# THE CRITIC IS ONLY A PROXY: REGRET OPTIMIZATION FOR REINFORCEMENT LEARNING

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

When direct evaluation is costly, learning systems can turn limited observations into a reusable source of feedback. Actor-critic reinforcement learning follows this approach, but its critic is only an informative, imperfect proxy for environmental return. The actor therefore faces an intrinsic robustness problem even without environment shift. We propose to protect against the worst plausible regret: the gap between the current policy and the best action under the same plausible critic. To develop a practical algorithm, we further replace hard hindsight search by a soft comparison and develop Distributionally Robust Relaxed Regret Optimization (DR3O), which can be efficiently estimated from counterfactual action samples. This principle can serve as a general framework when setting up the objective against which the actor optimizes. As an example, we use this approach to indicate "which $Q$ to give to the actor" in the well-known twin critics. As a result, across MuJoCo, MetaWorld, and the DeepMind Control Suite, changing this actor-facing signal improves matched training backbones without any additional efforts. A data-dependent ellipsoidal bonus provides further gains, supporting regret optimization as a practical way to learn from imperfect feedback.

## 1 INTRODUCTION

How a system can learn depends on both the data amount available and the cost of evaluating a new decision. Many applications offer neither abundant labeled data nor unlimited cheap evaluation. A useful response is to learn an evaluator from available observations and reuse its predictions as feedback. We refer to this broad pattern as *learning through learned feedback*. One well-known instance is the family of actor-critic methods, which have long used a learned value function called the *critic* to guide policy improvement of the *actor* (Barto et al., 1983; Konda & Tsitsiklis, 2000).

This reuse introduces a fundamental "proxy issue": the actor optimizes against the critic, but the critic itself is a fitted function from finite data. The critic can therefore contain valuable yet not perfect decision information. This proxy issue is intrinsic to the architecture, even when the training environment when we collect data is the same as the environment where we evaluate the algorithm, leading to problems such as the overestimation bias or the over-optimization.

To mitigate the proxy issue, a common strategy is to treat the critic's signals pessimistically, such as using twin-critic lower envelopes (Haarnoja et al., 2019), adding confidence penalties (Ghasemipour et al., 2022), or evaluating under truncated return distributions (Kuznetsov et al., 2020). However, despite the efforts in different ways of *how* to get pessimistic values, seldom has asked *if* the value is the correct objective to pessimize directly. In this work, we argue that what should be pessimized is the *relative* loss (the regret) rather than the *absolute* value. In other words, we ask: how much could the actor lose relative to the best decision under the same plausible critic? We formulate this problem as distributionally robust regret optimization (DRRO). Each plausible critic compares the actor with its own best action, and the actor minimizes the largest resulting regret. The comparison is policy-relative: an uncertain alternative matters through its potential improvement over what the actor already does.

Although minimax regret is a longstanding criterion, its "min-max-max" structure makes it difficult or even strongly NP-hard for those simplest decision models (Averbakh & Lebedev, 2005; Fiechtner & Blanchet, 2025). Turning this principle into a practical deep-learning update is thus the central challenge. To tackle this, we first formulate the inner problem inside the minimax regret as optimizing



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.

a linear objective over the policy space. This linear structure enables a clear duality, which reduces the double maximization to a single maximization. Later, relaxing the maximum makes it possible to be efficiently estimated via sampling. The resulting objective is a natural lower bound on the worst-case regret, and running policy gradient on that leads to a practical algorithm with minimal efforts. We call this framework *Distributionally Robust Relaxed Regret Optimization* (DR3O). We test the framework across multiple datasets, including MuJoCo, MetaWOrld, and the DeepMind Control Suite. By simply indicating "which $Q$ to give to the actor" on the well-known algorithm backbones such as SAC, TD3, TD7, and CrossQ, the outcome has witnessed a consistent improve. We also develop another variant of the DR3Ousing feature-ellipsoidal ambiguity geometry, which gains even more, implying strong evidence of the potential of the DR3Oprinciple.

Our contributions are:

- **A decision principle for imperfect feedback.** We formulate the actor's proxy issue as worst-case regret minimization, retaining useful critic information while protecting against costly action comparisons.
- **A tractable regret objective.** We derive a lower bound on hard worst-case regret, quantify its approximation gaps, and obtain a policy gradient estimable by counterfactual action sampling.
- **A minimal twin-critic intervention.** Selecting which critic to give the actor improves matched backbones across multiple benchmarks, while preserving all those remain, including critic training, replay mechanisms, and update schedules.
- **A data-dependent extension.** A feature-ellipsoidal ambiguity model produces an actor bonus that also yields consistent empirical gains, illustrating the reach of the same principle beyond critic selection.

## 2 Decision model and proxy uncertainty

Consider a Markov decision process (MDP) with states $s \in \mathcal{S}$, actions $a \in \mathcal{A}$, transition kernel $P$, and reward $r$. Without loss of generality, we consider the discounted MDP with discount $\gamma \in [0, 1)$, where our method also applies to the finite-horizon MDP. A policy $\pi(\cdot \mid s)$ is a probability distribution over actions. For a continuation policy $\pi_t$, the action value is

$$Q^{\pi_t}(s, a) = \mathbb{E}\left[\sum_{k=0}^{\infty} \gamma^k r(S_k, A_k) \,\middle|\, S_0 = s, A_0 = a,\ A_{k\geq 1} \sim \pi_t\right].$$

An actor-critic method fits an estimate $\widehat{Q}_t$ from interaction data and uses it to improve the actor via policy gradient updates. In our paper, we derive our algorithm in one single step of policy gradient: the current critic and continuation are fixed while the actor changes its action distribution for a fixed state $s$. The surrounding training loop then collects data and updates the critic again.

Fix a state and suppress $s$ and $t$. Write $\pi \in \mathcal{P}(\mathcal{A})$ for the actor's action distribution, $\mathbb{E}_\pi Q = \int Q(a)\pi(\,\mathrm{d}a)$ for its predicted value, and $\mathcal{Q}$ for a set of plausible critics where the oracle $Q$-function would probably lie in. The nominal objective maximizes $\mathbb{E}_\pi \widehat{Q}$. Two uncertainty-aware alternatives are

$$\underbrace{\max_{\pi \in \mathcal{P}(\mathcal{A})} \inf_{Q \in \mathcal{Q}} \mathbb{E}_\pi Q}_{\text{robust value (DRO)}} \qquad \text{and} \qquad \underbrace{\min_{\pi \in \mathcal{P}(\mathcal{A})} \sup_{Q \in \mathcal{Q}} \{ \sup_{q \in \mathcal{P}(\mathcal{A})} \mathbb{E}_q Q - \mathbb{E}_\pi Q \}}_{\text{robust regret (DRRO)}}. \tag{1}$$

Here, the "D" stands for the term "distributionally", where the plausible $Q$-functions are replaced by plausible distributions of $Q$-functions. However, it can be reduced to the non-distributional case since the objective is linear in $Q$. Appendix A.2 proves the reduction and explains the information structure. We use the term DRO and DRRO simply following the convention in the literature.

One thing to note is that the regret uses the same critic for both the actor and its comparator. With one known critic ($\mathcal{Q}$ being a singleton), its maximum is constant in $\pi$, so the two objectives agree. With several plausible critics, the best alternative changes across scenarios. We use a simple example to illustrate the difference behind those two approaches.

### 2.1 WHY ROBUSTIFYING REGRET RATHER THAN VALUE?

Worst-case value asks whether the payoff is high even in an unfavorable scenario. Worst-case regret asks whether another decision would have done much better in that scenario. This distinction lets regret use promising but uncertain information without committing fully to it. A one-step allocation problem illustrates the difference.

**Example 1** (An informative but imperfect return)**.** Split \$100 between an asset returning exactly $5\%$ and an asset estimated to return $6\%$, with a plausible range $[4\%, 8\%]$. Let $x$ be the fraction in the second asset. Nominal value maximizes $5 + x$ and chooses $x = 1$; worst-case value maximizes $5 - x$ and chooses $x = 0$. Worst-case regret instead solves

$$\min_{x\in[0,1]} \max\{x,\, 3(1-x)\}, \qquad x^\star = 3/4. \tag{2}$$

The two terms are the dollar losses relative to the better asset at returns $4\%$ and $8\%$. Equalizing them invests \$75 in the more promising asset and \$25 in the certain one, balancing the cost of either missed opportunity.

### 2.2 MODELING UNCERTAINTY IN THE PROXY

We continue to specify the model we use in this work. The uncertainty is modeled directly through its possible mean critics:

$$\mathcal{Q} = \{Q_h = \widehat{Q} + \delta h :\ h \in \mathcal{H},\ \|h\|_{\mathcal{H}} \le 1\}, \qquad \delta \ge 0. \tag{3}$$

Here $\widehat{Q}$ is the nominal estimate, $\delta$ sets the uncertainty size, and the function norm describes the geometry of the critics and how estimation errors across actions can vary together. This formulation focuses on the feedback the actor actually uses. We take $\mathcal{A}$ compact for the main development and state the regularity assumptions in Appendix A.1. As stated before, we focus on one step of the policy gradient, so the critic, ambiguity geometry, and replay-state distribution remain fixed within each actor step.

## 3 FROM HARD REGRET TO A TRACTABLE SOFT TARGET

The DRRO problem is known to be NP-hard in the literature for even simple models (Averbakh & Lebedev, 2005; Fiechtner & Blanchet, 2025). It leaves a computational question: how can an actor improve against a worst-case critic without solving a new computationally intractable optimization at every update? In this section, we are tackling the difficulty: We first eliminate the critic perturbation by duality. This exposes a hard search for the maximum over actions, which is still unfavorable, motivating a soft comparison based on expectations. A final relaxation separates the remaining coupled adversaries and gives an explicit sampling target.

### 3.1 THE HARD ADVERSARY: A POLICY-RELATIVE COMPARISON

The trickiest part of the DRRO problem in (1) is the double-maximum structure inside the outer minimization problem. In this subsection, we will analyze the problem and show that the double maximum can be compressed into one maximum using duality.

The hard worst-case regret has two maximizing choices, a perturbation $h$ and a comparator $q$. Since both are suprema over a product set, their order can be exchanged:

$$\begin{aligned}\mathcal{R}_0(\pi) &\coloneqq \sup_{\|h\|_{\mathcal{H}}\le 1} \sup_{q\in\mathcal{P}(\mathcal{A})} \left\langle q - \pi, \widehat{Q} + \delta h \right\rangle \\ &= \sup_{q\in\mathcal{P}(\mathcal{A})} \{\mathbb{E}_q \widehat{A}_\pi + \delta d_{\mathcal{H}}(q,\pi)\}, \qquad d_{\mathcal{H}}(q,\pi) \coloneqq \sup_{\|h\|_{\mathcal{H}}\le 1} \langle q - \pi, h\rangle\,. \end{aligned} \tag{4}$$

Here $\langle q, f\rangle = \int f \,\mathrm{d}q$. The inner optimization is the dual norm of the signed measure $q - \pi$ (or a pseudometric if the function class does not distinguish every distribution). Its meaning is the largest plausible error in the *comparison* between $q$ and $\pi$.

The set $\mathcal{P}(\mathcal{A})$ contains all Borel probability distributions on the compact action space, including the actor's distribution. It is convex and compact under weak convergence, its extreme points are the point masses $\delta_z$, and it is their closed convex hull. The objective in (4) is convex in $q$, because the dual term is a supremum of linear functions. Under our continuity assumptions, a maximizer can therefore be chosen at an extreme point. Geometrically, the adversary only needs to compare the actor with one action.

**Proposition 1** (Exact hard-adversary reduction)**.** *Let $\widehat{A}_\pi(z) \coloneqq \widehat{Q}(z) - \mathbb{E}_\pi \widehat{Q}$ denote the nominal advantage of action $z$ over the current actor. For $D_{\mathcal{H}}(z;\pi) \coloneqq d_{\mathcal{H}}(\delta_z, \pi)$,*

$$\mathcal{R}_0(\pi) = \sup_{z \in \mathcal{A}} \{\widehat{A}_\pi(z) + \delta D_{\mathcal{H}}(z;\pi)\}. \tag{5}$$

*An optimal inner adversary chooses a maximizing action $z^\star$ and a perturbation maximizing $h(z^\star) - \mathbb{E}_\pi h$ over the unit ball.*

This solves the function-space part exactly: the adversary seeks a large advantage plus an uncertainty allowance relative to the actor. The remaining global action maximization can still be expensive for a neural critic on a continuous space, or for an exponentially large completion space in RLHF. Evaluating a critic at sampled alternatives is much easier than certifying its best action, motivating the next step.

### 3.2 Canonical soft regret and the coupled adversaries

We replace the hard maximum by a soft maximum so that hindsight comparison can be expressed through an expectation and estimated by sampling. Choose a fixed reference probability distribution $\nu$ and a temperature $\tau > 0$, and define

$$\mathrm{S}_{\tau,\nu}(f) := \tau \log \mathbb{E}_\nu e^{f/\tau} = \sup_{q \ll \nu} \{\mathbb{E}_q f - \tau \,\mathrm{KL}(q\|\nu)\}. \tag{6}$$

For $n$ actions and uniform $\nu$, this is the usual log-sum-exp soft maximum minus the normalization constant $\tau \log n$. Smaller temperatures concentrate on better actions, yet introducing harder sampling procedures. The variational identity explains the role of $\nu$: the soft comparator trades predicted value against a KL penalty for moving away from its reference. The expectation on the left admits Monte Carlo estimation.

Replacing the best-action value within each regret comparison gives the *canonical soft regret*. Gibbs duality (6), followed by the perturbation duality in (4), yields

$$\begin{aligned}\mathcal{R}_\tau^{\mathrm{can}}(\pi) &:= \sup_{\|h\|_{\mathcal{H}} \le 1} \{\mathrm{S}_{\tau,\nu}(\widehat{Q} + \delta h) - \mathbb{E}_\pi(\widehat{Q} + \delta h)\} \\ &= \sup_{q \ll \nu} \{\mathbb{E}_q \widehat{A}_\pi + \delta d_{\mathcal{H}}(q,\pi) - \tau \,\mathrm{KL}(q\|\nu)\}.\end{aligned} \tag{7}$$

For a given perturbation, the best soft comparator is explicit. For a given comparator, the perturbation is a dual-norm maximizer. However, their jointly optimal choices satisfy the coupled conditions

$$\frac{\mathrm{d}q^\star}{\mathrm{d}\nu}(z) \propto e^{[\widehat{Q}(z) + \delta h^\star(z)]/\tau}, \qquad h^\star \in \operatorname*{arg\,max}_{\|h\|_{\mathcal{H}} \le 1} \langle q^\star - \pi, h \rangle . \tag{8}$$

The first cannot be evaluated without the second, and the second depends on the entire distribution in the first. Eliminating $h$ leaves a convex distance plus a concave negative-KL term, generally not a concave maximization problem. Alternating the two conditional optimizations need not recover the global optimum. Sampling alone therefore does not solve canonical regret: one still has to find a *shared* perturbation optimal for all sampled alternatives together.

### 3.3 Separating the perturbations with Jensen's inequality

The coupling suggests allowing each alternative its own perturbation. A distribution $q$ is a mixture of point masses, $q = \mathbb{E}_{Z\sim q}\delta_Z$. Convexity of the dual distance then gives

$$d_{\mathcal{H}}(q,\pi) = d_{\mathcal{H}}(\mathbb{E}_{Z\sim q}\delta_Z, \pi) \le \mathbb{E}_{Z\sim q} d_{\mathcal{H}}(\delta_Z, \pi) = \mathbb{E}_q D_{\mathcal{H}}(Z;\pi). \tag{9}$$

On the left, one perturbation separates the whole comparator distribution from the actor; on the right, each $Z$ is evaluated against its own most favorable perturbation. Substituting this upper bound into (7) makes the non-entropy part linear in $q$. Gibbs duality now solves the comparator explicitly:

$$\begin{aligned}\overline{\mathcal{R}}_\tau(\pi) &:= \sup_{q \ll \nu} \{\mathbb{E}_q[\widehat{A}_\pi + \delta D_{\mathcal{H}}(\cdot;\pi)] - \tau\,\mathrm{KL}(q\|\nu)\} \\ &= \mathrm{S}_{\tau,\nu}[\widehat{A}_\pi + \delta D_{\mathcal{H}}(\cdot;\pi)], \qquad (10)\end{aligned}$$

$$\frac{\mathrm{d}\bar{q}_\pi}{\mathrm{d}\nu}(z) = \frac{e^{[\widehat{Q}(z)+\delta D_{\mathcal{H}}(z;\pi)]/\tau}}{\mathbb{E}_\nu e^{[\widehat{Q}+\delta D_{\mathcal{H}}(\cdot;\pi)]/\tau}}. \qquad (11)$$

This is the relaxed regret target of DR3O. Once singleton distances are available, counterfactual actions can be scored separately and combined by exponential weighting. Equivalently, we have moved the perturbation supremum inside the soft maximum, using $\sup_h \mathrm{S}\, f_h \le \mathrm{S} \sup_h f_h$. The comparator emphasizes alternatives with high nominal advantage or substantial uncertainty relative to the current policy.

**Proposition 2** (Ordering and approximation gaps)**.** *The three targets satisfy*

$$\mathcal{R}_\tau^{\mathrm{can}}(\pi) \le \overline{\mathcal{R}}_\tau(\pi) \le \mathcal{R}_0(\pi). \qquad (12)$$

Thus the relaxation is an upper bound on the difficult soft problem and a lower bound on the original hard problem. We also include some more discussions on how large those gaps are in Appendix A.5 for the hard-to-soft gap and Appendix A.6 for the Jensen gap.

Finally, we have obtained the tractable objective $\overline{\mathcal{R}}_\tau(\pi)$ to be minimized for the actor $\pi$. We call this relaxed worst-case regret Distributionally Robust Relaxed Regret Optimization (DR3O).

**Choosing the reference.** Uniform $\nu$ on a bounded action box gives equal prior weight per unit volume. An expert distribution or a feasible-action prior can instead emphasize informed alternatives. The KL term regularizes the *comparator* toward this reference, and the actor reduces its regret against that comparator. We keep $\nu$ fixed during actor optimization; a separate sampling proposal may adapt to the actor without changing the objective. A fixed analytic reference, such as a uniform law, requires no additional policy network.

## 4 Which Q to give to the actor?

Before deriving a general algorithm, consider an important case following the widely-adopted twin critics. Twin critics are separately initialized and trained value estimates $Q_1, Q_2$, introduced to reduce the overestimation associated with function approximation. Twin Delayed Deep Deterministic Policy Gradient (TD3) uses their minimum in its bootstrapped target, but gives $Q_1$ to the actor (Fujimoto et al., 2018). Modern Soft Actor-Critic (SAC) uses a clipped twin target and gives the actor $\min(Q_1, Q_2)$, alongside its entropy term (Haarnoja et al., 2018; 2019). These are two different ways to use the same type of proxy information. In this section, we will show how we derive the DR3O principle on top of the twin critics, requiring only a choice between two existing critics. This choice exceeds the performance of the TD3 and the SAC update rule later in Section 6.

### 4.1 From pessimistic values to coherent critic uncertainty

Using a designated $Q_1$ treats that estimate as the actor's nominal objective. Using the lower envelope is a worst-case-value rule. Indeed, with $\ell(a) = \min_i Q_i(a)$ and $u(a) = \max_i Q_i(a)$, define the rectangular ambiguity set

$$\mathcal{Q}_{\mathrm{rect}} = \{Q : \ell(a) \le Q(a) \le u(a) \text{ for every } a \in \mathcal{A}\}. \qquad (13)$$

It permits the value at each action to vary independently, so $\inf_{Q \in \mathcal{Q}_{\mathrm{rect}}} \mathbb{E}_\pi Q = \mathbb{E}_\pi \ell$.

What would regret optimization do with this same set? For a non-atomic policy on a continuous action space, its worst-case regret is

$$\mathcal{R}_0^{\mathrm{rect}}(\pi) = \sup_z u(z) - \mathbb{E}_\pi \ell. \qquad (14)$$

To favor a comparator $z$, the adversary sets $Q(z) = u(z)$ and $Q(b) = \ell(b)$ elsewhere. The spike raises the comparator value but costs nothing under a policy assigning zero mass to $z$. The first term in (14) is independent of the actor, so hard regret and robust value induce the same optimization over non-atomic policies.

This collapse reflects the rectangular model's uncoupled errors: we cannot allow the plausible critics to be arbitrarily chosen among every action. Even if we force the critics to be continuous, it still allows arbitrarily narrow spikes and the same supremum. The observation suggests that we have to pose at least *some* kind of regularity on the plausible critics. For twin critics, a particularly simple way to preserve their actionwise structure is to use one coefficient across the entire function:

$$\mathcal{Q}_{\text{twin}} = \{\lambda Q_1 + (1-\lambda)Q_2 : \lambda \in [0,1]\}. \tag{15}$$

The disagreement between two fitted critics now describes a coherent direction of uncertainty, rather than independent intervals at each action. This is a special case of (3): set

$$\widehat{Q} = \tfrac{1}{2}(Q_1 + Q_2), \quad \Delta Q = \tfrac{1}{2}(Q_1 - Q_2), \quad \mathcal{H}_{\text{twin}} = \text{span}\{\Delta Q\}, \quad \|\xi \Delta Q\|_{\mathcal{H}_{\text{twin}}} = |\xi|, \tag{16}$$

and choose $\delta = 1$. Its unit ball gives exactly the segment (15).

### 4.2 Regret selects the critic with the larger missed opportunity

For each critic, compare its best predicted value with the value of the current policy. The adversary chooses the critic making this missed opportunity largest. Checking only $Q_1$ and $Q_2$ is exact: hard maximum and soft maximum are both convex in the critic, while subtracting $\mathbb{E}_\pi Q$ is linear. Their regret is therefore maximized at an endpoint of the segment.

