ADAPTIVE ENSEMBLE AGGREGATION FOR ACTOR-CRITICS

Anonymous authors
Paper under double-blind review

ABSTRACT

Ensembles are standard in off-policy actor-critic learning, but their efficacy depends critically on how they are aggregated. Many methods construct two ensemble-based reference estimators: a critic reference for bootstrapping and an actor reference for policy optimization. Existing approaches typically control these references through prescribed aggregation rules or externally specified hyperparameters. We introduce Adaptive Ensemble Aggregation (AEA), which calibrates both references based on actor-critic training dynamics. Extensive empirical evaluations study AEA across three settings: interactive learning, sample-efficient learning, and learning with external data collected by other behavior policies. Across multiple continuous-control tasks and external datasets of varying behavioral quality and coverage, AEA performs competitively against strong baselines while using the same AEA update throughout.

1 INTRODUCTION

Off-policy reinforcement learning (RL) has become a powerful framework for continuous control, with deep actor-critic methods forming a central class (Lillicrap et al., 2016; Haarnoja et al., 2017; Fujimoto et al., 2018). These methods jointly learn a *critic* (the Q-function) that estimates expected returns and an *actor* that optimizes its policy using the critic's estimates (Haarnoja et al., 2018a). A central difficulty is that policy improvement is conditioned on the reliability of the Q-value estimates; otherwise, the actor exploits estimation errors, and training becomes unstable. Off-policy bootstrapping with function approximation often leads to overestimation bias (Thrun & Schwartz, 1993; Fujimoto et al., 2018; Van Hasselt et al., 2018). To mitigate this, modern methods commonly use ensembles of critics and aggregate them conservatively to obtain more stable Q-value estimates (Fujimoto et al., 2018; Haarnoja et al., 2018a).

Although ensembles can yield negatively biased Q-value estimates (Van Hasselt, 2010; Chen et al., 2021), the stability benefits of conservative updates can come at the cost of reduced exploration and slower learning (Ciosek et al., 2019). Determining the optimum level of conservatism for the fastest adaptation remains a challenge. Existing methods balance critic stability and action exploration through aggregation choices and task-dependent hyperparameters. Early methods hard-coded conservatism via min-clipping: TD3 (Fujimoto et al., 2018) applied it only to the critic target while updating the actor through a single critic, and SAC (Haarnoja et al., 2018a) extended it symmetrically to both references, while subsequent works refined this balance. For instance, some sought to encourage exploration by introducing optimism (Ciosek et al., 2019) or by using bandit-based online alternation between conservatism and optimism (Moskovitz et al., 2021). Others refined critic conservatism by truncating quantiles (Kuznetsov et al., 2020) or learning a penalty based on ensemble disagreement (Cetin & Celiktutan, 2023). Most recently, Tasdighi et al. (2025) demonstrated the utility of decoupling actor optimism from critic pessimism. Yet choices such as truncation levels and optimism scales still require manual specification, and even adaptive methods retain task-dependent hyperparameters (Cetin & Celiktutan, 2023).

Searching for suitable critic ensemble aggregation hyperparameters is particularly challenging in high update-to-data (UTD) ratio regimes. While high UTD ratios can improve sample efficiency (Chen et al., 2021), repeated data reuse can increase overfitting to early experience (Nikishin et al., 2022). SAC-N's full-ensemble minimum (An et al., 2021) illustrates the difficulty: increasing ensemble size strengthens pessimism, and high-UTD ablations show that minimizing over too

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.

many critics can cause excessive underestimation and impair learning (Chen et al., 2021). REDQ mitigates this by minimizing over a small random subset, but its subset size remains a prescribed choice. On the other hand, incorporating external data collected by other behavior policies offers an opportunity to accelerate learning and reduce the need for new environment interactions by making use of previously collected experience (Ball et al., 2023). However, such data can come from very different behavior policies. Although incorporating them may seem natural and straightforward, selecting suitable aggregation parameters becomes more challenging. The appropriate balance between conservatism and optimism may then depend on the external data's behavioral quality and coverage, as well as the update regime. These challenges motivate three core questions:

- Can a single learning rule adapt critic and actor aggregation effectively across tasks and settings with different data availability and reuse?
- How can these aggregation parameters be learned directly from the actor-critic training signal without allowing large regression error signals to dominate their updates?
- What can the learned parameters tell us about how critic and actor conservatism and optimism vary across tasks, training stages, and data conditions?

We introduce Adaptive Ensemble Aggregation (AEA), a simple learning method that constructs critic and actor references in the form $\mu + \kappa\delta$, where μ is the ensemble mean and δ measures pairwise disagreement. Learning ensemble aggregation thus reduces to learning the critic and actor coefficients, $\bar{\kappa}$ and κ. The key to stable learning of these coefficients, and our answer to the second question, is a majority-vote calibration rule: each transition votes on whether to increase or decrease a coefficient, with no extra weight given to large regression errors. We derive this rule from disagreement-normalized absolute losses and establish convergence of policy evaluation and soft policy iteration in an idealized tabular setting with bounded coefficients and exact updates.

We evaluate AEA on MuJoCo and DeepMind Control Suite benchmarks across interactive learning, sample-efficient learning, and learning with external data. AEA uses the same update throughout, outperforming REDQ on most sample-efficient tasks and RLPD on most external-data conditions in final performance, while remaining competitive in interactive learning. Beyond performance, the learned trajectories address the third question by making the allocation of conservatism and optimism between the critic and actor interpretable. These choices vary across tasks and data conditions and evolve during training, offering a direct view of the non-stationary dynamics of actor-critic learning.

The paper is organized as follows. Section 2 introduces the problem setup and the necessary background, and Section 3 explains our algorithm AEA and its theoretical analysis. Section 4 reports the experiments and coefficient analysis, followed by discussion and conclusions in Section 5. The appendices contain extended related work, proofs, implementation details, additional results, and ablations.

2 Preliminaries and Problem Setup

2.1 Maximum-entropy actor-critic learning

We consider an infinite-horizon Markov Decision Process (MDP) defined by the tuple $\mathcal{M} = (\mathcal{S}, \mathcal{A}, p, r, \gamma)$ (Puterman, 2014), where $\mathcal{S}$ and $\mathcal{A}$ are continuous state and action spaces, $p(\cdot \mid s, a)$ is the transition kernel, $r(s, a)$ is the reward, and $\gamma \in (0, 1)$ is the discount factor. The agent's behavior is defined by a stochastic policy $\pi_\phi(\cdot \mid s)$ parameterized by ϕ, which maps a state $s \in \mathcal{S}$ to a probability density over actions $a \in \mathcal{A}$. We adopt the maximum-entropy RL framework (Ziebart, 2010; Haarnoja et al., 2018a), which augments cumulative discounted returns with an entropy bonus. The policy seeks to maximize

$$J(\pi_\phi) = \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t \left(r(s_t, a_t) - \alpha \log \pi_\phi(a_t \mid s_t)\right)\right], \tag{1}$$

where the expectation is over trajectories generated by the policy π_ϕ and the transition kernel p. The temperature $\alpha > 0$ balances the trade-off between the expected reward and the entropy bonus, and may be adjusted automatically during training (Haarnoja et al., 2018b). The entropy-regularized

action-value function $Q^\pi(s,a)$ is the expected discounted return obtained by taking action a in state s and then following π, with entropy bonuses included (Watkins & Dayan, 1992; Haarnoja et al., 2017).

A widely used class of methods for maximum-entropy RL is the soft actor-critic family (Haarnoja et al., 2018a), in which the agent jointly learns a Q-value function, the *critic*, and a policy, the *actor*. We introduce the framework following the generalized perspective of Tasdighi et al. (2025). The critic Q estimates the action-value function Q^{π_ϕ} of the policy by minimizing the Bellman residual (Silver et al., 2014; Bertsekas & Tsitsiklis, 1996; Haarnoja et al., 2018a):

$$\min_Q \mathbb{E}_{(s,a,r,s')\sim\mathcal{D}} \left[(Q(s,a) - y(s,a,r,s'))^2 \right], \tag{2}$$

using samples (s,a,r,s') drawn from a data distribution $\mathcal{D}$. We leave the construction of $\mathcal{D}$ unspecified here and discuss its possible data sources below. The bootstrapped regression target is

$$y(s,a,r,s') = r + \gamma \left[\overline{Q}(s',a') - \alpha \log \pi_\phi(a' \mid s') \right], \tag{3}$$

where $a' \sim \pi_\phi(\cdot \mid s')$ and the critic reference $\overline{Q}$ may differ from the critic Q and is commonly constructed from delayed target critics (Mnih et al., 2015). The actor is trained to maximize the expected action value:

$$\max_\phi \mathbb{E}_{s\sim\mathcal{D}, a\sim\pi_\phi(\cdot|s)} \left[\widetilde{Q}(s,a) - \alpha \log \pi_\phi(a \mid s) \right], \tag{4}$$

where $s \sim \mathcal{D}$ denotes the state marginal of the data distribution and $\widetilde{Q}$ is the actor reference used for training the actor. It may differ from both the critic Q and the critic reference $\overline{Q}$.

2.2 Off-policy data sources

These updates are off-policy because the data distribution $\mathcal{D}$ is not necessarily generated by the current actor. Our focus is online learning, in which the agent updates its policy while continuing to collect new experience through environment interaction. We write $\mathcal{B}_t^{\text{on}}$ for the replay buffer containing transitions collected by the learner up to time t. In some settings, a fixed external dataset $\mathcal{D}^{\text{off}}$, generated by an arbitrary behavior policy, is also available before the run begins. At each update, the sampling protocol induces $\mathcal{D}$ by drawing either from $\mathcal{B}_t^{\text{on}}$ alone or from both $\mathcal{B}_t^{\text{on}}$ and $\mathcal{D}^{\text{off}}$. We use B_t to denote the sampled minibatch for an actor-critic update. The objectives above apply to either construction, and AEA uses the same update rules without conditioning on the data regime or source labels. The experiments specify the exact construction of $\mathcal{D}$ and B_t for each evaluation setting.

