DUAL ADVANTAGE FIELDS

Anonymous authors
Paper under double-blind review

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 (DAF), 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, which directly specifies which action should be preferred. DAF introduces an action-effect model that predicts the discounted direction of change in state representation space induced by actions and scores them by the similarity between this change 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.

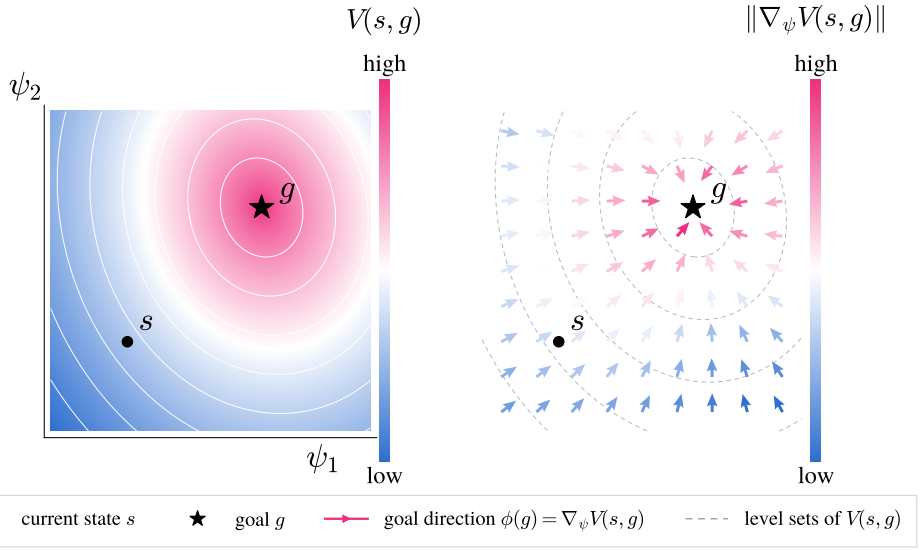

Figure 1: A dual goal-conditioned value model defines a global value surface over state representations. DAF uses the fact that the local action-comparative signal aligns with the goal direction.

1 INTRODUCTION

Goal-conditioned reinforcement learning (GCRL) aims to learn policies that reach arbitrary goals from a fixed dataset of prior experience (Liu et al., 2022; Ma et al., 2022; Park et al., 2025a). This requires solving two different problems at once. First, 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 (Park et al., 2023; 2025b). Second, the agent must act locally: at the current state, it must decide which available action makes the most progress toward the requested goal (Peng et al., 2019; Kostrikov et al., 2022). A good goal-conditioned agent therefore needs both a global map of reachability and a local compass for action selection.

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

Most recently, dual goal representations (Park et al., 2025c) provide one of the strong answers to the first problem. They parameterize a goal-conditioned value function as a bilinear interaction between a state embedding and a goal embedding $V_\theta(s, g) = \psi_\theta(s)^\top \phi_\theta(g)$, in the spirit of bilinear value networks (Hong et al., 2022). This structure induces a value surface for each goal, where states that are more reachable or desirable for the goal receive higher values. However, a value surface alone does not directly answer the local control question.

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 a local advantage-like quantity: a way to score whether an action changes the state in a direction that improves goal-conditioned value. The usual remedy is to build a second object for the purpose, such as an action-value function (Kostrikov et al., 2022; Ghosh et al., 2023) or a subgoal predictor (Park et al., 2023); notably, Park et al. (2025c) report that the dual field is insufficient as the object a policy is extracted from, and therefore use dual representations only as conditioners while training a separate value branch for extraction.

Our key observation is that this local signal is already implicit in the geometry of dual representations, and that no additional machinery is needed to read it out. 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. This insight shows that the goal embedding is not merely a descriptor of a target state: it is the dual object that evaluates local directions in representation space. Consequently, combining this goal embedding with an action-effect vector yields an advantage signal that is inherently local and action-comparative, the kind of advantage that suffices for policy iteration.

We introduce *Dual Advantage Fields* (DAF), a policy-extraction method that makes this geometry explicit (Figure 1). DAF learns an action-effect model that predicts the discounted change in the state representation caused by an action. The resulting score is local, goal-conditioned, and action-comparative.

Our contributions are:

- We show that, under the dual goal representation parameterization (Park et al., 2025c), the goal embedding is itself the gradient direction, in state representation space, of the goal-conditioned value field.
- We introduce *Dual Advantage Fields*, an action-effect model that scores actions by their alignment with the direction toward the goal and therefore produces a local advantage-like signal for goal-conditioned policy extraction. We prove that this signal can be viewed as the goal-conditioned Bellman advantage, and hence as a valid improvement operator, when the dual model is realizable (Proposition 4). We justify theoretically and empirically why the action effect should be learned rather than read off a single transition (Proposition 6).
- We use this signal to extract policies from offline data without training a separate goal-conditioned action-value function, and evaluate the resulting method on the 20 state-based OGBench tasks (Park et al., 2025a), where it improves both per-task and aggregate RLiable performance (Agarwal et al., 2021) over flat and hierarchical backbones alike.

2 Preliminaries

Goal-conditioned Reinforcement Learning. We study *offline* goal-conditioned reinforcement learning (GCRL) (Liu et al., 2022; Eysenbach et al., 2022; Ma et al., 2022; Myers et al., 2025): the learner has access to a fixed offline dataset of transitions but cannot collect new experience in the environment (Park et al., 2025a). The objective is to infer an optimal goal-conditioned policy even for state-goal pairs whose combination is unseen during training.

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 $P(s' \mid s, a)$, and a goal $g \in \mathcal{G}$ induces a reward $r(s, a, g)$, which in sparse goal-reaching problems is often zero until a success condition holds. With discount $\gamma \in (0, 1)$, the goal-conditioned value functions of a policy $\pi(a \mid s, g)$ satisfy

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

Recent GCRL methods combine several ideas, including representation learning, quasimetric objectives (Wang et al., 2023; Ke et al., 2025b; Myers et al., 2025), and hierarchical horizon reduction (Park et al., 2023; Giammarino & Qureshi, 2026; Myers et al., 2024) 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.

Dual Goal Representations (Park et al., 2025c). The goal representation determines what information the policy and value function use about the target state. Rather than conditioning 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, in the same spirit in which a point may be described by its distances to the rest of the space rather than by its coordinates. Formally, a goal g is represented not by the observation itself but by the function $\phi^\vee(g) : s \mapsto d^\star(s, g)$, where the superscript $\vee$ marks this functional, *dual* view of the goal and $d^\star(s, g)$ is the optimal discounted temporal distance introduced below. In practice we approximate this functional through a bilinear goal-conditioned potential (Hong et al., 2022):

$$V_\theta(s, g) = \psi_\theta(s)^\top \phi_\theta(g), \tag{2}$$

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

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 (Park et al., 2023). To address this issue, HIQL (Park et al., 2023) proposed a hierarchical policy structure that utilizes a value function learned with IQL (Kostrikov et al., 2022). 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:

$$\mathcal{L}(V) = \mathbb{E}_{(s,s')\sim\mathcal{D},\, g\sim p(g)} \left[L_2^\tau \left(r(s, g) + \gamma \bar{V}(s', g) - V(s, g) \right) \right], \tag{3}$$

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 (Andrychowicz et al., 2017; Wang et al., 2023; Park et al., 2023), 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* $d^\star(s, g)$, *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) (Peng et al., 2019; Wang et al., 2020) with the following objective:

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

$$\mathcal{J}(\pi^\ell) = \mathbb{E}_{(s_t, a_t, s_{t+1}, s_{t+k})\sim\mathcal{D}} \left[\exp\left(\beta^\ell \cdot A^\ell(s_t, s_{t+1}, s_{t+k})\right) \log \pi^\ell(a_t|s_t, s_{t+k}) \right], \tag{5}$$

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 (Park et al., 2025a).

3 Dual Advantage Fields

A dual goal representation encodes a goal by how it sits relative to other states. The same vector can be read locally. With the goal held fixed, the bilinear model is a scalar field on the state representation,

[1] Expectile regression with $\tau > 0.5$ is optimistic, so under stochastic dynamics it upper-biases the value; following prior work we therefore present the derivation for deterministic dynamics and return to the stochastic case in Section 5 and Section D.1.

