimetric estimation (MQE) improves horizon generalization—including long-horizon stitching in visual domains—but typically requires stronger structural assumptions on the value landscape than local TD methods and can be sensitive to misspecification of the quasimetric prior.

B.4 Conservative goal-conditioned implicit V-learning (GCIVL)

GCIQL-style methods extend IQL to goal-conditioned settings but can overestimate values for *unconnected* state–goal pairs produced by cross-trajectory pairing. GCIVL [10] introduces conservative penalties on such pairs together with a quasimetric formulation. Concretely, for a learned value or distance $v_\theta(s, g)$ and a connectivity indicator $c(s, g) \in \{0, 1\}$ (reachable from $\mathcal{D}$), the GCIVL loss augments Bellman terms with

$$\mathcal{L}_{\text{cons}}(\theta) = \lambda\, \mathbb{E}_{(s,g)\sim p_{\text{pair}}}\big[(1 - c(s, g))\,\big(\max\{0, v_\theta(s, g) - \delta\}\big)^2\big], \tag{22}$$

penalizing large estimates on likely-unreachable pairs. This improves robustness on goal-stitching tasks in OGBench, but depends on correctly identifying or regularizing unreachable pairs and still operates on scalar values rather than local action-effect structure.

B.5 Contrastive representation learning (CRL)

Contrastive RL methods treat goal-conditioned control as a representation learning problem: they learn embeddings so that inner products between state(-action) and goal features approximate a goal-conditioned value or reachability score [5]. A typical loss takes the form

$$\mathcal{L}_{\text{CRL}} = -\mathbb{E}\left[\log \frac{\exp\big(\phi(s, a)^\top \psi(g^+)/\tau\big)}{\sum_{g' \in \mathcal{N}} \exp\big(\phi(s, a)^\top \psi(g')/\tau\big)}\right], \tag{23}$$

where (s, a, g^+) is a positive triple and $\mathcal{N}$ is a set of negatives. Policies then act by choosing actions whose embeddings are closest to the goal embedding. These approaches can learn powerful, task-agnostic representations from unlabeled trajectories, but the contrastive loss is global rather than local in the sense of our dual advantage field: it encourages correct ordering over large batches of positive and negative pairs without explicitly privileging one-step action-induced displacements in representation space.

C Additional Environment and Evaluation Details

OGBench environments. We evaluate on goal-conditioned offline reinforcement learning tasks from OGBench [20]. OGBench is designed to test several capabilities that are central to offline GCRL, including long-horizon reasoning, trajectory stitching, generalization to unseen goals, robustness to suboptimal data, and control under stochasticity. In our main experiments, we focus on the state-based locomotion and manipulation tasks used in prior work.

The locomotion tasks include maze-style navigation domains such as `pointmaze`, `antmaze`, and `humanoidmaze`, as well as `antsoccer`. These tasks require the agent to reach target goal states from diverse initial configurations using only offline data. The difficulty varies with maze size, agent morphology, and dataset coverage. In particular, `humanoidmaze` requires full-body control and therefore combines low-level locomotion with long-horizon navigation, while `antsoccer` additionally requires controlling a ball while navigating.

The manipulation tasks include `cube`, `scene`, and `puzzle`. The `cube` environments test basic object manipulation through pick-and-place, stacking, swapping, and rearrangement of colored cubes. The `scene` environment contains multiple interacting objects, such as a cube, drawer, window, and buttons, and therefore requires sequencing several atomic behaviors to achieve the desired goal configuration. The `puzzle` environments instantiate a robotic version of the Lights Out puzzle, where pressing one button changes the state of neighboring buttons. These tasks are particularly challenging because the agent must combine continuous robotic control with combinatorial generalization over many possible configurations.

Dataset variants. For each environment, OGBench provides multiple dataset variants that differ in coverage, trajectory quality, and the extent to which successful behavior can be recovered directly from the dataset. In maze-style locomotion tasks, `navigate` datasets are collected from noisy expert policies that traverse the environment, while `stitch` datasets contain shorter trajectory segments and require the policy to compose partial behaviors into longer goal-reaching trajectories. Some locomotion domains also provide `explore` datasets, which contain highly exploratory and substantially suboptimal trajectories.