**Proposition 3** (Exact canonical selector)**.** *Let $M_{i,0} = \sup_a Q_i(a)$ be the hard maximum over $\mathcal{A}$ and $M_{i,\tau} = \mathrm{S}_{\tau,\nu}(Q_i)$ be the soft maximum. For the hard target ($t = 0$) and canonical soft target ($t = \tau$),*

$$\mathcal{R}_t^{\text{twin}}(\pi) = \max_{i=1,2}\{M_{i,t} - \mathbb{E}_\pi Q_i\}, \qquad i^\star \in \arg\max_i \{M_{i,\tau} - \mathbb{E}_\pi Q_i\}. \tag{17}$$

*At a unique active endpoint, $\nabla_\theta \mathcal{R}_\tau^{\text{twin}} = -\nabla_\theta \mathbb{E}_{\pi_\theta} Q_{i^\star}$. At a tie, either active endpoint gives a generalized gradient.*

Although $M_{i,0}$ can be difficult to maximize globally, $M_{i,\tau}$ is a log-expectation and can be estimated from counterfactual samples. The actor consequently receives the critic with the larger estimated soft regret. This segment admits an exact canonical reduction, so the Jensen relaxation is unnecessary. Sampling formulas and selector error bounds appear in Appendix B.6; Algorithm 2 gives the update. Later in Section 6, the numerical results show consistent gain.

## 5 A regret-derived actor signal

For richer ambiguity classes, we optimize the relaxed target (10) by policy gradient. The goal is to express its derivative in the form already used by actor-critic algorithms: an ordinary actor update driven by a critic signal. The result is the nominal critic plus a policy-dependent bonus, obtained by averaging the perturbations associated with challenging alternatives.

### 5.1 From the objective to the actor update

For a candidate action $z$, let $h_z^\pi$ be a perturbation attaining its largest advantage error:

$$h_z^\pi \in \arg\max_{\|h\|_{\mathcal{H}} \le 1} \{h(z) - \mathbb{E}_{b\sim\pi} h(b)\}. \tag{18}$$

We call it a *witness* because it realizes the dual distance $D_{\mathcal{H}}(z;\pi)$: it is the plausible perturbation certifying why $z$ challenges the current policy. To differentiate that distance, the envelope theorem holds the maximizing perturbation fixed and differentiates $-\mathbb{E}_\pi h_z^\pi$. Differentiating the outer soft maximum then weights these derivatives by the soft comparator $\bar{q}_\pi$.

**Proposition 4** (Policy gradient and ambiguity bonus)**.** *Let $\pi_\theta$ have differentiable density with score $\psi_\theta(b) = \nabla_\theta \log \pi_\theta(b)$. At differentiability points, with differentiation allowed through the expectations,*

$$\nabla_\theta \overline{\mathcal{R}}_\tau(\pi_\theta) = -\mathbb{E}_{b\sim\pi_\theta}\left[\psi_\theta(b)\widetilde{Q}_\pi(b)\right], \qquad \widetilde{Q}_\pi(b) = \widehat{Q}(b) + \delta \mathbb{E}_{z\sim\bar{q}_\pi} h_z^\pi(b). \tag{19}$$

*The same gradient can be computed by differentiating through reparameterized actor actions, with comparator and witness coefficients fixed at the current policy; see* (28)*.*

Minimizing relaxed regret is thus actor ascent on $\widetilde{Q}_\pi$. Two bonuses have distinct jobs. The candidate bonus $\delta D_{\mathcal{H}}(z;\pi)$ identifies threatening alternatives in (11); the actor bonus $\delta \mathbb{E}_{\bar{q}_\pi} h_z^\pi(b)$ moves the actor to reduce their advantage. Adding the candidate distance directly to the actor's reward would give a different gradient.

### 5.2 Uncertainty geometry examples

The function norm specifies what an error can look like; duality turns that choice into a distance and a witness. The following examples give concrete actor signals. Their estimators and derivations appear in Appendix D.

**Example 2** (Lipschitz perturbations and Wasserstein distance)**.** For a metric $c$, constrain $h$ to be 1-Lipschitz. Kantorovich–Rubinstein duality gives $d_{\mathcal{H}} = W_1$, and the singleton comparison has

$$D_{\mathcal{H}}(z;\pi) = \mathbb{E}_{B\sim\pi} c(z,B), \qquad h_z^\pi(b) = -c(z,b).$$

The candidate is challenging when far from current behavior; its witness rewards moving the actor toward that candidate.

**Example 3** (RKHS perturbations and MMD)**.** Let $\mathcal{H}$ be a reproducing kernel Hilbert space with kernel $k$ and feature map $\varphi$. Its unit-ball dual is maximum mean discrepancy (MMD). With $\mu_\pi = \mathbb{E}_\pi \varphi(B)$,

$$D_{\mathcal{H}}(z;\pi) = \|\varphi(z) - \mu_\pi\|_{\mathcal{H}}, \qquad h_z^\pi(b) = \frac{k(z,b) - \mathbb{E}_\pi k(B,b)}{D_{\mathcal{H}}(z;\pi)}.$$

The witness is the normalized difference between candidate and policy features; choose zero when the denominator is zero.

**Example 4** (Feature-ellipsoidal perturbations)**.** Let $x(b) \in \mathbb{R}^m$ be a critic feature vector and constrain $h_u(b) = u^\top x(b)$ by $u^\top H u \le 1$, for $H \succ 0$. Writing $v_z = x(z) - \mathbb{E}_\pi x(B)$ gives

$$D_{\mathcal{H}}(z;\pi) = \|v_z\|_{H^{-1}}, \quad u_z^\star = \frac{H^{-1} v_z}{\|v_z\|_{H^{-1}}}, \quad \widetilde{Q}_\pi(b) = \widehat{Q}(b) + c_\pi^\top x(b), \quad c_\pi = \delta \mathbb{E}_{\bar{q}_\pi} u_Z^\star. \tag{20}$$

Set $u_z^\star = 0$ for $v_z = 0$. Taking $H$ from regularized replay-feature second moments makes well-observed directions less uncertain. This parallels inverse-Gram confidence geometry in linear bandits (Abbasi-Yadkori et al., 2011), but the relevant direction is the *advantage feature* $x(z) - \mathbb{E}_\pi x$, rather than $x(z)$ alone.

With a fixed metric or kernel, the first two constructions are data-agnostic. Our initial exploration favored the data-dependent ellipsoid, which uses replay coverage to distinguish uncertain directions. The experiments therefore focus on this geometry alongside twin-critic selection. The ellipsoidal actor receives a direction $c_\pi$ averaged over challenging candidates; as the policy changes, both its feature mean and this direction change. Appendix D describes feature extraction, matrix solves, and sampling approximations.

**Sampling.** In the previous discussions, three distributions play separate roles: $\pi_\theta$ is the actor, $\nu$ defines the soft comparison, and $p$ proposes counterfactual actions. To ease the burden, in our implementations, we take $\nu$ as the uniform and $p$ as the actor itself to prevent using another neural network. Self-normalized importance sampling (SNIS) from $p$ estimates the expectation over $\bar{q}_\pi$ in (19); Appendix B gives the detailed derivation of the estimator and its finite-sample error bounds.

## 6 Empirical evaluation

We evaluate the regret-aware updates across three continuous-control suites and four actor–critic backbones.

**Implementation over the backbones.** For the twin-critic selector specified in Section 4, we abbreviate it as DR3O in this section. As noted earlier, the only modification compared to different backbones is to indicate which $Q$ to give to the actor. For the feature-ellipsoidal ambiguity geometry in Example 4, we derive an adaptive ambiguity set size $\delta$ to the mean-squared error of the critic, and we call it Adaptive Ellip in this section. As an ablation, we also set a constant $\delta$ control group called Ellip. For Adaptive Ellip and Ellip, we only train one single critic neural network, and keeps everything else the same as the backbone. We retain the backbone's entropy term if the backbone itself uses it. The extra computation is the critic's evaluation of the counterfactual actions (which is a batched forward pass) and bonus construction before the ordinary actor backward pass. Counterfactual actions are never executed in the environment. Algorithms 1 and 2 specify both interfaces while preserving the backbone's learning machinery. The additional computation cost is mild, mainly the counterfactual action sampling and one batch forward pass alongside the critic neural network.

### 6.1 Experimental design

**Coverage and controls.** The study contains 68 task–backbone settings: nine MuJoCo-v5 tasks with SAC, TD3, CrossQ, and TD7 (Bhatt et al., 2024; Fujimoto et al., 2023); ten MetaWorld MT1 tasks with SAC and TD3; and six DeepMind Control Suite (DMC) tasks with SAC and TD3. Each setting compares the matched baseline with the twin-critic selector (DR3O), an adaptive ellipsoidal bonus (Adaptive Ellip), and its fixed-scale counterpart (Ellip). Architectures, replay configurations, critic losses, optimizers, update schedules, and interaction budgets are matched. MuJoCo runs to one million steps; MetaWorld and DMC run to 500,000 steps.

**Metrics and inference.** We report final-checkpoint mean (SD) over training seeds 100–119; each seed endpoint averages 100 deterministic evaluation episodes. Improvements are paired with the baseline by training seed and summarized by two-sided 95% Student-$t$ confidence intervals over the twenty paired differences. Hyperparameters are selected by mean final return on the disjoint tuning seeds 200–203. MuJoCo and DMC use episodic return; MetaWorld uses success rate, which directly measures completion of its sparse manipulation objectives. MuJoCo percentage aggregates exclude negative-return Reacher because ratios are misleading. Complete task-level endpoints, adjusted seed-level win counts, configurations, and evaluation protocols appear in Appendix F.

### 6.2 Results

**Learning dynamics.** Figure 1 shows improvements on HalfCheetah–TD3 and Swimmer–SAC. DR3O, Ellip, and Adaptive Ellip raise final return by 6.82%, 7.82%, and 10.54% on the former, and 9.03%, 43.28%, and 48.79% on the latter. The actor correction can improve an otherwise similar learning profile or substantially change the attained solution.

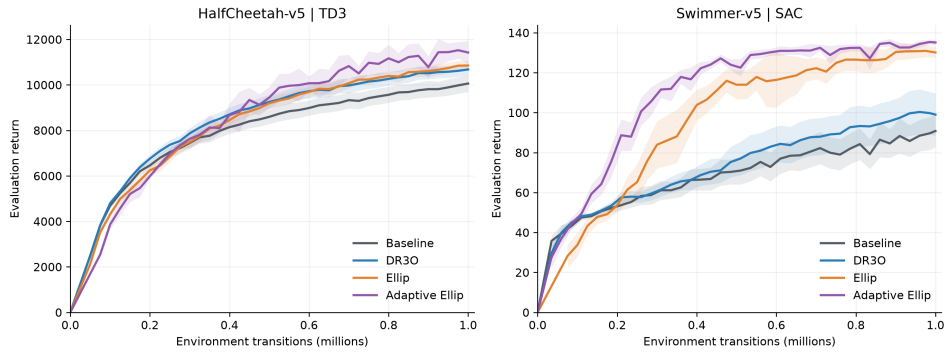



Figure 1: Representative learning curves on HalfCheetah-v5 with TD3 and Swimmer-v5 with SAC. Curves show evaluation return for the matched baseline, DR3O, Ellip, and Adaptive Ellip; shading shows one SD across training seeds.

Table 1: Endpoint gain over the matched baseline, reported as mean (SD) across task–backbone pairs. Bold marks the largest mean gain within each benchmark.

| Benchmark (gain metric) | DR3O | Ellip | Adaptive Ellip |
|---|---|---|---|
| MuJoCo return (%) | +3.79 (4.35) | +5.83 (7.93) | **+6.20 (8.42)** |
| MetaWorld success (pp) | +6.15 (7.25) | +4.07 (5.02) | **+7.01 (8.25)** |
| DMC return (points) | +26.92 (32.10) | +18.60 (22.82) | **+34.80 (28.88)** |

MuJoCo percentage aggregation excludes Reacher because its negative-return scale makes percentage normalization inappropriate; absolute and paired results remain in Appendix F.

**Aggregate performance.** All three variants improve the average matched baseline in every suite (Table 1). Adaptive Ellip is strongest throughout: $+6.20\%$ on MuJoCo, $+7.01$ success points on MetaWorld, and $+34.80$ return points on DMC. The sizeable SDs reveal task heterogeneity: near-solved tasks leave little headroom, whereas other tasks admit substantial gains. Task-resampled interquartile means (IQM) and cross-task consistency analyses appear in Appendix F.3.

**Transfer and task structure.** Adaptive Ellip improves 30 of 32 scale-comparable MuJoCo pairs, including every CrossQ pair and every scale-comparable TD7 pair. Gains extend to contact-rich MetaWorld manipulation and dynamic DMC tasks, while near-saturated tasks change little. Appendix F gives task-level returns and success rates, including cases where shaped return and task completion differ.

**Seed-paired evidence.** Adaptive Ellip has a wholly positive 95% paired interval in 27/36 MuJoCo, 13/20 MetaWorld, and 11/12 DMC settings. DR3O and fixed Ellip also yield broad gains. Improvement is therefore reflected in both aggregate magnitude and same-seed comparisons.

### 6.3 Choosing the radius: a scale-adaptation ablation

The radius $\delta$ controls the response to critic uncertainty. Fixed and Adaptive Ellip use the same task-specific base radius and ellipsoid temperature selected on tuning seeds. Fixed Ellip applies the scheduled base radius directly; Adaptive Ellip multiplies it by a clipped controller computed from running normalized Bellman MSE, so larger critic error permits a larger perturbation. Appendix F.2 specifies the normalization, clipping, and ramp schedule.

Adaptation raises the average gain from $5.83\%$ to $6.20\%$ on MuJoCo, from $4.07$ to $7.01$ success points on MetaWorld, and from $18.60$ to $34.80$ return points on DMC. The fixed-scale correction is already useful; adapting its strength improves the cross-suite average further.

### 6.4 Computational cost

Measured baseline wall time ranges from 74.7 minutes for DMC–TD3 to 483.5 minutes for MuJoCo–CrossQ. Across suite–backbone averages, DR3O adds 6.9–17.0% wall time, fixed Ellip ranges from $-3.2$ to $+1.0\%$, and Adaptive Ellip ranges from $-1.1$ to $+2.6\%$, with no additional environment interactions. Wall time includes setup, warmup, evaluation, concurrent GPU scheduling, and system variation, so small negative ratios indicate timing variation rather than negative computational cost. Complete wall-clock and per-update measurements appear in Appendix F.

## 7 Conclusion

DR3O turns uncertainty in learned feedback into a policy-relative regret objective. Relaxed regret enables sampling-based actor updates, and twin critics admit an exact endpoint selector. Improvements across three suites and four backbones support regret optimization as an effective way to use imperfect critics, without additional environment interactions.

## REPRODUCIBILITY STATEMENT

The ambiguity classes and actor objectives are specified in Sections 2–5. Appendix A gives the proofs, Appendix B defines the sampling estimators and analyzes their errors, and Appendix C provides pseudocode, implementation conventions, and the anonymous code repository. Appendix F documents the complete three-suite, four-backbone evaluation matrix, software and optimization configurations, the disjoint tuning and evaluation seed protocol, robust cross-task aggregates, all endpoint results, paired 95% confidence intervals, seed-level win counts, hardware, and measured training times.

## AI USE STATEMENT

OpenAI Codex was used to assist with expanding and organizing the manuscript, formulating and checking mathematical claims and proofs, suggesting evaluation diagnostics, identifying related literature, editing the prose, and preparing and validating the LaTeX document.

## REFERENCES

Yasin Abbasi-Yadkori, Dávid Pál, and Csaba Szepesvári. Improved algorithms for linear stochastic bandits. In *Advances in Neural Information Processing Systems*, volume 24, 2011.

Sergios Agapiou, Omiros Papaspiliopoulos, Daniel Sanz-Alonso, and Andrew M. Stuart. Importance sampling: Intrinsic dimension and computational cost. *Statistical Science*, 32(3):405–431, 2017. doi: 10.1214/17-STS611.

Alekh Agarwal and Tong Zhang. Minimax regret optimization for robust machine learning under distribution shift. In *Proceedings of the Thirty-Fifth Conference on Learning Theory*, volume 178, pp. 2704–2729. PMLR, 2022.

Feras Al Taha, Shuhao Yan, and Eilyan Bitar. A distributionally robust approach to regret optimal control using the wasserstein distance. *arXiv preprint arXiv:2304.06783*, 2023.

Igor Averbakh and Vasilij Lebedev. On the complexity of minmax regret linear programming. *European Journal of Operational Research*, 160(1):227–231, 2005. doi: 10.1016/j.ejor.2003.07.007.

Andrew G. Barto, Richard S. Sutton, and Charles W. Anderson. Neuronlike adaptive elements that can solve difficult learning control problems. *IEEE Transactions on Systems, Man, and Cybernetics*, SMC-13(5):834–846, 1983. doi: 10.1109/TSMC.1983.6313077.

Aditya Bhatt, Daniel Palenicek, Boris Belousov, Max Argus, Artemij Amiranashvili, Thomas Brox, and Jan Peters. CrossQ: Batch normalization in deep reinforcement learning for greater sample efficiency and simplicity. In *International Conference on Learning Representations*, 2024.

Eilyan Bitar. Distributionally robust regret minimization. *arXiv preprint arXiv:2412.15406*, 2024.

Lukas-Benedikt Fiechtner and Jose Blanchet. Wasserstein distributionally robust regret optimization. *arXiv preprint arXiv:2504.10796*, 2025.

Scott Fujimoto, Herke van Hoof, and David Meger. Addressing function approximation error in actor-critic methods. In *International Conference on Machine Learning*, pp. 1587–1596, 2018.

Scott Fujimoto, Wei-Di Chang, Edward J. Smith, Shixiang Shane Gu, Doina Precup, and David Meger. For SALE: State-action representation learning for deep reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 36, 2023.

Kamyar Ghasemipour, Shixiang Shane Gu, and Ofir Nachum. Why so pessimistic? estimating uncertainties for offline RL through ensembles, and why their independence matters. In *Advances in Neural Information Processing Systems*, volume 35, 2022. URL https://papers.nips.cc/paper_files/paper/2022/hash/7423902b5534e2b267438c85444a54b1-Abstract-Conference.html.

Arthur Gretton, Karsten M. Borgwardt, Malte J. Rasch, Bernhard Schölkopf, and Alexander Smola. A kernel two-sample test. *Journal of Machine Learning Research*, 13:723–773, 2012. URL https://jmlr.org/papers/v13/gretton12a.html.

Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International Conference on Machine Learning*, pp. 1861–1870, 2018.

Tuomas Haarnoja, Aurick Zhou, Kristian Hartikainen, George Tucker, Sehoon Ha, Jie Tan, Vikash Kumar, Henry Zhu, Abhishek Gupta, Pieter Abbeel, and Sergey Levine. Soft actor-critic algorithms and applications. *arXiv preprint arXiv:1812.05905*, 2019.

Tim Hesterberg. Weighted average importance sampling and defensive mixture distributions. *Technometrics*, 37(2):185–194, 1995.

Garud N. Iyengar. Robust dynamic programming. *Mathematics of Operations Research*, 30(2): 257–280, 2005.

Ying Jin, Zhuoran Yang, and Zhaoran Wang. Is pessimism provably efficient for offline rl? In *International Conference on Machine Learning*, volume 139, pp. 5084–5096. PMLR, 2021.

Vijay R. Konda and John N. Tsitsiklis. Actor-critic algorithms. In *Advances in Neural Information Processing Systems*, volume 12, pp. 1008–1014, 2000.

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

Arsenii Kuznetsov, Pavel Shvechikov, Alexander Grishin, and Dmitry Vetrov. Controlling overestimation bias with truncated mixture of continuous distributional quantile critics. In *International Conference on Machine Learning*, volume 119, pp. 5556–5566. PMLR, 2020.

Miao Lu, Han Zhong, Tong Zhang, and Jose Blanchet. Distributionally robust reinforcement learning with interactive data collection: Fundamental hardness and near-optimal algorithms. In *Advances in Neural Information Processing Systems*, volume 37, 2024.

Arnab Nilim and Laurent El Ghaoui. Robust control of markov decision processes with uncertain transition matrices. *Operations Research*, 53(5):780–798, 2005.

Art B. Owen. *Monte Carlo Theory, Methods and Examples*. Self-published, 2013.

Lerrel Pinto, James Davidson, Rahul Sukthankar, and Abhinav Gupta. Robust adversarial reinforcement learning. In *International Conference on Machine Learning*, volume 70, pp. 2817–2826. PMLR, 2017.

Kevin Regan and Craig Boutilier. Robust policy computation in reward-uncertain mdps using nondominated policies. In *Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence*, pp. 1127–1133, 2010.

Marc Rigter, Bruno Lacerda, and Nick Hawes. Minimax regret optimisation for robust planning in uncertain markov decision processes. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pp. 11930–11938, 2021.

Hado van Hasselt. Double q-learning. In *Advances in Neural Information Processing Systems*, volume 23, 2010.

Sharan Vaswani, Amirreza Kazemi, Reza Babanezhad Harikandeh, and Nicolas Le Roux. Decision-aware actor-critic with function approximation and theoretical guarantees. In *Advances in Neural Information Processing Systems*, volume 36, 2023.

Shengbo Wang, Nian Si, Jose Blanchet, and Zhengyuan Zhou. On the foundation of distributionally robust reinforcement learning. *arXiv preprint arXiv:2311.09018*, 2023.

Yikai Wang, Shang Liu, and Jose Blanchet. Wasserstein distributionally robust regret optimization for reinforcement learning from human feedback. *arXiv preprint arXiv:2605.00155*, 2026.

Wolfram Wiesemann, Daniel Kuhn, and Berç Rustem. Robust markov decision processes. *Mathematics of Operations Research*, 38(1):153–183, 2013.

Huan Xu and Shie Mannor. Distributionally robust markov decision processes. *Mathematics of Operations Research*, 37(2):288–300, 2012.

Lily Xu, Andrew Perrault, Fei Fang, Haipeng Chen, and Milind Tambe. Robust reinforcement learning under minimax regret for green security. In *Proceedings of the Thirty-Seventh Conference on Uncertainty in Artificial Intelligence*, volume 161, pp. 257–267. PMLR, 2021.

# A Decision model and proofs

## A.1 Assumptions, measures, and existence

The principal setting is a compact metric action space $\mathcal{A}$ with Borel probability measures $\mathcal{P}(\mathcal{A})$. Weak convergence $q_n \Rightarrow q$ means $\int f \, \mathrm{d}q_n \to \int f \, \mathrm{d}q$ for every continuous bounded $f$. Under this topology, $\mathcal{P}(\mathcal{A})$ is compact and convex. Its extreme points are exactly the point masses: a non-Dirac probability measure can be split into its normalized restrictions to a measurable set with mass strictly between zero and one and its complement. Conversely, a nontrivial mixture cannot equal a point mass unless both components are that point mass.