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

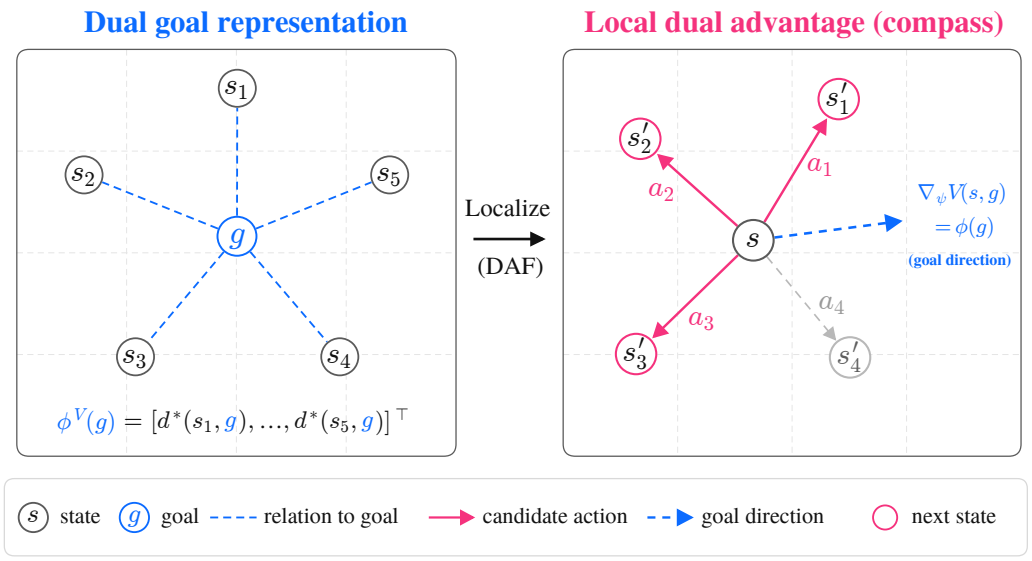

Figure 2: DAF scores an action by projecting its induced feature displacement onto the goal direction, yielding a local advantage signal for policy improvement.

and the direction of steepest increase is a comparison among actions rather than a statement of long-horizon reachability. Dual Advantage Fields make that comparison explicit. Holding the goal fixed and viewing the value as a function of the state embedding ψ, we have

Proposition 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; 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', g) provides a sample estimate of this quantity:

Corollary 2. *The sample-level Dual Advantage Field score is*

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

Note that (9) is an expectation over successors while (10) is a single draw from it; Section 4 replaces the draw by a learned module, for theoretical and empirical reasons.

In (10), 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, so $\widehat{A}_\theta$ is 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 (Dayan & Singh, 1995).

Local policy improvement. In the realizable case, the DAF score is exactly the goal-conditioned Bellman advantage: 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 with positive DAF score is a standard policy-improvement step, and repeated exact DAF improvement recovers an optimal goal-conditioned policy; in particular, its limiting policy is at least as good as any policy restricted to a fixed hierarchical class. We give the formal statement, the assumptions it needs, and the proof in Proposition 4.

4 Goal-conditioned policy extraction

Dayan & Singh (1995) argue that policy iteration should only require relative measures of action utility, not absolute values. In their continuous deterministic setting they define an advantage $A^w(x,u) = r(x,u) + f(x,u) \cdot \nabla_x V^w(x)$ and improve the policy by maximizing A^w. They emphasize that learning these derivatives can be underconstrained, and introduce conservative-field structure, a curl-free gradient, as an additional constraint.

We take this as intuition for extending their model-based comparison to goal-conditioned control, with the empirical findings of provable performance in continuous scenario. The local improvement step addresses the discounted change in the state representation, $\gamma\psi_\theta(s') - \psi_\theta(s)$, which serves as a conceptual analogue to velocity: the Bellman advantage (8) is the inner product of that change with the goal direction (9), in the spirit of $r + f \cdot \nabla V$.

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). The raw dual score is

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

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. Everything that follows is standard offline-RL machinery assembled around this one scalar, and Algorithm 1 summarizes the resulting training loop.

We learn $(\psi_\theta, \phi_\theta)$, and the displacement map u_ξ from offline tuples (s,a,s',g) (Park et al., 2025a). For stability we used a common approach in offline RL (Kostrikov et al., 2022) that learns twin critics $Q_\eta^{(1)}, Q_\eta^{(2)}$, and the bilinear head $V_\theta(s,g) = \psi_\theta(s)^\top \phi_\theta(g)$ is tied to pessimistic Q-estimates via expectile regression (3) and to Bellman backups on $Q_\eta^{(j)}$.

Fitting a value to $\max_a Q_\theta(s,a,g)$ is the step that most often destabilizes continuous-control critics: the maximum over a continuous action set has to be approximated by an actor, which introduces maximization bias (van Hasselt et al., 2016; Fujimoto et al., 2018) and, in deterministic sparse-reward problems, can deadlock actor and critic (Matheron et al., 2019). DAF reads a *direction* out of the field, so an error can rotate $\phi_\theta(g)$ and invert the ranking of actions. We ground u_ξ with the auxiliary loss

$$\mathcal{L}_{\text{ae}} \;=\; \mathbb{E}\Big[\big\|u_\xi(s,a) - \mathrm{sg}\big(\gamma\psi_\theta(s') - \psi_\theta(s)\big)\big\|_2^2\Big]\,, \tag{12}$$

with sg stopping gradients through the target, so u_ξ tracks one-step feature dynamics on $\mathcal{D}$. (9) asks for a conditional expectation over successors, and a dataset transition supplies one draw of it. By learning the action-effect network we amortize that expectation across the dataset instead of a single-sample draw. The theoretical comparison of these scenarios is given in Proposition 6.

We therefore add an actor-free coupling between the value function V_θ and the dual score z_θ of (11), following Perrin-Gilbert (2024). In continuous control the usual critic backup maximizes a Q-function over actions, and that maximum has to be approximated by an actor. The actor is what injects maximization bias and, in deterministic sparse-reward problems, what can deadlock the critic (Matheron et al., 2019). Actor-free updates avoid that path: they split the Bellman target into a state-value part and a non-positive advantage part, and fit the two with an asymmetric piecewise loss. When the backup is above the current estimate the value and the advantage are coupled; when it is below, they are fit separately, so a pessimistic error in one cannot be cancelled by an optimistic error in the other. The policy parameters receive no gradient from this loss, so the critic is updated with no actor and no policy gradient in the backup (Appendix E).

Algorithm 1 DAF training (with optional high-level actor).

Input: offline dataset $\mathcal{D}$ of tuples (s, a, s', g); discount γ; expectile τ; AWR temperature α and cap $W_{\max}$; subgoal stride k.
Initialize: dual factors ψ_θ, ϕ_θ with $V_\theta(s, g) = \psi_\theta(s)^\top \phi_\theta(g)$ (2); twin critics $Q_\eta^{(j)}$; action-effect map u_ξ; policies $\pi_\omega, \pi^h_{\omega_h}$; targets.
while not converged **do**
  Sample a minibatch $\{(s, a, s', g)\} \sim \mathcal{D}$.
  $\min_\eta \ \mathbb{E}\Big[\big(Q_\eta^{(j)}(s, a, g) - \big(r(s, g) + \gamma V_\theta^{\mathrm{tgt}}(s', g)\big)\big)^2\Big]$.
  $\min_\theta \ \mathbb{E}\Big[L_2^\tau\Big(\min_j Q_\eta^{\mathrm{tgt},(j)}(s, a, g) - V_\theta(s, g)\Big)\Big]$ (3), where $L_2^\tau(u) = |\tau - \mathbf{1}(u < 0)|u^2$.
  $z_\theta(s, a, g) = u_\xi(s, a)^\top \phi_\theta(g)$ (11).
  $\min \ \mathcal{L}_{\mathrm{afu}}$ (30) {Appendix E}.
  $\min_\xi \ \mathbb{E}\Big[\big\|u_\xi(s, a) - \mathrm{sg}\big(\gamma\psi_\theta(s') - \psi_\theta(s)\big)\big\|_2^2\Big]$ (12).
  set $w_\theta \leftarrow \min\{\exp(\alpha z_\theta), W_{\max}\}$ (13) and minimize $-\mathbb{E}_\mathcal{D}[w_\theta \log \pi_\omega(a \mid s, c)]$ over ω.
  (optional) set $w_\theta^h \leftarrow \min\big\{\exp\big(\beta^h\big(V_\theta(s_{t+k}, g) - V_\theta(s_t, g)\big)\big), W_{\max}\big\}$ and
    minimize $-\mathbb{E}[w_\theta^h \log \pi^h_{\omega_h}(s_{t+k} \mid s_t, g)]$ over ω_h (4).
  Polyak-update the target networks.
end while
Return π_ω and $\pi^h_{\omega_h}$.

Let $\pi_\omega(a \mid s, c)$ denote the policy, conditioned on g through $\phi_\theta(g)$. Advantage-weighted regression (Peng et al., 2019) uses weights

$$w_\theta(s, a, g) \ = \ \min\Big\{\exp\big(\alpha\, z_\theta(s, a, g)\big),\, W_{\max}\Big\} \tag{13}$$

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 $\phi_\theta(g)$.