For manipulation tasks, OGBench provides `play` and `noisy` datasets. The `play` datasets contain natural interaction trajectories generated by scripted policies with temporally correlated behavior. These datasets often contain useful local skills but do not necessarily demonstrate each evaluation task end-to-end. The `noisy` datasets are collected with larger, less structured exploration noise, which increases state-action coverage but also makes the data more suboptimal. Together, these dataset variants test whether an offline GCRL method can learn useful local behaviors, compose them over long horizons, and remain robust when the data are imperfect or only partially aligned with the evaluation goals.

Evaluation protocol. We follow the standard OGBench protocol and report success-based performance on each task. For a method m, environment e, and random seed r, let $s_{m,e,r} \in [0, 1]$ denote the resulting success rate, averaged over the evaluation episodes and goals for that environment. Higher values indicate better goal-reaching performance. Unless otherwise stated, all methods are trained purely offline on the provided datasets and are evaluated without additional environment interaction during training.

Aggregate metrics with RLiable. In addition to per-environment results, we report aggregate statistics using the RLiable evaluation framework [1]. RLiable is useful in the few-seed regime because it summarizes performance across tasks while also quantifying uncertainty with stratified-bootstrap confidence intervals. Importantly, RLiable does not discard "noisy" runs or remove experiments. Instead, it estimates how sensitive aggregate conclusions are to the finite set of tasks and random seeds.

Let $S_m = \{s_{m,e,r}\}_{e,r}$ denote the collection of scores for method m across environments and seeds. We report the following aggregate metrics:

$$\mathrm{Mean}(m) = \frac{1}{|\mathcal{E}||\mathcal{R}|} \sum_{e \in \mathcal{E}} \sum_{r \in \mathcal{R}} s_{m,e,r}, \tag{24}$$

$$\mathrm{Median}(m) = \mathrm{median}\left(\{s_{m,e,r}\}_{e,r}\right), \tag{25}$$

$$\mathrm{IQM}(m) = \mathrm{mean}\left(\{s_{m,e,r} : s_{m,e,r} \text{ lies between the 25th and 75th percentiles}\}\right), \tag{26}$$

$$\mathrm{OptimalityGap}(m) = \frac{1}{|\mathcal{E}||\mathcal{R}|} \sum_{e \in \mathcal{E}} \sum_{r \in \mathcal{R}} \max(0, 1 - s_{m,e,r}). \tag{27}$$

The interquartile mean (IQM) averages the middle 50% of outcomes, making it less sensitive to extreme outlier tasks than the mean, while being more statistically efficient than the median. The optimality gap measures the average shortfall from the maximum normalized score of 1; thus, lower values are better. Since our scores are success rates in $[0, 1]$, the optimality gap is directly interpretable as the average remaining failure mass. If scores are reported as percentages, they are first divided by 100 before computing the RLiable metrics.

For confidence intervals, we use stratified bootstrap resampling over tasks and seeds. Each bootstrap replicate preserves the task structure: for every environment, we resample seeds with replacement and then recompute the aggregate metric on the resampled score matrix. The reported intervals correspond to the empirical percentiles of the bootstrap distribution. This procedure avoids treating all scores as exchangeable independent samples and prevents environments with more runs from dominating the uncertainty estimate.

D Additional Results

We include the additional Rliable [1] plots in Figure 6.

E AFU-style coupling of the bilinear value and dual score

This section provides the actor-free coupling we use between the bilinear value $V_\theta(s, g) = \psi_\theta(s)^\top \phi_\theta(g)$ and the dual score, following the separation of roles emphasized by Perrin-Gilbert [24]. The policy parameters do not receive gradients through this objective; policy learning uses only the weighted regression step.