2.3 Ensemble critic and actor references

To build ensembles, we instantiate the critic Q above with N active critics $\{Q_{\theta_i}\}_{i=1}^N$ and N target critics $\{Q_{\bar{\theta}_i}\}_{i=1}^N$, maintained as delayed copies of the active critics. The critic reference $\overline{Q}$ aggregates the target critics and is shared across critic updates, whereas the actor reference $\widetilde{Q}$ aggregates the active critics for policy improvement (Lillicrap et al., 2016; Mnih et al., 2015; Fujimoto et al., 2018). Each active critic is trained with Equation (2) using the shared critic reference. Section 3 defines how AEA constructs and adapts both references. In the idealized case in which all active and target critics equal Q^{π_ϕ}, these constructions make both references equal Q^{π_ϕ} (Sutton & Barto, 2018).

3 Adaptive Ensemble Aggregation

Separating the critic and actor references exposes two aggregation coefficients, but does not determine how they should be selected. Prior approaches span prescribed aggregation rules and adaptive controllers, but their aggregation behavior still depends on externally specified controls, such as coefficients selected through task-level performance comparisons or candidate values specified in advance (Tasdighi et al., 2025). AEA instead calibrates both reference coefficients from the actor-critic learning signal itself. This is difficult because the evolving Q-values both produce that signal and are changed by the resulting coefficients.

In this section, we explain how AEA learns the reference estimators on the fly. Section 3.1 defines these estimators in the form $\mu + \kappa\delta$, using standard ensemble statistics to construct μ and δ. Section 3.2 then addresses a central challenge in our design: updating the coefficient κ reliably and stably during training. Finally, Section 3.3 assembles the complete algorithm, presents idealized policy-evaluation and policy-iteration results, and discusses use beyond SAC, replay protocols, and computational cost.

3.1 AEA REFERENCE ESTIMATORS

AEA constructs the critic and actor references by shifting their corresponding ensemble means according to a pairwise-disagreement statistic. For the active critics, define

$$\mu(s,a) = \frac{1}{N}\sum_{i=1}^{N} Q_{\theta_i}(s,a), \qquad \delta(s,a) = \frac{1}{N(N-1)}\sum_{i<j}\left|Q_{\theta_i}(s,a) - Q_{\theta_j}(s,a)\right|. \tag{5}$$

Here, μ is the active-ensemble mean, while δ measures disagreement through absolute pairwise gaps. We write $\bar{\mu}$ and $\bar{\delta}$ for the corresponding statistics of the delayed target critics, obtained by replacing θ_i with $\bar{\theta}_i$. The disagreement statistic is a scaled form of the mean absolute pairwise difference (Yitzhaki, 2003); its linear dependence on each gap avoids further amplifying a large pairwise difference through squaring.

The two AEA references are

$$\overline{Q}_{\bar{\kappa}}(s,a) = \bar{\mu}(s,a) + \bar{\kappa}\,\bar{\delta}(s,a), \qquad \widetilde{Q}_{\kappa}(s,a) = \mu(s,a) + \kappa\,\delta(s,a). \tag{6}$$

Here, $\bar{\kappa}$ and κ are the learnable critic-reference and actor-reference coefficients, respectively. A zero coefficient recovers the corresponding ensemble mean. When the corresponding disagreement is positive, positive and negative coefficients place the reference above and below that mean. For $\kappa > 0$, the actor objective also favors actions with greater critic disagreement when their ensemble means are equal. Throughout the paper, conservatism and optimism are defined relative to the corresponding ensemble mean. This parameterization exposes the two reference choices separately; Section 3.2 defines how AEA updates these coefficients.

The following proposition relates this parameterization to the reference choices of SAC and REDQ (Chen et al., 2021), and the proof is provided in Section B.1.

Proposition 1 (Connection to SAC and REDQ)**.** *Under the ensemble statistics in Equation (5), the references in Equation (6) recover the following choices:*

- *(i) For $N = 2$, the coefficient pair $(\bar{\kappa}, \kappa) = (-1, -1)$ exactly recovers the critic and actor references of twin-critic SAC (Haarnoja et al., 2018b).*
- *(ii) For any $N \geq 2$, the pair $(\bar{\kappa}, \kappa) = (-1, 0)$ recovers REDQ's conditional expected critic reference for $M = 2$ and its exact actor reference (Chen et al., 2021). The expectation is conditional on the target-ensemble outputs.*

3.2 LEARNING THE AGGREGATION PARAMETERS BY MAJORITY VOTE

As noted in the introduction to Section 3, learning the reference coefficients $\bar{\kappa}$ and κ on the fly is challenging. Although the coefficients are scalars, their update signal must be formed from Q-value estimates and Bellman regression targets that can change sharply during actor-critic learning. An update proportional to the regression error can turn a temporary error in the value estimates into a large coefficient change that feeds back into learning, especially when data are reused (Fujimoto et al., 2018; Nikishin et al., 2022). Earlier methods also use explicit controls for uncertainty-sensitive references: OAC constrains its optimistic exploration policy through a KL budget (Ciosek et al., 2019), while USAC selects its reference coefficients through task-level performance comparisons (Tasdighi et al., 2025). We now explain how AEA is designed to keep the coefficient updates stable.

For the following coefficient-update derivations, assume $\delta(s,a) > 0$ and $\bar{\delta}(s',a') > 0$ for every transition in B_t and its sampled next action a'. For the absolute calibration losses below, ∇ denotes a selected subgradient, with $\operatorname{sign}(0) = 0$.

Critic-reference calibration. We compare the critic target with the actor reference because these are the values used for critic bootstrapping and policy improvement, respectively; their agreement provides the calibration signal. We introduce the normalized objective:

$$\mathcal{L}_{\text{crit}}(\bar{\kappa}) = \mathbb{E}_{B_t}\left[|\widetilde{Q}_\kappa(s,a) - y_{\bar{\kappa}}(s,a,r,s')|/\bar{\delta}(s',a')\right]. \tag{7}$$

Here, $\mathbb{E}_{B_t}$ denotes the average over the minibatch introduced in Section 2.2, and $y_{\bar{\kappa}}$ is the regression target in Equation (3). We update $\bar{\kappa}$ via subgradient descent on this objective. Given that $\partial y_{\bar{\kappa}}/\partial\bar{\kappa} = \gamma\bar{\delta}(s',a')$, the selected subgradient with respect to $\bar{\kappa}$ simplifies as follows:

$$\nabla_{\bar{\kappa}}\mathcal{L}_{\text{crit}}(\bar{\kappa}) = \mathbb{E}_{B_t}\left[\frac{-\operatorname{sign}(\widetilde{Q}_\kappa(s,a) - y_{\bar{\kappa}})}{\bar{\delta}(s',a')} \cdot \frac{\partial y_{\bar{\kappa}}}{\partial\bar{\kappa}}\right] = -\gamma\mathbb{E}_{B_t}\left[\operatorname{sign}(\widetilde{Q}_\kappa(s,a) - y_{\bar{\kappa}})\right]. \tag{8}$$

We note that the update direction is determined by a scale-invariant majority vote: whether the current target $y_{\bar{\kappa}}$ lies above or below the active reference $\widetilde{Q}_\kappa$ across the minibatch.

Actor-reference calibration. As discussed in Section 2.1, $\widetilde{Q}_\kappa$ shapes the actor's objective in Equation (4) and can serve a different role from the critic reference $\overline{Q}_{\bar{\kappa}}$. We therefore update the actor-reference coefficient κ through the following objective:

$$\min_{\kappa}\mathcal{L}_{\text{act}}(\kappa) = \min_{\kappa}\mathbb{E}_{B_t}\left[|\widetilde{Q}_\kappa(s,a) - y_{\bar{\kappa}}(s,a,r,s')|/\delta(s,a)\right], \tag{9}$$

yielding the selected subgradient

$$\nabla_{\kappa}\mathcal{L}_{\text{act}}(\kappa) = \mathbb{E}_{B_t}\left[\operatorname{sign}(\widetilde{Q}_\kappa(s,a) - y_{\bar{\kappa}})\right]. \tag{10}$$

The two coefficient updates have opposite directions for the same regression error: the $\bar{\kappa}$ update moves the regression target toward the actor reference $\widetilde{Q}_\kappa$, while the κ update moves the actor reference toward the regression target.

Remark (Stability). Each transition contributes only the sign of its regression error to the coefficient update. Extreme regression errors therefore do not receive extra weight; the subgradients with respect to $\bar{\kappa}$ and κ have magnitudes at most γ and 1, respectively.

The stability remark describes the influence of each sample on a coefficient update. It does not yet tell us which coefficient values minimize the calibration losses over a minibatch. Normalizing by disagreement gives each transition an equal vote in coefficient space; without it, the loss would select a disagreement-weighted median. The following informal proposition answers this second question, and the formal statement and proof are in Section B.2.

Proposition 2 (Median solutions; informal). Assume all other quantities are fixed. Minimizing $\mathcal{L}_{\text{crit}}(\bar{\kappa})$ selects a median of the $\bar{\kappa}$ values that give zero regression error on individual transitions. The same holds for κ under $\mathcal{L}_{\text{act}}$.

3.3 Complete actor-critic update

The preceding subsections defined AEA's critic and actor references and described how their coefficients are updated. We now combine these components in the complete actor-critic update. We then examine its convergence under idealized policy evaluation and policy iteration, before discussing its applicability beyond SAC, data sampling, and computational overhead.