Let $\widehat{Q} \in C(\mathcal{A})$ and let $\mathcal{B} \subset C(\mathcal{A})$ be a symmetric perturbation unit ball. We assume its image, after any constant normalization, has compact closure in the uniform norm. Uniform boundedness and equicontinuity imply this by Arzela–Ascoli. Replacing the ball by this closure leaves every supremum unchanged and guarantees witness attainment. For a Lipschitz seminorm, one can require $h(a_0) = 0$ at a fixed $a_0$; this removes only constants, which have zero pairing with $q - \pi$. Continuous bounded RKHS kernels on compact $\mathcal{A}$ also give uniformly bounded, equicontinuous unit balls as functions on $\mathcal{A}$.

The dual distance is convex in either measure and satisfies the triangle inequality because it is the support function of a symmetric function ball. It is a genuine metric when this class separates probability measures. All arguments only require the induced seminorm, so finite-dimensional feature embeddings and non-characteristic kernels are included. Convexity here follows from the dual representation, rather than from the word "distance."

With these assumptions, $D_{\mathcal{H}}(z; \pi)$ is continuous in $z$ and the hard maximum is attained. The regret objectives are also convex and continuous as functions of the action measure $\pi$, so an outer minimizer over $\mathcal{P}(\mathcal{A})$ exists. Neural parameterizations $\pi_\theta$ inherit the usual nonconvex optimization problem. On noncompact spaces the algebraic identities continue to hold whenever the displayed integrals and suprema are finite; compactness, attainment, and uniform hard-to-soft rates then require separate moment and tightness conditions.

## A.2 The distributional reduction

To make the distributional interpretation explicit, let $\mathfrak{U}$ be a set of probability laws $\Lambda$ over random critics $\mathsf{Q}$. Under mean performance, a comparator chooses its action using $\mathbb{E}_\Lambda \mathsf{Q}$, rather than observing a realization of $\mathsf{Q}$ before acting. Only the attainable mean functions then matter.

**Proposition 5** (Reduction to mean critics)**.** *Let $\mathcal{Q} = \{\mathbb{E}_\Lambda \mathsf{Q} : \Lambda \in \mathfrak{U}\}$, with integrable evaluations and valid interchange of expectations. For every $\pi$,*

$$\sup_{\Lambda \in \mathfrak{U}} \sup_{q \in \mathcal{P}(\mathcal{A})} \mathbb{E}_\Lambda[\mathbb{E}_q \mathsf{Q} - \mathbb{E}_\pi \mathsf{Q}] = \sup_{Q \in \mathcal{Q}} \{\sup_a Q(a) - \mathbb{E}_\pi Q\}. \tag{21}$$

*The corresponding worst-case expected value is* $\inf_{\Lambda \in \mathfrak{U}} \mathbb{E}_\Lambda \mathbb{E}_\pi \mathsf{Q} = \inf_{Q \in \mathcal{Q}} \mathbb{E}_\pi Q$.

*Proof of Proposition 5.* Let $\mathfrak{U}$ be an ambiguity set of probability laws $\Lambda$ over random bounded critic functions $\mathsf{Q}$. Its set of mean functions is

$$\mathcal{Q} = \{\bar{Q}_\Lambda : \bar{Q}_\Lambda(a) = \mathbb{E}_\Lambda \mathsf{Q}(a), \ \Lambda \in \mathfrak{U}\}.$$

When a comparator chooses a policy using the distribution's expected values, distributionally robust regret is

$$\begin{aligned} \sup_{\Lambda \in \mathfrak{U}} \sup_q \mathbb{E}_\Lambda[\mathbb{E}_q \mathsf{Q} - \mathbb{E}_\pi \mathsf{Q}] &= \sup_{\Lambda \in \mathfrak{U}} \sup_q \left\langle q - \pi, \bar{Q}_\Lambda \right\rangle \\ &= \sup_{Q \in \mathcal{Q}} \{\sup_a Q(a) - \mathbb{E}_\pi Q\}. \end{aligned} \tag{22}$$

Interchanging expectations also gives $\mathbb{E}_\Lambda \mathbb{E}_\pi \mathsf{Q} = \mathbb{E}_\pi \bar{Q}_\Lambda$, proving the value identity. □

This is the mean-performance criterion used in the paper. Its comparator is optimal for the mean critic of each plausible law. A comparator that observes each realized random critic before acting would instead place the action supremum inside $\mathbb{E}_\Lambda$, yielding a different information structure. The norm ball in (3) specifies the possible means relevant to the present decision problem.

### A.3 Exact solution of the hard inner problem

*Proof of Proposition 1.* For any bounded critic, $\sup_q \mathbb{E}_q Q = \sup_z Q(z)$. The two adversarial choices form a product set, so their suprema commute without a minimax theorem:

$$\sup_h \sup_q \left\langle q - \pi, \widehat{Q} + \delta h \right\rangle = \sup_q \{ \left\langle q - \pi, \widehat{Q} \right\rangle + \delta \sup_h \langle q - \pi, h \rangle \}.$$

Alternatively, eliminate the comparator first and exchange the two remaining suprema:

$$\sup_h \sup_z [\widehat{A}_\pi(z) + \delta(h(z) - \mathbb{E}_\pi h)] = \sup_z [\widehat{A}_\pi(z) + \delta D_{\mathcal{H}}(z; \pi)].$$

This proves the pointwise identity without any relaxation. At a maximizing $z^\star$, any maximizing perturbation $h^\pi_{z^\star}$ gives a jointly optimal pair $(\delta_{z^\star}, h^\pi_{z^\star})$. □

The extreme-point interpretation describes the same reasoning geometrically. The perturbation ball's uniform compactness makes $q \mapsto d_{\mathcal{H}}(q, \pi)$ weakly continuous: a finite uniform-norm net reduces uniform convergence of its integrals to weak convergence for finitely many continuous functions. The convex continuous objective on compact $\mathcal{P}(\mathcal{A})$ therefore has an extreme-point maximizer. The direct supremum calculation above also proves equality when maxima are not attained. It solves the adversary up to an action maximization; the outer actor still requires optimization.

### A.4 Gibbs duality and canonical soft regret

For bounded $f$ and a probability reference $\nu$, put $Z_f = \mathbb{E}_\nu e^{f/\tau}$ and $q_f(\,\mathrm{d}z) = Z_f^{-1} e^{f(z)/\tau} \nu(\,\mathrm{d}z)$. For any $q \ll \nu$ with finite KL,

$$\mathbb{E}_q f - \tau \,\mathrm{KL}(q\|\nu) = \tau \log Z_f - \tau \,\mathrm{KL}(q\|q_f) \le \tau \log Z_f. \tag{23}$$

Equality holds uniquely at $q = q_f$ up to null sets. Apply this identity to $f = \widehat{Q} + \delta h$ and subtract $\mathbb{E}_\pi f$. Exchanging $\sup_q$ and $\sup_h$ gives (7). Any jointly optimal pair must maximize each coordinate conditionally, which gives (8). Conditional optimality alone need not imply global optimality.

There is a second useful view of the difficulty. For a fixed actor, the map $h \mapsto \mathrm{S}_{\tau,\nu}(\widehat{Q} + \delta h) - \mathbb{E}_\pi(\widehat{Q} + \delta h)$ is convex. In a finite-dimensional parametrization $h_u = u^\top x$, its Hessian is

$$\nabla_u^2 \,\mathrm{S}_{\tau,\nu}(\widehat{Q} + \delta h_u) = \frac{\delta^2}{\tau} \,\mathrm{Cov}_{q_u}(x(Z)) \succeq 0.$$

Thus the inner task maximizes a convex function over a convex ambiguity body. A one-dimensional segment has only two endpoints to check, but a feature ellipsoid has a continuum of boundary points. This explains both the special tractability of twin critics and the utility of the pointwise relaxation for richer geometries.

### A.5 Ordering and the hard-to-soft approximation

The Jensen step (9), followed by Gibbs duality, proves $\mathcal{R}_\tau^{\mathrm{can}} \le \overline{\mathcal{R}}_\tau$. Since $\nu$ is a probability measure, $\mathrm{S}_{\tau,\nu} f \le \sup f$, giving $\overline{\mathcal{R}}_\tau \le \mathcal{R}_0$. A normalized soft regret can be negative because its comparator pays a KL cost; all ordering and gradient statements apply to this regularized criterion.

To further bound the gap between the hard-to-soft approximation, we have: For finite $\mathcal{A}$ with $\nu_{\min} := \min_a \nu(a) > 0$,

$$0 \le \mathcal{R}_0(\pi) - \mathcal{R}_\tau^{\mathrm{can}}(\pi) \le \tau \log(1/\nu_{\min}). \tag{24}$$

For finite $\mathcal{A}$, let $z^\star$ maximize $f$. Then

$$\mathbb{E}_\nu e^{f/\tau} \ge \nu(z^\star) e^{\sup f/\tau}, \qquad 0 \le \sup f - \mathrm{S}_{\tau,\nu} f \le \tau \log(1/\nu_{\min}).$$

The bound is uniform in $h$, so it survives the supremum defining the canonical regret, proving (24). In particular, uniform $\nu$ gives $\tau \log |\mathcal{A}|$.

For continuous $\mathcal{A}$, define the near-optimal set and its reference mass by

$$A_\epsilon(f) = \{z : f(z) \geq \sup f - \epsilon\}, \qquad \alpha_\epsilon(f) = \nu(A_\epsilon(f)).$$

Integrating only over this set gives the replacement bound

$$0 \leq \sup f - \mathrm{S}_{\tau,\nu}\, f \leq \inf_{\epsilon>0:\alpha_\epsilon(f)>0} \{\epsilon + \tau \log[1/\alpha_\epsilon(f)]\}. \tag{25}$$

If $\alpha_\epsilon = \inf_{h\in\mathcal{B}} \alpha_\epsilon(Q_h) > 0$, the same bound with $\alpha_\epsilon$ controls $\mathcal{R}_0 - \mathcal{R}_\tau^{\mathrm{can}}$ uniformly in $\pi$. A sufficient condition is a common Lipschitz constant $L_Q$ and a reference with a uniform lower bound on the mass of action balls: then $\alpha_\epsilon \geq \inf_a \nu(B(a, \epsilon/L_Q))$. On a regular bounded $d$-dimensional domain with a reference density bounded below, this yields a bound of order $\tau d \log(1/\tau)$ for fixed geometry, by choosing $\epsilon$ proportional to $\tau$.

For convergence without a uniform rate, suppose $\nu$ has full support and each $Q_h$ is continuous. Given $\eta > 0$, select a perturbation within $\eta$ of hard optimality. The soft maximum of this fixed critic converges to its maximum as $\tau \downarrow 0$ by (25). Hence $\liminf_{\tau\downarrow 0} \mathcal{R}_\tau^{\mathrm{can}} \geq \mathcal{R}_0 - \eta$. Letting $\eta \downarrow 0$ and using the ordering proves that both soft criteria converge to hard regret.

### A.6 The Jensen gap and symmetrization

The Jensen gap between the $\overline{\mathcal{R}}_\tau(\pi)$ and the $\mathcal{R}_\tau^{\mathrm{can}}(\pi)$ can also be bounded as: For finite or compact continuous actions, define $J(q,\pi) = \mathbb{E}_q D_{\mathcal{H}}(Z;\pi) - d_{\mathcal{H}}(q,\pi)$. Then

$$0 \leq \overline{\mathcal{R}}_\tau(\pi) - \mathcal{R}_\tau^{\mathrm{can}}(\pi) \leq \delta J(\bar{q}_\pi, \pi) \leq \delta \mathbb{E}_{Z,Z' \overset{\mathrm{iid}}{\sim} \bar{q}_\pi} d_{\mathcal{H}}(\delta_Z, \delta_{Z'}). \tag{26}$$

Let $\bar{q} = \bar{q}_\pi$ solve the relaxed variational problem. Evaluating the canonical variational objective at this same feasible distribution gives

$$\mathcal{R}_\tau^{\mathrm{can}}(\pi) \geq \mathbb{E}_{\bar{q}} \widehat{A}_\pi + \delta d_{\mathcal{H}}(\bar{q}, \pi) - \tau\, \mathrm{KL}(\bar{q} \| \nu).$$

Subtracting this expression from the relaxed value proves the first upper bound in (26). To prove the last bound, work in the dual space of normalized perturbations. Let $X_z$ be the evaluation functional $X_z(h) = h(z)$ and let $m_q = \mathbb{E}_q X_Z$, $m_\pi = \mathbb{E}_\pi X_B$. Then

$$\begin{aligned} J(q,\pi) &= \mathbb{E}_q \|X_Z - m_\pi\|_{\mathcal{H}^*} - \|m_q - m_\pi\|_{\mathcal{H}^*} \\ &\leq \mathbb{E}_q \|X_Z - m_q\|_{\mathcal{H}^*} \\ &= \mathbb{E}_Z \|\mathbb{E}_{Z'}[X_Z - X_{Z'} \mid Z]\|_{\mathcal{H}^*} \\ &\leq \mathbb{E}_{Z,Z' \overset{\mathrm{iid}}{\sim} q} \|X_Z - X_{Z'}\|_{\mathcal{H}^*} = \mathbb{E}_{Z,Z'} d_{\mathcal{H}}(\delta_Z, \delta_{Z'}). \end{aligned} \tag{27}$$

The first inequality is the triangle inequality and the second is Jensen applied to the norm, conditional on $Z$. The same argument can be written directly with support functions if a Bochner embedding is unavailable. This proves Proposition 2.

If $d_{\mathcal{H}}(\delta_z, \delta_{z'}) \leq D_{\max}$, the relaxation gap is at most $\delta D_{\max}$. For a comparator supported in a ball of radius $r$ in this geometry, it is at most $2\delta r$. If $q$ is a point mass, its Jensen gap vanishes exactly. These statements capture concentration geometrically, so they apply equally to Wasserstein and feature-space uncertainty.

**From objective approximation to a policy comparison.** Suppose $\mathcal{R}_0(\pi) - \overline{\mathcal{R}}_\tau(\pi) \leq \epsilon_\tau$ uniformly over a chosen policy class, and $\widehat{\pi}$ is an $\epsilon_{\mathrm{opt}}$-minimizer of $\overline{\mathcal{R}}_\tau$ over that class. The one-sided ordering gives

$$\mathcal{R}_0(\widehat{\pi}) \leq \inf_\pi \mathcal{R}_0(\pi) + \epsilon_\tau + \epsilon_{\mathrm{opt}}.$$

For a canonical comparison instead, a uniform Jensen gap $\epsilon_J$ gives $\mathcal{R}_\tau^{\mathrm{can}}(\widehat{\pi}) \leq \inf_\pi \mathcal{R}_\tau^{\mathrm{can}}(\pi) + \epsilon_J + \epsilon_{\mathrm{opt}}$. These are statewise surrogate comparisons, with the optimization accuracy stated explicitly.

### A.7 Policy-gradient derivation

Fix $\theta_0$ and choose witnesses at $\pi_{\theta_0}$. Danskin's theorem gives

$$\nabla_\theta D_{\mathcal{H}}(z;\pi_\theta)\big|_{\theta_0} = -\nabla_\theta \mathbb{E}_{b\sim\pi_\theta} h_z^{\pi_{\theta_0}}(b)\big|_{\theta_0}.$$

Writing the relaxed objective as $-\mathbb{E}_{\pi_\theta}\widehat{Q} + \tau \log \mathbb{E}_\nu \exp\{[\widehat{Q} + \delta D_{\mathcal{H}}(\cdot;\pi_\theta)]/\tau\}$ yields

$$\begin{aligned}\nabla_\theta \overline{\mathcal{R}}_\tau &= -\nabla_\theta \mathbb{E}_{\pi_\theta}\widehat{Q} + \delta \mathbb{E}_{z\sim\bar{q}_{\pi_{\theta_0}}} \nabla_\theta D_{\mathcal{H}}(z;\pi_\theta) \\ &= -\nabla_\theta \mathbb{E}_{b\sim\pi_\theta}[\widehat{Q}(b) + \delta \mathbb{E}_{z\sim\bar{q}_{\pi_{\theta_0}}} h_z^{\pi_{\theta_0}}(b)], \qquad \text{at } \theta = \theta_0.\end{aligned}$$

The score identity gives (19). For a differentiable reparameterization $b = T_\theta(\epsilon)$, differentiation through the actor action gives

$$\nabla_\theta \overline{\mathcal{R}}_\tau(\pi_\theta) = -\mathbb{E}_\epsilon\left[(\nabla_\theta T_\theta(\epsilon))^\top \nabla_b \widetilde{Q}_\pi(b)\big|_{b=T_\theta(\epsilon)}\right], \tag{28}$$

with comparator and witness coefficients evaluated at the current policy and held fixed. A state-dependent constant can be subtracted from the actor signal without changing the score gradient. For pathwise differentiation, constants in $b$ also disappear.

The witness optimizer and Gibbs law are evaluated at $\theta_0$ and act as coefficients of this first derivative. Differentiating them again while implementing the shaped-critic expression would add terms that do not belong to (19). In contrast, the witness *evaluation* must remain differentiable through the actor action. At nondifferentiability points, compactness and the usual Danskin directional-derivative conditions give active-witness generalized gradients. In finite-dimensional feature norms the only additional case is a zero feature difference; any maximizing direction, including zero, is a valid support-function subgradient there.

### A.8 Rectangular ambiguity and narrow spikes

Let $\ell, u$ be bounded measurable functions with $\ell \le u$, and let $\mathcal{Q}_{\mathrm{rect}}$ contain all measurable functions between them. For any probability measure $\pi$, write $m_z = \pi(\{z\})$. The exact hard regret is

$$\mathcal{R}_0^{\mathrm{rect}}(\pi) = \sup_z \{u(z) - \mathbb{E}_\pi \ell - m_z[u(z) - \ell(z)]\}. \tag{29}$$

Indeed, after exchanging $\sup_Q$ and $\sup_z$, the signed comparison measure $\delta_z - \pi$ has nonnegative coefficient $1 - m_z$ at $z$ and is nonpositive elsewhere. Its maximizer sets $Q(z) = u(z)$ and $Q(b) = \ell(b)$ off $z$. Substitution gives (29). For a non-atomic actor, $m_z = 0$ for all $z$, proving (14). Since $\inf_Q \mathbb{E}_\pi Q = \mathbb{E}_\pi \ell$, minimizing this regret is the same as maximizing robust value over any class of non-atomic actors.

Suppose now that $\mathcal{A}$ is compact metric and $\ell, u$ are continuous. Fix $z$ and take continuous bumps $b_n(a) = \max\{0, 1 - nc(a, z)\}$, with $c$ the action metric. Then $Q_n = \ell + b_n(u - \ell)$ lies between the envelopes, is continuous, and satisfies $Q_n(z) = u(z)$. By dominated convergence,

$$\mathbb{E}_\pi Q_n \longrightarrow \mathbb{E}_\pi \ell + m_z[u(z) - \ell(z)].$$

Thus restricting to continuous critics leaves the same supremum (29), though a maximizing critic need not exist in that class. On a compact space each $Q_n$ is individually uniformly continuous; what fails is a *common* modulus of continuity for the class as $n$ grows. A uniform Lipschitz constraint, or an appropriate bounded RKHS norm, couples nearby errors and excludes fixed-height spikes of arbitrarily small width.

For policies with atoms, the mass correction in (29) records the cost of raising the actor's own value at $z$. On ordinary continuous action domains without isolated points, continuous $u$ nevertheless permits comparator points approaching a maximizer while avoiding the countably many atoms. In that setting the supremum again equals $\sup u - \mathbb{E}_\pi \ell$, even for atomic actors. The formula (29) covers both these domains and discrete or mixed action spaces.

### A.9 Twin-critic endpoint proof and invariances

When $\Delta Q \neq 0$, (16) defines a norm on a one-dimensional space. When $Q_1 = Q_2$, use the zero-dimensional space $\{0\}$ and the ambiguity set is a singleton. For $\xi \in [-1, 1]$, define

$$F_\tau(\xi) = \mathrm{S}_{\tau,\nu}(\widehat{Q} + \xi \Delta Q) - \mathbb{E}_\pi(\widehat{Q} + \xi \Delta Q).$$

It is convex, and therefore $F_\tau(\xi) \le \frac{1+\xi}{2} F_\tau(1) + \frac{1-\xi}{2} F_\tau(-1) \le \max\{F_\tau(1), F_\tau(-1)\}$. Both endpoints are feasible, proving the soft identity in Proposition 3. Replacing soft maximum by

supremum gives the same proof for the hard identity. Since $M_{i,\tau}$ is constant with respect to the actor, differentiating an active endpoint yields the stated update. This also extends immediately to the convex hull of any finite number of critics.

Endpoint regret is invariant to an action-independent shift of either critic: $\mathrm{S}(Q_i+c_i)-\mathbb{E}_\pi(Q_i+c_i) = \mathrm{S}(Q_i)-\mathbb{E}_\pi Q_i$. Selection therefore depends on relative decision quality rather than on arbitrary value offsets. By comparison, the actionwise minimum can change when only one critic is shifted.

For this class the pointwise relaxation is a different, also well-defined objective:

$$\overline{\mathcal{R}}_\tau^{\mathrm{twin}}(\pi) = \mathrm{S}_{\tau,\nu}[\max\{Q_1 - \mathbb{E}_\pi Q_1,\, Q_2 - \mathbb{E}_\pi Q_2\}].$$

Its witness switches by candidate, so its actor signal is an averaged mixture of the two critics, with the mixture coefficient given by the relaxed comparator's mass on the corresponding active region. The endpoint selector in the main text instead optimizes the canonical target, which uses one common critic.

# B Sampling guarantees

All results in this section condition on a fixed state, the current actor, and the current critic and geometry. The samples introduced for an actor step are independent conditional on those quantities. This separates Monte Carlo estimation error from the evolving learning problem.

## B.1 Estimating the relaxed-regret gradient

The actor $\pi_\theta$, reference $\nu$, and proposal $p$ have different roles. The actor is optimized; the fixed reference specifies which alternatives enter the soft comparison; the proposal is a computational choice with $\nu \ll p$. An analytic reference needs no policy network. For example, the defensive mixture $p = (1-\varepsilon)\pi_\theta + \varepsilon\nu$ combines local actor samples with broad reference coverage (Hesterberg, 1995). Draw $Z_{1:L} \overset{\mathrm{iid}}{\sim} p$ and form

$$w_\ell = \frac{\mathrm{d}\nu}{\mathrm{d}p}(Z_\ell) e^{[\widehat{Q}(Z_\ell)+\delta D_{\mathcal{H}}(Z_\ell;\pi)]/\tau}, \quad \bar{w}_\ell = \frac{w_\ell}{\sum_m w_m}, \quad \widehat{B}_\pi(b) = \delta \sum_{\ell=1}^{L} \bar{w}_\ell h^\pi_{Z_\ell}(b). \tag{30}$$

Independent actor samples estimate (19), or a pathwise backward pass uses $\widehat{Q} + \widehat{B}_\pi$. Weights and witness coefficients are detached; the actor action remains differentiable. With exact witnesses, this estimates the gradient of (10). When a witness needs a policy mean, a separate pilot sample estimates it before candidate sampling, with the additional error analyzed in Appendix B.5.