5 Experiments

5.1 Localization Illustration

We illustrate the local geometry captured by Dual Advantage Fields on the `cube-single-play-v0-task1` manipulation task from OGBench (Park et al., 2025a), which 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, so a direction that points directly toward the terminal object location may be globally plausible but locally unhelpful. We compare the resulting directions with option-aware temporally abstracted value learning (OTA (Ahn et al., 2025)), a state-of-the-art hierarchical method and a standard backbone among strong goal-conditioned approaches.

The raw dual score (11) favors actions whose predicted feature displacement is aligned with the local direction of value increase toward the goal.

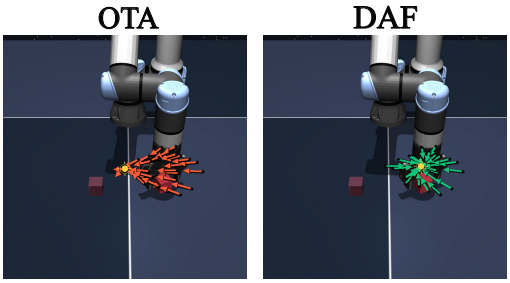

Figure 3: **Pre-grasp vector field in `cube-single`.** Arrows show decoded high-level directions from sampled gripper positions around the cube, with the cube and final goal fixed. DAF points locally toward the cube before grasping, while OTA points toward the terminal placement goal. The yellow marker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

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

$$\hat{x}_i^m = D_m(h_m(\tilde{s}_i, g)), \tag{14}$$

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

$$d_i^m = \frac{\hat{x}_i^m - x_{\mathrm{ee}}(\tilde{s}_i)}{\|\hat{x}_i^m - x_{\mathrm{ee}}(\tilde{s}_i)\|_2}, \tag{15}$$

drawn from the gripper position $x_{\mathrm{ee}}(\tilde{s}_i)$. Near the cube, DAF produces directions that point toward the object, matching the pre-grasp behavior required before transport, whereas OTA points toward the terminal placement region, which is appropriate only after grasping. This is why local advantage fields can be more useful than a high-level subgoal alone: they select actions by whether they locally improve the goal-conditioned potential, precisely the comparison that a value level set cannot make.

5.2 OGBench benchmark

We empirically validate the findings developed above on the OGBench benchmark (Park et al., 2025a). We focus on the state-based locomotion and manipulation tasks, which lets us test whether DAF improves consistently across domains with different control structure. All methods are trained purely offline and evaluated without additional environment interaction; we report success-based performance in $[0, 1]$ averaged over 5 seeds. In maze-style locomotion, `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, `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. The temperature α is the only extraction hyperparameter; following the OGBench and HIQL protocols we select it per environment family from $\{0.1, 3.0, 10.0\}$, and report the full sweep in Appendix A.

Baselines. We compare against a representative set of recent methods for offline GCRL: HIQL (Park et al., 2023), OTA (Ahn et al., 2025), MQE (Myers et al., 2024), CRL (Eysenbach et al., 2022), the goal-conditioned implicit Q- and V-learning baselines GCIQL and GCIVL as implemented in OGBench (Kostrikov et al., 2022; Park et al., 2025a), and, where applicable, their dual-goal-representation variants (Park et al., 2025c). These cover the main families used in offline GCRL, including horizon reduction (Park et al., 2025b) and representation priors such as quasimetrics.

Maze locomotion: long-horizon navigation and stitching. We first evaluate on `humanoidmaze` and `antmaze`, the top block of 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, with `humanoidmaze` the harder of the two because it combines full-body humanoid control with long-horizon goal reaching. DAF is competitive with the strongest horizon-reduction baselines on `navigate` datasets and obtains the best results on the harder `stitch` cases.

Object manipulation: local control from imperfect demonstrations. Next, we evaluate on `cube` and `scene`. The largest margins are on `cube-triple` and `scene`, precisely the settings in which many actions sit at a similar value level.

Puzzle rearrangement: continuous control with combinatorial structure. Finally, we evaluate on `puzzle`, robotic versions of Lights Out in which pressing one button changes the state of neighboring buttons, combining continuous control with combinatorial generalization over discrete configurations; the `3x3` and `4x4` variants further increase the configuration space.

Hierarchical goals. For long horizons, a high-level policy over subgoals can be trained alongside the low-level stack, with value differences along options as in hierarchical offline GCRL (Park et al., 2023); option-aware temporally abstracted value learning offers a related hierarchical baseline (Ahn et al., 2025). In our main experiments the high level is extracted with the value-gap objective (4) and the low level replaces the value gap in (5) by z_θ. The hierarchy and the dual advantage address different bottlenecks: the former improves how value is *propagated* over long horizons, the latter how actions are *compared* once a value field is available. Because z_θ consumes only a bilinear field

Table 1: Results on the 20 state-based OGBench tasks. The `navigate` and `play` datasets evaluate learning from coherent near-expert data, while `stitch` requires composing trajectory segments and `noisy` tests robustness under less structured exploration. Best values are highlighted in blue.

Task	DAF	HIQL	OTA	MQE	CRL	CRL Dual	GCIQL	GCIVL	GCIVL Dual
humanoidmaze-medium-navigate	**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
humanoidmaze-large-navigate	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
humanoidmaze-medium-stitch	**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
humanoidmaze-large-stitch	**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-teleport-navigate	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
antmaze-medium-navigate	**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
antmaze-teleport-stitch	**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
antmaze-medium-stitch	**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
cube-double-play	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
cube-triple-play	**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
cube-quadruple-play	**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
cube-double-noisy	**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
cube-triple-noisy	**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
cube-quadruple-noisy	**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
scene-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
puzzle-3x3-play	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
puzzle-4x4-play	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
puzzle-3x3-noisy	**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
puzzle-4x4-noisy	**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

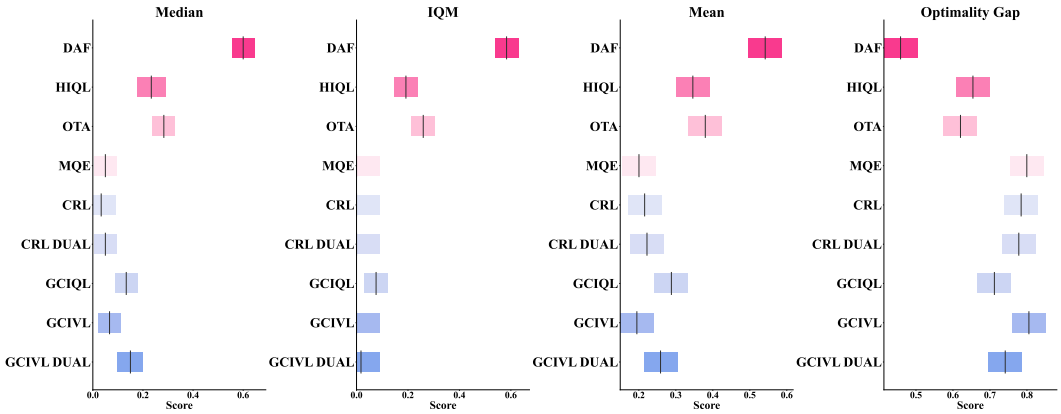

Figure 4: **Performance comparison.** Following Agarwal et al. (2021), we report Median, IQM, Mean and Optimality Gap with stratified-bootstrap confidence intervals; colored segments denote confidence intervals and dark markers point estimates.

and a set of transitions, it is agnostic to which backbone produces the field, and Appendix A shows the same gain when it is attached to a flat critic instead. Concrete instantiations differ by which of the optional terms above are active; experimental details are in Appendix A.

Table 1 shows that DAF performs strongly across different kinds of offline coverage and control structure. Figure 4 summarizes the same runs with RLiable metrics (Agarwal et al., 2021): the interquartile mean reduces sensitivity to outlier tasks and the optimality gap measures the average shortfall from perfect success. DAF improves all four aggregate metrics, the gain is not driven by a single environment family, and the performance profile (Figure 5) shows the same ordering at every score threshold.

Ablations. Appendix A removes each design choice in turn (Table 6). Dropping the actor-free coupling, the learned action effect or the hierarchy costs 0.33, 0.28 and 0.29 success on average. The sharpest comparison is the last column: the same representations and hierarchy, extracted conventionally, lose 0.45. Conversely, the dual advantage on a flat critic already beats the flat goal-conditioned baselines on most tasks (Table 7), so it improves extraction from either backbone.

Extraction from a directly learned dual critic. The gradient identity says that the local signal is already the goal direction of the state–goal field, so a separate action-conditioned critic is unnecessary. The natural alternative is to set that identity aside and learn a bilinear critic $Q(s, a, g) = \psi(s, a)^\top \phi(g)$ directly, then extract with Q or with $Q - V$. This keeps the dual factorization and drops only the claim that the state–goal field already contains the action comparison. Table 2 compares the two under the same training budget, hyperparameters, evaluation protocol and seeds, on nine tasks. We report the better of the two extraction variants, which was $Q - V$ on every task. DAF is ahead on all nine, by 0.07–0.41. The gap is consistent with the observation that bilinear value models degrade as the interaction they represent becomes more complex (Hong et al., 2022): letting the state factor depend on the action strictly increases that complexity. It is also consistent with the finding that a dual representation alone is not an object conventional extraction can act on (Park et al., 2025c), which is why that line of work adds a separate value branch. The gradient identity removes the need for either. The state–goal field is enough: one reads a direction out of it rather than a number.