In the SAC framework introduced in Section 2, AEA uses the critic reference $\overline{Q}_{\bar{\kappa}}$ to form the regression target

$$y_{\bar{\kappa}} = r + \gamma\left[\overline{Q}_{\bar{\kappa}}(s',a') - \alpha\log\pi_\phi(a' \mid s')\right]. \tag{11}$$

Each active critic fits this target using Equation (2), while the actor maximizes $\widetilde{Q}_\kappa$ together with the entropy bonus in Equation (4). In each update, the critics and both coefficient losses use the same replay minibatch B_t. We tune the temperature α automatically following Haarnoja et al. (2018b). Algorithm 1 summarizes the complete procedure, with G complete updates per environment interaction, where G denotes the update-to-data ratio. We clip both coefficients to $[-1,1]$ after each update to satisfy the boundedness condition in Theorem 3, using a range that contains the SAC and REDQ coefficient choices in Proposition 1.

Algorithm 1 Adaptive Ensemble Aggregation

Require: Ensemble size N, UTD ratio G, target-update rate τ, and step sizes $\eta_\theta, \eta_{\bar{\kappa}}, \eta_\kappa, \eta_\phi$

Initialize: critic networks $\{Q_{\theta_i}\}_{i=1}^N$ and actor π_ϕ with random parameters; target parameters $\bar{\theta}_i \leftarrow \theta_i$ for all $i \in [N]$; coefficients $(\bar{\kappa}, \kappa)$ and temperature α

Initialize the online replay buffer $\mathcal{B}_0^{\text{on}} = \emptyset$; a fixed external dataset $\mathcal{D}^{\text{off}}$ may also be available

for each environment interaction t **do**

  Take action $a_t \sim \pi_\phi(\cdot \mid s_t)$, observe reward r_t and next state s_{t+1}, and add (s_t, a_t, r_t, s_{t+1}) to the online replay buffer $\mathcal{B}_t^{\text{on}}$

  for $g = 1, \ldots, G$ **do**

    Sample B_t using the replay protocol, and sample next actions $a' \sim \pi_\phi(\cdot \mid s')$

    $\theta_i \leftarrow \theta_i - \eta_\theta \nabla_{\theta_i} \left\{ \frac{1}{|B_t|} \sum_{B_t} \left(Q_{\theta_i}(s,a) - y_{\bar{\kappa}}(s,a,r,s')\right)^2 \right\}, \forall i \in [N]$ ▷ critic

    When scheduled, $\bar{\theta}_i \leftarrow \tau\theta_i + (1-\tau)\bar{\theta}_i, \forall i \in [N]$ ▷ target critic

    $\bar{\kappa} \leftarrow \bar{\kappa} - \eta_{\bar{\kappa}} \nabla_{\bar{\kappa}} \left\{ \frac{1}{|B_t|} \sum_{B_t} |\widetilde{Q}_\kappa(s,a) - y_{\bar{\kappa}}(s,a,r,s')| / \bar{\delta}(s',a') \right\}$ ▷ AEA: critic

    $\kappa \leftarrow \kappa - \eta_\kappa \nabla_\kappa \left\{ \frac{1}{|B_t|} \sum_{B_t} |\widetilde{Q}_\kappa(s,a) - y_{\bar{\kappa}}(s,a,r,s')| / \delta(s,a) \right\}$ ▷ AEA: actor

    Sample new actor actions $a_\phi(s) \sim \pi_\phi(\cdot \mid s)$ for the states in B_t

    $\phi \leftarrow \phi + \eta_\phi \nabla_\phi \left\{ \frac{1}{|B_t|} \sum_{B_t} \left(\widetilde{Q}_\kappa(s, a_\phi(s)) - \alpha \log \pi_\phi(a_\phi(s) \mid s) \right) \right\}$ ▷ policy

    Update the entropy temperature α using the standard SAC temperature update

  end for

end for

Convergence guarantees. As in the standard policy iteration analyses, we consider an idealized tabular setting. We first establish convergence of policy evaluation with bounded aggregation coefficients in Theorem 3, then show in Corollary 4 that combining this evaluation procedure with exact policy improvement yields convergence of policy iteration. Since the full statements include several standard but lengthy technical conditions, we present informal versions here and defer the formal statements and proofs to Sections B.3 and B.4.

We first define some notations. At evaluation iteration t, let $Q_{i,t}$ and $\bar{Q}_{i,t}$ denote the active and target critics, and let $\bar{\kappa}_t$ and κ_t denote the reference coefficients. The convergence of policy evaluation is established as follows:

Theorem 3 (Policy evaluation; informal). Consider exact synchronous tabular evaluation of a fixed policy π_ϕ at a fixed temperature α, using a shared Bellman target and target-critic smoothing as specified in the formal statement. If the coefficients $\bar{\kappa}_t$ and κ_t remain bounded, then, for every critic i,

$$\lim_{t\to\infty} \|Q_{i,t} - Q^{\pi_\phi}\|_\infty = 0, \qquad \lim_{t\to\infty} \|\bar{Q}_{i,t} - Q^{\pi_\phi}\|_\infty = 0.$$

Both the critic and actor references also converge to Q^{π_ϕ}.

The shared Bellman target and target-critic smoothing in Theorem 3 reflect AEA's critic updates. Under exact synchronous tabular policy evaluation, critic disagreement vanishes, so bounded coefficients need not converge for their effect on the references to vanish. Together with the Bellman contraction, this gives the usual policy-evaluation limit Q^{π_ϕ} for the critics and references.

Since the AEA references converge to the evaluated policy's value function, they are compatible with the standard soft policy iteration guarantee under exact policy improvement (Haarnoja et al., 2018a). The formal statement and proof are in Section B.4.

Corollary 4 (Policy iteration; informal). In a finite tabular MDP, suppose each policy is evaluated to convergence as in Theorem 3, followed by exact soft policy improvement at a fixed temperature. Then $Q^{\pi_{m+1}} \geq Q^{\pi_m}$ pointwise, $Q^{\pi_m} \to Q^*$, and $\pi_m \to \pi^*$, where Q^* and π^* are the soft-optimal action-value function and policy.

These results establish convergence under exact tabular evaluation with value-space target smoothing and, for policy iteration, exact policy improvement at a fixed temperature. The practical procedure in Algorithm 1 instead updates the critics, coefficients, and actor from replay minibatches

Table 1: Interquartile mean (IQM) of final return (Agarwal et al., 2021) in the sample-efficient setting, with a UTD ratio of 20 and an interaction budget of 300K. Results are based on 10 independent runs. ± denotes standard deviation over seeds. Highest mean is bold. The same results are reproduced in Table 9.

MuJoCo-v5	AEA (ours)	REDQ (Chen et al., 2021)	SAC-N (An et al., 2021)	DRND (Yang et al., 2024)	DSAC (Ma et al., 2025)
Ant	4744±260	**4828**±398	1±1	797±560	776±177
HalfCheetah	**9661**±395	8678±813	5425±191	7892±1180	9059±1191
Hopper	**3555**±32	3415±153	2063±1034	3015±840	2640±783
Humanoid	5376±87	**5404**±61	5341±124	2853±1684	908±171
Walker2d	4652±267	4749±170	1035±830	5054±414	**5362**±349
DMC					
Cheetah-run	827±9	866±6	635±26	**873**±21	753±37
Hopper-hop	**86**±75	54±81	0±1	3±7	54±58
Hopper-stand	**935**±16	109±268	0±1	251±339	39±58
Humanoid-run	**144**±5	131±8	1±1	1±0	17±32
Humanoid-stand	**665**±55	559±31	9±1	213±144	284±190
Humanoid-walk	**484**±14	454±35	2±1	166±160	2±0
Quadruped-run	**856**±19	847±24	43±52	224±379	536±80
Quadruped-walk	**947**±14	932±16	112±90	184±285	555±371
Walker-run	**741**±27	649±56	258±84	689±93	658±86

without running policy evaluation to convergence; it also uses neural function approximation and temperature adaptation.

Beyond maximum-entropy framework. Although we present AEA in the maximum-entropy framework, its reference construction and coefficient updates can also be used in other ensemble actor-critic methods. In particular, for actor-critic learning without entropy regularization, the target becomes $y_{\bar{\kappa}} = r + \gamma \overline{Q}_{\bar{\kappa}}(s', a')$, and the same coefficient updates apply.

Data minibatches. As discussed in Section 2.2, the minibatch B_t can be constructed using different replay protocols, which we specify in the experiments section. The settings differ not only in where transition data come from but also in how often they are reused. We emphasize that the same AEA update rules apply to all settings without using setting-specific labels.

Computational overhead. AEA uses an ensemble actor-critic backbone, as in REDQ and RLPD (Chen et al., 2021; Ball et al., 2023), and learns two additional scalar parameters without introducing auxiliary networks. Computing ensemble disagreement requires $O(N^2)$ scalar arithmetic per state-action pair. For $N = 10$, the statistic averages differences over 45 distinct critic pairs, a small arithmetic cost relative to neural-network forward and backward passes. Network evaluations remain the main computational cost; implementation details and running times are provided in the experimental-setup appendix.

4 Experiments

We evaluate AEA on Gymnasium's MuJoCo benchmarks (Towers et al., 2024) and the DeepMind Control Suite (DMC) (Tassa et al., 2018) to study its performance as data availability and reuse change. The same AEA update is used across three learning settings, with implementation details provided in Section C. Each baseline comparison uses 10 independent runs. We evaluate the deterministic policy given by the mean of the learned action distribution.