**Proposition 6** (Finite-sample SNIS error)**.** *Let $r = \mathrm{d}\bar{q}_\pi/\mathrm{d}p$, $\kappa = \mathbb{E}_p r^2 < \infty$, and let $F$ be a fixed vector-valued integrand with $\|F(z)\|_2 \le G$. For the SNIS estimator $\widehat{m}_L = \sum_\ell \bar{w}_\ell F(Z_\ell)$,*

$$\mathbb{E}\,\|\widehat{m}_L - \mathbb{E}_{\bar{q}_\pi} F\|_2^2 \le \frac{4G^2\kappa}{L}, \qquad \kappa = 1 + \chi^2(\bar{q}_\pi \| p). \tag{31}$$

*If $|\widehat{Q}| \le Q_{\max}$, $|h^\pi_z| \le H_{\max}$, and $\|\psi_\theta\|_2 \le S$, the score estimator using $J$ independent actor samples and exact witnesses satisfies*

$$\mathbb{E}\,\big\|\widehat{g} - \nabla_\theta \overline{\mathcal{R}}_\tau\big\|_2^2 \le \frac{S^2(Q_{\max}+\delta H_{\max})^2}{J} + \frac{4\delta^2 S^2 H_{\max}^2 \kappa}{L}. \tag{32}$$

The first term is actor Monte Carlo error; the second measures the cost of sampling the comparator. Smaller temperatures sharpen the comparison but can increase $\kappa$, exposing an approximation–sampling tradeoff. SNIS is generally biased at finite sample size and consistent as the number of candidates grows.

## B.2 A vector-valued SNIS bound

*Proof of* (31). Let $r = \mathrm{d}\bar{q}_\pi/\mathrm{d}p$, so $\mathbb{E}_p r = 1$ and $\mathbb{E}_p r^2 = \kappa$. Define

$$A_L = L^{-1}\sum_{\ell=1}^{L} r(Z_\ell)F(Z_\ell), \quad B_L = L^{-1}\sum_{\ell=1}^{L} r(Z_\ell), \quad m = \mathbb{E}_p[rF].$$

Then $\widehat{m}_L = A_L/B_L$, since the unknown normalizing constant cancels. If $B_L = 0$, define $\widehat{m}_L$ to be any vector of norm at most $G$. The identity $A_L = B_L\widehat{m}_L$ still holds, and $\|\widehat{m}_L\|_2 \le G$ in either case. Therefore

$$\widehat{m}_L - m = (A_L - m) + (1 - B_L)\widehat{m}_L,$$

and independence implies

$$\begin{aligned}\mathbb{E}\,\|\widehat{m}_L - m\|_2^2 &\le 2\mathbb{E}\,\|A_L - m\|_2^2 + 2G^2\mathbb{E}|B_L - 1|^2 \\ &\le \frac{2}{L}\mathbb{E}_p[r^2\,\|F\|_2^2] + \frac{2G^2}{L}(\kappa - 1) \le \frac{4G^2\kappa}{L}.\end{aligned}$$

This argument applies to any finite-dimensional Euclidean output and, with the same assumptions, to Hilbert-valued integrands. It is the bounded-integrand second-moment mechanism underlying classical SNIS error bounds (Agapiou et al., 2017). □

Markov's inequality gives an immediate high-probability consequence:

$$\Pr\left(\|\widehat{m}_L - m\|_2 > 2G\sqrt{\frac{\kappa}{L\eta}}\right) \le \eta.$$

The bound is conservative but makes the required sample scale explicit. The strong law also gives consistency when $\mathbb{E}_p|rF| < \infty$ and $\mathbb{E}_p r = 1$. SNIS is a ratio estimator, so finite-sample unbiasedness is not assumed.

### B.3 Error of the actor-gradient estimate

For exact witnesses and candidates $Z_{1:L}$, take independent actor samples $B_{1:J} \sim \pi_\theta$ and define

$$\widehat{g} = -\frac{1}{J}\sum_{j=1}^{J}\psi_\theta(B_j)\left[\widehat{Q}(B_j) + \delta\sum_{\ell=1}^{L}\bar{w}_\ell h^\pi_{Z_\ell}(B_j)\right]. \tag{33}$$

Conditional on the candidate set, its mean $g_L$ equals the exact nominal gradient plus SNIS of $F(z) = -\delta\mathbb{E}_{B\sim\pi_\theta}\psi_\theta(B)h^\pi_z(B)$. The norm of this integrand is at most $\delta S H_{\max}$. Moreover, each summand in (33) has norm at most $S(Q_{\max} + \delta H_{\max})$. Conditional variance decomposition gives

$$\begin{aligned}\mathbb{E}\,\|\widehat{g} - g\|_2^2 &= \mathbb{E}\,\|\widehat{g} - g_L\|_2^2 + \mathbb{E}\,\|g_L - g\|_2^2 \\ &\le \frac{S^2(Q_{\max} + \delta H_{\max})^2}{J} + \frac{4\delta^2 S^2 H_{\max}^2\kappa}{L},\end{aligned}$$

which proves (32). The score boundedness condition is a sufficient condition for this displayed rate. For pathwise gradients, the same proof uses bounds on the corresponding action-Jacobian integrands in place of the score bounds. Unbounded scores can instead be treated through appropriate weighted moment assumptions.

### B.4 Proposal coverage, temperature, and effective sample size

Write $f(z) = \widehat{Q}(z) + \delta D_{\mathcal{H}}(z;\pi)$, with oscillation $\mathrm{osc}(f) = \sup f - \inf f$. If $p \ge \varepsilon\nu$ as measures, then

$$r(z) = \frac{\mathrm{d}\nu}{\mathrm{d}p}(z)\frac{e^{f(z)/\tau}}{\mathbb{E}_\nu e^{f/\tau}} \le \varepsilon^{-1}e^{\mathrm{osc}(f)/\tau}, \qquad \kappa \le \varepsilon^{-1}e^{\mathrm{osc}(f)/\tau}.$$

A defensive mixture therefore prevents loss of reference support, while its usefulness still depends on overlap with the tilted comparator. The usual empirical effective sample size $\mathrm{ESS} = 1/\sum_\ell \bar{w}_\ell^2$ satisfies $\mathrm{ESS}/L \to 1/\kappa$ under the relevant laws of large numbers. It is a useful diagnostic for overly concentrated weights. Lowering $\tau$ decreases the hard-to-soft error but can sharply raise the number of candidates needed for a stable gradient.

### B.5 Pilot estimates of distances and witnesses

In feature-based implementations, a pilot sample $B_{1:K}^{\mathrm{pilot}} \sim \pi$ estimates the policy feature mean. Conditional on this sample, write $\widehat{D}(z)$ and $\widehat{h}_z(b)$ for the resulting functions, and let $\widehat{q}$ be the Gibbs distribution based on $\widehat{D}$. Candidate sampling then estimates $\widehat{q}$, rather than the ideal $\bar{q}$.

Suppose the pilot satisfies $\sup_z |\widehat{D}(z) - D(z)| \le e_D$. Its two log-partitions differ by at most $\delta e_D$, and consequently

$$e^{-2\delta e_D/\tau} \le \frac{\mathrm{d}\widehat{q}}{\mathrm{d}\bar{q}}(z) \le e^{2\delta e_D/\tau}. \tag{34}$$

For vector integrands satisfying $\sup_z \left\|\widehat{F}(z) - F(z)\right\|_2 \le e_F$ and $\sup_z \|F(z)\|_2 \le G$, this gives

$$\left\|\mathbb{E}_{\widehat{q}}\widehat{F} - \mathbb{E}_{\bar{q}}F\right\|_2 \le e_F + G\min\{2, e^{2\delta e_D/\tau} - 1\}. \tag{35}$$

Indeed, add and subtract $\mathbb{E}_{\widehat{q}}F$ and bound the remaining integral by $G\int |\,\mathrm{d}\widehat{q} - \mathrm{d}\bar{q}|$. Conditional SNIS error is controlled by (31) with $\widehat{q}$ and $\widehat{F}$; (35) then supplies the additional bias relative to the ideal objective. For small $e_D$, the contribution from the Gibbs weights is of order $\delta e_D/\tau$.

For the feature ellipsoid, put $\bar{x} = \mathbb{E}_\pi x$ and assume $\left\|\widehat{\bar{x}} - \bar{x}\right\|_{H^{-1}} \le e_x$. The reverse triangle inequality gives $e_D \le e_x$. If $\|x(z) - \bar{x}\|_{H^{-1}} \ge r_0 > 0$ on the candidates, normalization of the whitened witness direction gives

$$\left\| \frac{H^{-1/2}(x(z) - \widehat{\bar{x}})}{\left\|x(z) - \widehat{\bar{x}}\right\|_{H^{-1}}} - \frac{H^{-1/2}(x(z) - \bar{x})}{\|x(z) - \bar{x}\|_{H^{-1}}} \right\|_2 \le \frac{2e_x}{r_0},$$

provided the estimated direction is nonzero. Thus bounded whitened actor features turn the mean-estimation error into a witness error. Bounded Hilbert-space features give an expected squared pilot-mean error of order $K^{-1}$. Near zero difference, the support-function subgradient is set-valued; numerical implementations can use zero at equality or a stated smoothing rule. Keeping this case explicit avoids treating an unstable normalized direction as an exact, uniformly smooth witness.

### B.6 Estimating and selecting twin-critic regrets

With $Z_{1:L} \sim p$, write $u_\ell = (\,\mathrm{d}\nu/\,\mathrm{d}p)(Z_\ell)$ and estimate the two canonical endpoint regrets using

$$\widehat{M}_{i,\tau} = \tau \log \frac{\sum_\ell u_\ell e^{Q_i(Z_\ell)/\tau}}{\sum_\ell u_\ell}, \qquad \widehat{V}_i = K^{-1}\sum_{k=1}^{K} Q_i(B_k), \quad B_k \sim \pi. \tag{36}$$

Select the larger $\widehat{M}_{i,\tau} - \widehat{V}_i$ and use its critic in the actor loss. For a deterministic actor, evaluate $\widehat{V}_i = Q_i(a_\theta(s))$ exactly while retaining a broad candidate proposal. The same candidate set can be evaluated by both critics. Algorithm 2 implements this rule with log-sum-exp stabilization.

Let $\nu \ll p$ and $\kappa_\nu = \mathbb{E}_p(\,\mathrm{d}\nu/\,\mathrm{d}p)^2 < \infty$, with $\,\mathrm{d}\nu/\,\mathrm{d}p > 0$ $p$-almost surely so that the sampled normalizer is positive. Suppose each critic has range at most $B$, and estimate its soft maximum as in (36). Shifting each $Q_i$ by its lower bound makes $f_i = \exp((Q_i - \inf Q_i)/\tau)$ lie in $[1, e^{B/\tau}]$. Both the true expectation and its SNIS average are at least one, where logarithm is 1-Lipschitz. Therefore

$$\mathbb{E}|\widehat{M}_{i,\tau} - M_{i,\tau}|^2 \le \frac{4\tau^2 e^{2B/\tau}\kappa_\nu}{L}. \tag{37}$$

For $K$ independent policy samples, $\mathbb{E}|\widehat{V}_i - V_i|^2 \le B^2/(4K)$. Writing $r_i = M_{i,\tau} - V_i$ and $\widehat{r}_i = \widehat{M}_{i,\tau} - \widehat{V}_i$ yields

$$\mathbb{E}|\widehat{r}_i - r_i|^2 \le e_i^2 := \frac{8\tau^2 e^{2B/\tau}\kappa_\nu}{L} + \frac{B^2}{2K}.$$

The policy term vanishes for exact evaluation of a deterministic actor. If the true endpoint-regret margin is $m = |r_1 - r_2| > 0$, a wrong selection requires an error at least $m/2$ in one of the two regrets. Hence

$$\Pr(\widehat{i} \ne i^\star) \le \min\{1,\, 4(e_1^2 + e_2^2)/m^2\}. \tag{38}$$

The bound allows shared candidates between the critics; their estimation errors need not be independent. With separately sampled actor-loss actions, a wrong selection is the only additional source of bias in the selected expected gradient. If the two branch gradients have distance at most $G_\Delta$, its norm is at most $G_\Delta \Pr(\widehat{i} \neq i^\star)$. At a true tie, both branches are active. These statements distinguish estimating a soft maximum from estimating the general relaxed-gradient bonus.

# C Algorithms and implementation

**Anonymous code repository.** The implementation, experiment configurations, and reproduction scripts are available at `https://anonymous.4open.science/r/RL_DR3O_public-B519/`.

## C.1 General relaxed-regret actor update

Algorithm 1 replaces the actor objective inside a standard training loop. The critic remains a learned proxy updated by its original loss. The geometry and radius can be refreshed between actor steps, and are held fixed within each step. A mini-batch of replay states estimates the fixed state-distribution objective described in Section 5.

**Algorithm 1** DR3O: a sampling-based relaxed-regret actor update

**Require:** Backbone actor $\pi_\theta$, critic $\widehat{Q}_\phi$, replay and update schedule
**Require:** Perturbation geometry $\mathcal{H}_s$, radius $\delta_s$, soft temperature $\tau_s > 0$
**Require:** Fixed reference $\nu_s$; proposal $p_s$ with positive, evaluable density ratio $\mathrm{d}\nu_s/\,\mathrm{d}p_s$
**Require:** State batch size $N_s$, pilot size $K$, candidate size $L$, actor sample size $J$
**Require:** Actor optimizer and learning rate; backbone entropy coefficient $\alpha$ if used
**while** the backbone training loop is active **do**
  Collect transitions and train the critic using the backbone's original rules
  **if** an actor update is due **then**
    Sample $N_s$ replay states; freeze critic parameters and ambiguity geometry
    **for** each sampled state $s$ **do**
      Draw $K$ pilot actions from $\pi_\theta(\cdot \mid s)$ if the witness needs policy statistics
      Estimate those statistics; construct the map $z \mapsto (\widehat{D}_s(z), \widehat{h}_{s,z})$
      Draw $L$ independent candidates $Z_\ell \sim p_s$ and evaluate their critic values
      $\ell_\ell \leftarrow \log(\,\mathrm{d}\nu_s/\,\mathrm{d}p_s)(Z_\ell) + [\widehat{Q}_\phi(s, Z_\ell) + \delta_s \widehat{D}_s(Z_\ell)]/\tau_s$
      $\bar{w}_\ell \leftarrow \exp(\ell_\ell - \mathrm{LSE}(\ell_{1:L}))$
      Freeze $\bar{w}_\ell$ and witness coefficients at their current values
      Form $\widetilde{Q}_s(b) = \widehat{Q}_\phi(s, b) + \delta_s \sum_\ell \bar{w}_\ell \widehat{h}_{s,Z_\ell}(b)$
      Draw $J$ fresh actor actions; estimate the score or pathwise gradient of $-\mathbb{E}_\pi \widetilde{Q}_s$
    **end for**
    Average state gradients, add the original entropy gradient if used, and update $\theta$
  **end if**
**end while**

**Inputs and tuning.** The ambiguity radius $\delta$ determines the strength of regret protection. The soft temperature $\tau$ trades a sharp comparator against reliable sampling. The geometry controls which errors are considered plausible; for an ellipsoid this includes a ridge parameter, feature dimension, and curvature refresh schedule. The reference $\nu$ is part of the objective. The proposal $p$, defensive mixing weight $\varepsilon$, pilot size $K$, and candidate size $L$ are numerical choices. The remaining learning rates, state batch size, entropy coefficient, and actor sample size are inherited from the backbone unless explicitly varied.

**Gradient bookkeeping.** The actor optimizer sees a local surrogate whose gradient is the desired gradient at the current parameters. Freeze all candidate values, estimated policy means, weights, and witness parameters. During the actor loss evaluation, allow gradients through the actor action into both the nominal critic and the witness evaluation. Freezing critic *parameters* must not disable the

action derivative. For a score-function implementation, detach the sampled signal multiplying $\log \pi_\theta$; for a reparameterized implementation, differentiate its dependence on the sampled actor action.

**Independent samples.** Separating pilot, candidate, and actor samples makes the conditional targets and the error analysis explicit. Shared samples can be used as an engineering variant, but introduce dependence into the finite-sample estimator. The bounds in Appendix B correspond to the independent construction in Algorithm 1.

**State distributions and continuation.** For off-policy training, $s$ is drawn from the replay distribution and the actor step optimizes $\mathbb{E}_{s\sim d}\overline{\mathcal{R}}_{\tau,s}$. Critic training then tracks the updated continuation policy. This is a policy-improvement objective based on the current proxy, in the same operational sense that ordinary actor updates optimize their current learned $Q$. If a state distribution is itself differentiated as a function of the policy, its derivative must also be included; the replay-based update holds it fixed.

### C.2 Canonical twin-critic update

The twin-critic construction avoids the general witness calculation and solves the canonical endpoint problem instead. It uses one set of proposal actions for both critics, which also reduces the noise in their comparison.

**Algorithm 2** Twin-critic canonical soft-regret selector

**Require:** Twin critics $Q_1, Q_2$; current actor $\pi_\theta$; fixed reference $\nu_s$; temperature $\tau_s$
**Require:** Proposal $p_s$ with positive reference density ratio; candidate count $L$; policy-mean count $K$
**for** each state $s$ in an ordinary actor-update batch **do**
  Draw independent candidates $Z_{1:L} \sim p_s$ and policy actions $B_{1:K} \sim \pi_\theta$
  Evaluate both critics on both sets without parameter gradients
  $v_\ell \leftarrow \log(\mathrm{d}\nu_s/\mathrm{d}p_s)(Z_\ell)$
  **for** $i \in \{1, 2\}$ **do**
    $\widehat{M}_i \leftarrow \tau_s[\mathrm{LSE}(v_\ell + Q_i(s, Z_\ell)/\tau_s) - \mathrm{LSE}(v_\ell)]$
    $\widehat{V}_i \leftarrow K^{-1}\sum_k Q_i(s, B_k)$
    For deterministic $\pi_\theta = \delta_{a_\theta(s)}$, use $\widehat{V}_i = Q_i(s, a_\theta(s))$
  **end for**
  $\widehat{i}(s) \leftarrow \arg\max_i(\widehat{M}_i - \widehat{V}_i)$; freeze the selector
  Use $Q_{\widehat{i}(s)}$ in the backbone's actor loss, with fresh actor-loss samples
**end for**
Retain all critic targets, replay operations, and update schedules of the backbone

For SAC, the resulting actor loss is

$$L_{\mathrm{actor}} = \mathbb{E}_{s,a\sim\pi_\theta}[\alpha \log \pi_\theta(a \mid s) - Q_{\widehat{i}(s)}(s, a)].$$

For deterministic TD3 it is $-\mathbb{E}_s Q_{\widehat{i}(s)}(s, a_\theta(s))$. The selector varies by state and remains fixed during the backward pass. The clipped Bellman target remains part of the original critic update in either case.

**A local-policy selector variant.** The initial TD3 implementation used a local tanh–Gaussian distribution $g_{\theta,s}$ around the deterministic actor to estimate $\widehat{V}_i$, while retaining the deterministic action in the actor loss. This smooths endpoint selection around the actor's current action. Its selector is based on the regret of $g_{\theta,s}$, and its subsequent update follows the selected critic at $a_\theta(s)$. Algorithm 2 uses $Q_i(s, a_\theta(s))$ for exact deterministic-policy regret; alternatively, averaging the actor loss over the same reparameterized $g_{\theta,s}$ gives the exact local-policy objective. This identifies precisely which policy is being compared in each implementation.

**Reference versus proposal.** On a normalized action box $[-1, 1]^{d_a}$, uniform $\nu_s$ has density $2^{-d_a}$ and can be stored analytically. SAC can use $p_s = (1 - \varepsilon)\pi_\theta + \varepsilon\nu_s$. A deterministic actor requires a continuous local component $g_{\theta,s}$ if density-based mixture sampling is desired. These proposals

give positive reference ratios almost surely, making the sampled normalizers positive. Evaluate tanh–Gaussian densities with the change-of-variables Jacobian, and evaluate mixture densities using log-sum-exp. The proposal may change at every update without changing the target reference. Setting $\nu_s = \pi_\theta$ instead changes the objective and its derivative.

### C.3 Why an adaptive reference changes the objective

With a fixed reference and $\delta = 0$, $\mathcal{R}_\tau^{\mathrm{can}} = \overline{\mathcal{R}}_\tau = \mathrm{S}_{\tau,\nu}\,\widehat{Q} - \mathbb{E}_\pi \widehat{Q}$, so the first term is constant in the actor and the update is ordinary value ascent. If the reference is instead the current policy, even this nominal limit becomes

$$\tau \log \mathbb{E}_{\pi_\theta} e^{\widehat{Q}/\tau} - \mathbb{E}_{\pi_\theta} \widehat{Q}.$$

Its gradient contains the derivative of the reference in the log-partition. Moreover, this expression is zero for every deterministic policy, regardless of its value. A fixed broad reference preserves a meaningful comparison across actions; an adaptive proposal supplies the computational convenience of sampling near the current actor without introducing this degeneracy.

## D Concrete uncertainty geometries

### D.1 Lipschitz perturbations and Wasserstein distance

Let $c$ be a metric on $\mathcal{A}$, and take the anchored 1-Lipschitz unit ball. Kantorovich–Rubinstein duality identifies

$$d_{\mathcal{H}}(q, \pi) = W_{1,c}(q, \pi).$$

When one marginal is a Dirac measure, the transport plan is unique: every action sampled from $\pi$ must be transported to $z$. Thus

$$D_{\mathcal{H}}(z; \pi) = \mathbb{E}_{b\sim\pi} c(z, b), \qquad h_z^\pi(b) = -c(z, b) + c(z, a_0). \tag{39}$$

The triangle inequality ensures that the displayed witness is 1-Lipschitz, and substituting it attains the distance. The anchoring term cancels from the gap and from actor gradients. Pilot samples give $\widehat{D}(z) = K^{-1} \sum_k c(z, B_k)$; the shaped critic is

$$\widetilde{Q}(b) = \widehat{Q}(b) - \delta \sum_{\ell=1}^{L} \bar{w}_\ell c(Z_\ell, b) \quad \text{up to an action-independent constant.}$$

No general optimal-transport solver is needed for this singleton calculation. For Euclidean cost, the action derivative uses normalized differences away from coincident actions; a subgradient handles coincidence. For a general non-metric transport cost, the appropriate dual is a pair of transport potentials rather than the single Lipschitz ball used here.