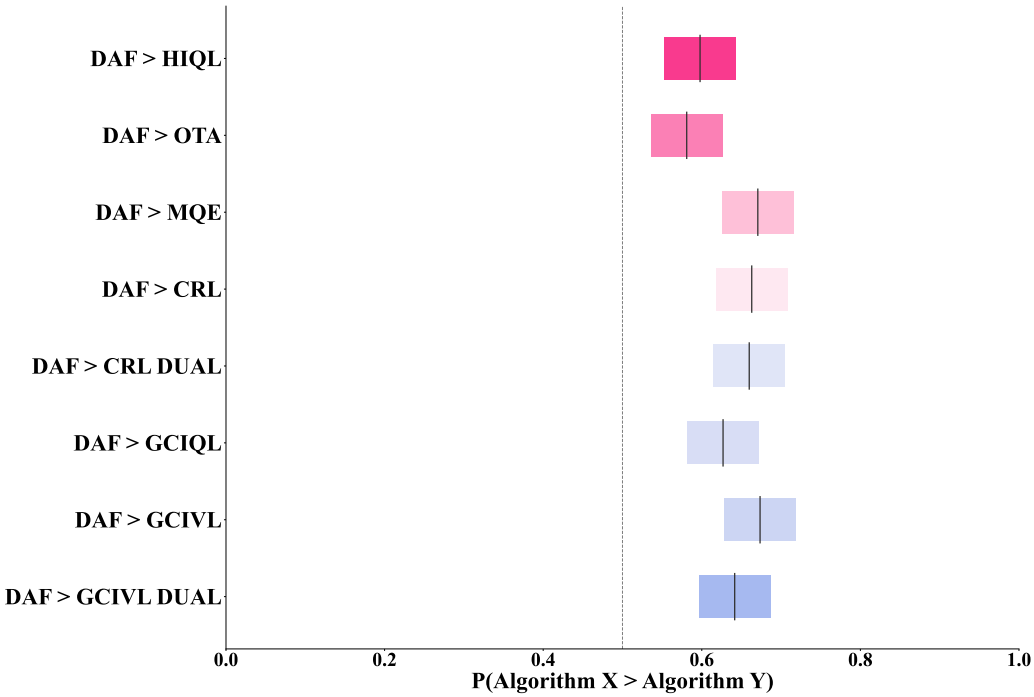

Figure 6: **Rliable Probability of Improvement.**

Surrogate dual score for the coupling. The main text defines the raw dual score z_θ in (14). In the AFU objective below it is convenient to use a non-positive surrogate

$$\widetilde{A}_\theta(s,a,g) \;:=\; h(z_\theta(s,a,g))\,, \qquad h:\mathbb{R}\to(-\infty,0], \tag{28}$$

where h is any monotone transformation used in implementation to keep the coupling term bounded on the optimistic side while preserving action ordering. In our experiments we use `softplus` function. The same z_θ can still be used directly in advantage-weighted regression, as in the main text; (28) is only required for the piecewise coupling with V_θ.

Scalar Bellman target. Let

$$T(s,a,g) \;:=\; r(s,g) + \gamma\, V_\theta^{\mathrm{tgt}}(s',g)\,, \tag{29}$$

with V_θ^{tgt} a slowly updated target network for the bilinear head.

Conditional scaling of V_θ. Let $U = \mathbb{1}[V_\theta + \widetilde{A}_\theta < T]$ and $\rho \in (0,1)$. Define

$$\widetilde{V} \;:=\; (1-\rho U)\, V_\theta + \rho U\, \mathrm{stopgrad}(V_\theta)\,. \tag{30}$$

When the optimistic sum $V_\theta + \widetilde{A}_\theta$ falls short of the Bellman target T, the mask down-weights direct updates to V_θ so that $\widetilde{A}_\theta$ can absorb slack in the near-optimistic regime.

Piecewise coupling loss. With $x = \widetilde{V} - T$ and $y = \widetilde{A}_\theta$, set

$$Z(x,y) \;=\; \begin{cases} (x+y)^2\,, & x \ge 0\,, \\ x^2 + y^2\,, & x < 0\,. \end{cases} \tag{31}$$

Training minimizes $\mathbb{E}_{\mathcal{D}}[Z(x,y)]$ jointly over the parameters of V_θ (equivalently ψ_θ and, where tied, ϕ_θ) and of the heads that define $\widetilde{A}_\theta$ (including u_ξ and ϕ_θ as used in z_θ). The asymmetric split between $x \ge 0$ and $x < 0$ mirrors the AFU construction: pessimistic errors on V and the dual score are not forced to cancel spuriously when the backup is optimistic.

Feature dynamics auxiliary loss. The loss $\mathcal{L}_{\text{ae}}$ in (15) complements the coupling above: the AFU-style term enforces Bellman consistency between V_θ and $\widetilde{A}_\theta$, while $\mathcal{L}_{\text{ae}}$ grounds u_ξ in explicit one-step feature dynamics on the offline dataset.