Experimental settings. We recall that N denotes the critic ensemble size and G denotes the update-to-data (UTD) ratio, the number of updates per environment interaction. In interactive learning, AEA uses $(N, G) = (2, 1)$ for 1M environment interactions, drawing each minibatch B_t from the online replay buffer $\mathcal{B}_t^{\text{on}}$. Sample-efficient learning uses $(N, G) = (10, 20)$ over 300K interactions, with minibatches drawn from the same online replay source. Learning with external data uses

Table 2: Interquartile mean (IQM) of final return (Agarwal et al., 2021) on Ant, HalfCheetah, Hopper, and Humanoid with external data and an online interaction budget of 1M. Results are based on 10 independent runs. ± denotes the standard deviation across seeds. Highest mean is bold. These are selected results from Table 12.

MuJoCo-v5	AEA (ours)	RLPD (Ball et al., 2023)	Cal-QL (Nakamoto et al., 2023)	SACfD (Ball et al., 2023)
ant-expert	7116±137	**7417**±36	5791±1106	6221±111
ant-medium	7080±23	**7162**±55	−1784±366	6137±142
ant-simple	6970±67	**7214**±43	4630±71	5202±103
halfcheetah-expert	**17127**±40	16725±27	9009±302	1630±469
halfcheetah-medium	**15769**±44	14330±47	12695±416	13184±95
halfcheetah-simple	**13467**±116	12292±47	8108±49	11153±66
hopper-expert	**3819**±180	3640±242	1789±229	2069±148
hopper-medium	3258±243	**3754**±181	3474±94	2858±190
hopper-simple	2793±333	**3354**±257	3154±18	3197±142
humanoid-expert	**9933**±153	5375±16	6762±597	191±57
humanoid-medium	**8746**±218	5454±30	7149±237	346±26
humanoid-simple	**7249**±426	6988±82	5780±46	5671±24

$(N, G) = (10, 20)$ for 1M interactions, with each minibatch containing equal numbers of transitions from $\mathcal{B}_t^{\text{on}}$ and a fixed external dataset $\mathcal{D}^{\text{off}}$. These sampling protocols define the data distribution $\mathcal{D}$ in each setting. Table 3 summarizes the benchmarks and interaction protocols.

Interactive and sample-efficient settings. SAC (Haarnoja et al., 2018a) and REDQ (Chen et al., 2021) provide the fixed aggregation choices in Proposition 1, SAC-N (An et al., 2021) tests full-ensemble minimization, and DRND (Yang et al., 2024) and DSAC (Ma et al., 2025) provide strong baselines based on distillation-based uncertainty estimation and distributional value learning, respectively. In the sample-efficient setting, AEA achieves the highest IQM on a majority of tasks, with particularly strong results on DMC (Table 1). It improves over REDQ on most tasks and over SAC-N throughout the benchmark under matched ensemble sizes and update rates. In the interactive setting, AEA remains competitive with SAC, REDQ, DRND, and DSAC. This standard twin-critic, single-update setting checks AEA under limited data reuse; to leave room for the high-reuse and external-data comparisons, we provide its complete results in Section D.1. Together, these results demonstrate the effectiveness of the same aggregation update under both heavy and limited data reuse. Complete sample-efficient results are in Section D.2.

Learning with external data. AEA also enjoys strong performance when incorporating external data, using the same coefficient update without offline pretraining. In comparison to RLPD, which uses the same ensemble size and update rate, AEA achieves higher IQM across all HalfCheetah and Humanoid dataset qualities, with especially significant gains on Humanoid with medium and expert data (Table 2). We also compare with Cal-QL (Nakamoto et al., 2023), a strong offline-to-online baseline that performs 1M offline pretraining gradient updates before online fine-tuning, and SACfD (Ball et al., 2023), which initializes the replay buffer with external data and applies SAC without offline pretraining. Both use $(N, G) = (2, 1)$ during online learning. Complete task-level results are in Section D.3.

Learned coefficient trajectories. Figure 1 shows how AEA learns different aggregation parameters from the common initialization $(\bar{\kappa}, \kappa) = (-0.8, 0)$; initialization sensitivity is examined in Section E.3. The actor coefficient κ rises on HalfCheetah but stays near zero on Hopper after the initial transient. In both tasks, the critic coefficient $\bar{\kappa}$ settles near -0.9 in the interactive setting, while sample-efficient trajectories remain less negative and more variable. The signs of these coefficients clarify the different roles of the two references: a negative critic coefficient lowers the bootstrap reference relative to its ensemble mean, while a positive actor coefficient raises the reference used for policy improvement. On Hopper, the critic coefficient remains conservative while the actor coefficient stays nearly neutral. AEA thus learns distinct aggregation choices for the critic and actor, with their allocation varying across tasks.

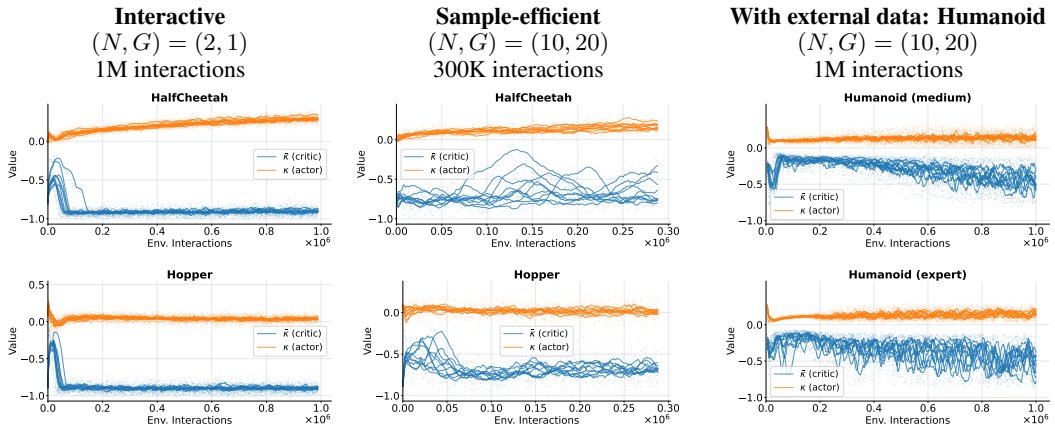

Figure 1: Selected AEA coefficient trajectories. The first two columns show HalfCheetah (top) and Hopper (bottom); the third shows Humanoid with medium (top) and expert (bottom) external data. Blue and orange denote the critic and actor coefficients $\bar{\kappa}$ and κ, respectively. Points show recorded values; lines show per-seed moving averages across 10 runs.

The external-data panels show further differences from the interactive and sample-efficient examples. On Humanoid, for example, the critic coefficients generally stay closer to zero and continue changing over the 1M-interaction training period, while the actor coefficients remain positive and comparatively steady. Together, these examples show how the same AEA update learns different aggregation choices across tasks and settings, ranging from nearly settled coefficients to continuing adjustment during training. Further examples in Section E.1 show that coefficient trajectories can also differ across external datasets within the same environment and update protocol.

Fixed-coefficient ablation. Two natural questions arise from AEA: do we need to keep adapting the coefficients during training, and can AEA's calibration mechanism find useful fixed coefficient pairs? For each task condition, we choose a pair in hindsight by averaging AEA's late-training coefficients and keep it fixed throughout independent runs. These pairs can perform well against standard baselines, showing that AEA's calibration can identify useful constants. Continuing adaptation gives better performance than the selected fixed pair on some tasks. Section E.2 provides the selection procedure, the resulting pairs, and a detailed discussion of the results.

5 Discussion and Conclusion

AEA brings ensemble aggregation into the actor-critic learning process. Its disagreement-normalized majority-vote rule learns the critic and actor reference coefficients from the existing training signal while limiting each transition's influence on their updates. Our idealized tabular analysis establishes convergence of policy evaluation and soft policy iteration. Empirically, the same update achieves strong results in sample-efficient learning and learning with external data, while remaining competitive in the interactive setting.

The learned trajectories also provide an interpretable view of aggregation during training. Critic and actor coefficients follow distinct paths, with some approaching nearly constant values and others continuing to change across training stages. The fixed-pair analysis further shows that AEA can discover useful task-specific constants, although holding these choices fixed does not preserve adaptive performance on every task. AEA allows aggregation to respond to the evolving training process and makes the resulting choices available for inspection.

AI USE STATEMENT

In this work, we used generative AI tools to help refine theoretical statements and their assumptions, check and revise proof details, and provide feedback on the interpretation of experimental results. The research idea, the algorithm, and overall proof strategy were developed by the authors. We did not use generative AI to generate the reported experimental results. Additionally, we used generative AI tools to improve the clarity of the manuscript and to help identify relevant literature. The authors have carefully reviewed the AI-assisted text, checked suggested references, and checked the revised theoretical statements, assumptions, and proof details. We take responsibility for the final content of this work, including all text, claims, and artifacts produced with the assistance of generative AI.

ETHICS STATEMENT

To the best of the authors' knowledge, this work has no specific ethical concerns. The experiments use publicly available benchmark datasets, and do not involve human participants, personal data, or sensitive information.

REPRODUCIBILITY STATEMENT

Section 3 and Algorithm 1 describe AEA and its update rules. Appendix C documents the benchmarks, external dataset identifiers, implementation details, hyperparameters, and evaluation protocols. Our implementation will be made publicly available upon acceptance. Appendix B provides proofs of the theoretical results.

REFERENCES

Zaheer Abbas, Rosie Zhao, Joseph Modayil, Adam White, and Marlos C Machado. Loss of plasticity in continual deep reinforcement learning. In *Conference on Lifelong Learning Agents (CoLLAs)*, 2023.

Rishabh Agarwal, Max Schwarzer, Pablo Samuel Castro, Aaron C Courville, and Marc Bellemare. Deep reinforcement learning at the edge of the statistical precipice. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

Gaon An, Seungyong Moon, Jang-Hyun Kim, and Hyun Oh Song. Uncertainty-based offline reinforcement learning with diversified q-ensemble. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

Oron Anschel, Nir Baram, and Nahum Shimkin. Averaged-dqn: Variance reduction and stabilization for deep reinforcement learning. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2017.