Table 2: DAF versus extraction from a directly learned bilinear $Q(s, a, g) = \psi(s, a)^\top \phi(g)$, under an identical training and evaluation protocol with 5 seeds. Best values are highlighted in blue.

Task	DAF	Direct dual-Q
`scene-play`	**0.81** ± 0.04	0.63 ± 0.03
`cube-triple-play`	**0.17** ± 0.03	0.10 ± 0.03
`cube-double-noisy`	**0.33** ± 0.05	0.25 ± 0.02
`puzzle-4x4-noisy`	**0.47** ± 0.03	0.06 ± 0.05
`humanoidmaze-large-stitch`	**0.48** ± 0.06	0.24 ± 0.06
`antmaze-large-navigate`	**0.93** ± 0.02	0.78 ± 0.02
`antmaze-large-stitch`	**0.56** ± 0.04	0.33 ± 0.04
`antmaze-teleport-stitch`	**0.50** ± 0.05	0.36 ± 0.03
`antmaze-teleport-navigate`	**0.51** ± 0.08	0.42 ± 0.04

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, and DAF uses this to score actions by the alignment between their predicted feature displacement and the goal direction. The resulting rule is a single inner product, needs no additional action-value function, and in the realizable case is provably the goal-conditioned Bellman advantage. Empirically it improves aggregate performance across offline GCRL benchmarks, is especially effective in manipulation tasks where local directional choices matter, and carries over to flat and hierarchical backbones alike. Dual representations should therefore be used not only as global value maps, but also as local advantage fields for goal-conditioned policy extraction.

AI use statement

We used generative AI tools, in particular a large language model (LLM), as a writing and coding assistant. Concretely, we used them to improve the clarity and grammar of the exposition and to

produce the plotting scripts. We have reviewed all AI-assisted text and code and take full responsibility for the final content of this work.

ETHICS STATEMENT

This work studies goal-reaching policies learned from fixed datasets in simulated robotic benchmarks. It involves no human subjects, no personal data and no newly released dataset. The main foreseeable risk is the generic one for offline RL: a policy extracted from data that does not cover the behavior required for a task can rank actions confidently and incorrectly, and DAF provides no uncertainty estimate that would flag this. We discuss this in Section 6 and consider uncertainty-aware action-effect models an important direction before any deployment on physical systems.

REPRODUCIBILITY STATEMENT

Our method and baselines are implemented on top of the public OGBench (Park et al., 2025a) and Dual Goal Representations (Park et al., 2025c) codebases, and all results use the unmodified OGBench datasets and evaluation protocol. Algorithm 1 gives the full training loop with a cross-reference to the defining equation for every line; Section E gives the explicit form of the actor-free coupling. Table 3 lists the hyperparameters shared by all runs and Table 4 the single per-environment hyperparameter, with the grid search that produced it in Table 5. Section C specifies the environments, dataset variants, seeds and the exact definitions of the aggregate metrics and bootstrap procedure, and Section G reports the compute used. All assumptions behind the theoretical claims are stated formally in Assumptions 1 to 3 and each proposition in Section F is proved there. Source code is provided as anonymous supplementary material.

REFERENCES

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.

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.

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.

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

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

Scott Fujimoto, Herke van Hoof, and David Meger. Addressing function approximation error in actor-critic methods. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 1587–1596. PMLR, 2018.

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*, pp. 11321–11339. PMLR, 2023.

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.

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

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

Kaiqiang Ke, Qian Lin, Zongkai Liu, Shenghong He, and Chao Yu. Conservative offline goal-conditioned implicit v-learning. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 29591–29607. PMLR, 2025a. URL https://proceedings.mlr.press/v267/ke25a.html.

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 (MLNN)*, pp. 34–41. ACM, 2025b. doi: 10.1145/3747227.3747233.

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

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

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

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.

Guillaume Matheron, Nicolas Perrin, and Olivier Sigaud. The problem with DDPG: understanding failures in deterministic environments with sparse rewards, 2019. URL https://arxiv.org/abs/1911.11679.

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

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

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.

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

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

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

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.

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

Hado van Hasselt, Arthur Guez, and David Silver. Deep reinforcement learning with double Q-learning. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, pp. 2094–2100, 2016.

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

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.

Haoran Xu, Li Jiang, Jianxiong Li, Zhuoran Yang, Zhaoran Wang, Victor Wai Kin Chan, and Xianyuan Zhan. Offline RL with no OOD actions: In-sample learning via implicit value regularization. In *International Conference on Learning Representations*, 2023. arXiv:2303.15810.

A Implementation and Reproducibility

Our method and baselines are implemented on top of the implementations given in OGBench (Park et al., 2025a) and Dual Goal Representations (Park et al., 2025c) codebases. Our method is employed upon a hierarchy of actors, with the low-level actor being updated by the dual score (Equation (11)) and the high-level actor by AWR (Equation (4)).

Table 3 details the common hyperparameters for all methods on OGBench. Table 4 shows the α regularization hyperparameter that was found to be the best for performance of DAF, and Table 5 the grid search behind it. We also report the ablation studies on important architectural aspects of our proposed method: AFU coupling Perrin-Gilbert (2024), presence of the action-effect module (Equation (12)), hierarchical actor Park et al. (2023) and integrating dual representations Park et al. (2025c) upon the hierarchical backbone.

Table 3: Network configuration for DAF on OGBench.

Configuration	Value
Gradient steps	10^6
Optimizer	Adam (Kingma & Ba, 2015)
Nonlinearity	GELU (Hendrycks & Gimpel, 2016)
Target network update rate	0.005
Goal representation dimensionality d	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
Seeds per task	5

Table 4: 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

Sensitivity to the extraction temperature. The AWR temperature α of Equation (13) is the single hyperparameter we select per environment family, following the protocol of OGBench (Park et al., 2025a) and HIQL (Park et al., 2023), which both identify it as one of the most important hyperparameters in this benchmark and publish per-environment values; Park et al. (2025c) follow the same convention. Table 5 reports the full grid over $\{0.1, 3.0, 10.0\}$ with the training budget, evaluation protocol and seeds held identical to the main tables. The ordering is genuinely environment-dependent - $\alpha = 10$ for mazes and `scene`, 3 for `cube` and `puzzle-3x3`, 0.1 for `puzzle-4x4` - and moving to the selected value changes success by 0.22-0.36 on the maze and `scene` tasks, which is why it is tuned per family rather than globally.

Table 5: Sensitivity of DAF to the AWR temperature α. Selected settings (Table 4) are highlighted in blue.

Task	$\alpha = 0.1$	$\alpha = 3.0$	$\alpha = 10.0$
humanoidmaze-medium-navigate	0.68 ±0.05	0.85 ±0.04	**0.93** ±0.03
antmaze-medium-stitch	0.61 ±0.06	0.87 ±0.04	**0.97** ±0.02
antmaze-teleport-stitch	0.28 ±0.06	0.42 ±0.06	**0.50** ±0.05
scene-play	0.54 ±0.06	0.73 ±0.05	**0.81** ±0.04
cube-double-noisy	0.12 ±0.04	**0.33** ±0.05	0.28 ±0.05
cube-triple-play	0.09 ±0.04	**0.17** ±0.03	0.12 ±0.03
puzzle-3x3-play	0.54 ±0.05	**0.74** ±0.04	0.63 ±0.05
puzzle-4x4-noisy	**0.47** ±0.03	0.28 ±0.04	0.22 ±0.03

Table 6: Ablation on OGBench. We report full DAF and four ablations: removing AFU coupling, removing the action-effect model (using the direct one-sample value-difference score of Equation (10)), removing hierarchy, and a dual-representation hierarchical baseline that extracts conventionally rather than with the dual advantage. Best values are highlighted in blue.