F Theoretical Analysis

This section establishes two complementary properties of Dual Advantage Fields (DAF). First, we show that under exact representability DAF recovers the true Bellman advantage and therefore constitutes a valid policy-improvement operator (Section F.1). Second, we analyse a didactic 1-D example and prove that, even when the learned goal embedding is corrupted by noise in irrelevant directions, DAF's local advantage remains significantly more robust than both flat and hierarchical value-difference extraction (Section F.2).

F.1 DAF as exact policy-improvement signal

Fix a goal g and consider the goal-conditioned MDP with reward $r_g(s, a) := r(s, a, g)$. For a policy π, define the usual Bellman advantage

$$A^\pi(s, a, g) := \mathbb{E}_{s' \sim P(\cdot|s,a)} \left[r(s, a, g) + \gamma V^\pi(s', g) - V^\pi(s, g) \right].$$

This is the relative quantity that drives policy improvement: only the ordering of actions at a given state matters, not the absolute level of V^π.

Assume that the policy value is realisable by the bilinear dual field,

$$V^\pi(s, g) = \psi(s)^\top \phi(g),$$

and that the action-effect model is exact,

$$u(s, a, g) = \mathbb{E}_{s' \sim P(\cdot|s,a)} \left[\gamma \psi(s') - \psi(s) \right].$$

Then the DAF score

$$D^\pi(s, a, g) := r(s, a, g) + u(s, a, g)^\top \phi(g)$$

equals the true goal-conditioned advantage:

$$D^\pi(s, a, g) = A^\pi(s, a, g).$$

Proposition F.1 (DAF local policy improvement)**.** *Let π^+ be any goal-conditioned policy satisfying*

$$\mathbb{E}_{a \sim \pi^+(\cdot|s,g)} \left[D^\pi(s, a, g) \right] \geq 0 \qquad \textit{for all } s, g.$$

Under the realizability and exact action-effect assumptions above,

$$V^{\pi^+}(s, g) \geq V^\pi(s, g) \qquad \textit{for all } s, g.$$

Proof. Since $D^\pi = A^\pi$, the assumption gives $\mathbb{E}_{a \sim \pi^+(\cdot|s,g)}[A^\pi(s, a, g)] \geq 0$. This is exactly

$$(T_{\pi^+} V^\pi)(s, g) - V^\pi(s, g) = \mathbb{E}_{a \sim \pi^+(\cdot|s,g)}[A^\pi(s, a, g)] \geq 0,$$

where T_{π^+} is the Bellman operator for policy π^+. Hence $T_{\pi^+} V^\pi \geq V^\pi$ pointwise, and by monotonicity of the Bellman operator, $T_{\pi^+}^k V^\pi \geq V^\pi$ for every $k \geq 1$. Taking $k \to \infty$ and using the contraction property of T_{π^+} yields $V^{\pi^+} \geq V^\pi$. □

The advantage-weighted regression (AWR) update used by DAF is one such improvement in the exact on-policy case. If

$$\pi_\alpha^+(a \mid s, g) = \frac{\pi(a \mid s, g) \exp(\alpha D^\pi(s, a, g))}{\sum_b \pi(b \mid s, g) \exp(\alpha D^\pi(s, b, g))}, \qquad \alpha \geq 0,$$

then a standard argument shows $\mathbb{E}_{a \sim \pi_\alpha^+}[D^\pi(s, a, g)] \geq \mathbb{E}_{a \sim \pi}[D^\pi(s, a, g)] = 0$, so the AWR policy satisfies the condition of Proposition F.1.

Corollary F.2 (Exact DAF policy iteration)**.** *In a finite discounted goal-conditioned MDP, suppose each iteration k uses exact representations for V^{π_k} and an exact action-effect model, and define*

$$\pi_{k+1}(\cdot \mid s, g) \in \arg\max_{\pi'} \mathbb{E}_{a\sim\pi'(\cdot|s,g)}[D^{\pi_k}(s, a, g)].$$

Then π_{k+1} is the standard greedy policy-improvement step with respect to Q^{π_k}. Consequently, repeated exact DAF improvement is policy iteration and converges to an optimal goal-conditioned policy.