Philip J Ball, Laura Smith, Ilya Kostrikov, and Sergey Levine. Efficient online reinforcement learning with offline data. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2023.

Gulcin Baykal, Abdullah Akgül, Manuel Haussmann, Bahareh Tasdighi, Nicklas Werge, Yi-Shan Wu, and Melih Kandemir. Objectrl: An object-oriented reinforcement learning codebase. *arXiv preprint arXiv:2507.03487*, 2025.

Dimitri Bertsekas and John N Tsitsiklis. *Neuro-dynamic programming*. Athena Scientific, 1996.

James Bradbury, Roy Frostig, Peter Hawkins, Matthew James Johnson, Yash Katariya, Chris Leary, Dougal Maclaurin, George Necula, Adam Paszke, Jake VanderPlas, Skye Wanderman-Milne, and Qiao Zhang. JAX: composable transformations of Python+NumPy programs, 2018.

Edoardo Cetin and Oya Celiktutan. Learning pessimism for reinforcement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2023.

Richard Y Chen, Szymon Sidor, Pieter Abbeel, and John Schulman. Ucb exploration via q-ensembles. *arXiv preprint arXiv:1706.01502*, 2017.

Xinyue Chen, Che Wang, Zijian Zhou, and Keith W Ross. Randomized ensembled double Q-learning: Learning fast without a model. In *International Conference on Learning Representations (ICLR)*, 2021.

Kamil Ciosek, Quan Vuong, Robert Loftin, and Katja Hofmann. Better exploration with optimistic actor critic. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

Leif Döring, Benedikt Wille, Maximilian Birr, Mihail Bîrsan, and Martin Slowik. ADDQ: Adaptive distributional double Q-learning. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2025.

Nicolai Dorka, Tim Welschehold, Joschka Boedecker, and Wolfram Burgard. Adaptively calibrated critic estimates for deep reinforcement learning. *IEEE Robotics and Automation Letters*, 2023.

Scott Fujimoto, Herke van Hoof, and David Meger. Addressing function approximation error in actor-critic methods. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2018.

Tuomas Haarnoja, Haoran Tang, Pieter Abbeel, and Sergey Levine. Reinforcement learning with deep energy-based policies. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2017.

Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2018a.

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*, 2018b.

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

Arsenii Kuznetsov, Pavel Shvechikov, Alexander Grishin, and Dmitry Vetrov. Controlling overestimation bias with truncated mixture of continuous distributional quantile critics. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2020.

Arsenii Kuznetsov, Alexander Grishin, Artem Tsypin, Arsenii Ashukha, Artur Kadurin, and Dmitry Vetrov. Automating control of overestimation bias for reinforcement learning, 2021. arXiv:2110.13523.

Qingfeng Lan, Yangchen Pan, Alona Fyshe, and Martha White. Maxmin q-learning: Controlling the estimation bias of q-learning. In *International Conference on Learning Representations (ICLR)*, 2020.

Kimin Lee, Michael Laskin, Aravind Srinivas, and Pieter Abbeel. Sunrise: A simple unified framework for ensemble learning in deep reinforcement learning. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2021.

Litian Liang, Yaosheng Xu, Stephen McAleer, Dailin Hu, Alexander Ihler, Pieter Abbeel, and Roy Fox. Reducing variance in temporal-difference value estimation via ensemble of deep networks. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2022.

T.P. Lillicrap, J.J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra. Continuous control with deep reinforcement learning. In *International Conference on Learning Representations (ICLR)*, 2016.

Xiaoteng Ma, Junyao Chen, Li Xia, Jun Yang, Qianchuan Zhao, and Zhengyuan Zhou. DSAC: distributional soft actor-critic for risk-sensitive reinforcement learning. *Journal of Artificial Intelligence Research*, 2025.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *Nature*, 2015.

Ted Moskovitz, Jack Parker-Holder, Aldo Pacchiano, Michael Arbel, and Michael Jordan. Tactical optimism and pessimism for deep reinforcement learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

Mitsuhiko Nakamoto, Simon Zhai, Anikait Singh, Max Sobol Mark, Yi Ma, Chelsea Finn, Aviral Kumar, and Sergey Levine. Cal-ql: Calibrated offline rl pre-training for efficient online fine-tuning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

Michal Nauman and Marek Cygan. Decoupled policy actor-critic: Bridging pessimism and risk awareness in reinforcement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2025.

Michal Nauman, Michał Bortkiewicz, Piotr Miłoś, Tomasz Trzcinski, Mateusz Ostaszewski, and Marek Cygan. Overestimation, overfitting, and plasticity in actor-critic: the bitter lesson of reinforcement learning. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2024.

Michał Nauman, Mateusz Ostaszewski, and Marek Cygan. A case for validation buffer in pessimistic actor-critic. In *Proceedings of the Thirty-Fourth International Joint Conference on Artificial Intelligence (IJCAI)*, 2025.

Evgenii Nikishin, Max Schwarzer, Pierluca D'Oro, Pierre-Luc Bacon, and Aaron Courville. The primacy bias in deep reinforcement learning. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2022.

Ian Osband, Charles Blundell, Alexander Pritzel, and Benjamin Van Roy. Deep exploration via bootstrapped DQN. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2016.

A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, Trevor Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

Martin L Puterman. *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons, 2014.

Wenling Shang, Kihyuk Sohn, Diogo Almeida, and Honglak Lee. Understanding and improving convolutional neural networks via concatenated rectified linear units. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2016.

David Silver, Guy Lever, Nicolas Heess, Thomas Degris, Daan Wierstra, and Martin Riedmiller. Deterministic policy gradient algorithms. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2014.

Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT Press, 2018.

Denis Tarasov, Alexander Nikulin, Dmitry Akimov, Vladislav Kurenkov, and Sergey Kolesnikov. Corl: Research-oriented deep offline reinforcement learning library. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

Bahareh Tasdighi, Nicklas Werge, Yi-Shan Wu, and Melih Kandemir. Improving actor-critic training with steerable action-value approximation errors. In *European Conference on Artificial Intelligence (ECAI)*, 2025.

Y. Tassa, Y. Doron, A. Muldal, T. Erez, Y. Li, D. de Las Casas, D. Budden, A. Abdolmaleki, J. Merel, A. Lefrancq, et al. Deepmind control suite. *arXiv preprint arXiv:1801.00690*, 2018.

Sebastian Thrun and Anton Schwartz. Issues in using function approximation for reinforcement learning. In *Proceedings of the Fourth Connectionist Models Summer School*, 1993.

Emanuel Todorov, Tom Erez, and Yuval Tassa. Mujoco: A physics engine for model-based control. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2012.

Mark Towers, Ariel Kwiatkowski, Jordan Terry, John U Balis, Gianluca De Cola, Tristan Deleu, Manuel Goulao, Andreas Kallinteris, Markus Krimmel, Arjun KG, et al. Gymnasium: A standard interface for reinforcement learning environments. *arXiv preprint arXiv:2407.17032*, 2024.

Hado Van Hasselt. Double Q-learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2010.

Hado Van Hasselt, Arthur Guez, and David Silver. Deep reinforcement learning with double q-learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2016.

Hado Van Hasselt, Yotam Doron, Florian Strub, Matteo Hessel, Nicolas Sonnerat, and Joseph Modayil. Deep reinforcement learning and the deadly triad. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2018. Deep Reinforcement Learning Workshop.

Matej Vecerik, Todd Hester, Jonathan Scholz, Fumin Wang, Olivier Pietquin, Bilal Piot, Nicolas Heess, Thomas Rothörl, Thomas Lampe, and Martin Riedmiller. Leveraging demonstrations for deep reinforcement learning on robotics problems with sparse rewards. *arXiv preprint arXiv:1707.08817*, 2017.

Hang Wang, Sen Lin, and Junshan Zhang. Adaptive ensemble Q-learning: Minimizing estimation bias via error feedback. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

Christopher JCH Watkins and Peter Dayan. Q-learning. *Machine Learning*, 1992.

Yanqiu Wu, Xinyue Chen, Che Wang, Yiming Zhang, and Keith W Ross. Aggressive Q-learning with ensembles: Achieving both high sample efficiency and high asymptotic performance. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022. Deep Reinforcement Learning Workshop.

Derek Yang, Li Zhao, Zichuan Lin, Tao Qin, Jiang Bian, and Tieyan Liu. Fully parameterized quantile function for distributional reinforcement learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

Kai Yang, Jian Tao, Jiafei Lyu, and Xiu Li. Exploration and anti-exploration with distributional random network distillation. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2024.

Shlomo Yitzhaki. Gini's mean difference: A superior measure of variability for non-normal distributions. *Metron*, 2003.

Omar G. Younis, Rodrigo Perez-Vicente, John U. Balis, Will Dudley, Alex Davey, and Jordan K. Terry. Minari. Zenodo, 2024. URL https://doi.org/10.5281/zenodo.13767625.

Kai Zhao, Jianye Hao, Yi Ma, Jinyi Liu, Yan Zheng, and Zhaopeng Meng. ENOTO: Improving offline-to-online reinforcement learning with Q-ensembles. In *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence (IJCAI)*, 2024.

Zhiyuan Zhou, Andy Peng, Qiyang Li, Sergey Levine, and Aviral Kumar. Efficient online reinforcement learning fine-tuning need not retain offline data. In *International Conference on Learning Representations (ICLR)*, 2025.

Brian D Ziebart. *Modeling purposeful adaptive behavior with the principle of maximum causal entropy*. PhD thesis, Carnegie Mellon University, 2010.