Task	DAF	No AFU Coupling	No Action-Effect	DAF w/o Hierarchy	Dual-Rep + Hierarchy
humanoidmaze-medium-navigate	**0.93** ±0.03	0.18 ±0.03	0.35 ±0.03	0.38 ±0.04	0.06 ±0.01
humanoidmaze-large-navigate	**0.66** ±0.03	0.02 ±0.01	0.04 ±0.01	0.39 ±0.05	0.01 ±0.00
humanoidmaze-medium-stitch	**0.90** ±0.04	0.47 ±0.06	0.50 ±0.07	0.32 ±0.06	0.05 ±0.02
humanoidmaze-large-stitch	0.48 ±0.06	0.06 ±0.02	0.08 ±0.01	**0.65** ±0.07	0.01 ±0.01
antmaze-teleport-navigate	**0.51** ±0.08	0.33 ±0.05	0.35 ±0.05	0.39 ±0.05	0.18 ±0.07
antmaze-medium-navigate	**0.98** ±0.01	0.78 ±0.05	0.93 ±0.05	0.19 ±0.04	0.89 ±0.02
antmaze-teleport-stitch	**0.50** ±0.05	0.19 ±0.04	0.17 ±0.04	0.00 ±0.01	0.09 ±0.02
antmaze-medium-stitch	**0.97** ±0.02	0.42 ±0.05	0.42 ±0.10	0.00 ±0.01	0.21 ±0.06
cube-double-play	0.41 ±0.04	0.36 ±0.07	**0.51** ±0.05	0.39 ±0.05	0.02 ±0.01
cube-triple-play	**0.17** ±0.03	0.02 ±0.01	0.04 ±0.02	0.07 ±0.02	0.01 ±0.01
cube-quadruple-play	**0.03** ±0.01	0.00 ±0.00	0.00 ±0.00	0.01 ±0.01	0.00 ±0.00
cube-double-noisy	0.33 ±0.05	0.26 ±0.05	**0.39** ±0.03	0.35 ±0.04	0.03 ±0.01
cube-triple-noisy	**0.23** ±0.01	0.01 ±0.01	0.05 ±0.02	0.02 ±0.01	0.01 ±0.00
cube-quadruple-noisy	**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
scene-noisy	**0.43** ±0.03	0.29 ±0.03	0.40 ±0.04	0.37 ±0.05	0.05 ±0.02
puzzle-3x3-play	**0.74** ±0.04	0.10 ±0.01	0.15 ±0.01	0.02 ±0.01	0.07 ±0.02
puzzle-4x4-play	0.40 ±0.05	0.18 ±0.03	0.17 ±0.03	**0.75** ±0.06	0.01 ±0.00
puzzle-3x3-noisy	**0.98** ±0.04	0.15 ±0.03	0.17 ±0.02	0.37 ±0.06	0.05 ±0.01
puzzle-4x4-noisy	**0.47** ±0.03	0.07 ±0.01	0.09 ±0.01	0.03 ±0.01	0.00 ±0.01

The dual advantage is backbone-agnostic. The same extraction applies whether the value field comes from a flat backbone or from a hierarchy. Adding the dual advantage to the flat backbone improves over the flat goal-conditioned methods, and adding it to the hierarchy improves over extracting from that hierarchy in the usual way (Tables 6 and 7). In both cases the method improves what can be extracted; the hierarchy remains a separate choice about how value is propagated over long horizons.

Table 7: The dual advantage on a flat backbone, against the flat goal-conditioned methods of the main tables. Best values are highlighted in blue.

Task	Flat DAF	GCIQL	GCIVL
humanoidmaze-medium-navigate	**0.38** ±0.04	0.31 ±0.04	0.31 ±0.03
humanoidmaze-large-navigate	**0.39** ±0.05	0.04 ±0.01	0.05 ±0.01
humanoidmaze-medium-stitch	**0.32** ±0.06	0.15 ±0.03	0.14 ±0.01
humanoidmaze-large-stitch	**0.65** ±0.07	0.02 ±0.00	0.02 ±0.01
antmaze-teleport-navigate	0.39 ±0.05	0.37 ±0.02	**0.44** ±0.02
antmaze-medium-navigate	0.19 ±0.04	**0.76** ±0.06	0.72 ±0.06
antmaze-teleport-stitch	0.00 ±0.01	0.23 ±0.02	**0.45** ±0.05
antmaze-medium-stitch	0.00 ±0.01	0.39 ±0.06	**0.48** ±0.03
cube-double-play	**0.39** ±0.05	0.35 ±0.06	0.33 ±0.05
cube-triple-play	**0.07** ±0.02	0.02 ±0.01	0.01 ±0.01
cube-quadruple-play	**0.01** ±0.01	0.00 ±0.00	0.00 ±0.00
cube-double-noisy	**0.35** ±0.04	0.24 ±0.06	0.17 ±0.04
cube-triple-noisy	0.02 ±0.01	0.05 ±0.01	**0.11** ±0.02
cube-quadruple-noisy	**0.01** ±0.00	0.00 ±0.00	0.00 ±0.00
scene-play	0.45 ±0.04	**0.53** ±0.02	0.51 ±0.05
scene-noisy	**0.37** ±0.05	0.29 ±0.02	0.31 ±0.05
puzzle-3x3-play	0.02 ±0.01	**0.98** ±0.02	0.08 ±0.02
puzzle-4x4-play	**0.75** ±0.06	0.31 ±0.02	0.26 ±0.02
puzzle-3x3-noisy	0.37 ±0.06	**0.95** ±0.01	0.44 ±0.15
puzzle-4x4-noisy	0.03 ±0.01	**0.33** ±0.11	0.24 ±0.02

The comparison is reported in Section 5.

B Related Works

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

Implicit Q-Learning (IQL) Kostrikov et al. (2022) 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 $Q_\vartheta(s,a)$. The value functions are trained via:

$$\mathcal{L}_Q(\vartheta) = \mathbb{E}_{(s,a,s')\sim\mathcal{D}}\left[\left(r(s,a) + \gamma V_\psi(s') - Q_\vartheta(s,a)\right)^2\right], \tag{16}$$

$$\mathcal{L}_V(\psi) = \mathbb{E}_{(s,a)\sim\mathcal{D}}\left[L_2^\tau\left(Q_{\bar{\vartheta}}(s,a) - V_\psi(s)\right)\right], \tag{17}$$

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 $\bar{\vartheta}$ are the parameters of the target Q network. The policy $\pi_\varphi(a|s)$ is then extracted via advantage-weighted regression (AWR) Peng et al. (2019); Wang et al. (2020):

$$J_\pi(\varphi) = \mathbb{E}_{(s,a)\sim\mathcal{D}}\left[\exp\left(\beta \cdot A(s,a)\right)\log\pi_\varphi(a|s)\right], \tag{18}$$

with $A(s,a) = Q_\vartheta(s,a) - V_\psi(s)$, 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 Ghosh et al. (2023); Xu et al. (2023). GCIVL, as used in OGBench (Park et al., 2025a), is the value-only counterpart: it fits $V_\theta(s,g)$ directly with the expectile loss of Equation (3) and extracts with AWR, so no action-value function is ever formed.

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

HIQL (Park et al., 2023) addresses long horizons by introducing a hierarchy over subgoals, but still relies on flat temporal-difference updates to a high-level value. OTA (Ahn et al., 2025) 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

$$V(s,g) \approx \mathbb{E}\big[r^{(o)}(s,g) + \gamma^{k(o)} V(s',g)\big], \tag{19}$$

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 (Wang et al., 2023; Myers et al., 2025) directly fit such distances with multistep returns: instead of bootstrapping only from immediate successors, they regress

$$d_\theta(s,g) \approx \mathbb{E}\Big[\sum_{t=0}^{K-1} c(s_t,g) + d_\theta(s_K,g) \;\Big|\; s_0 = s\Big] \tag{20}$$

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

GCIQL- and GCIVL-style methods can overestimate values for *unconnected* state-goal pairs produced by cross-trajectory pairing, because expectile regression is optimistic. CGCIVL (Ke et al., 2025a) 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 loss augments Bellman terms with

$$\mathcal{L}_{\text{cons}}(\theta) = \lambda\, \mathbb{E}_{(s,g)\sim p_{\text{pair}}}\big[(1-c(s,g))\,\big(\max\{0, v_\theta(s,g) - \delta\}\big)^2\big], \tag{21}$$

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 (Eysenbach et al., 2022). A typical loss takes the form

$$\mathcal{L}_{\text{CRL}} = -\mathbb{E}\left[\log \frac{\exp\big(\phi(s,a)^\top \psi(g^+)/\varsigma\big)}{\sum_{g'\in\mathcal{N}} \exp\big(\phi(s,a)^\top \psi(g')/\varsigma\big)}\right], \tag{22}$$

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 (Park et al., 2025a). 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 (Agarwal et al., 2021). 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.

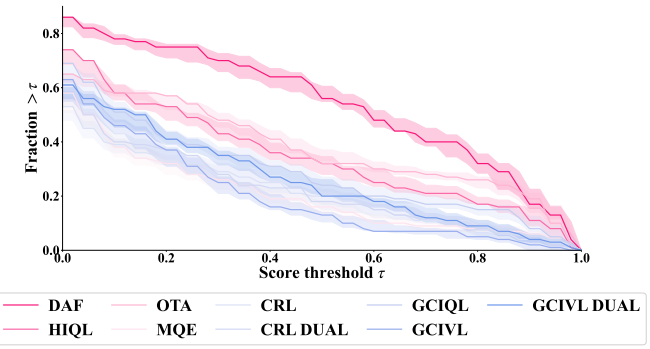

Figure 5: **Performance profile** across all tasks and environments. DAF achieves a better distribution of scores than the baselines across the OGBench evaluation suite: the fraction of runs exceeding a given normalized score is higher at every threshold.