### D.2 RKHS perturbations and maximum mean discrepancy

Let $\mathcal{H}_k$ be an RKHS with kernel $k$ and feature map $\varphi(a) = k(a, \cdot)$. The reproducing property gives $h(a) = \langle h, \varphi(a)\rangle_{\mathcal{H}_k}$. Assuming bounded kernel diagonal, the mean embedding $\mu_q = \mathbb{E}_q \varphi(Z)$ exists and

$$d_{\mathcal{H}_k}(q, \pi) = \sup_{\|h\|_{\mathcal{H}_k} \le 1} \langle h, \mu_q - \mu_\pi \rangle_{\mathcal{H}_k} = \|\mu_q - \mu_\pi\|_{\mathcal{H}_k}. \tag{40}$$

This is MMD (Gretton et al., 2012). For $D(z) = \|\varphi(z) - \mu_\pi\|_{\mathcal{H}_k} > 0$, the normalized representer gives

$$h_z^\pi(b) = \frac{k(z, b) - \mathbb{E}_{B\sim\pi} k(B, b)}{D(z)}, \quad D(z)^2 = k(z, z) - 2\mathbb{E}_B k(z, B) + \mathbb{E}_{B,B'} k(B, B'). \tag{41}$$

At $D(z) = 0$, zero is a maximizing witness. Using the empirical embedding $\widehat{\mu} = K^{-1} \sum_k \varphi(B_k)$ gives

$$\widehat{D}(z)^2 = k(z, z) - \frac{2}{K} \sum_k k(z, B_k) + \frac{1}{K^2} \sum_{k,j} k(B_k, B_j).$$

The diagonal-inclusive empirical expression is nonnegative in exact arithmetic because it is a squared Hilbert norm. Its square root estimates the norm entering DR3O. A squared MMD defines another discrepancy but is not itself the support function of the same unit ball; the norm, witness, and gradient here use unsquared MMD.

For a Gaussian kernel, action differences are compared on the kernel bandwidth scale. More generally,

$$|h(a) - h(b)| \le \|h\|_{\mathcal{H}_k} \sqrt{k(a,a) + k(b,b) - 2k(a,b)}.$$

For a continuous kernel on a compact domain, a bounded RKHS norm therefore imposes a uniform modulus of continuity. It excludes fixed-height spikes of arbitrarily small width. This couples uncertainty across actions, which is precisely the structure lost in a pointwise interval ambiguity set.

### D.3 Feature ellipsoids and learned representations

Let $x_\phi(s,a) \in \mathbb{R}^m$ be a critic feature vector, frozen during each actor update, and take

$$h_u(s,a) = u^\top x_\phi(s,a), \qquad u^\top H u \le 1, \qquad H \succ 0.$$

Writing $v_z = x_\phi(s,z) - \mu_\pi(s)$ and $\mu_\pi(s) = \mathbb{E}_{a\sim\pi} x_\phi(s,a)$, Cauchy–Schwarz in the $H$ geometry yields

$$D(z;\pi) = \sqrt{v_z^\top H^{-1} v_z}, \qquad u_z^\star = \frac{H^{-1} v_z}{\sqrt{v_z^\top H^{-1} v_z}}, \qquad h_z^\pi(b) = (u_z^\star)^\top x_\phi(s,b). \tag{42}$$

Use $u_z^\star = 0$ when $v_z = 0$. The aggregated direction $c_s = \delta \sum_\ell \bar{w}_\ell u_{Z_\ell}^\star$ then gives a compact actor signal

$$\widetilde{Q}(s,b) = \widehat{Q}_\phi(s,b) + c_s^\top x_\phi(s,b).$$

Only the feature evaluation at $b$ is differentiated in the bonus. The candidate directions and $c_s$ are fixed coefficients during the actor step.

**Constructing the ellipsoid.** If the last critic layer is linear, $Q(s,a) = w^\top x_\phi(s,a)$, perturbing $w$ gives an exact linear change for frozen features. With frozen regression targets, the squared-loss curvature in $w$ is proportional to a feature Gram matrix. A regularized, recency-weighted estimate is therefore

$$H_t = \lambda I + C_t, \qquad C_t = \beta C_{t-1} + (1-\beta)\frac{1}{B}\sum_{j=1}^{B} x_j x_j^\top, \qquad \lambda > 0. \tag{43}$$

Directions well represented in replay have larger curvature and smaller dual distances. The forgetting factor accommodates the changing representation. This is an adaptive geometry; a frequentist confidence interpretation additionally requires assumptions on regression noise, target construction, and representation drift. The radius remains a separately specified modeling or tuning parameter.

**Computing the dual norm.** Factor $H = LL^\top$ and solve $LY = V$, where the columns of $V$ are the candidate differences. Their squared norms are the column sums of $Y^2$. A second triangular solve computes $H^{-1}V$ for the witness directions. A factorization costs $O(m^3)$, and all $L$ candidate solves cost $O(m^2 L)$. Reusing the factorization for several actor steps amortizes its cost. This avoids full-network parameter gradients for every counterfactual action: the network supplies features through forward passes.

For rank-one updates $H^+ = H + xx^\top$, Sherman–Morrison gives

$$(H^+)^{-1} = H^{-1} - \frac{H^{-1} x x^\top H^{-1}}{1 + x^\top H^{-1} x},$$

with $O(m^2)$ arithmetic per update. Rank-$B$ Woodbury updates replace the scalar denominator by a $B \times B$ solve. Explicit inverse updates can accumulate floating-point asymmetry and cancellation, so a cached Cholesky factor with periodic refresh is a simple alternative. These are computational choices for the same dual norm, with costs governed by the feature dimension rather than the full neural-network parameter count.

### D.4 Finite actions and coordinate uncertainty

For $n$ actions, choose the perturbation ball $\|h\|_1 \le 1$. The dual is $d_{\mathcal{H}}(q,\pi) = \|q-\pi\|_\infty$, and

$$D_{\mathcal{H}}(i;\pi) = \|e_i - \pi\|_\infty = 1 - \pi_i.$$

The coordinate witness $h_i = e_i$ attains the supremum. The relaxed comparator is $\bar{q}_i \propto \nu_i \exp\{[\widehat{Q}_i + \delta(1-\pi_i)]/\tau\}$, and the actor bonus is $\delta\bar{q}_i$. This recovers the finite-action form associated with the RLHF construction (Wang et al., 2026) and shows why candidate uncertainty and actor correction are distinct quantities.

### D.5 Why coherent uncertainty matters for twin critics

The rectangular class in (13) allows errors at different actions to vary independently. Appendix A.8 derives the resulting spike adversary, its formula for atomic policies, and the conditions under which regret collapses to pessimistic value. The geometries above instead couple actionwise errors: a common Lipschitz or RKHS norm bound prevents fixed-height arbitrarily narrow bumps, while a coherent twin segment uses one coefficient across all actions. The choice of uncertainty class therefore determines which comparisons the actor is asked to protect.

## E Related literature

### E.1 Positioning and core related work

**Learning with an imperfect critic.** Double Q-learning, clipped double-Q learning, and distributional truncation control optimistic estimates (van Hasselt, 2010; Fujimoto et al., 2018; Haarnoja et al., 2019; Kuznetsov et al., 2020); conservative offline learning penalizes unsupported values (Kumar et al., 2020). Decision-aware actor-critic methods emphasize the decisions a critic induces (Vaswani et al., 2023). Our focus is the criterion applied to plausible critics: their values or their policy-relative regrets.

**Robustness and regret.** Robust MDPs commonly optimize worst-case return over transition uncertainty (Iyengar, 2005; Nilim & El Ghaoui, 2005; Xu & Mannor, 2012). Minimax-regret planning compares a policy with a scenario-specific optimum (Regan & Boutilier, 2010; Rigter et al., 2021). Distributionally robust regret has also been studied in optimization (Fiechtner & Blanchet, 2025). The closest work to ours probably is Wang et al. (2026), which studies the worst-case regret in RLHF, yet our works differs in two ways. First, we bring this criterion to repeated actor updates, where the learned action-value function already summarizes future consequences. Second, their derivation of the practical objective remains heuristic and provides no concrete guarantees. Appendix E discusses the broader literature.

**Robust and distributionally robust MDPs.** Robust dynamic programming studies uncertain transition laws and the structure needed for tractable Bellman recursions (Iyengar, 2005; Nilim & El Ghaoui, 2005; Wiesemann et al., 2013). Distributionally robust MDPs put ambiguity on distributions of model parameters (Xu & Mannor, 2012); foundations and interactive data collection are developed by Wang et al. (2023) and Lu et al. (2024). Robust adversarial reinforcement learning also trains policies against adversarial disturbances (Pinto et al., 2017). These approaches place robustness in the environment model or interaction process. The present construction operates on the action-value information supplied to an actor, which already summarizes downstream rewards and transitions for its current continuation policy.

### E.2 Additional related literature

**Regret as a decision criterion.** Regret compares a decision with the best decision under the same realized or hypothesized objective. Reward-uncertain planning (Regan & Boutilier, 2010), robust MDP planning (Rigter et al., 2021), and green-security applications (Xu et al., 2021) use this criterion in sequential problems. Minimax excess risk provides an analogous criterion in robust machine learning (Agarwal & Zhang, 2022). Distributionally robust regret has also been studied in general

optimization and control (Bitar, 2024; Al Taha et al., 2023; Fiechtner & Blanchet, 2025). The statewise actor regret here is a decision criterion at an update; cumulative learning regret instead sums performance gaps over interactions, as in bandit learning (Abbasi-Yadkori et al., 2011).

**RLHF and large action spaces.** Wang et al. (2026) treat a reward model as an informative proxy and derive a regret-based alternative to worst-case reward optimization for RLHF. A completion is a single action at the contextual-bandit level, with a very large discrete action space. Our function-space formulation extends the actor-facing principle to continuous actions and evolving action-value critics. The canonical-versus-relaxed distinction is particularly useful here: it identifies precisely when smoothing yields a globally solvable inner problem and when a further relaxation is needed.

**Value pessimism and decision-aware learning.** Beyond the twin-Q methods discussed in the introduction, Truncated Quantile Critics combines distributional critics and upper-tail truncation to address overestimation (Kuznetsov et al., 2020). Conservative Q-learning and pessimistic value iteration address insufficient coverage in offline learning (Kumar et al., 2020; Jin et al., 2021). Decision-aware actor-critic learning emphasizes the consequences of approximation error for policy improvement (Vaswani et al., 2023). These ideas motivate treating critic error according to its effect on decisions. DR3O does so through a worst-case gap to a critic-dependent comparator, producing an actor signal whose uncertainty correction depends on the current policy.

**Sampling and dual geometries.** Self-normalized importance sampling permits expectations under an unnormalized target to be estimated from a proposal (Hesterberg, 1995; Owen, 2013). Its second-moment ratio characterizes computational difficulty and effective sample size (Agapiou et al., 2017). Kernel mean embeddings give the RKHS support function used by MMD (Gretton et al., 2012); feature ellipsoids yield the familiar inverse-Gram geometry of linear uncertainty models (Abbasi-Yadkori et al., 2011). In DR3O these tools are applied to the difference between a comparator and the current actor, giving one derivation for several practical bonus constructions.

# F Full continuous-control results

This appendix reports the complete verified experiment record supporting Section 6. It separates three levels of evidence: the endpoint attained by each method, the within-seed change relative to the matched baseline, and the computational cost of producing that change. Results are grouped by benchmark so that the tables and their interpretation remain adjacent.

Absolute performance is reported as mean (SD) across training seeds. Paired entries report the method-minus-baseline mean difference with a two-sided 95% Student-$t$ confidence interval. The adjusted seed-level win count assigns one point to a positive paired difference, one-half point to an exact tie, and zero to a negative difference, then sums over the twenty training seeds.

**Interpretive strategy.** We read effect magnitude, paired intervals, and seed-level wins together. The mean describes the practical size of a change, while interval width and adjusted wins describe how consistently that change appears across seeds. We also distinguish ceiling effects, where an already strong baseline leaves little numerical room, from floor effects, where every method fails to complete the task. These distinctions prevent a suite average from hiding qualitatively different outcomes.

## F.1 Evaluation coverage

**Benchmark suites.** The evaluation spans three continuous-control families with distinct sources of difficulty. MuJoCo-v5 contributes nine locomotion and control tasks: Ant, HalfCheetah, Hopper, Humanoid, InvertedPendulum, InvertedDoublePendulum, Reacher, Swimmer, and Walker2d. MetaWorld MT1 contributes ten manipulation tasks: reach, push, pick-place, door-open, drawer-open, drawer-close, button-press-topdown, peg-insert-side, window-open, and window-close. DMC contributes six visually simple but dynamically different control tasks: ball-in-cup catch, cartpole swingup, cheetah run, finger spin, reacher easy, and walker walk.

**Backbone coverage.** We study four off-policy actor–critic frameworks. SAC and CrossQ provide stochastic, entropy-regularized backbones, while TD3 and TD7 provide deterministic TD-style backbones. The full four-backbone transfer study is available for MuJoCo. MetaWorld and DMC use SAC and TD3; the absence of CrossQ and TD7 rows for those suites reflects the evaluated matrix rather than omitted results.

Table 2: Complete benchmark–backbone evaluation matrix. The task count is the number of distinct environments.

| Benchmark | Tasks | Backbones | Endpoint and reported outcome |
|---|---|---|---|
| MuJoCo-v5 | 9 | SAC, TD3, CrossQ, TD7 | 1M steps; return |
| MetaWorld MT1 | 10 | SAC, TD3 | 500k steps; success (primary) and return |
| DMC | 6 | SAC, TD3 | 500k steps; return |

**Compared actor updates.** Every evaluated task–backbone cell contains four matched methods: the original backbone (Baseline), DR3O, fixed-scale feature-ellipsoidal regret optimization (Ellip), and its MSE-adaptive version (Adaptive Ellip). Adaptive Ellip estimates the ambiguity scale online from critic mean-squared error on replay batches. The matrix therefore contains 36 MuJoCo, 20 MetaWorld, and 12 DMC task–backbone comparisons, for 68 comparisons in total. Within each comparison, interaction budgets, critic-training rules, replay configurations, architectures, optimizers, and update schedules are held fixed; only the actor-facing decision rule changes.

**Endpoints and metrics.** MuJoCo uses the 1M-step endpoint, while MetaWorld and DMC use 500k steps. Return is the primary outcome for MuJoCo and DMC. MetaWorld reports both dense shaped return and task success, with success treated as primary because return can increase without completing the manipulation.

**Sources of variation.** The 68 comparisons vary along four axes: stochastic versus deterministic actors, low- versus high-capacity critic updates, tasks with substantial headroom versus near-saturated tasks, and smooth-return control versus thresholded manipulation success. The purpose of this matrix is therefore not only to increase the number of runs. It tests whether the same actor-side principle survives changes in exploration, critic construction, task geometry, and evaluation metric. Because raw reward scales differ, all claims are made within matched task–backbone cells before being summarized across a suite.

**Task-selection rationale.** The task set was fixed before evaluation on seeds 100–119. MuJoCo supplies established locomotion, balance, and low-dimensional control problems. MetaWorld adds contact-rich manipulation, articulated objects, insertion, and thresholded task completion. DMC adds complementary dynamical systems, including swing-up, spinning, catching, reaching, and walking. MetaWorld and DMC tasks were chosen to reduce behavioral duplication with MuJoCo and were included only when a reliable matched SAC and TD3 baseline was available; selection did not use the held-out method outcomes.

### F.2 Experimental protocol and configurations

**Software and observations.** The experiments use Gymnasium 1.3.0 with the MuJoCo-v5 tasks, MetaWorld 3.1.1, and DeepMind Control 1.0.46 through Shimmy 2.0.1. Policies receive state observations rather than pixels. Dictionary-valued DMC observations are flattened before being passed to the actor and critic, and continuous actions are represented internally on the normalized action box. True environment terminations suppress bootstrapping, whereas time-limit truncations reset the environment without suppressing the bootstrap.

**Backbone configurations.** Table 4 records the settings shared by the baseline and all three regret-aware updates. SAC and TD3 use two 256-unit ReLU hidden layers, a replay capacity of $10^6$, batch size 256, discount $0.99$, one gradient update per environment step, learning rate $3 \times 10^{-4}$ for the actor and critic, and target update rate $0.005$. SAC learns its temperature from $0.2$ with target entropy $-d_a$. TD3 uses exploration standard deviation $0.1$, target smoothing standard deviation $0.2$ clipped to $0.5$, and an actor/target update every two critic updates. CrossQ uses a 256-unit actor, a 2,048-unit critic,

Table 3: Suite-level training and evaluation protocol. The notation 10/25k means ten deterministic evaluation episodes every 25,000 environment steps, including the final checkpoint.

| Suite and backbone | Horizon | Learning start / warmup | Evaluation | Reported outcome |
|---|---|---|---|---|
| MuJoCo SAC | 1M | 25k | 10/25k; 100 at final | Return |
| MuJoCo TD3 | 1M | 25k | 10/25k; 100 at final | Return |
| MuJoCo CrossQ | 1M | 5k | 10/25k; 100 at final | Return |
| MuJoCo TD7 | 1M | 25k | 10/25k; 100 at final | Return |
| MetaWorld SAC/TD3 | 500k | 50k | 10/25k; 100 at final | Success (primary), return |
| DMC SAC/TD3 | 500k | 50k | 10/25k; 100 at final | Return |

learning rate $10^{-3}$, Adam first-moment coefficient $0.5$, batch-normalization momentum $0.01$, and policy delay three; its initial entropy temperature is $1.0$. TD7 uses 256-dimensional state embeddings and 256-unit encoder, actor, and critic hidden representations with learning rate $3 \times 10^{-4}$. CrossQ uses discount $0.9999$ on Swimmer and $0.99$ elsewhere; all other listed backbones use $0.99$.

Table 4: Shared backbone configuration. Regret-aware variants change only the actor-facing decision rule; replay, critic loss, optimizer, data collection, and update frequency remain matched to the baseline.

| Backbone | Actor width | Critic width | Batch | Learning rate | Principal backbone setting |
|---|---|---|---|---|---|
| SAC | $2 \times 256$ | $2 \times 256$ | 256 | $3 \times 10^{-4}$ | learned $\alpha$, target $-d_a$ |
| TD3 | $2 \times 256$ | $2 \times 256$ | 256 | $3 \times 10^{-4}$ | delay 2; noise $0.1/0.2$ |
| CrossQ | $2 \times 256$ | $2 \times 2048$ | 256 | $10^{-3}$ | BN momentum $0.01$; delay 3 |
| TD7 | 256-d representations | 256 | 256 | $3 \times 10^{-4}$ | checkpointed TD7 update |

**Regret-update configurations.** All regret-aware variants mix policy-centered samples with a uniform reference component of mass $\rho = 0.1$. SAC and TD3 use $L/K = 128/64$ candidate and independent policy samples, while CrossQ and TD7 use $64/32$. Deterministic proposals use standard deviation $0.1$ for TD3 and $0.2$ for TD7. Fixed and Adaptive Ellip use covariance decay $0.995$, relative ridge $0.1$, Cholesky refresh every 50 tracker updates, and numerical stabilizers $10^{-6}$. In SAC and TD3 the ellipsoidal radius activates after 50k transitions and ramps linearly for another 50k; CrossQ and TD7 activate when learning starts.

Adaptive Ellip uses the same task-specific base radius and ellipsoid temperature as fixed Ellip. It samples normalized Bellman MSE every 100 replay updates and averages a window of 100 samples. With the first complete active window frozen as $\overline{m}_{\mathrm{ref}}$, the implemented multiplier is

$$\mathrm{clip}\big(1 + 0.5\,(\overline{m}_t / \max(\overline{m}_{\mathrm{ref}}, 10^{-8}) - 1)\,, 0.5, 1.5\big)\,.$$

The scheduled base radius is multiplied by this quantity; fixed Ellip uses multiplier one.

**Selected MuJoCo hyperparameters.** For DR3O, SAC uses $T = 6$ on Ant, 12 on HalfCheetah, $0.1$ on Reacher, $0.25$ on Swimmer, 5 on Walker2d, and 1 on the remaining four tasks. TD3 uses $T = 5$ on Ant and Walker2d, $0.1$ on Reacher, 2 on Swimmer, and 1 elsewhere. CrossQ uses $T = 1$ throughout; TD7 uses $T = 0.1$ on Reacher and 1 elsewhere.

Fixed and Adaptive Ellip share the same $(\tau, \delta)$ in every task–backbone cell. For SAC, the selected pairs are $(10, 3)$ on Ant, $(2, 0.02)$ on HalfCheetah, $(1, 0.05)$ on Hopper and Swimmer, $(10, 0.3)$ on Humanoid, $(1, 0.03)$ on InvertedPendulum and InvertedDoublePendulum, $(0.1, 0.01)$ on Reacher, and $(10, 1)$ on Walker2d. TD3 uses the same pairs except for HalfCheetah, where it uses $(1, 0.3)$. CrossQ uses $(1, 0.03)$ on the two inverted-pendulum tasks and $(10, 1)$ elsewhere. TD7 additionally uses $(0.1, 0.01)$ on Reacher; its other settings match CrossQ.

**Selected MetaWorld hyperparameters.** For DR3O with SAC, $T = 0.5$ on reach, 1 on pick-place and button-press-topdown, and 5 on the other seven tasks. With TD3, $T = 0.5$ on reach, 1 on push, pick-place, door-open, and button-press-topdown, and 5 on drawer-open, drawer-close, peg-insert-side, window-open, and window-close. Both Ellip variants use $\tau = 10$ throughout and share $\delta = 1$ on push, door-open, and peg-insert-side, with $\delta = 0.3$ on the remaining tasks.

**Selected DMC hyperparameters.** For DR3O with SAC, $T = 5$ on cartpole swingup, cheetah run, and finger spin, and $T = 1$ on ball-in-cup catch, walker walk, and reacher easy. TD3 uses $T = 1$

throughout. Fixed and Adaptive Ellip use $(\tau, \delta) = (10, 0.3)$ by default. Reacher easy uses $(5, 0.1)$ with both backbones, and TD3 cheetah run uses $(10, 0.03)$.