Proof. Because $D^{\pi_k} = A^{\pi_k} = Q^{\pi_k} - V^{\pi_k}$, maximising D^{π_k} over actions is equivalent to maximising Q^{π_k}. The result follows from classical policy iteration for finite discounted MDPs, applied separately for each goal g. □

Relation to hierarchical policies. Let Π denote the class of all stationary goal-conditioned primitive-action policies, and let $\Pi_{\text{hier}} \subseteq \Pi$ be any hierarchically constrained class (e.g. subgoal or option policies). The optimal primitive-action policy $\pi^\star \in \arg\max_{\pi\in\Pi} V^\pi$ satisfies

$$V^{\pi^\star}(s, g) \geq \sup_{\pi\in\Pi_{\text{hier}}} V^\pi(s, g) \quad \text{for all } s, g.$$

Thus, in the exact realisable limit, DAF policy iteration reaches a policy that is at least as good as the best policy in any fixed hierarchical class.

This comparison is a representational statement: hierarchy may improve learning by reducing the effective horizon, but a fixed hierarchy can also introduce subgoal-level constraints that exclude the true optimal primitive-action policy. DAF instead performs improvement directly at the primitive-action level using the local dual advantage, while preserving the long-horizon reachability information encoded in the dual value field.

F.2 Robustness to learned embedding noise: a didactic example

We now turn to a more practical regime where the representation is learned from finite data and inevitably contains noise.

F.2.1 Environment and representation model

Line-world dynamics. Consider deterministic states $s \in \{0, 1, \ldots, T\}$ with a fixed goal $g = T > 0$. Two actions are available: right ($a = +1$, $s \to s + 1$) and left ($a = -1$, $s \to s - 1$). The episode terminates upon reaching g; the reward is 0 at the goal and -1 otherwise. Hence the optimal policy always moves right for $s < T$, and the optimal (negative) value function is

$$V^\star(s, g) = s - T, \qquad s \leq T.$$

Fixed state embedding. The environment provides a feature map $\psi : \mathbb{Z} \to \mathbb{R}^d$ with $d = m + 2$ ($m \geq 0$):

$$\psi(s) = \big[s,\ 1,\ f_1(s), \ldots, f_m(s)\big]^\top,$$

where $\{f_i\}_{i=1}^m$ are bounded C^2 functions (or, in the discrete case, functions with well-defined first and second differences). The first two coordinates are "essential" for representing the linear optimal value; the remaining ones are *nuisance* dimensions that are irrelevant for the control task (e.g., visual textures, lighting gradients).

True goal embedding. The optimal value can be expressed via an inner product:

$$\phi^\star(g) = \big[1,\ -T,\ 0, \ldots, 0\big]^\top \quad \Longrightarrow \quad \psi(s)^\top \phi^\star(g) = s - T = V^\star(s, g).$$

F.2.2 Noise model for the learned goal embedding

In offline training, the goal embedding $\phi(g)$ is estimated from a finite dataset. Because the temporal-difference loss only weakly constrains the coefficients of the nuisance coordinates (especially if those coordinates vary slowly), the learned embedding can accumulate significant noise along those directions. We model this by an additive perturbation confined to the nuisance components:

$$\phi(g) = \phi^\star(g) + \boldsymbol{\varepsilon}, \qquad \boldsymbol{\varepsilon} = \big[0,\ 0,\ \eta_1, \ldots, \eta_m\big]^\top,$$

where $\eta_i \sim \mathcal{N}(0, \sigma_i^2)$ are independent. The essential coordinates are assumed to be learned accurately for simplicity; allowing noise there would not change the qualitative conclusions.

Consequently the noisy value estimate at any state s is

$$\widehat{V}(s, g) = \psi(s)^\top \phi(g) = s - T + \sum_{i=1}^{m} \eta_i \, f_i(s).$$

For the subgoal s_{sub} we assume the same embedding function $\phi(\cdot)$ is applied and that its noise is independent of $\phi(g)$:

$$\phi(s_{\text{sub}}) = \phi^\star(s_{\text{sub}}) + \tilde{\varepsilon}, \qquad \tilde{\eta}_i \sim \mathcal{N}(0, \sigma_i^2) \text{ independent of } \eta_i.$$

F.2.3 Action-effect model and policy extraction rules

We assume that a separate action-effect model $u(s, a)$ has been trained to regress to the true one-step feature change $\psi(s + a) - \psi(s)$ and has converged to the exact quantity (realistic because the model sees abundant transitions and the dynamics are deterministic).