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:

$$\text{Mean}(m) = \frac{1}{|\mathcal{E}||\mathcal{R}|} \sum_{e \in \mathcal{E}} \sum_{r \in \mathcal{R}} s_{m,e,r}, \tag{23}$$

$$\text{Median}(m) = \text{median}\left(\{s_{m,e,r}\}_{e,r}\right), \tag{24}$$

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

$$\text{OptimalityGap}(m) = \frac{1}{|\mathcal{E}||\mathcal{R}|} \sum_{e \in \mathcal{E}} \sum_{r \in \mathcal{R}} \max(0, 1 - s_{m,e,r}). \tag{26}$$

The interquartile mean (IQM) averages the middle 50% of scores and is therefore 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 (Agarwal et al., 2021) plots in Figures 5 and 6.

D.1 Deterministic-to-stochastic transfer

Because u_ξ estimates a conditional expectation, a multi-modal transition kernel makes it predict an averaged feature displacement, and the local direction could in principle become uninformative. `antmaze-teleport` is constructed to create exactly this situation: certain state-action pairs enter a teleporter that emits the agent at a randomly chosen destination, making the kernel genuinely multi-modal (Park et al., 2025a).

We isolate the effect by following the OGBench protocol of comparing `antmaze-large` with `antmaze-teleport`, which have the same size and differ only by the stochastic teleporters. Table 8 trains DAF, HIQL and OTA on `antmaze-large` and compares against the `antmaze-teleport` results of Table 1. On `stitch`, DAF retains 89% of its deterministic-maze performance against 58% for HIQL and 44% for OTA; on `navigate` all three degrade similarly. In absolute terms DAF is best on `teleport-stitch` and fourth of nine on `teleport-navigate`.

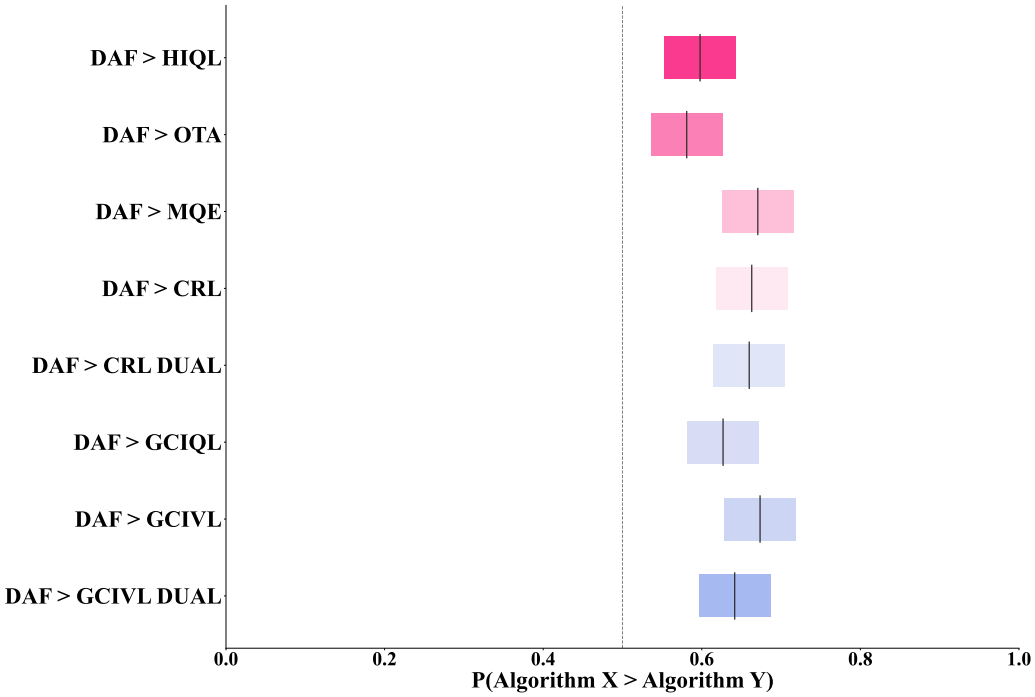

Figure 6: **RLiable Probability of Improvement.**

This is consistent with the OGBench observation that optimistically biased methods struggle under stochasticity: DAF mitigates it through the AFU pessimism on the value side (Table 6 shows the cost of removing it) and through dual embeddings that are fit to whole reachability relations rather than to single successors.

Table 8: Transfer from deterministic to stochastic dynamics. `antmaze-large` and `antmaze-teleport` are the same size and differ only by the teleporters, so the retention ratio isolates the cost of stochasticity. Retention is the `teleport` score divided by the `large` score.

Dataset	Method	`antmaze-large`	`antmaze-teleport`	Retention
navigate	DAF	0.93 ± 0.02	0.51 ± 0.08	55%
	HIQL	0.95 ± 0.01	0.46 ± 0.03	48%
	OTA	0.94 ± 0.01	0.53 ± 0.03	56%
stitch	DAF	0.56 ± 0.04	0.50 ± 0.05	**89%**
	HIQL	0.65 ± 0.04	0.38 ± 0.02	58%
	OTA	0.87 ± 0.03	0.38 ± 0.01	44%

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 (2024). The policy parameters do not receive gradients through this objective; policy learning uses only the weighted regression step of Section 4.

Surrogate dual score for the coupling. The main text defines the raw dual score z_θ in (11). In the AFU objective below it is convenient to use a non-positive surrogate

$$\widetilde{A}_\theta(s, a, g) \;:=\; h(z_\theta(s, a, g))\,, \qquad h : \mathbb{R} \to (-\infty, 0], \tag{27}$$

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

$h(x) = -\mathrm{softplus}(-x)$, which is nearly the identity for strongly negative x and saturates for positive x. The same z_θ can still be used directly in advantage-weighted regression, as in the main text; (27) is only required for the piecewise coupling with V_θ.

Scalar Bellman target. Let

$$T(s,a,g) \;:=\; r(s,g) + \gamma\, V_\theta^{\mathrm{tgt}}(s',g)\,, \tag{28}$$

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

$$\widetilde{V} \;:=\; (1-\rho U)\, V_\theta + \rho U\, \mathrm{stopgrad}(V_\theta)\,. \tag{29}$$

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

$$Z(x,y) \;=\; \begin{cases} (x+y)^2\,, & x \ge 0\,, \\ x^2 + y^2\,, & x < 0\,, \end{cases} \qquad \mathcal{L}_{\mathrm{afu}} \;=\; \mathbb{E}_{\mathcal{D}}\big[Z(x,y)\big]. \tag{30}$$

Training minimizes $\mathcal{L}_{\mathrm{afu}}$ 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 $\mathcal{L}_{\mathrm{ae}}$ in (12) complements the coupling above: the AFU-style term enforces Bellman consistency between V_θ and $\widetilde{A}_\theta$, while $\mathcal{L}_{\mathrm{ae}}$ grounds u_ξ in explicit one-step feature dynamics on the offline dataset. Removing either one degrades performance (Table 6).

F THEORETICAL ANALYSIS

This section establishes three properties of Dual Advantage Fields. Section F.1 shows that under exact representability the DAF score *is* the Bellman advantage, so DAF is a valid policy-improvement operator. Section F.2 quantifies why the action effect is learned rather than read off a single transition. Section F.3 analyses a didactic one-dimensional example, comparing DAF against flat and hierarchical value-difference extraction under a learned, noisy goal embedding.

F.1 DAF AS AN 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, recall the Bellman advantage (8). Only the ordering of actions at a given state matters for improvement, not the absolute level of V^π.

Assumption 1 (Exact dual realizability)**.** For the policy π under consideration,

(i) **Value realizability:** there exist $\psi : \mathcal{S} \to \mathbb{R}^d$ and $\phi : \mathcal{G} \to \mathbb{R}^d$ with $V^\pi(s,g) = \psi(s)^\top \phi(g)$ for all $s \in \mathcal{S}$, $g \in \mathcal{G}$;

(ii) **Exact action effect:** the action-effect model satisfies $u(s,a) = \mathbb{E}_{s' \sim P(\cdot|s,a)}\big[\gamma\psi(s') - \psi(s)\big]$ for all (s,a).

Define the population DAF score $D^\pi(s,a,g) := r(s,a,g) + u(s,a)^\top \phi(g)$.