APPENDIX

A RELATED WORK

Fixed aggregation rules for the critic reference. The concern that motivates ensemble aggregation is overestimation: maximizing over noisy action-value estimates systematically inflates them (Thrun & Schwartz, 1993), and double estimators were introduced to decouple action selection from action evaluation (Van Hasselt, 2010; Van Hasselt et al., 2016). In continuous control, the dominant remedy is clipped double Q-learning, which sets the critic reference to the minimum of two critics (Fujimoto et al., 2018) and was adopted by SAC (Haarnoja et al., 2018b); in our parameterization it is the single point $(\bar{\kappa}, \kappa) = (-1, -1)$ at $N = 2$ (Proposition 1). Larger ensembles have been aggregated through a variety of prescribed rules: the minimum over all N estimators, with N itself trading overestimation against underestimation (Lan et al., 2020); an average over previously computed target estimates (Anschel et al., 2017); and the plain ensemble mean, motivated by variance reduction (Liang et al., 2022). REDQ takes the minimum over a randomly drawn subset of size M for the critic target while using the mean of all N critics for the actor (Chen et al., 2021), AQE averages the K smallest target estimates (Wu et al., 2022), and SAC-N minimizes over the full ensemble in the offline setting (An et al., 2021). The distributional counterpart is TQC, which drops the largest atoms of the pooled quantile mixture, with the number of dropped atoms tuned per environment (Kuznetsov et al., 2020). What these rules share is that the degree of conservatism is fixed before training by a discrete design choice (M, K, N, or the number of dropped atoms) and remains fixed throughout. AEA replaces that choice by a continuous coefficient learned online. Our experiments include SAC, REDQ, and SAC-N to compare AEA with standard twin-critic aggregation, subset-based critic aggregation with ensemble-mean actor updates, and full-ensemble minimization, respectively.

Mean-disagreement parameterizations. A second group of methods makes the degree of conservatism explicit by shifting the ensemble mean by a multiple of its spread, $\mu_Q \pm \beta\, \sigma_Q$ (Ciosek et al., 2019; Moskovitz et al., 2021; Cetin & Celiktutan, 2023; Nauman et al., 2024). OAC uses $\sigma_Q = \frac{1}{2}|Q^1 - Q^2|$ and fixed confidence coefficients: it trains its target policy using a lower confidence bound and collects experience with a separate optimistic exploration policy constrained to stay close to the target policy in KL divergence (Ciosek et al., 2019). AEA instead learns the aggregation coefficient in a single actor's objective, without a separate KL-constrained exploration step. Ensemble disagreement also guides exploration through UCB bonuses (Chen et al., 2017; Lee et al., 2021) and posterior-sampling-style ensembles (Osband et al., 2016). Closest to our parameterization is USAC, which likewise maintains separate steerable critic and actor references, but selects their coefficients by comparing task-level performance over a grid (Tasdighi et al., 2025). AEA retains the two-reference structure and learns the coefficients from training data.

Online adaptation of pessimism. The most closely related line of work adapts the degree of pessimism during training. TOP uses a bandit to select among prescribed optimism levels based on changes in episodic return (Moskovitz et al., 2021). GPL learns a pessimism weight by dual TD-learning on replay data; its actor uses the same learned weight with an optional, separately prescribed optimism shift (Cetin & Celiktutan, 2023). However, its reported configurations retain task-specific choices: GPL-SAC uses environment-specific target entropies for its five MuJoCo tasks, and the pixel-based GPL-DrQ uses three-step returns except on Walker-run, where it uses one-step returns. Thus, learning the pessimism weight does not remove environment-specific configuration. OPL adapts pessimism using recent k-step returns (Kuznetsov et al., 2021), while VPL minimizes a squared TD objective on held-out transitions (Nauman et al., 2025). Other approaches adjust TQC's truncation using recent Monte Carlo returns (Dorka et al., 2023), adapt the in-target ensemble size using estimated approximation error (Wang et al., 2021), or control overestimation locally using distributional double Q-learning (Döring et al., 2025).

AEA differs in both its calibration signal and the data it requires. GPL uses an average TD residual, so large errors can dominate its update. OPL also uses sign-based updates, but takes the sign after aggregating its bias estimates and requires recent k-step rollouts, introducing a fresh-trajectory buffer and return-estimation step beyond ordinary transition minibatches. AEA instead averages transition-wise votes from the same minibatch B_t used to train the actor and critics. Its

disagreement-normalized absolute losses yield median-based calibration (Proposition 2), without additional return estimation or held-out data. AEA also learns separate critic and actor coefficients, rather than sharing one learned weight with a prescribed actor offset. Across our tasks and three settings, we keep the calibration step size, initialization, and coefficient bounds fixed (Table 4).

The actor reference, optimism, and exploration. A pessimistic actor objective can discourage exploration of uncertain actions (Ciosek et al., 2019). Decoupled Policy Actor-Critic addresses this by maintaining separate optimistic and pessimistic actors coupled through a KL constraint (Nauman & Cygan, 2025). AEA keeps a single actor and admits optimism through κ: when $\kappa > 0$, its actor reference favors, among actions with equal ensemble mean, those on which the critics disagree more. Neither a second actor nor a KL term is required.

Uncertainty estimation and value learning. DRND estimates uncertainty by distilling a distribution of random networks and reports strong results against established exploration methods on Adroit and Fetch control tasks (Yang et al., 2024). DSAC models the return distribution within a soft actor-critic framework and reports higher average peak returns than SAC and TD3 across the MuJoCo tasks in its original evaluation (Ma et al., 2025). We include these strong baselines to compare AEA with alternative approaches to uncertainty estimation and value learning, extending the evaluation beyond prescribed ensemble aggregation rules.

Critic regularization provides a complementary approach. In some settings, layer normalization has been reported to remove the need for clipped double Q-learning (Nauman et al., 2024). AEA uses CReLU activations (Shang et al., 2016) following Nauman et al. (2024) and learns how to aggregate the resulting critic estimates.

Learning with external data. Our third evaluation setting connects AEA to methods that incorporate previously collected experience. Retaining such data in replay has a long history in learning from demonstrations (Vecerik et al., 2017). The SACfD baseline evaluated by Ball et al. (2023) initializes SAC's replay buffer with external data and provides a direct baseline for incorporating such data without offline pretraining. RLPD combines symmetric sampling from external and online buffers with a layer-normalized critic ensemble, also without offline pretraining (Ball et al., 2023). It provides our main comparison under the same ensemble size and update-to-data ratio as AEA. Cal-QL instead calibrates the offline Q-function before online fine-tuning and reports the best fine-tuned performance on nine of eleven tasks in its original evaluation (Nakamoto et al., 2023). We include it as a strong offline-to-online baseline to compare AEA with a method that uses offline pretraining. Related approaches switch the ensemble aggregation rule at the offline-to-online boundary (Zhao et al., 2024) or investigate whether offline data must be retained during fine-tuning (Zhou et al., 2025). AEA calibrates aggregation throughout training, using the same update in all three settings and no offline pretraining.

B Proofs of Statements in Section 3

B.1 Proof of Proposition 1

Proof of Proposition 1. Fix a state-action pair (s, a) and write $q_i = Q_{\bar{\theta}_i}(s, a)$ for the target-critic outputs. For a uniformly sampled pair of distinct indices $\{I, J\}$, the identity $\min(q_i, q_j) = (q_i + q_j - |q_i - q_j|)/2$ gives

$$\mathbb{E}_{\{I,J\}}[\min(q_I, q_J) \mid q_{1:N}] = \frac{1}{\binom{N}{2}} \sum_{i<j} \frac{q_i + q_j - |q_i - q_j|}{2} = \bar{\mu}(s, a) - \bar{\delta}(s, a), \tag{12}$$

where the last equality follows because each q_i appears in $N - 1$ pairs and $2\binom{N}{2} = N(N - 1)$.

When $N = 2$, there is only one pair, so $\bar{\mu} - \bar{\delta}$ equals the minimum of the two target critics. Applying the same identity to the active critics gives $\mu - \delta = \min(Q_{\theta_1}, Q_{\theta_2})$. Thus, $(\bar{\kappa}, \kappa) = (-1, -1)$ exactly recovers SAC's two references.

For REDQ with $M = 2$, Equation (12) shows that the conditional expected critic reference is $\bar{\mu} - \bar{\delta}$, corresponding to $\bar{\kappa} = -1$. Its actor reference is the mean of all N active critics, which is exactly μ and corresponds to $\kappa = 0$. □

B.2 Median Solutions of the Calibration Losses

Proposition 2 (Median solutions)**.** *Assume all quantities other than the coefficient being optimized are fixed, and that $\delta(s,a) > 0$ and $\bar{\delta}(s',a') > 0$ for every transition in B_t and its sampled next action a'. Let Med_{B_t} denote the set of empirical medians over the minibatch, and let y_0 denote the regression target $y_{\bar{\kappa}}$ evaluated at $\bar{\kappa} = 0$. Then, the losses in Equations (7) and (9) satisfy*

$$\arg\min_{\bar{\kappa}\in\mathbb{R}} \mathcal{L}_{\mathrm{crit}}(\bar{\kappa}) = \mathrm{Med}_{B_t}\left[\frac{\widetilde{Q}_\kappa(s,a) - y_0(s,a,r,s')}{\gamma\bar{\delta}(s',a')}\right],$$
$$\arg\min_{\kappa\in\mathbb{R}} \mathcal{L}_{\mathrm{act}}(\kappa) = \mathrm{Med}_{B_t}\left[\frac{y_{\bar{\kappa}}(s,a,r,s') - \mu(s,a)}{\delta(s,a)}\right].$$