Thus

$$u(s, +1) = \psi(s + 1) - \psi(s), \qquad u(s, -1) = \psi(s - 1) - \psi(s).$$

We compare three policy extraction methods, all built upon the same learned bilinear value $\widehat{V}$ and the same u.

1. **Flat value-difference.** Choose the action that leads to the highest estimated next-state value:

   $$a_{\text{V}}(s) = \operatorname*{arg\,max}_{a \in \{-1, +1\}} \widehat{V}(s + a, g).$$

   This corresponds to the implicit advantage used in HIQL's flat baseline (comparing $V(s + 1, g)$ and $V(s - 1, g)$).
2. **DAF local advantage.** Score each action by the inner product of its predicted feature displacement and the goal embedding (Eq. 14 in the main paper):

   $$a_{\text{DAF}}(s) = \operatorname*{arg\,max}_{a \in \{-1, +1\}} u(s, a)^\top \phi(g).$$

   (The sparse reward, identical for both actions, is omitted from the comparison.)
3. **Hierarchical HIQL.** The hierarchical policy first selects a subgoal at distance $k \geq 2$ (to the right, $s_{\text{sub}} = s + k$) by comparing values of the candidate subgoals:

   $$s_{\text{sub}} = \operatorname*{arg\,max}_{x \in \{s+k, s-k\}} \big[\widehat{V}(x, g) - \widehat{V}(s, g)\big].$$

   Subsequently a low-level controller attempts to reach that subgoal, using the subgoal's own embedding $\phi(s_{\text{sub}})$ and the same flat value-difference rule:

   $$a_\ell(s) = \operatorname*{arg\,max}_{a \in \{-1, +1\}} \widehat{V}(s + a, s_{\text{sub}}).$$

   An error occurs if either the subgoal choice is wrong or the low-level action is wrong; we bound this with a union argument as in Park et al. [19, Proposition 4.1].

F.2.4 Error probabilities

For any nuisance function f, define the first and second discrete differences at state s:

$$\Delta f(s) := f(s + 1) - f(s - 1), \qquad \Delta^2 f(s) := f(s + 1) + f(s - 1) - 2f(s).$$

Flat value-difference.

$$\Delta_{\text{V}}(s) = \widehat{V}(s + 1, g) - \widehat{V}(s - 1, g) = 2 + \sum_{i=1}^{m} \eta_i \, \Delta f_i(s).$$

DAF.

$$\Delta_{\mathrm{DAF}}(s) = \big(u(s,+1) - u(s,-1)\big)^\top \phi(g) = 2 + \sum_{i=1}^{m} \eta_i \, \Delta^2 f_i(s).$$

Hierarchical high-level.

$$\Delta_{\mathrm{high}}(s) = \widehat{V}(s+k,g) - \widehat{V}(s-k,g) = 2k + \sum_{i=1}^{m} \eta_i \, \big(f_i(s+k) - f_i(s-k)\big).$$

Hierarchical low-level. Conditioned on the subgoal $s+k$ being selected,

$$\Delta_{\mathrm{low}}(s) = \widehat{V}(s+1,s+k) - \widehat{V}(s-1,s+k) = 2 + \sum_{i=1}^{m} \tilde{\eta}_i \, \Delta f_i(s).$$

All decision statistics are Gaussian. Let Φ be the standard normal c.d.f.

Proposition F.3 (Error probabilities)**.** *For any state $s \in \{1,\dots,T-1\}$ and subgoal step k,*

$$\varepsilon_{\mathrm{flat}}(s) = \Phi\left(-\frac{2}{\sqrt{\sum_i \sigma_i^2 \left(\Delta f_i(s)\right)^2}}\right),$$

$$\varepsilon_{\mathrm{DAF}}(s) = \Phi\left(-\frac{2}{\sqrt{\sum_i \sigma_i^2 \left(\Delta^2 f_i(s)\right)^2}}\right),$$

$$\varepsilon_{\mathrm{high}}(s) = \Phi\left(-\frac{2k}{\sqrt{\sum_i \sigma_i^2 \left(f_i(s+k) - f_i(s-k)\right)^2}}\right),$$

$$\varepsilon_{\mathrm{low}}(s) = \Phi\left(-\frac{2}{\sqrt{\sum_i \sigma_i^2 \left(\Delta f_i(s)\right)^2}}\right).$$

The overall hierarchical error is bounded by

$$\varepsilon_{\mathrm{hier}}(s) \;\le\; \varepsilon_{\mathrm{high}}(s) \;+\; \varepsilon_{\mathrm{low}}(s).$$