Lemma 3 (The score is the advantage)**.** *Under Assumption 1, $D^\pi(s,a,g) = A^\pi(s,a,g)$ for all (s,a,g).*

Proof. By Assumption 1(i), $\gamma V^\pi(s',g) - V^\pi(s,g) = \phi(g)^\top(\gamma\psi(s') - \psi(s))$ for every s'. Taking the expectation over $s' \sim P(\cdot \mid s,a)$ and using linearity together with Assumption 1(ii) gives $\mathbb{E}_{s'}[\gamma V^\pi(s',g) - V^\pi(s,g)] = u(s,a)^\top \phi(g)$. Adding $r(s,a,g)$ and comparing with (8) yields the claim. □

Proposition 4 (DAF local policy improvement)**.** *Let π^+ be any goal-conditioned policy satisfying $\mathbb{E}_{a\sim\pi^+(\cdot|s,g)}[D^\pi(s,a,g)] \geq 0$ for all s, g. Then, under Assumption 1, $V^{\pi^+}(s,g) \geq V^\pi(s,g)$ for all s, g.*

Proof. By Lemma 3 the hypothesis reads $\mathbb{E}_{a\sim\pi^+(\cdot|s,g)}[A^\pi(s,a,g)] \geq 0$, which 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 of π^+. Hence $T_{\pi^+}V^\pi \geq V^\pi$ pointwise, and by monotonicity of T_{π^+}, $T^k_{\pi^+}V^\pi \geq V^\pi$ for every $k \geq 1$. Letting $k \to \infty$ and using that T_{π^+} is a γ-contraction with fixed point V^{π^+} gives $V^{\pi^+} \geq V^\pi$. □

AWR satisfies the hypothesis. The exponential reweighting of (13) is one such π^+ in the exact on-policy case. Let $\pi^+_\alpha(a \mid s,g) \propto \pi(a \mid s,g)\exp(\alpha D^\pi(s,a,g))$ with $\alpha \geq 0$ and $Z(\alpha) = \mathbb{E}_{a\sim\pi}[\exp(\alpha D^\pi(s,a,g))]$. Then

$$\frac{d}{d\alpha}\log Z(\alpha) = \mathbb{E}_{a\sim\pi^+_\alpha}[D^\pi(s,a,g)], \qquad \frac{d^2}{d\alpha^2}\log Z(\alpha) = \mathrm{Var}_{a\sim\pi^+_\alpha}[D^\pi(s,a,g)] \geq 0.$$

Since $\log Z$ is convex and its derivative at $\alpha = 0$ equals $\mathbb{E}_{a\sim\pi}[D^\pi(s,a,g)] = \mathbb{E}_{a\sim\pi}[A^\pi(s,a,g)] = 0$, the derivative is nonnegative for all $\alpha \geq 0$. Hence $\mathbb{E}_{a\sim\pi^+_\alpha}[D^\pi(s,a,g)] \geq 0$ and Proposition 4 applies.

Corollary 5 (Exact DAF policy iteration)**.** *In a finite discounted goal-conditioned MDP, suppose Assumption 1 holds at every iteration k for V^{π_k} 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}, and repeated exact DAF improvement is policy iteration and converges to an optimal goal-conditioned policy.*

Proof. By Lemma 3, $D^{\pi_k} = A^{\pi_k} = Q^{\pi_k} - V^{\pi_k}$, and $V^{\pi_k}(s,g)$ does not depend on a, so maximizing D^{π_k} over actions is equivalent to maximizing Q^{π_k}. The result follows from classical convergence of policy iteration in finite discounted MDPs, applied separately for each g. □

Relation to hierarchical policy classes. Let Π be the class of all stationary goal-conditioned primitive-action policies and $\Pi_{\text{hier}} \subseteq \Pi$ any hierarchically constrained subclass induced by a fixed subgoal or option parameterization. Since $\pi^\star \in \arg\max_{\pi\in\Pi} V^\pi$ satisfies $V^{\pi^\star}(s,g) \geq \sup_{\pi\in\Pi_{\text{hier}}} V^\pi(s,g)$ for all s, g, Corollary 5 implies that in the exact realizable limit DAF policy iteration reaches a policy at least as good as the best policy in any fixed hierarchical class. This is a *representational* statement and not a finite-sample one: hierarchy can accelerate learning by reducing the effective horizon, which is why we use it, but a fixed hierarchy can also exclude the optimal primitive-action policy.

F.2 Why learn the action-effect module?

Assumption 1(ii) asks for a conditional expectation. A single dataset transition provides one draw from $P(\cdot \mid s,a)$, so the obvious alternative - scoring with $\widehat{z}(s,a,s',g) = (\gamma\psi(s') - \psi(s))^\top\phi(g)$, which is (10) up to the reward term - is an unbiased but noisy estimate of the same quantity. The following makes the cost precise.

Proposition 6 (One-sample scoring can invert the action ranking)**.** *Fix s and g, let $a, b \in \mathcal{A}$ and write $\sigma_a^2 := \mathrm{Var}_{s'\sim P(\cdot|s,a)}\big(\phi(g)^\top\psi(s')\big)$, similarly σ_b^2. Let $m := D^\pi(s,a,g) - D^\pi(s,b,g) > 0$, so that a is genuinely the better action, and let $\widehat{z}$ be evaluated at independent draws $s'_a \sim P(\cdot \mid s,a)$ and $s'_b \sim P(\cdot \mid s,b)$ with the reward term included. Then*

$$\mathbb{E}\big[\widehat{z}(s,a,s'_a,g)\big] = D^\pi(s,a,g), \qquad \mathrm{Var}\big(\widehat{z}(s,a,s'_a,g)\big) = \gamma^2\sigma_a^2, \tag{31}$$

and the probability that the one-sample rule prefers b to a satisfies

$$\mathbb{P}\big[\widehat{z}(s,b,s'_b,g) \geq \widehat{z}(s,a,s'_a,g)\big] \;\leq\; \frac{\gamma^2(\sigma_a^2+\sigma_b^2)}{m^2}, \tag{32}$$

with the bound replaced by the exact value $\Phi\big(-m/(\gamma\sqrt{\sigma_a^2+\sigma_b^2})\big)$ when $\phi(g)^\top\psi(s')$ is Gaussian under both kernels, where Φ is the standard normal c.d.f. By contrast, the population minimizer of $\mathcal{L}_{\text{ae}}$ (12) satisfies $u_\xi(s,a) = \mathbb{E}_{s'}[\gamma\psi(s') - \psi(s)]$ exactly, so the amortized score z_θ of (11) prefers a to b with probability one.

Proof. Equation (31) is immediate: $\psi(s)$ and $\phi(g)$ are deterministic given (s,g), so $\widehat{z} = r(s,a,g) - \psi(s)^\top\phi(g) + \gamma\,\phi(g)^\top\psi(s')$ is an affine function of the scalar random variable $\phi(g)^\top\psi(s')$ with slope γ. For Equation (32), write $\widehat{z}(s,a,s'_a,g) = D^\pi(s,a,g) + \xi_a$ and likewise for b, where ξ_a, ξ_b are independent, zero-mean, with variances $\gamma^2\sigma_a^2$ and $\gamma^2\sigma_b^2$. The event in question is $\xi_b - \xi_a \geq m$, and $\mathrm{Var}(\xi_b - \xi_a) = \gamma^2(\sigma_a^2+\sigma_b^2)$, so Chebyshev's inequality gives the bound; under joint Gaussianity $\xi_b - \xi_a$ is normal and the probability is exactly $\Phi(-m/(\gamma\sqrt{\sigma_a^2+\sigma_b^2}))$. The final claim is the standard fact that the minimizer of a squared-error regression objective is the conditional mean of its target, here $\mathbb{E}[\gamma\psi(s') - \psi(s) \mid s,a]$; substituting into (11) recovers D^π up to the action-independent reward term, whose ordering is unaffected. □

F.3 Robustness to learned embedding noise: a didactic example

We now move to the regime where the representation is learned from finite data and therefore contains noise, and compare three extraction rules built on the *same* learned value and the same action-effect model.

Assumption 2 (Line world with nuisance coordinates)**.** (i) **Dynamics.** States $s \in \{0,\dots,T\}$ with fixed goal $g = T > 0$; actions $a \in \{-1,+1\}$ move deterministically to $s+a$; the episode terminates at g; the reward is 0 at the goal and -1 elsewhere. Hence the optimal action is $+1$ for all $s < T$ and $V^\star(s,g) = s - T$.

(ii) **Embedding.** $\psi(s) = [\,s,\ 1,\ f_1(s),\dots,f_m(s)\,]^\top \in \mathbb{R}^{m+2}$, where the f_i are bounded with well-defined first and second differences. The first two coordinates are *essential*; the remaining m are *nuisance* coordinates, irrelevant to control (*e.g.* visual textures or lighting gradients).

(iii) **True goal embedding.** $\phi^\star(g) = [\,1,\ -T,\ 0,\dots,0\,]^\top$, so that $\psi(s)^\top\phi^\star(g) = s - T = V^\star(s,g)$.