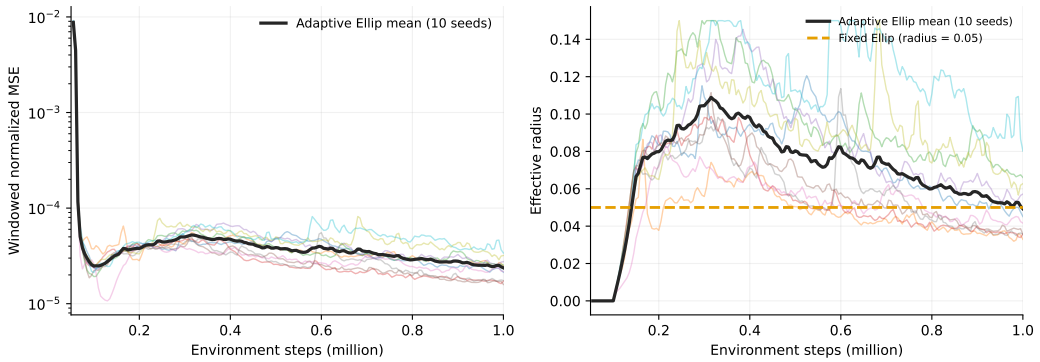



Figure 2: Adaptive-radius dynamics on SAC Swimmer. Left: windowed Bellman MSE normalized by the mean squared TD target. Right: the resulting Adaptive Ellip effective radius. Thin colored curves show individual seeds and the black curve shows their mean; the dashed orange line marks the fixed Ellip radius $\delta = 0.05$. Following activation and ramp-in, the adaptive radius expands when normalized critic error is elevated and gradually contracts toward the fixed scale as training progresses.

**Adaptive-radius dynamics.** Figure 2 illustrates how the MSE-driven rule changes the ambiguity scale during training rather than selecting another fixed constant. On SAC Swimmer, the radius is inactive during the initial calibration period, rises above the fixed-Ellip value as the normalized critic error increases, and then contracts as the error subsides. The seed-level trajectories retain meaningful variation, while their mean follows the same expand-then-contract pattern. This behavior connects the adaptive construction to an observable training signal and explains how it can remain conservative when critic error is high without enforcing the same radius throughout optimization.

**Hyperparameter selection and held-out evaluation.** Hyperparameters are selected on four tuning seeds, 200–203. For every candidate configuration, the selection score is mean deterministic return at the final training checkpoint; intermediate peaks are not used. The search is task-specific: follow-up screens refine the initial candidates when needed, so a single rectangular grid would not describe the actual search. The accompanying configuration manifests enumerate the candidates and candidate count for every task–backbone cell, while the paragraphs above record the selected settings. Only full-horizon candidates with all four tuning seeds are eligible for selection. All endpoint means, SDs, paired intervals, and seed-level win counts reported below use the disjoint training seeds 100–119. Baseline and regret-aware runs are paired by training seed within each task–backbone cell; tuning runs and incomplete runs do not enter the reported tables.

**Endpoint and statistical definitions.** An endpoint is the deterministic evaluation at the final training checkpoint: one million environment steps for MuJoCo and 500,000 for MetaWorld and DMC. Evaluation episodes are averaged within each training seed, after which the reported mean and sample SD are computed over the twenty training seeds. A paired entry is formed by subtracting the matched baseline endpoint from the method endpoint for each seed and applying a two-sided Student-$t$ interval to those twenty differences. MetaWorld success is the fraction of evaluation episodes that attain the environment success signal; success differences are therefore measured in percentage points.

**Hardware and scheduling.** All reported training runs use NVIDIA RTX A6000 GPUs. The scheduler maintains up to four concurrent training processes per GPU. Baseline and regret-aware variants within a task–backbone cell use the same GPU class, software environment, and packing policy, so the timing comparisons preserve the same scheduling conditions.

### F.3 Robust cross-task aggregation

Figure 3 complements the arithmetic means in Table 1 with two summaries computed from the reported endpoint means. The upper row reports the interquartile mean (IQM) of the matched task–backbone improvement. The lower row reports the fraction of task–backbone cells whose endpoint mean exceeds its matched baseline. Intervals are 95% percentile bootstrap intervals from 10,000 resamples of tasks, retaining all evaluated backbones for a sampled task. MuJoCo uses percentage return gain after excluding negative-return Reacher, MetaWorld uses success-rate gain in percentage points, and DMC uses return-point gain.

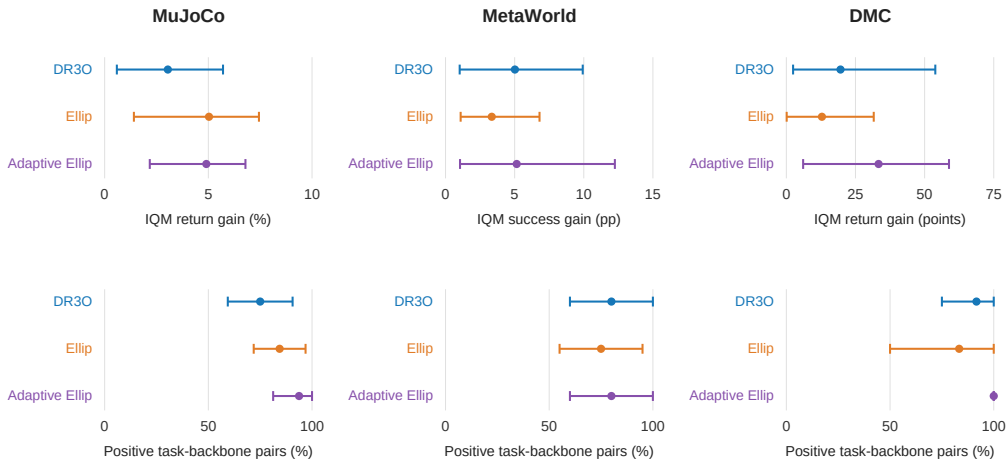



Figure 3: Robust effect size and cross-task consistency. Points show IQM improvement (top) and the percentage of task–backbone cells improved (bottom); horizontal bars are task-bootstrap 95% intervals. Adaptive Ellip improves 30/32 scale-comparable MuJoCo cells, 16/20 MetaWorld cells, and all 12 DMC cells, while retaining positive IQM gains in every suite.

### F.4 Absolute endpoint performance

**How to read the absolute tables.** The mean records endpoint performance and the SD records variation across training seeds. Comparisons should be made within an environment and backbone because reward scales differ across tasks. These tables answer whether a method changes the achieved return or success level; the paired tables in Section F.5 provide the corresponding uncertainty assessment.

#### F.4.1 MuJoCo: transfer across four actor–critic backbones

The MuJoCo block evaluates the same nine tasks with SAC, TD3, CrossQ, and TD7. Reporting all four backbones tests whether the actor-side correction transfers beyond the two algorithms used in the initial study. Returns are shown in the native task units; Reacher retains its conventional negative-return scale.

Table 5: MuJoCo SAC — return, mean (SD)

| Environment | Baseline | DR3O | Ellip | Adaptive Ellip |
|---|---|---|---|---|
| Ant | 4,489.1 (772.3) | 4,429.1 (691.0) | 4,888.7 (854.0) | 4,739.8 (801.1) |
| HalfCheetah | 11,071.7 (599.8) | 11,015.0 (618.7) | 11,908.2 (606.6) | 11,890.1 (1,056.1) |
| Hopper | 3,027.2 (773.9) | 3,267.5 (728.3) | 3,320.7 (817.8) | 3,159.4 (791.8) |
| Humanoid | 5,035.0 (400.3) | 5,361.5 (500.0) | 5,131.9 (411.6) | 5,250.9 (469.0) |
| InvertedPendulum | 988.2 (9.4) | 992.2 (9.5) | 983.2 (26.1) | 981.7 (12.9) |
| InvertedDoublePendulum | 9,060.9 (183.2) | 9,135.7 (223.1) | 9,052.7 (208.5) | 9,175.8 (157.7) |
| Reacher | -3.9 (0.9) | -3.9 (1.3) | -4.0 (1.1) | -3.2 (1.0) |
| Swimmer | 90.8 (25.2) | 99.0 (20.8) | 130.1 (8.0) | 135.1 (7.7) |
| Walker2d | 3,972.1 (993.4) | 4,358.8 (547.3) | 4,220.8 (1,098.0) | 4,165.3 (691.5) |

Table 6: MuJoCo TD3 — return, mean (SD)

| Environment | Baseline | DR3O | Ellip | Adaptive Ellip |
|---|---|---|---|---|
| Ant | 4,552.8 (760.4) | 5,130.2 (684.4) | 4,766.6 (771.9) | 4,865.8 (863.7) |
| HalfCheetah | 10,048.7 (1,222.0) | 10,733.9 (1,195.2) | 10,834.3 (914.6) | 11,108.2 (1,271.2) |
| Hopper | 2,658.1 (1,244.1) | 3,041.6 (853.9) | 2,917.0 (1,147.3) | 2,808.5 (1,200.4) |
| InvertedPendulum | 989.9 (6.3) | 976.9 (31.3) | 997.6 (1.8) | 989.1 (8.5) |
| InvertedDoublePendulum | 9,010.0 (256.1) | 9,031.2 (314.1) | 9,137.2 (251.0) | 9,012.6 (439.3) |
| Humanoid | 4,144.0 (458.5) | 4,083.8 (412.6) | 4,070.6 (459.3) | 4,520.6 (497.8) |
| Reacher | -4.6 (1.6) | -3.7 (1.3) | -4.6 (1.4) | -3.2 (1.1) |
| Swimmer | 93.8 (30.6) | 94.5 (30.3) | 103.3 (26.5) | 103.5 (31.3) |
| Walker2d | 3,927.7 (510.3) | 4,123.1 (498.1) | 4,220.1 (569.6) | 4,178.7 (512.8) |

Table 7: MuJoCo CrossQ — return, mean (SD)

| Environment | Baseline | DR3O | Ellip | Adaptive Ellip |
|---|---|---|---|---|
| Ant | 6,699.2 (395.4) | 7,140.7 (476.1) | 7,250.0 (583.8) | 7,006.2 (447.5) |
| HalfCheetah | 12,831.7 (2,658.7) | 13,439.0 (2,604.6) | 13,666.0 (2,776.5) | 14,223.5 (2,247.1) |
| Hopper | 2,650.1 (780.6) | 2,911.5 (736.9) | 2,939.9 (735.9) | 2,854.6 (847.0) |
| Humanoid | 9,964.6 (535.9) | 10,492.5 (750.4) | 10,961.4 (536.7) | 10,379.0 (591.8) |
| InvertedPendulum | 992.9 (7.8) | 998.2 (5.1) | 994.4 (5.0) | 998.5 (1.8) |
| InvertedDoublePendulum | 9,242.5 (85.3) | 9,222.6 (151.2) | 9,166.1 (169.1) | 9,315.5 (31.1) |
| Reacher | -4.6 (1.4) | -3.8 (1.6) | -5.2 (1.7) | -4.1 (1.7) |
| Swimmer | 94.6 (27.9) | 99.3 (32.3) | 91.8 (22.8) | 103.7 (22.2) |
| Walker2d | 6,566.9 (1,003.9) | 7,062.9 (1,101.9) | 6,953.1 (979.7) | 7,082.6 (967.1) |

Table 8: MuJoCo TD7 — return, mean (SD)

| Environment | Baseline | DR3O | Ellip | Adaptive Ellip |
|---|---|---|---|---|
| Ant | 8,557.5 (1,095.3) | 8,535.3 (1,222.8) | 8,857.4 (970.6) | 8,861.6 (901.3) |
| HalfCheetah | 17,221.5 (385.9) | 17,805.1 (545.6) | 18,098.9 (914.8) | 17,879.2 (590.5) |
| Hopper | 3,684.3 (300.6) | 3,846.9 (286.5) | 3,814.6 (315.5) | 3,895.0 (320.0) |
| Humanoid | 6,989.0 (225.1) | 6,971.8 (190.2) | 7,300.8 (275.3) | 7,228.8 (320.6) |
| InvertedPendulum | 992.1 (11.4) | 989.0 (10.9) | 996.9 (2.9) | 997.1 (2.3) |
| InvertedDoublePendulum | 9,051.7 (191.5) | 9,102.4 (168.4) | 9,060.5 (163.0) | 9,308.0 (99.6) |
| Reacher | -5.8 (2.3) | -4.4 (1.4) | -6.4 (2.1) | -6.2 (1.7) |
| Swimmer | 114.2 (13.1) | 116.6 (15.3) | 124.2 (11.1) | 118.9 (12.3) |
| Walker2d | 6,365.6 (544.4) | 6,524.0 (594.9) | 6,710.4 (535.3) | 6,598.1 (451.9) |

**MuJoCo interpretation: backbone transfer.** The main phenomenon is transfer across actor parameterizations rather than dominance by one implementation. With SAC, fixed Ellip produces the largest HalfCheetah and Swimmer gains (+836.5 and +39.3), while Adaptive Ellip is strongest on Reacher and InvertedDoublePendulum. With TD3, DR3O is strongest on Ant and Hopper, whereas Adaptive Ellip gives the largest HalfCheetah, Humanoid, Reacher, and Swimmer means. CrossQ is especially informative because its baseline critic and update rule differ substantially from SAC and TD3: Adaptive Ellip raises the mean on all nine CrossQ tasks, including +1,391.8 on HalfCheetah and +515.7 on Walker2d. TD7 gives the same qualitative transfer result; its largest gains alternate between fixed and adaptive Ellip instead of collapsing to a single method.

**MuJoCo interpretation: headroom and task scale.** The largest absolute changes occur on locomotion tasks whose policies are strong but not saturated. HalfCheetah improves under every backbone and all three regret-aware variants, and Walker2d improves in mean direction in every cell. Ant, Hopper, and Humanoid show a more method-dependent ordering, which is consistent with their higher seed variability and different failure modes. InvertedPendulum is already near its nominal ceiling of 1,000, so changes of a few return points can be statistically stable without being practically large. InvertedDoublePendulum is also high-performing at baseline, although Adaptive Ellip still produces a clear +256.3 TD7 gain.

**MuJoCo interpretation: scope.** The few regressions are concentrated in saturated pendulum settings and selected Reacher cells, whereas the large locomotion gains recur across backbones. The main MuJoCo phenomenon is therefore broad positive transfer with task-dependent method ordering. Reacher is interpreted in absolute return units and excluded from percentage aggregation because its baseline return is negative.

### F.4.2 METAWORLD: SHAPED RETURN AND TASK COMPLETION

MetaWorld contains both dense shaped returns and binary task success. We therefore report both views for SAC and TD3. Success is the primary measure of whether the manipulation was completed, while return reveals partial progress when the success threshold is not reached.

Table 9: MetaWorld 10 tasks (MT1) SAC — return, mean (SD)

| Environment | Baseline | DR3O | Ellip | Adaptive Ellip |
|---|---|---|---|---|
| reach-v3 | 4,487.4 (396.1) | 4,547.0 (379.8) | 4,545.6 (368.5) | 4,480.8 (344.1) |
| push-v3 | 658.4 (519.7) | 651.5 (521.9) | 603.7 (490.6) | 732.0 (622.8) |
| pick-place-v3 | 6.7 (3.0) | 7.4 (3.4) | 6.7 (3.1) | 7.6 (3.8) |
| door-open-v3 | 3,538.7 (748.4) | 3,654.2 (683.6) | 3,493.8 (791.7) | 3,682.6 (685.9) |
| drawer-open-v3 | 3,451.6 (540.5) | 3,573.2 (531.4) | 3,494.7 (508.9) | 3,408.5 (520.1) |
| drawer-close-v3 | 4,815.2 (79.0) | 4,846.5 (60.5) | 4,875.4 (42.3) | 4,863.4 (51.5) |
| button-press-topdown-v3 | 3,702.7 (342.4) | 3,954.5 (599.6) | 4,043.3 (525.4) | 3,998.9 (462.4) |
| peg-insert-side-v3 | 1,078.0 (611.9) | 1,138.4 (877.5) | 1,167.5 (824.8) | 1,155.6 (803.9) |
| window-open-v3 | 4,031.3 (575.1) | 4,329.0 (483.7) | 4,272.3 (504.7) | 4,237.1 (447.6) |
| window-close-v3 | 4,214.9 (505.8) | 4,507.6 (407.9) | 4,445.1 (341.0) | 4,260.2 (484.9) |

Table 10: MetaWorld 10 tasks (MT1) TD3 — return, mean (SD)

| Environment | Baseline | DR3O | Ellip | Adaptive Ellip |
|---|---|---|---|---|
| reach-v3 | 4,225.2 (599.4) | 4,313.8 (463.6) | 4,372.9 (317.4) | 4,170.7 (727.6) |
| push-v3 | 553.3 (337.5) | 878.2 (615.0) | 580.5 (479.9) | 813.5 (547.3) |
| pick-place-v3 | 5.9 (2.2) | 7.1 (2.6) | 6.4 (2.2) | 4.9 (1.9) |
| door-open-v3 | 3,186.4 (946.1) | 3,327.8 (872.0) | 3,258.1 (893.8) | 3,374.5 (851.7) |
| drawer-open-v3 | 3,062.8 (746.3) | 3,238.4 (730.2) | 3,016.7 (769.6) | 3,189.3 (682.0) |
| drawer-close-v3 | 4,720.8 (98.0) | 4,749.1 (88.5) | 4,765.6 (51.4) | 4,786.9 (77.0) |
| button-press-topdown-v3 | 3,263.4 (1,300.1) | 3,531.0 (1,207.1) | 3,455.9 (1,182.2) | 3,560.4 (1,151.2) |
| peg-insert-side-v3 | 444.1 (748.0) | 477.0 (773.1) | 464.5 (670.4) | 485.4 (661.9) |
| window-open-v3 | 3,955.8 (594.6) | 4,027.3 (623.9) | 4,052.1 (712.2) | 4,275.8 (438.1) |
| window-close-v3 | 4,275.1 (443.6) | 4,128.9 (498.4) | 4,049.0 (498.1) | 4,134.1 (464.0) |

Table 11: MetaWorld 10 tasks (MT1) SAC — success rate (%), mean (SD)

| Environment | Baseline | DR3O | Ellip | Adaptive Ellip |
|---|---|---|---|---|
| reach-v3 | 99.25 (2.45) | 99.75 (1.12) | 100.00 (0.00) | 99.75 (1.12) |
| push-v3 | 18.75 (21.64) | 28.75 (22.88) | 24.00 (20.17) | 34.50 (23.56) |
| pick-place-v3 | 0.00 (0.00) | 0.00 (0.00) | 0.00 (0.00) | 0.00 (0.00) |
| door-open-v3 | 76.25 (21.39) | 84.50 (13.95) | 76.00 (21.37) | 84.00 (15.18) |
| drawer-open-v3 | 50.25 (21.85) | 63.50 (18.43) | 61.50 (15.74) | 58.00 (15.93) |
| drawer-close-v3 | 96.8 (5.2) | 98.0 (3.8) | 99.3 (2.4) | 98.8 (3.2) |
| button-press-topdown-v3 | 97.3 (3.4) | 98.8 (2.8) | 99.3 (1.8) | 98.8 (2.2) |
| peg-insert-side-v3 | 7.50 (14.82) | 23.50 (17.85) | 22.00 (18.67) | 26.00 (14.92) |
| window-open-v3 | 88.50 (19.27) | 97.75 (4.44) | 93.75 (8.56) | 97.00 (4.10) |
| window-close-v3 | 97.25 (4.13) | 98.25 (2.94) | 98.25 (2.45) | 94.75 (6.58) |

Table 12: MetaWorld 10 tasks (MT1) TD3 — success rate (%), mean (SD)

| Environment | Baseline | DR3O | Ellip | Adaptive Ellip |
|---|---|---|---|---|
| reach-v3 | 89.25 (18.52) | 94.00 (14.47) | 96.25 (7.05) | 85.75 (18.66) |
| push-v3 | 13.50 (14.96) | 38.50 (23.29) | 23.75 (17.98) | 39.75 (20.10) |
| pick-place-v3 | 0.00 (0.00) | 0.00 (0.00) | 0.00 (0.00) | 0.00 (0.00) |
| door-open-v3 | 57.25 (30.54) | 67.25 (23.92) | 64.00 (26.54) | 71.75 (22.32) |
| drawer-open-v3 | 64.75 (20.74) | 69.75 (18.10) | 62.75 (22.03) | 72.25 (17.13) |
| drawer-close-v3 | 97.3 (4.7) | 97.8 (3.8) | 99.5 (1.5) | 97.8 (3.4) |
| button-press-topdown-v3 | 75.00 (34.18) | 83.75 (26.15) | 82.25 (28.90) | 86.25 (21.76) |
| peg-insert-side-v3 | 15.00 (29.60) | 28.75 (30.43) | 26.25 (23.78) | 34.25 (27.83) |
| window-open-v3 | 90.00 (10.51) | 85.00 (17.01) | 85.75 (18.52) | 90.75 (12.28) |
| window-close-v3 | 86.75 (10.17) | 86.00 (11.07) | 87.50 (10.70) | 90.75 (8.32) |

**MetaWorld interpretation: where completion improves.** The clearest gains occur on tasks that are neither solved nor completely out of reach. Adaptive Ellip improves TD3 success by 26.25 percentage points on push, 14.50 on door-open, 7.50 on drawer-open, 11.25 on button-press-topdown,

and 19.25 on peg-insert-side. Under SAC, the same method improves push and peg-insert-side by 15.75 and 18.50 points. DR3O is also strong on these intermediate-difficulty tasks, including +25.00 points on TD3 push and +16.00 on SAC peg-insert-side. The repeated gains on contact-rich pushing, insertion, and door manipulation suggest that the correction is most useful when the critic provides a meaningful direction but ranks nearby actions imperfectly.

**MetaWorld interpretation: return is not success.** Several cells show why shaped return and completion must be reported separately. On SAC drawer-open, Adaptive Ellip changes return by $-43.1$ but success by +7.75 points; on SAC window-close, its return mean increases by 45.3 while success decreases by 2.50 points. Under TD3, Adaptive Ellip adds 320.0 return on window-open but only 0.75 success points, whereas window-close loses 141.0 return but gains 4.00 success points. These are not contradictions: dense rewards record trajectory-level progress, while success records whether the terminal task criterion was crossed. The success tables therefore anchor the MetaWorld claim, and the return tables explain how behavior can change below or around the threshold.