Proof. Each decision margin is a normal random variable with the stated mean and variance; misclassification is the event "margin < 0". The hierarchical bound follows from a union bound over the two decision stages, exactly as in Park et al. [19, Proposition 4.1]. □

F.2.5 Why DAF can be more robust

The formulas in Proposition F.3 show that DAF's noise enters through the *second differences* $\Delta^2 f_i(s)$, whereas all value-difference methods (flat and low-level) involve the *first differences* $\Delta f_i(s)$. The high-level comparison involves the even larger span $f_i(s+k) - f_i(s-k)$.

For many realistic nuisance functions, the second difference is much smaller than the first difference. Two concrete regimes make this quantitative.

Corollary F.4 (Affine nuisance coordinates are eliminated by DAF)**.** *If $f_i(s) = \alpha_i s + \beta_i$ for all i, then $\Delta^2 f_i(s) = 0$ for every s; hence $\Delta_{\mathrm{DAF}}(s) \equiv 2$ and $\varepsilon_{\mathrm{DAF}}(s) = 0$. In contrast,*

$$\varepsilon_{\mathrm{flat}}(s) = \varepsilon_{\mathrm{low}}(s) = \Phi\left(-\frac{1}{\sqrt{\sum_i \sigma_i^2 \alpha_i^2}}\right), \qquad \varepsilon_{\mathrm{high}}(s) = \Phi\left(-\frac{1}{\sqrt{\sum_i \sigma_i^2 \alpha_i^2}}\right).$$

Thus DAF makes zero *mistakes regardless of the horizon, while the flat and hierarchical baselines can suffer significant error whenever $\sum_i \sigma_i^2 \alpha_i^2$ is large.*

Corollary F.5 (Low-curvature nuisance coordinates)**.** *Suppose each f_i is twice differentiable with $|f_i''(s)| \leq C$ and that over a short interval the first difference can be expressed as $\Delta f_i(s) = 2f_i'(s)+O(C)$, $\Delta^2 f_i(s) = 2f_i''(s)+O(C)$. If the local slope $f_i'(s)$ is large (e.g., a strong linear trend) while the curvature remains bounded, then $\sigma^2_{\mathrm{DAF}}(s) = O(C^2)$ whereas $\sigma^2_{\mathrm{flat}}(s) = 4\sum_i \sigma_i^2 f_i'(s)^2$ can be arbitrarily large. Consequently $\varepsilon_{\mathrm{DAF}}(s)$ stays close to zero while $\varepsilon_{\mathrm{flat}}(s)$ and $\varepsilon_{\mathrm{low}}(s)$ may approach $\frac{1}{2}$.*

Comparison with the hierarchical baseline. Even with a well-chosen subgoal step k, the low-level controller still relies on first differences (Proposition F.3), inheriting the same vulnerability as the flat extraction. Moreover, the high-level stage introduces an additional source of error that scales with the span of the nuisance functions. As a result, a *single* DAF flat policy can achieve a lower error rate than a hierarchical policy that employs two value-difference decisions.

Illustrative quantitative example. Let $m = 1$ and $f_1(s) = s^2$. Then $\Delta f_1(s) = 4s,\ \Delta^2 f_1(s) = 2$, and

$$\varepsilon_{\mathrm{DAF}}(s) = \Phi\left(-\frac{2}{\sigma_1 \cdot 2}\right), \qquad \varepsilon_{\mathrm{flat}}(s) = \varepsilon_{\mathrm{low}}(s) = \Phi\left(-\frac{2}{\sigma_1 \cdot 4s}\right).$$

For a state far from the goal ($s = T - 1 \gg 1$), $\varepsilon_{\mathrm{flat}}$ and $\varepsilon_{\mathrm{low}}$ are close to 0.5 if σ_1 is large, while $\varepsilon_{\mathrm{DAF}}$ remains bounded by a constant that does not grow with T.

G Compute Resources

All experiments were performed on servers with a single H100 GPU with 80 GB of GPU memory, 12 CPU cores, and 244 GB of RAM. All metrics for the experiments were logged using the Weights & Biases platform. Overall, the Weights & Biases project of the paper had 17,359 tracked experiments at the time of submission and used an estimated ~ 407 days of GPU compute in total.