(iv) **Learned goal embedding.** $\phi(g) = \phi^\star(g) + \varepsilon$ with $\varepsilon = [0, 0, \eta_1, \ldots, \eta_m]^\top$ and $\eta_i \sim \mathcal{N}(0, \sigma_i^2)$ independent: the temporal-difference loss constrains the nuisance coefficients only weakly, so the noise is confined to them. The essential coordinates are assumed exact; allowing noise there does not change the conclusions but obscures the algebra. Consequently $\widehat{V}(s, g) = \psi(s)^\top \phi(g) = s - T + \sum_{i=1}^m \eta_i f_i(s)$.

(v) **Subgoal embedding.** Subgoals are embedded by the same map with independent noise, $\phi(s_{\text{sub}}) = \phi^\star(s_{\text{sub}}) + \tilde{\varepsilon}$, $\tilde{\eta}_i \sim \mathcal{N}(0, \sigma_i^2)$ independent of η_i.

(vi) **Exact action effect.** $u(s, a) = \gamma\psi(s + a) - \psi(s)$, which is attainable here because the dynamics are deterministic and transitions are abundant.

The three rules under Assumption 2 are: *flat value-difference* $a_{\text{V}}(s) = \arg\max_a \widehat{V}(s + a, g)$, which is the extraction implicit in HIQL's flat baseline; *DAF* $a_{\text{DAF}}(s) = \arg\max_a u(s, a)^\top \phi(g)$; and *hierarchical* extraction, which first picks $s_{\text{sub}} = \arg\max_{x \in \{s+k, s-k\}} [\widehat{V}(x, g) - \widehat{V}(s, g)]$ for a stride $k \geq 2$, then applies the flat rule toward the chosen subgoal, $a_\ell(s) = \arg\max_a \widehat{V}(s + a, s_{\text{sub}})$. Write $\Delta f(s) := f(s + 1) - f(s - 1)$ for the first difference.

The first observation is a negative one, and it is worth being explicit about because it bounds what the didactic model can support.

Lemma 7 (In this model, DAF and flat extraction make the same decision)**.** *Under Assumption 2, the DAF and flat decision margins satisfy*

$$\Delta_{\text{DAF}}(s) := \big(u(s, +1) - u(s, -1)\big)^\top \phi(g) = \gamma \left(\widehat{V}(s + 1, g) - \widehat{V}(s - 1, g)\right) = \gamma\, \Delta_{\text{V}}(s), \quad (33)$$

so the two rules select the same action at every state and have identical error probabilities.

Proof. $u(s, +1) - u(s, -1) = (\gamma\psi(s+1) - \psi(s)) - (\gamma\psi(s-1) - \psi(s)) = \gamma(\psi(s+1) - \psi(s-1))$. Taking the inner product with $\phi(g)$ and using $\widehat{V}(x, g) = \psi(x)^\top \phi(g)$ gives the identity. Since $\gamma > 0$, the two margins have the same sign. □

Remark 8. Lemma 7 rules out the tempting claim that DAF filters representation noise through *second* differences $f_i(s+1) + f_i(s-1) - 2f_i(s)$: the action-effect vectors differ by a first difference, not a second one. In this idealized model, therefore, the advantage of DAF over flat extraction is *not* a noise-filtering effect. The two places where DAF provably differs are Proposition 6, which concerns stochastic kernels and is invisible here by Assumption 2(i), and the comparison with hierarchy below, which concerns the number of noisy decisions that have to be made.

Proposition 9 (Error probabilities of the three rules)**.** *Under Assumption 2, for any $s \in \{1, \ldots, T-1\}$ and stride k, with Φ the standard normal c.d.f.,*

$$\begin{aligned} \varepsilon_{\text{DAF}}(s) = \varepsilon_{\text{flat}}(s) = \varepsilon_{\text{low}}(s) &= \Phi\Big(-2\big(\textstyle\sum_i \sigma_i^2 (\Delta f_i(s))^2\big)^{-1/2}\Big), \\ \varepsilon_{\text{high}}(s) &= \Phi\Big(-2k\big(\textstyle\sum_i \sigma_i^2 (f_i(s+k) - f_i(s-k))^2\big)^{-1/2}\Big). \end{aligned} \quad (34)$$

Proof. Each decision margin is an affine function of the Gaussian vector $\boldsymbol{\eta}$ (or $\tilde{\boldsymbol{\eta}}$ for the low level) and is therefore Gaussian. Direct computation gives $\Delta_{\text{V}}(s) = 2 + \sum_i \eta_i \Delta f_i(s)$, $\Delta_{\text{high}}(s) = 2k + \sum_i \eta_i (f_i(s+k) - f_i(s-k))$ and $\Delta_{\text{low}}(s) = 2 + \sum_i \tilde{\eta}_i \Delta f_i(s)$ conditioned on the subgoal $s+k$ being chosen. Each rule errs exactly when its margin is negative, and $\mathbb{P}[\mathcal{N}(\mu, \varsigma^2) < 0] = \Phi(-\mu/\varsigma)$. The first chain of equalities uses Lemma 7 and the fact that η and $\tilde{\eta}$ have the same variances. □

Assumption 3 (Hierarchical failure model)**.** The hierarchical decision at s is counted as correct only if *both* stages are correct: (i) the two stages' noise sources are independent, which holds under Assumption 2(v); and (ii) when the high level selects the wrong subgoal, the resulting primitive action is wrong - a low-level error is not credited with accidentally recovering the correct action.

Assumption 3(ii) is the conservative bookkeeping convention used in the hierarchical analysis of Park et al. (2023, Proposition 4.1); without it, the two errors could cancel and the hierarchical error would be the symmetric difference of the two events rather than their union.

Proposition 10 (Exact hierarchical error and two-sided bounds)**.** *Under Assumptions 2 and 3, the hierarchical error probability is*

$$\varepsilon_{\text{hier}}(s) = \varepsilon_{\text{high}}(s) + \big(1 - \varepsilon_{\text{high}}(s)\big)\,\varepsilon_{\text{low}}(s), \tag{35}$$

and consequently

$$\max\{\varepsilon_{\text{high}}(s), \varepsilon_{\text{low}}(s)\} \;\le\; \varepsilon_{\text{hier}}(s) \;\le\; \varepsilon_{\text{high}}(s) + \varepsilon_{\text{low}}(s). \tag{36}$$

In particular $\varepsilon_{\text{DAF}}(s) = \varepsilon_{\text{low}}(s) \le \varepsilon_{\text{hier}}(s)$*, with strict inequality whenever* $\varepsilon_{\text{high}}(s) > 0$ *and* $\varepsilon_{\text{low}}(s) < 1$.

Proof. Let E_{high} and E_{low} be the two stage-error events. By Assumption 3 the hierarchical rule errs on $E_{\text{high}} \cup E_{\text{low}}$, and by independence $\mathbb{P}[E_{\text{high}} \cup E_{\text{low}}] = \varepsilon_{\text{high}} + \varepsilon_{\text{low}} - \varepsilon_{\text{high}}\varepsilon_{\text{low}}$, which is Equation (35). Writing Equation (35) in the two equivalent forms $\varepsilon_{\text{high}} + (1 - \varepsilon_{\text{high}})\varepsilon_{\text{low}}$ and $\varepsilon_{\text{low}} + (1 - \varepsilon_{\text{low}})\varepsilon_{\text{high}}$ and using $\varepsilon \in [0, 1]$ gives $\varepsilon_{\text{hier}} \ge \varepsilon_{\text{high}}$ and $\varepsilon_{\text{hier}} \ge \varepsilon_{\text{low}}$, hence the lower bound in Equation (36); the upper bound follows by dropping the nonnegative product term. The final statement combines this with $\varepsilon_{\text{DAF}} = \varepsilon_{\text{low}}$ from Proposition 9, and the inequality is strict because the discarded term $\varepsilon_{\text{high}}(1 - \varepsilon_{\text{low}})$ is then strictly positive. □

Proposition 10 replaces a union bound with an identity, which matters for the direction of the comparison: an upper bound on $\varepsilon_{\text{hier}}$ cannot show that hierarchy is worse than flat extraction, whereas the lower bound $\varepsilon_{\text{hier}} \ge \varepsilon_{\text{low}} = \varepsilon_{\text{DAF}}$ can.

Remark 11 (Scope of the didactic model). Assumption 2 grants every rule an accurate value field, corrupted only in control-irrelevant directions. That is exactly the regime in which hierarchy has nothing to contribute, since its purpose is to improve the *propagation* of value over long horizons rather than the reliability of a single decision. The model therefore says something narrow but useful: whatever benefit hierarchy provides in practice must come from value quality, not from decision reliability, because in the decision-reliability accounting it strictly loses. This is consistent with Table 6, where removing the hierarchy helps on the two tasks whose value fields are apparently already adequate (`humanoidmaze-large-stitch`, `puzzle-4x4-play`) and collapses on `antmaze-stitch`, where long-horizon propagation is the bottleneck.

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.