**MetaWorld interpretation: ceilings and floors.** Reach and drawer-close are close to saturation for at least one backbone, leaving little room for success gains. At the opposite extreme, pick-place remains at zero success for every method and both backbones despite small return differences. Adaptive Ellip has negative success changes on TD3 reach ($-3.50$ points) and SAC window-close ($-2.50$ points). The largest benefits occur on unsaturated tasks where the learned feedback already supports successful behavior.

### F.4.3 DMC: CONSISTENCY ACROSS CONTROL DOMAINS

DMC covers six control tasks under SAC and TD3. The reward scale is shared more closely within this suite than in MuJoCo, but conclusions are still drawn from task-wise matched comparisons rather than from pooling raw returns.

Table 13: DMC SAC — return, mean (SD)

| Environment | Baseline | DR3O | Ellip | Adaptive Ellip |
|---|---|---|---|---|
| ball_in_cup-catch | 982.0 (1.9) | 984.7 (3.2) | 985.0 (3.0) | 984.6 (2.3) |
| cartpole-swingup | 868.8 (11.3) | 905.0 (20.4) | 915.1 (25.9) | 917.1 (18.6) |
| cheetah-run | 731.8 (45.5) | 823.3 (50.9) | 800.1 (52.0) | 808.9 (41.5) |
| finger-spin | 916.1 (30.4) | 954.6 (26.4) | 947.8 (45.2) | 964.0 (18.3) |
| reacher-easy | 950.0 (28.6) | 936.9 (40.4) | 946.6 (39.2) | 981.4 (10.6) |
| walker-walk | 964.3 (5.1) | 968.9 (4.6) | 970.6 (5.3) | 971.0 (7.5) |

Table 14: DMC TD3 — return, mean (SD)

| Environment | Baseline | DR3O | Ellip | Adaptive Ellip |
|---|---|---|---|---|
| ball_in_cup-catch | 981.9 (2.6) | 985.0 (3.6) | 984.4 (2.8) | 982.3 (3.0) |
| cartpole-swingup | 866.8 (9.7) | 897.3 (11.0) | 886.3 (20.8) | 927.4 (15.7) |
| cheetah-run | 809.1 (51.0) | 871.3 (64.1) | 822.2 (41.4) | 866.9 (36.9) |
| finger-spin | 858.0 (123.9) | 919.9 (105.1) | 895.5 (104.7) | 928.6 (78.9) |
| reacher-easy | 955.2 (23.4) | 955.5 (17.1) | 950.0 (21.9) | 963.2 (19.0) |
| walker-walk | 964.3 (6.6) | 968.9 (6.6) | 967.9 (8.9) | 970.5 (4.9) |

**DMC interpretation: dynamic tasks versus saturated tasks.** Adaptive Ellip improves the mean return on all 12 task–backbone pairs. The largest TD3 gains occur on finger-spin (+70.6), cartpole-swingup (+60.6), and cheetah-run (+57.8), where a policy must repeatedly rank dynamically different actions over long trajectories. Under SAC, cartpole, cheetah, and finger again show substantial gains. By contrast, ball-in-cup and walker-walk begin near 980 and 964 return, respectively; their improvements are numerically small because little reward remains available, although most paired intervals are still positive because seed variance is low.

**DMC interpretation: calibration matters.** Reacher-easy provides the sharpest separation among variants. Under SAC, DR3O changes return by $-13.1$ and fixed Ellip by $-3.4$, while Adaptive Ellip gains +31.4 with all 20 adjusted seed comparisons favoring it. Under TD3, the corresponding adaptive

gain is +8.0 and its interval remains positive. The adaptive scale can therefore correct an overly aggressive or poorly matched fixed correction on this task. At the same time, DR3O is strongest on SAC cheetah-run (+91.5), so adaptation improves consistency rather than selecting the largest method in every cell.

### F.5 Paired improvements over baseline

**Statistical reading.** Each row pairs a regret-aware run with the baseline run using the same training seed, then reports the mean paired difference and its two-sided 95% confidence interval. Intervals above zero give the strongest paired evidence. The adjusted win count is complementary: values above one half show that improvement is typical across seeds rather than driven only by a few large runs.

#### F.5.1 MuJoCo paired evidence

The four tables below preserve native return units for statistical interpretation. The accompanying figure divides each paired difference by the absolute baseline mean solely for visualization across heterogeneous task scales; this rescaling does not change interval signs.

Table 15: MuJoCo SAC return — Δ [95% CI], adjusted wins/20

| Environment | DR3O Δ [CI] | Adj. Win | Ellip Δ [CI] | Adj. Win | Adaptive Δ [CI] | Adj. Win |
|---|---|---|---|---|---|---|
| Ant | -60.0 [-293.5, +173.5] | 8/20 | +399.6 [+137.6, +661.5] | 16/20 | +250.7 [+50.3, +451.1] | 15/20 |
| HalfCheetah | -56.7 [-172.2, +58.7] | 10/20 | +836.5 [+709.3, +963.7] | 20/20 | +818.4 [+554.4, +1,082.4] | 17/20 |
| Hopper | +240.3 [+47.9, +432.7] | 14/20 | +293.5 [+105.4, +481.7] | 15/20 | +132.2 [-84.1, +348.5] | 13/20 |
| Humanoid | +326.5 [+174.8, +478.1] | 16/20 | +96.9 [+2.8, +191.0] | 15/20 | +215.9 [+102.0, +329.7] | 17/20 |
| InvertedPendulum | +4.0 [-0.1, +8.1] | 15/20 | -5.0 [-15.0, +5.0] | 12/20 | -6.5 [-10.5, -2.5] | 6/20 |
| InvertedDoublePendulum | +74.8 [+3.7, +145.9] | 14/20 | -8.2 [-113.3, +96.9] | 12/20 | +114.9 [+51.4, +178.4] | 16/20 |
| Reacher | +0.0 [-0.5, +0.5] | 11/20 | -0.1 [-0.5, +0.3] | 10/20 | +0.7 [+0.4, +1.0] | 16/20 |
| Swimmer | +8.2 [+0.4, +16.0] | 12/20 | +39.3 [+30.9, +47.7] | 19/20 | +44.3 [+35.1, +53.5] | 19/20 |
| Walker2d | +386.7 [+90.6, +682.9] | 12/20 | +248.7 [-93.6, +591.0] | 11/20 | +193.2 [-47.2, +433.5] | 11/20 |

Table 16: MuJoCo TD3 return — Δ [95% CI], adjusted wins/20

| Environment | DR3O Δ [CI] | Adj. Win | Ellip Δ [CI] | Adj. Win | Adaptive Δ [CI] | Adj. Win |
|---|---|---|---|---|---|---|
| Ant | +577.4 [+358.5, +796.3] | 16/20 | +213.8 [-3.3, +430.9] | 12/20 | +313.0 [+64.7, +561.2] | 14/20 |
| HalfCheetah | +685.2 [+365.3, +1,005.1] | 16/20 | +785.6 [+349.4, +1,221.9] | 15/20 | +1,059.5 [+536.5, +1,582.4] | 15/20 |
| Hopper | +383.5 [+135.3, +631.6] | 14/20 | +258.9 [+98.3, +419.6] | 14/20 | +150.4 [-44.3, +345.1] | 11/20 |
| InvertedPendulum | -13.0 [-23.7, -2.3] | 5/20 | +7.7 [+5.5, +9.9] | 18/20 | -0.8 [-2.8, +1.2] | 12/20 |
| InvertedDoublePendulum | +21.2 [-85.9, +128.3] | 11/20 | +127.2 [+63.3, +191.1] | 17/20 | +2.6 [-123.1, +128.3] | 15/20 |
| Humanoid | -60.2 [-204.5, +84.1] | 9/20 | -73.4 [-256.3, +109.5] | 8/20 | +376.6 [+264.5, +488.7] | 18/20 |
| Reacher | +0.9 [+0.5, +1.3] | 17/20 | -0.0 [-0.5, +0.5] | 10/20 | +1.4 [+0.8, +2.0] | 18/20 |
| Swimmer | +0.7 [-7.2, +8.7] | 14/20 | +9.5 [+4.1, +14.9] | 15/20 | +9.7 [+0.1, +19.3] | 13/20 |
| Walker2d | +195.4 [+83.2, +307.7] | 15/20 | +292.4 [+119.8, +465.1] | 16/20 | +251.0 [+97.7, +404.3] | 14/20 |

Table 17: MuJoCo CrossQ return — Δ [95% CI], adjusted wins/20

| Environment | DR3O Δ [CI] | Adj. Win | Ellip Δ [CI] | Adj. Win | Adaptive Δ [CI] | Adj. Win |
|---|---|---|---|---|---|---|
| Ant | +441.5 [+339.8, +543.1] | 19/20 | +550.8 [+407.3, +694.3] | 19/20 | +307.0 [+162.3, +451.6] | 14/20 |
| HalfCheetah | +607.3 [-3.7, +1,218.3] | 12/20 | +834.3 [+234.6, +1,434.0] | 14/20 | +1,391.8 [+711.8, +2,071.9] | 15/20 |
| Hopper | +261.4 [+68.0, +454.8] | 14/20 | +289.8 [+120.2, +459.3] | 13/20 | +204.5 [-24.4, +433.5] | 15/20 |
| Humanoid | +527.9 [+326.6, +729.1] | 18/20 | +996.8 [+829.5, +1,164.1] | 20/20 | +414.4 [+273.2, +555.7] | 17/20 |
| InvertedPendulum | +5.3 [+3.6, +7.0] | 20/20 | +1.5 [-1.2, +4.2] | 10/20 | +5.6 [+3.1, +8.1] | 20/20 |
| InvertedDoublePendulum | -19.9 [-72.8, +33.0] | 10/20 | -76.4 [-145.1, -7.7] | 6/20 | +73.0 [+37.5, +108.5] | 17/20 |
| Reacher | +0.8 [+0.2, +1.4] | 13/20 | -0.6 [-1.1, -0.1] | 5/20 | +0.5 [+0.0, +1.0] | 13/20 |
| Swimmer | +4.7 [-5.8, +15.2] | 12/20 | -2.8 [-10.9, +5.3] | 10/20 | +9.1 [+0.1, +18.1] | 12/20 |
| Walker2d | +496.0 [+241.8, +750.3] | 15/20 | +386.2 [+163.8, +608.7] | 15/20 | +515.7 [+327.5, +703.9] | 18/20 |

**MuJoCo paired interpretation.** Adaptive Ellip is positive in 33 of 36 task–backbone comparisons and has a wholly positive paired interval in 27. DR3O and fixed Ellip are positive in 27 of 36 comparisons, with wholly positive intervals in 19 and 22, respectively. The strongest agreement among mean direction, interval sign, and adjusted wins occurs on HalfCheetah, Hopper, Humanoid, and Walker2d: the mean changes are positive, the intervals usually exclude zero, and most same-seed comparisons favor the regret-aware update. This agreement matters because it rules out the explanation that a suite mean is driven by a few unusually favorable seeds.

Table 18: MuJoCo TD7 return — Δ [95% CI], adjusted wins/20

| Environment | DR3O Δ [CI] | Adj. Win | Ellip Δ [CI] | Adj. Win | Adaptive Δ [CI] | Adj. Win |
|---|---|---|---|---|---|---|
| Ant | -22.2 [-340.5, +296.0] | 7/20 | +299.9 [-9.3, +609.1] | 14/20 | +304.1 [+78.9, +529.3] | 13/20 |
| HalfCheetah | +583.6 [+362.6, +804.5] | 18/20 | +877.4 [+581.8, +1,173.0] | 18/20 | +657.7 [+426.9, +888.4] | 18/20 |
| Hopper | +162.6 [+73.5, +251.6] | 15/20 | +130.3 [+54.1, +206.5] | 14/20 | +210.7 [+131.7, +289.7] | 17/20 |
| Humanoid | -17.2 [-86.6, +52.2] | 9/20 | +311.8 [+215.0, +408.6] | 18/20 | +239.8 [+121.0, +358.7] | 15/20 |
| InvertedPendulum | -3.1 [-7.2, +1.0] | 6/20 | +4.8 [+0.5, +9.1] | 13.5/20 | +5.0 [+1.1, +8.9] | 18/20 |
| InvertedDoublePendulum | +50.7 [-8.9, +110.3] | 15/20 | +8.8 [-60.5, +78.1] | 11/20 | +256.3 [+179.3, +333.3] | 19/20 |
| Reacher | +1.4 [+0.8, +2.0] | 16/20 | -0.6 [-1.6, +0.4] | 7/20 | -0.4 [-1.1, +0.3] | 7/20 |
| Swimmer | +2.4 [-1.7, +6.5] | 11/20 | +10.0 [+6.8, +13.2] | 18/20 | +4.7 [+1.8, +7.6] | 17/20 |
| Walker2d | +158.4 [-32.3, +349.0] | 12/20 | +344.8 [+182.2, +507.5] | 16/20 | +232.5 [+70.1, +394.8] | 13/20 |

The paired evidence reveals two favorable regimes. HalfCheetah, Hopper, Humanoid, and Walker2d combine large gains with strong seed-level agreement, while saturated pendulum tasks show smaller but often tightly concentrated changes. CrossQ HalfCheetah illustrates a third useful pattern: a large mean benefit accompanied by broader across-seed variation. Reading magnitude and seed wins together therefore preserves both practical improvement and repeatability.

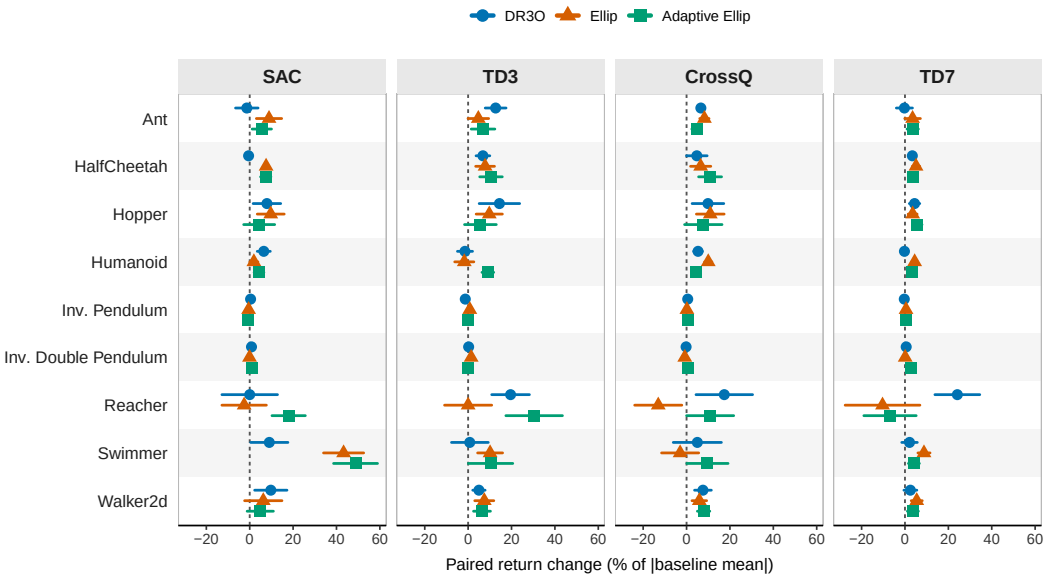



Figure 4: Task-level paired improvements on MuJoCo. Points show method-minus-baseline return differences and horizontal bars show paired 95% confidence intervals. For display, each difference and interval is divided by the absolute mean return of its matched baseline so that tasks with different return scales share one axis; this rescaling does not change whether an interval crosses zero. Positive values favor the regret-aware actor update, and the dashed line denotes no change. The preceding tables report the same results in the original return units.

**What the normalized interval plot adds.** Figure 4 makes cross-task coverage visible without replacing the raw-unit tables. Points to the right of zero indicate improvement, while bar width visualizes sensitivity to the paired seeds. The plot shows that adaptive gains extend across SAC, TD3, CrossQ, and TD7, with particularly dense positive coverage on the locomotion tasks. The figure is therefore a coverage and repeatability summary rather than a ranking by raw return magnitude.

### F.5.2 MetaWorld paired evidence

Return and success are paired separately because they answer different questions. We treat success as primary and use return to diagnose movement toward the goal. Zero-success tasks can still exhibit a nonzero return difference, but such a change is not counted as task completion.

**MetaWorld paired interpretation.** Using success, DR3O, fixed Ellip, and Adaptive Ellip improve 16, 15, and 16 of the 20 task–backbone pairs, respectively; their intervals are wholly positive in 13,

Table 19: MetaWorld 10 tasks (MT1) SAC return — Δ [95% CI], adjusted wins/20

| Environment | DR3O Δ [CI] | Adj. Win | Ellip Δ [CI] | Adj. Win | Adaptive Δ [CI] | Adj. Win |
|---|---|---|---|---|---|---|
| reach-v3 | +59.6 [-115.1, +234.3] | 11/20 | +58.2 [-125.9, +242.3] | 13/20 | -6.6 [-206.0, +192.8] | 12/20 |
| push-v3 | -6.9 [-66.5, +52.7] | 10/20 | -54.7 [-156.0, +46.6] | 10/20 | +73.6 [-24.4, +171.6] | 11/20 |
| pick-place-v3 | +0.7 [+0.3, +1.1] | 16/20 | +0.0 [-0.4, +0.4] | 11/20 | +0.9 [+0.4, +1.4] | 15/20 |
| door-open-v3 | +115.5 [+59.9, +171.1] | 16/20 | -44.9 [-107.1, +17.3] | 9/20 | +143.9 [+90.0, +197.8] | 17/20 |
| drawer-open-v3 | +121.6 [+61.7, +181.5] | 15/20 | +43.1 [-4.9, +91.1] | 13/20 | -43.1 [-90.9, +4.7] | 10/20 |
| drawer-close-v3 | +31.3 [+13.9, +48.7] | 17/20 | +60.2 [+35.2, +85.2] | 19/20 | +48.2 [+28.6, +67.7] | 18/20 |
| button-press-topdown-v3 | +251.8 [+46.5, +457.1] | 16/20 | +340.6 [+183.0, +498.3] | 16/20 | +296.2 [+162.5, +429.9] | 18/20 |
| peg-insert-side-v3 | +60.4 [-152.1, +272.8] | 9/20 | +89.5 [-93.0, +271.9] | 11/20 | +77.6 [-85.4, +240.6] | 9/20 |
| window-open-v3 | +297.7 [+66.5, +528.9] | 14/20 | +241.0 [-44.4, +526.4] | 11/20 | +205.8 [-79.0, +490.6] | 12/20 |
| window-close-v3 | +292.7 [+27.8, +557.6] | 14/20 | +230.2 [-41.5, +501.9] | 12/20 | +45.3 [-247.9, +338.5] | 11/20 |

Table 20: MetaWorld 10 tasks (MT1) TD3 return — Δ [95% CI], adjusted wins/20

| Environment | DR3O Δ [CI] | Adj. Win | Ellip Δ [CI] | Adj. Win | Adaptive Δ [CI] | Adj. Win |
|---|---|---|---|---|---|---|
| reach-v3 | +88.6 [-150.6, +327.8] | 10/20 | +147.7 [-100.3, +395.7] | 9/20 | -54.5 [-275.9, +166.9] | 8/20 |
| push-v3 | +324.9 [+172.9, +476.9] | 18/20 | +27.2 [-103.4, +157.8] | 7/20 | +260.2 [+105.2, +415.2] | 15/20 |
| pick-place-v3 | +1.2 [+0.8, +1.6] | 18/20 | +0.5 [+0.0, +1.0] | 14/20 | -1.0 [-1.5, -0.5] | 3/20 |
| door-open-v3 | +141.4 [+81.1, +201.7] | 16/20 | +71.7 [+24.8, +118.6] | 14/20 | +188.1 [+136.5, +239.7] | 18/20 |
| drawer-open-v3 | +175.6 [+129.9, +221.3] | 18/20 | -46.1 [-91.7, -0.5] | 6/20 | +126.5 [+66.6, +186.4] | 14/20 |
| drawer-close-v3 | +28.3 [+11.7, +45.0] | 17/20 | +44.8 [+20.7, +69.0] | 17/20 | +66.1 [+37.7, +94.5] | 18/20 |
| button-press-topdown-v3 | +267.6 [+104.9, +430.3] | 17/20 | +192.5 [+43.4, +341.7] | 15/20 | +297.0 [+127.0, +467.0] | 16/20 |
| peg-insert-side-v3 | +32.9 [-20.6, +86.3] | 8/20 | +20.4 [-91.3, +132.2] | 10/20 | +41.3 [-35.4, +118.0] | 13/20 |
| window-open-v3 | +71.5 [-211.6, +354.6] | 12/20 | +96.3 [-210.4, +403.0] | 10/20 | +320.0 [+89.5, +550.5] | 12/20 |
| window-close-v3 | -146.2 [-374.6, +82.2] | 9/20 | -226.1 [-459.4, +7.2] | 8/20 | -141.0 [-375.0, +93.0] | 8/20 |

Table 21: MetaWorld 10 tasks (MT1) SAC success rate — Δ [95% CI], adjusted wins/20

| Environment | DR3O Δ [CI] | Adj. Win | Ellip Δ [CI] | Adj. Win | Adaptive Δ [CI] | Adj. Win |
|---|---|---|---|---|---|---|
| reach-v3 | +0.50 [-0.10, +1.10] | 11/20 | +0.75 [-0.20, +1.70] | 11/20 | +0.50 [-0.57, +1.57] | 10.5/20 |
| push-v3 | +10.00 [+6.93, +13.07] | 18/20 | +5.25 [+1.84, +8.66] | 15/20 | +15.75 [+12.66, +18.84] | 19/20 |
| pick-place-v3 | +0.00 [+0.00, +0.00] | 10/20 | +0.00 [+0.00, +0.00] | 10/20 | +0.00 [+0.00, +0.00] | 10/20 |
| door-open-v3 | +8.25 [+4.27, +12.23] | 15.5/20 | -0.25 [-3.29, +2.79] | 8.5/20 | +7.75 [+4.12, +11.38] | 15/20 |
| drawer-open-v3 | +13.25 [+8.98, +17.52] | 16.5/20 | +11.25 [+7.65, +14.85] | 17/20 | +7.75 [+4.40, +11.10] | 15.5/20 |
| drawer-close-v3 | +1.3 [+0.2, +2.3] | 12/20 | +2.5 [+0.8, +4.2] | 13/20 | +2.0 [+0.8, +3.2] | 13.5/20 |
| button-press-topdown-v3 | +1.5 [+0.4, +2.6] | 13/20 | +2.0 [+0.8, +3.2] | 13.5/20 | +1.5 [+0.6, +2.4] | 13/20 |
| peg-insert-side-v3 | +16.00 [+12.53, +19.47] | 20/20 | +14.50 [+10.54, +18.46] | 19.5/20 | +18.50 [+15.76, +21.24] | 20/20 |
| window-open-v3 | +9.25 [+2.31, +16.19] | 13.5/20 | +5.25 [-0.38, +10.88] | 12/20 | +8.50 [+1.92, +15.08] | 13.5/20 |
| window-close-v3 | +1.00 [-0.73, +2.73] | 11/20 | +1.00 [-0.61, +2.61] | 11/20 | -2.50 [-4.72, -0.28] | 7.5/20 |

Table 22: MetaWorld 10 tasks (MT1) TD3 success rate — Δ [95% CI], adjusted wins/20

| Environment | DR3O Δ [CI] | Adj. Win | Ellip Δ [CI] | Adj. Win | Adaptive Δ [CI] | Adj. Win |
|---|---|---|---|---|---|---|
| reach-v3 | +4.75 [+0.29, +9.21] | 12.5/20 | +7.00 [+2.09, +11.91] | 14/20 | -3.50 [-6.51, -0.49] | 5.5/20 |
| push-v3 | +25.00 [+19.68, +30.32] | 20/20 | +10.25 [+6.31, +14.19] | 18.5/20 | +26.25 [+21.82, +30.68] | 20/20 |
| pick-place-v3 | +0.00 [+0.00, +0.00] | 10/20 | +0.00 [+0.00, +0.00] | 10/20 | +0.00 [+0.00, +0.00] | 10/20 |
| door-open-v3 | +10.00 [+6.45, +13.55] | 17.5/20 | +6.75 [+4.07, +9.43] | 17/20 | +14.50 [+10.59, +18.41] | 18.5/20 |
| drawer-open-v3 | +5.00 [+2.49, +7.51] | 14.5/20 | -2.00 [-4.02, +0.02] | 8/20 | +7.50 [+4.04, +10.96] | 15/20 |
| drawer-close-v3 | +0.5 [-0.4, +1.4] | 11/20 | +2.3 [+0.8, +3.7] | 13.5/20 | +0.5 [-1.1, +2.1] | 10/20 |
| button-press-topdown-v3 | +8.75 [+2.94, +14.56] | 15.5/20 | +7.25 [+3.21, +11.29] | 16/20 | +11.25 [+4.89, +17.61] | 16/20 |
| peg-insert-side-v3 | +13.75 [+10.44, +17.06] | 20/20 | +11.25 [+7.88, +14.62] | 18/20 | +19.25 [+14.72, +23.78] | 19.5/20 |
| window-open-v3 | -5.00 [-9.78, -0.22] | 8/20 | -4.25 [-9.39, +0.89] | 8/20 | +0.75 [-2.87, +4.37] | 10.5/20 |
| window-close-v3 | -0.75 [-4.63, +3.13] | 9/20 | +0.75 [-2.00, +3.50] | 11/20 | +4.00 [+1.22, +6.78] | 14.5/20 |

11, and 13 comparisons. The clearest adaptive evidence appears on push, door-open, drawer-open, button-press-topdown, and peg-insert-side. On push and peg insertion, the intervals are well separated from zero and the adjusted win counts reach 19–20 of 20, showing that the effect is typical across seeds rather than produced by isolated successes.