Proof. Since $y_{\bar{\kappa}} = y_0 + \gamma\bar{\kappa}\bar{\delta}(s',a')$ and $\widetilde{Q}_\kappa(s,a) = \mu(s,a) + \kappa\delta(s,a)$, substitution into the two losses gives

$$\mathcal{L}_{\mathrm{crit}}(\bar{\kappa}) = \gamma\mathbb{E}_{B_t}\left[\left|\bar{\kappa} - \frac{\widetilde{Q}_\kappa(s,a) - y_0(s,a,r,s')}{\gamma\bar{\delta}(s',a')}\right|\right],$$
$$\mathcal{L}_{\mathrm{act}}(\kappa) = \mathbb{E}_{B_t}\left[\left|\kappa - \frac{y_{\bar{\kappa}}(s,a,r,s') - \mu(s,a)}{\delta(s,a)}\right|\right].$$

A sum of absolute deviations is convex, and its subdifferential contains zero exactly when at most half of the observations lie strictly below the candidate value and at most half lie strictly above it. This is the defining condition for an empirical median. If the solution is not unique, both the argmin and Med_{B_t} denote the entire closed interval between the two middle ordered values. This establishes both identities. □

B.3 Convergence of Policy Evaluation

We stated the informal convergence of the AEA critic ensemble in Section 3.3. We now formalize this statement and provide a proof. Consider a finite tabular MDP with bounded rewards, and fix the policy π_ϕ and temperature α. Let $\mu_t, \delta_t, \bar{\mu}_t, \bar{\delta}_t$ denote the ensemble statistics at evaluation iteration t. We consider the following exact synchronous evaluation dynamics. At each iteration, form the shared exact expected target

$$y_t(s,a) = r(s,a) + \gamma\mathbb{E}_{s'\sim p(\cdot|s,a),\, a'\sim\pi_\phi(\cdot|s')}\left[\bar{\mu}_t(s',a') + \bar{\kappa}_t\bar{\delta}_t(s',a') - \alpha\log\pi_\phi(a' \mid s')\right], \quad (13)$$

and update all critics at every state-action pair by

$$Q_{i,t+1} = (1-\eta_\theta)Q_{i,t} + \eta_\theta y_t, \qquad \bar{Q}_{i,t+1} = (1-\tau)\bar{Q}_{i,t} + \tau Q_{i,t+1}, \quad (14)$$

where $\eta_\theta, \tau \in (0,1]$ are constant and shared across critics. These dynamics are idealized tabular counterparts of the critic and target-critic updates in Algorithm 1: the update of $Q_{i,t}$ is the value-space form of a gradient step on the squared Bellman loss, with η_θ denoting its effective interpolation rate, while the update of $\bar{Q}_{i,t}$ is an exponential moving average.

Theorem 3 (Policy evaluation)**.** *Assume there exists $K < \infty$ such that $|\bar{\kappa}_t| \le K$ and $|\kappa_t| \le K$ for every t. Apart from this bound, the coefficients may change arbitrarily across iterations and need not converge. Under the dynamics defined in Equations (13) and (14), from any finite initial critic values,*

$$\lim_{t\to\infty}\|Q_{i,t} - Q^{\pi_\phi}\|_\infty = \lim_{t\to\infty}\|\bar{Q}_{i,t} - Q^{\pi_\phi}\|_\infty = 0, \qquad i \in [N].$$

Moreover, the two references satisfy

$$\lim_{t\to\infty}\|\bar{\mu}_t + \bar{\kappa}_t\bar{\delta}_t - Q^{\pi_\phi}\|_\infty = \lim_{t\to\infty}\|\mu_t + \kappa_t\delta_t - Q^{\pi_\phi}\|_\infty = 0.$$

Proof. We first show that ensemble disagreement vanishes. Since all active critics use the same target,

$$Q_{i,t+1} - Q_{j,t+1} = (1-\eta_\theta)(Q_{i,t} - Q_{j,t}), \qquad \delta_t = (1-\eta_\theta)^t\delta_0.$$

Thus, $\|\delta_t\|_\infty \to 0$. Applying the triangle inequality to each pair of target critics and averaging gives

$$\bar{\delta}_{t+1} \le (1-\tau)\bar{\delta}_t + \tau\delta_{t+1}$$

pointwise. Because $\tau > 0$ and $\|\delta_t\|_\infty \to 0$, iterating this inequality also gives $\|\bar{\delta}_t\|_\infty \to 0$.

We next show that the ensemble means converge to the correct value function. Define

$$E_t = \max\left\{\|\mu_t - Q^{\pi_\phi}\|_\infty,\ \|\bar{\mu}_t - Q^{\pi_\phi}\|_\infty\right\}.$$

Since $|\bar{\kappa}_t| \le K$, averaging the active-critic updates and subtracting the Bellman equation for Q^{π_ϕ} yields

$$\|\mu_{t+1} - Q^{\pi_\phi}\|_\infty \le [1 - \eta_\theta(1-\gamma)]E_t + \eta_\theta\gamma K\|\bar{\delta}_t\|_\infty.$$

The target-mean update then gives

$$\|\bar{\mu}_{t+1} - Q^{\pi_\phi}\|_\infty \le [1 - \tau\eta_\theta(1-\gamma)]E_t + \tau\eta_\theta\gamma K\|\bar{\delta}_t\|_\infty.$$

Taking the maximum and using $\tau \le 1$, we obtain

$$E_{t+1} \le [1 - \tau\eta_\theta(1-\gamma)]E_t + \eta_\theta\gamma K\|\bar{\delta}_t\|_\infty.$$

The coefficient of E_t is strictly below one, and the remaining term tends to zero. Iterating this inequality therefore gives $E_t \to 0$.

The convergence of both ensemble means, together with vanishing pairwise disagreement, implies convergence of each active and target critic to Q^{π_ϕ}. Finally, because $|\bar{\kappa}_t| \le K$ and $|\kappa_t| \le K$, vanishing disagreement implies $\|\bar{\kappa}_t\bar{\delta}_t\|_\infty \to 0$ and $\|\kappa_t\delta_t\|_\infty \to 0$. Both references consequently converge to Q^{π_ϕ} as well. □

B.4 Policy Iteration

Corollary 4 (Policy iteration)**.** *Consider an MDP with finite state and action spaces, bounded rewards, discount factor $\gamma \in (0,1)$, and fixed temperature $\alpha > 0$. Starting from any policy π_0, at each outer iteration m hold π_m fixed and run the evaluation dynamics of Theorem 3 to convergence from finite critic values, allowing any bounded coefficient sequences. Denote the limiting actor reference by R_m. At every state, exactly maximize the entropy-regularized actor objective over all distributions on $\mathcal{A}$:*

$$\pi_{m+1}(\cdot \mid s) = \arg\max_{\nu \in \Delta(\mathcal{A})} \sum_{a\in\mathcal{A}} \nu(a)\left[R_m(s,a) - \alpha\log\nu(a)\right],$$

where $\Delta(\mathcal{A})$ is the probability simplex and $0\log 0 = 0$. Then $Q^{\pi_{m+1}} \ge Q^{\pi_m}$ pointwise and $\|Q^{\pi_m} - Q^*\|_\infty \to 0$, where Q^* is the optimal entropy-regularized action-value function at temperature α. Moreover, π_m converges at every state to the soft optimal policy

$$\pi^*(a \mid s) = \frac{\exp(Q^*(s,a)/\alpha)}{\sum_{b\in\mathcal{A}}\exp(Q^*(s,b)/\alpha)}.$$

Proof. Write $Q_m = Q^{\pi_m}$. By Theorem 3, $R_m = Q_m$. The exact actor maximizer is therefore $\pi_{m+1}(a \mid s) = \exp(Q_m(s,a)/\alpha)/\sum_{b\in\mathcal{A}}\exp(Q_m(s,b)/\alpha)$. Thus, the AEA references yield the usual exact soft policy improvement step.

To establish improvement and convergence, let $\mathcal{T}^\pi$ be the soft policy-evaluation operator and $\mathcal{T}$ the soft optimality operator:

$$(\mathcal{T}^\pi Q)(s,a) = r(s,a) + \gamma\mathbb{E}_{s'\sim p(\cdot|s,a)}\left[\sum_{b\in\mathcal{A}}\pi(b \mid s')\left(Q(s',b) - \alpha\log\pi(b \mid s')\right)\right],$$

$$(\mathcal{T}Q)(s,a) = r(s,a) + \gamma\mathbb{E}_{s'\sim p(\cdot|s,a)}\left[\alpha\log\sum_{b\in\mathcal{A}}\exp(Q(s',b)/\alpha)\right].$$

Both operators are monotone γ-contractions in the sup norm, with fixed points Q^π and Q^*, respectively. Exact maximization gives

$$Q_m = \mathcal{T}^{\pi_m}Q_m \le \mathcal{T}Q_m = \mathcal{T}^{\pi_{m+1}}Q_m \le Q_{m+1} \le Q^*.$$

The penultimate inequality follows by iterating the monotone operator $\mathcal{T}^{\pi_{m+1}}$ from Q_m to its fixed point. The last follows by iterating the same operator from Q^*, since $\mathcal{T}^{\pi_{m+1}}Q^* \le \mathcal{T}Q^* = Q^*$. Consequently,

$$0 \le Q^* - Q_{m+1} \le \mathcal{T}Q^* - \mathcal{T}Q_m, \qquad \|Q^* - Q_{m+1}\|_\infty \le \gamma\|Q^* - Q_m\|_\infty.$$

Hence Q_m increases pointwise to Q^* and converges in the sup norm. Continuity of the softmax map gives $\pi_m \to \pi^*$ at every state. Finally, $\mathcal{T}^{\pi^*}Q^* = \mathcal{T}Q^* = Q^*$, so π^* attains the optimal value. □

C Experimental Setup, Hyperparameters, and Implementation Details

Experimental settings. We evaluate three settings with different data availability and data reuse. Interactive learning uses only online data, a UTD ratio of 1, and 1M environment interactions. Sample-efficient learning also uses only online data, but uses a UTD ratio of 20 and 300K environment interactions to study learning under heavier data reuse. In learning with external data, the algorithms have access to a fixed external dataset in addition to the 1M transitions collected through environment interaction. Table 3 summarizes the data availability, benchmarks, interaction budgets, and UTD protocols. Network configurations and method-specific replay protocols are described separately below.