The paired return tables provide a complementary diagnostic. Return can improve while success remains unchanged, as on pick-place, and a strong success gain can accompany a modest shaped-return change, as on SAC push. Reach, drawer-close, and window tasks mix saturation with threshold sensitivity: small behavioral changes can flip completion, while already successful seeds cannot improve further. These patterns explain why success, rather than return alone, determines the MetaWorld reliability count.

### F.5.3 DMC PAIRED EVIDENCE

The DMC tables test whether the aggregate improvement reflects repeated within-seed gains rather than a small number of unusually strong runs.

Table 23: DMC SAC return — Δ [95% CI], adjusted wins/20

| Environment | DR3O Δ [CI] | Adj. Win | Ellip Δ [CI] | Adj. Win | Adaptive Δ [CI] | Adj. Win |
|---|---|---|---|---|---|---|
| ball_in_cup-catch | +2.7 [+1.8, +3.6] | 18/20 | +3.0 [+1.8, +4.2] | 15.5/20 | +2.6 [+1.8, +3.4] | 19/20 |
| cartpole-swingup | +36.2 [+27.2, +45.2] | 19/20 | +46.3 [+36.0, +56.7] | 20/20 | +48.3 [+40.2, +56.4] | 20/20 |
| cheetah-run | +91.5 [+72.1, +110.8] | 19/20 | +68.3 [+49.2, +87.4] | 19/20 | +77.1 [+64.5, +89.7] | 20/20 |
| finger-spin | +38.5 [+24.0, +52.9] | 16/20 | +31.7 [+16.2, +47.2] | 16/20 | +47.9 [+36.8, +59.0] | 20/20 |
| reacher-easy | -13.1 [-22.3, -3.9] | 6/20 | -3.4 [-13.9, +7.1] | 10/20 | +31.4 [+23.6, +39.2] | 20/20 |
| walker-walk | +4.6 [+2.1, +7.1] | 15/20 | +6.3 [+3.8, +8.9] | 17/20 | +6.7 [+4.8, +8.5] | 18/20 |

Table 24: DMC TD3 return — Δ [95% CI], adjusted wins/20

| Environment | DR3O Δ [CI] | Adj. Win | Ellip Δ [CI] | Adj. Win | Adaptive Δ [CI] | Adj. Win |
|---|---|---|---|---|---|---|
| ball_in_cup-catch | +3.1 [+1.8, +4.4] | 18/20 | +2.5 [+1.6, +3.4] | 17/20 | +0.4 [-0.5, +1.3] | 12/20 |
| cartpole-swingup | +30.5 [+26.5, +34.6] | 20/20 | +19.5 [+12.0, +27.0] | 16/20 | +60.6 [+55.3, +65.9] | 20/20 |
| cheetah-run | +62.2 [+41.8, +82.6] | 16/20 | +13.1 [+2.8, +23.4] | 13/20 | +57.8 [+42.6, +73.0] | 20/20 |
| finger-spin | +61.9 [+31.9, +91.9] | 16/20 | +37.5 [+14.9, +60.1] | 16/20 | +70.6 [+48.6, +92.5] | 19/20 |
| reacher-easy | +0.3 [-7.7, +8.3] | 10/20 | -5.2 [-13.3, +2.9] | 9/20 | +8.0 [+1.4, +14.6] | 12/20 |
| walker-walk | +4.6 [+1.7, +7.5] | 13/20 | +3.6 [+1.2, +6.1] | 15/20 | +6.2 [+2.9, +9.5] | 15/20 |

**DMC paired interpretation.** Adaptive Ellip improves all 12 SAC/TD3 task pairs and has a wholly positive interval in 11. On cartpole-swingup, cheetah-run, and finger-spin, 19–20 adjusted seed comparisons favor the method, so the large means are also highly repeatable. The remaining TD3 ball-in-cup change is only +0.4 because the baseline already scores 981.9. DR3O and fixed Ellip improve 11 and 10 pairs in mean direction, respectively; DR3O remains the strongest method on SAC cheetah-run, while the adaptive rule supplies the broadest coverage.

**Cross-suite synthesis.** The paired analysis supports the same ordering as the aggregate table in the main text while exposing its limits. Adaptive Ellip has the broadest positive coverage across all three suites: it is most consistently reliable when task difficulty and critic error vary during training. Fixed Ellip is competitive on several MuJoCo settings and incurs essentially no additional compute, suggesting that a constant conservative geometry is sufficient when critic quality is comparatively stable. DR3O achieves the best result on selected tasks, showing that the general regret correction can be powerful even when the simpler ellipsoidal parameterization is not uniformly optimal.

**Task-regime interpretation.** The cross-suite pattern is organized by task regime. Intermediate-difficulty tasks provide both usable critic information and enough headroom for the actor correction to matter; saturated tasks compress the numerical gain; and floor tasks such as MetaWorld pick-place lack successful behavior to refine. This explains more of the variation than backbone identity alone. CrossQ and TD7 are evaluated only on MuJoCo, so extending their MetaWorld and DMC coverage remains future work.

### F.6 Measured training time

All methods use the same environment-interaction budgets as their matched baselines. The measurements below separate total wall-clock time from host-clock time per replay update. Wall time includes setup, warmup, evaluation, and system effects; per-update time more directly reflects the additional actor-side computation. Neither DR3O nor the ellipsoidal variants change the number of environment steps.

#### F.6.1 One-million-step MuJoCo timing

These tables report the complete 1M-step runs for SAC, TD3, CrossQ, and TD7. Each cell contains total wall time in minutes and host time per replay update in milliseconds.

**MuJoCo timing interpretation.** The backbone determines most of the absolute cost: the baseline average ranges from 101.9 minutes for TD3 to 483.5 minutes for CrossQ. Relative to each matched baseline, DR3O adds 10.9–17.0% mean wall-clock time. Fixed Ellip ranges from $-3.2\%$ to +0.7%, and Adaptive Ellip from $-1.1\%$ to +2.2%. Their corresponding host-time changes per replay update are similarly small ($-3.7\%$ to +0.6% for fixed Ellip and $-1.3\%$ to +2.3% for Adaptive Ellip), so differences of this size are best interpreted as negligible method overhead under the packed-GPU schedule rather than as systematic speedups.

Table 25: MuJoCo SAC — measured training time at 1M steps, one replay update per step. Each entry: wall time (min) / host time per update (ms).

| Task | Baseline | DR3O | Ellip | Adaptive Ellip |
|---|---|---|---|---|
| Ant | 235.97 / 11.95 | 256.80 / 13.20 | 206.47 / 10.18 | 209.44 / 10.36 |
| HalfCheetah | 231.18 / 12.61 | 254.18 / 13.99 | 256.51 / 14.13 | 260.06 / 14.34 |
| Hopper | 233.02 / 12.71 | 259.35 / 14.29 | 244.35 / 13.39 | 247.67 / 13.59 |
| Humanoid | 261.49 / 12.05 | 299.16 / 14.31 | 255.32 / 11.68 | 255.57 / 11.70 |
| Swimmer | 233.20 / 12.72 | 255.70 / 14.07 | 248.37 / 13.63 | 251.67 / 13.83 |
| Walker2d | 228.65 / 11.51 | 248.65 / 12.71 | 241.32 / 12.27 | 245.18 / 12.50 |
| InvertedPendulum | 227.88 / 12.43 | 260.05 / 14.36 | 250.88 / 13.81 | 254.84 / 14.05 |
| InvertedDoublePendulum | 223.12 / 12.17 | 254.45 / 14.05 | 201.45 / 10.87 | 206.02 / 11.14 |
| Reacher | 216.52 / 11.81 | 232.35 / 12.76 | 200.69 / 10.86 | 206.15 / 11.19 |

Table 26: MuJoCo TD3 — measured training time at 1M steps, one replay update per step. Each entry: wall time (min) / host time per update (ms).

| Task | Baseline | DR3O | Ellip | Adaptive Ellip |
|---|---|---|---|---|
| Ant | 96.69 / 5.15 | 118.52 / 6.46 | 104.52 / 5.62 | 107.47 / 5.80 |
| HalfCheetah | 107.35 / 5.06 | 129.52 / 6.39 | 110.85 / 5.27 | 111.39 / 5.30 |
| Hopper | 98.43 / 5.07 | 113.26 / 5.96 | 86.76 / 4.37 | 89.28 / 4.52 |
| Swimmer | 91.68 / 4.87 | 106.51 / 5.76 | 89.01 / 4.71 | 93.29 / 4.97 |
| Walker2d | 95.19 / 5.00 | 108.86 / 5.82 | 85.52 / 4.42 | 88.86 / 4.62 |
| InvertedPendulum | 90.32 / 4.82 | 112.15 / 6.13 | 82.65 / 4.36 | 87.32 / 4.64 |
| InvertedDoublePendulum | 98.58 / 5.19 | 114.08 / 6.12 | 88.58 / 4.59 | 91.07 / 4.74 |
| Humanoid | 125.13 / 4.74 | 141.63 / 5.73 | 120.96 / 4.49 | 118.38 / 4.34 |
| Reacher | 113.65 / 4.94 | 125.82 / 5.67 | 120.82 / 5.37 | 120.14 / 5.33 |

Table 27: MuJoCo CrossQ — measured training time at 1M steps, one replay update per step. Each entry: wall time (min) / host time per update (ms).

| Task | Baseline | DR3O | Ellip | Adaptive Ellip |
|---|---|---|---|---|
| Ant | 505.87 / 25.64 | 549.70 / 28.27 | 468.37 / 23.39 | 487.45 / 24.53 |
| HalfCheetah | 474.83 / 24.86 | 525.16 / 27.88 | 459.83 / 23.96 | 459.39 / 23.93 |
| Hopper | 479.78 / 26.17 | 525.28 / 28.90 | 535.61 / 29.52 | 538.45 / 29.69 |
| Humanoid | 530.87 / 25.56 | 575.87 / 28.26 | 472.37 / 22.05 | 505.52 / 24.04 |
| Swimmer | 486.12 / 25.35 | 555.12 / 29.49 | 521.29 / 27.46 | 528.24 / 27.88 |
| Walker2d | 473.79 / 25.45 | 520.46 / 28.25 | 489.79 / 26.41 | 488.65 / 26.34 |
| InvertedPendulum | 452.83 / 24.70 | 521.83 / 28.84 | 452.16 / 24.66 | 436.26 / 23.71 |
| InvertedDoublePendulum | 463.67 / 24.83 | 527.50 / 28.66 | 408.50 / 21.52 | 400.40 / 21.03 |
| Reacher | 483.39 / 25.51 | 545.89 / 29.26 | 532.56 / 28.46 | 537.74 / 28.77 |

Table 28: MuJoCo TD7 — measured training time at 1M steps, one replay update per step. Each entry: wall time (min) / host time per update (ms).

| Task | Baseline | DR3O | Ellip | Adaptive Ellip |
|---|---|---|---|---|
| Ant | 369.98 / 14.98 | 395.48 / 16.51 | 368.81 / 14.91 | 367.75 / 14.85 |
| HalfCheetah | 342.89 / 16.18 | 384.56 / 18.68 | 373.39 / 18.01 | 376.85 / 18.22 |
| Hopper | 346.83 / 15.52 | 398.66 / 18.63 | 356.00 / 16.07 | 358.76 / 16.24 |
| Humanoid | 348.38 / 15.51 | 383.05 / 17.59 | 316.05 / 13.57 | 318.54 / 13.72 |
| Swimmer | 339.38 / 14.75 | 370.21 / 16.60 | 316.71 / 13.39 | 320.81 / 13.64 |
| Walker2d | 351.10 / 16.20 | 400.27 / 19.15 | 331.43 / 15.02 | 333.45 / 15.14 |
| InvertedPendulum | 328.29 / 14.91 | 374.46 / 17.68 | 356.79 / 16.62 | 362.99 / 16.99 |
| InvertedDoublePendulum | 330.52 / 16.43 | 360.69 / 18.24 | 326.35 / 16.18 | 332.12 / 16.53 |
| Reacher | 322.06 / 15.56 | 355.73 / 17.58 | 284.89 / 13.33 | 292.33 / 13.78 |

DR3O pays for additional candidate evaluation at each actor update; its host-time overhead ranges from 12.5% to 20.5%. CrossQ remains slowest in absolute minutes because its baseline update is already expensive. The close agreement between wall time and per-update host time supports separating the measurable DR3O premium from the near-baseline cost of the two ellipsoidal variants.

### F.6.2 Five-hundred-thousand-step MetaWorld and DMC timing

MetaWorld and DMC use the 500k-step endpoint. Presenting the two suites together makes it possible to compare method overhead under manipulation and control workloads without mixing their performance metrics.

Table 29: MetaWorld SAC — measured training time at 500k steps, one replay update per step. Each entry: wall time (min) / host time per update (ms).

| Task | Baseline | DR3O | Ellip | Adaptive Ellip |
|---|---|---|---|---|
| door-open-v3 | 161.45 / 8.27 | 173.87 / 9.76 | 163.12 / 8.47 | 161.93 / 8.33 |
| drawer-close-v3 | 149.40 / 9.07 | 160.73 / 10.43 | 152.90 / 9.49 | 159.44 / 10.27 |
| button-press-topdown-v3 | 153.90 / 9.16 | 160.90 / 10.00 | 155.98 / 9.41 | 159.49 / 9.83 |
| peg-insert-side-v3 | 173.63 / 8.87 | 186.71 / 10.44 | 182.46 / 9.93 | 174.55 / 8.98 |
| reach-v3 | 141.12 / 8.90 | 151.62 / 10.16 | 130.04 / 7.57 | 142.66 / 9.08 |
| push-v3 | 163.07 / 9.08 | 175.40 / 10.56 | 155.49 / 8.17 | 153.35 / 7.91 |
| pick-place-v3 | 168.10 / 8.37 | 180.43 / 9.85 | 170.60 / 8.67 | 165.62 / 8.07 |
| drawer-open-v3 | 156.61 / 9.04 | 170.11 / 10.66 | 149.03 / 8.13 | 150.80 / 8.34 |
| window-open-v3 | 155.07 / 9.05 | 168.40 / 10.65 | 157.40 / 9.33 | 160.16 / 9.66 |
| window-close-v3 | 151.15 / 8.53 | 165.23 / 10.22 | 140.65 / 7.27 | 146.00 / 7.91 |

Table 30: MetaWorld TD3 — measured training time at 500k steps, one replay update per step. Each entry: wall time (min) / host time per update (ms).

| Task | Baseline | DR3O | Ellip | Adaptive Ellip |
|---|---|---|---|---|
| door-open-v3 | 124.46 / 4.60 | 130.54 / 5.33 | 119.88 / 4.05 | 121.84 / 4.29 |
| drawer-close-v3 | 116.90 / 4.61 | 122.98 / 5.34 | 117.90 / 4.73 | 119.98 / 4.98 |
| button-press-topdown-v3 | 119.12 / 4.71 | 128.62 / 5.85 | 114.12 / 4.11 | 116.16 / 4.35 |
| peg-insert-side-v3 | 138.84 / 4.54 | 147.01 / 5.52 | 142.51 / 4.98 | 144.29 / 5.19 |
| reach-v3 | 110.44 / 4.56 | 118.27 / 5.50 | 110.02 / 4.51 | 112.21 / 4.77 |
| push-v3 | 123.76 / 4.74 | 135.09 / 6.10 | 123.84 / 4.75 | 125.81 / 4.99 |
| pick-place-v3 | 129.85 / 4.23 | 135.10 / 4.86 | 130.18 / 4.27 | 132.07 / 4.50 |
| drawer-open-v3 | 117.08 / 4.62 | 127.41 / 5.86 | 121.33 / 5.13 | 123.41 / 5.38 |
| window-open-v3 | 123.86 / 4.61 | 132.61 / 5.66 | 123.03 / 4.51 | 125.00 / 4.75 |
| window-close-v3 | 117.66 / 4.41 | 128.16 / 5.67 | 121.16 / 4.83 | 123.23 / 5.08 |

Table 31: DMC SAC — measured training time at 500k steps, one replay update per step. Each entry: wall time (min) / host time per update (ms).

| Task | Baseline | DR3O | Ellip | Adaptive Ellip |
|---|---|---|---|---|
| ball_in_cup-catch | 88.82 / 8.92 | 102.74 / 10.59 | 98.24 / 10.05 | 101.23 / 10.41 |
| cartpole-swingup | 92.62 / 8.94 | 100.87 / 9.93 | 95.04 / 9.23 | 96.65 / 9.42 |
| cheetah-run | 98.17 / 8.51 | 108.00 / 9.69 | 90.75 / 7.62 | 90.54 / 7.59 |
| finger-spin | 93.11 / 8.84 | 106.78 / 10.48 | 94.03 / 8.95 | 95.47 / 9.12 |
| walker-walk | 100.62 / 8.98 | 112.62 / 10.42 | 107.20 / 9.77 | 106.25 / 9.66 |
| reacher-easy | 87.35 / 8.60 | 95.60 / 9.59 | 81.10 / 7.85 | 84.66 / 8.28 |

Table 32: DMC TD3 — measured training time at 500k steps, one replay update per step. Each entry: wall time (min) / host time per update (ms).

| Task | Baseline | DR3O | Ellip | Adaptive Ellip |
|---|---|---|---|---|
| ball_in_cup-catch | 69.80 / 4.24 | 76.72 / 5.07 | 74.47 / 4.80 | 74.86 / 4.85 |
| cartpole-swingup | 70.64 / 4.76 | 78.64 / 5.72 | 67.56 / 4.39 | 68.04 / 4.45 |
| cheetah-run | 80.14 / 4.46 | 86.22 / 5.19 | 75.06 / 3.85 | 76.36 / 4.01 |
| finger-spin | 74.77 / 4.56 | 84.77 / 5.76 | 69.94 / 3.98 | 70.80 / 4.08 |
| walker-walk | 83.53 / 4.54 | 92.95 / 5.67 | 87.86 / 5.06 | 89.40 / 5.24 |
| reacher-easy | 69.53 / 4.60 | 77.53 / 5.56 | 73.78 / 5.11 | 74.14 / 5.15 |

**MetaWorld and DMC timing interpretation.** Across these runs, DR3O adds 6.9–11.8% mean wall-clock time. Fixed Ellip remains between $-1.1\%$ and $+1.0\%$, while Adaptive Ellip ranges from $+0.1\%$ to $+2.6\%$. The Adaptive Ellip host-time change per replay update ranges from approximately 0% to +5.9%, remaining far smaller than the DR3O premium. The adaptive controller therefore adds little end-to-end cost, requires no additional environment interaction, and leaves the replay and critic-update budgets unchanged.

**Performance–cost trade-off.** Taken together, the timing and performance tables identify three operating points. Fixed Ellip is the lowest-cost choice and remains strong on several MuJoCo tasks; it is appropriate when compute is constrained or a stable ambiguity scale is adequate. DR3O can deliver the largest task-specific gain, as on SAC cheetah-run, at a moderate computational premium. Adaptive Ellip provides the most consistent cross-suite improvement and is the strongest performance-oriented default, with mean wall-clock overhead below 2.7% in every reported suite–backbone average. Its broader positive paired coverage therefore does not require either additional

data or a substantial increase in end-to-end training time. This distinction is why the main text reports both aggregate performance and measured training time rather than treating all regret-aware updates as computationally interchangeable.