Table 3: Experimental settings. The fourth column reports initial random interactions followed by total environment interactions. G denotes the update-to-data ratio. Method-specific UTD ratios for learning with external data are given in Section C.3.

Setting	Available data	Benchmarks	Random / total	G
Interactive	Online data	MuJoCo, DMC	10K / 1M	1
Sample-efficient	Online data	MuJoCo, DMC	10K / 300K	20
With external data	Online and external data	MuJoCo	5K / 1M	Method-specific

Implementation frameworks and benchmarks. For the interactive and sample-efficient settings, we implement all methods in `PyTorch` (Paszke et al., 2019) to match the existing implementations of the corresponding baselines. We evaluate continuous-control tasks from Gymnasium's MuJoCo-`v5` environments (Towers et al., 2024) and the DeepMind Control Suite (DMC) (Tassa et al., 2018). All evaluated tasks use the MuJoCo physics engine (Todorov et al., 2012). For learning with external data, we implement all methods in JAX (Bradbury et al., 2018) to match the implementation framework of the corresponding baselines, and evaluate the same `v5` MuJoCo environments. We obtain the fixed external datasets from the public MuJoCo collection distributed through `Minari` (Younis et al., 2024).[1] For each task, we use the datasets with identifiers `mujoco/{task}/simple-v0`, `mujoco/{task}/medium-v0`, and `mujoco/{task}/expert-v0`.

AEA hyperparameters. We provide pseudocode for the complete AEA update in Algorithm 1. We summarize the architecture and hyperparameters used by AEA in Table 4. Apart from the critic ensemble size, these values are fixed across all tasks and experimental settings. The initial random interactions, interaction budgets, and UTD protocols are given in Table 3. In learning with external data, AEA uses a UTD ratio of 20 and draws equal numbers of external and online transitions in each minibatch. Following Nauman et al. (2024), we use CReLU activations (Shang et al., 2016) in view of the loss of plasticity observed in deep RL (Abbas et al., 2023). We tune the entropy temperature automatically (Haarnoja et al., 2018b) using target entropy $-\dim(\mathcal{A})$.

[1] https://minari.farama.org/

Table 4: Architecture and hyperparameters used for AEA. Data availability, interaction budgets, and UTD protocols are reported in Table 3.

Hyperparameter	Value
Network architecture (actor and critics)	
Hidden layers	2
Hidden units per layer	256
Activation function	CReLU
Optimization	
Batch size	256
Actor, critic, and temperature optimizer	Adam (Kingma & Ba, 2015)
Actor, critic, and temperature learning rate	3×10^{-4}
Aggregation-parameter step size	0.1
Training configuration	
Critic ensemble size N	2 (interactive), 10 (otherwise)
Replay buffer capacity	10^6
Discount factor γ	0.99
Target smoothing coefficient τ	5×10^{-3}
Initial entropy temperature α_0	0.2
Target entropy	$-\dim(\mathcal{A})$

Aggregation-parameter implementation. We optimize the aggregation coefficients $\bar{\kappa}$ and κ directly, initializing $(\bar{\kappa}, \kappa) = (-0.8, 0)$. After each coefficient update, we clip the updated coefficient to $[-1, 1]$. The endpoints correspond to the conditional expected minimum and maximum of a uniformly sampled critic pair, and the interval contains the SAC and REDQ reference choices in Proposition 1. This clipping also ensures the bounded-coefficient condition in Theorem 3. We use the same initialization, aggregation-parameter step size, and bounds across tasks and experimental settings. An ablation of initialization sensitivity is provided in Section E.

Evaluation. For every method, we evaluate the deterministic policy given by the mean of its learned action distribution. Each evaluation consists of 20 episodes in the interactive and sample-efficient settings and 10 episodes in the setting with external data. Across 10 independent runs, we report the mean and standard deviation of the final undiscounted return, the interquartile mean (IQM) of final return (Agarwal et al., 2021), and the area under the learning curve. IQM provides a robust summary with reduced sensitivity to extreme returns.

We next describe the baseline implementations and computational costs for each of the three experimental settings.

C.1 Interactive Learning

Baselines and implementation. We compare AEA with SAC (Haarnoja et al., 2018a), REDQ (Chen et al., 2021), DRND (Yang et al., 2024), and DSAC (Ma et al., 2025). All methods use 2 critics and a UTD ratio of 1. SAC and REDQ use the common actor-critic hyperparameters listed in Table 4; the aggregation-parameter step size applies only to AEA. DSAC models the return distribution through implicit quantile learning. We use an enhanced version of DSAC that integrates the fully parameterized quantile proposal mechanism of Yang et al. (2019), which learns the quantile fractions rather than sampling them uniformly. DRND estimates uncertainty by distilling a distribution defined by an ensemble of fixed random target networks. For the baselines we relied on the implementations in the ObjectRL library (Baykal et al., 2025).

Computation. AEA, SAC, and REDQ use 2 critics and a UTD ratio of 1, so their actor-critic network workloads are comparable. In addition to the common actor-critic computations, AEA computes the disagreement statistics used by its two references and updates two scalar aggregation parameters. DRND and DSAC include method-specific network components, so matching the critic count does not make their total computation identical. Each seed was run on one NVIDIA GeForce RTX 4090 GPU and required approximately 90 to 120 minutes of wall-clock time.

C.2 Sample-Efficient Learning

Baselines and implementation. We compare AEA with REDQ (Chen et al., 2021), SAC-N (An et al., 2021), DRND (Yang et al., 2024), and DSAC (Ma et al., 2025). AEA, REDQ, and SAC-N use 10 critics and a UTD ratio of 20. DRND and DSAC use the same implementations as in the interactive setting and retain 2 critics. DSAC represents uncertainty through its learned return distribution, whereas DRND uses distillation error; scaling either method to 10 critics would combine its original uncertainty mechanism with a large critic ensemble and change the method being compared. All methods use a UTD ratio of 20 and an interaction budget of 300K. For the baselines, we relied on the implementations in the ObjectRL library (Baykal et al., 2025).

Computation. AEA, REDQ, and SAC-N use 10 critics and a UTD ratio of 20, so their actor-critic network workloads are comparable. In addition to the common actor-critic computations, AEA computes the disagreement statistics used by its two references and updates two scalar aggregation parameters. DRND and DSAC use 2 critics and include method-specific network components, so their total computation is not matched to the 10-critic methods. Each seed was run on one NVIDIA GeForce RTX 4090 GPU and required approximately 6 to 8 hours of wall-clock time.

C.3 Learning with External Data

Baselines and implementation. We compare AEA with SACfD (following the baseline evaluated by Ball et al. (2023)), RLPD (Ball et al., 2023), and Cal-QL (Nakamoto et al., 2023). Our implementation will be made publicly available upon acceptance. SACfD extends SAC to external data in the most direct way: it initializes the replay buffer with the complete external dataset, retains those transitions as online experience is added, and applies the standard SAC update without offline pretraining. It therefore provides a direct baseline for isolating the benefit of more specialized treatments of external data. RLPD provides a strong practical baseline for learning with external data without offline pretraining. Our RLPD baseline is a JAX reimplementation adapted from the public implementation released with A3RL[2]. For AEA and SACfD, we initialize the entropy temperature to $\alpha_0 = 0.2$ and use target entropy $-\dim(\mathcal{A})$. Following RLPD's original hyperparameter selection, we initialize its entropy temperature to $\alpha_0 = 1$ and use target entropy $-\dim(\mathcal{A})/2$. AEA begins online learning without offline pretraining and uses the same coefficient update as in the two online-only settings. Cal-QL provides a strong offline-to-online baseline that uses offline pretraining followed by online fine-tuning. Our Cal-QL implementation is adapted from the public CORL implementation (Tarasov et al., 2023). For the MuJoCo tasks, we use discounted Monte Carlo returns within each recorded trajectory as calibration references. During online fine-tuning, Cal-QL uses a UTD ratio of 1, retains the calibrated CQL objective, and draws equal numbers of external and online transitions in each minibatch. Following the official implementation, we set the CQL regularization weight to 5, initialize the entropy temperature to $\alpha_0 = 1$, and tune it automatically using target entropy $-\dim(\mathcal{A})$.

Computation. AEA uses 10 critics and a UTD ratio of 20 for 1M environment interactions, so its aggregation cost is the same as in the sample-efficient setting. Cal-QL performs 1M gradient updates for offline pretraining before 1M environment interactions with a UTD ratio of 1. Experiments in this setting use an NVIDIA RTX PRO 6000 Blackwell Server Edition and require approximately 2 to 4 hours of wall-clock time per seed, depending on the method.

[2] https://github.com/xuefeng-cs/A3RL

D Additional Experimental Results

Across all three learning settings, AEA remains competitive with the strongest corresponding baselines while using the same coefficient update throughout. This appendix reports complete task-level performance results for the three settings summarized in Table 3. Sections D.1 and D.2 report results on MuJoCo and DMC for interactive and sample-efficient learning, respectively, whereas Section D.3 reports MuJoCo results for learning with external data across datasets with varying behavioral quality and coverage. For each setting, we report final return, the interquartile mean (IQM) of final return (Agarwal et al., 2021), and area under the learning curve (AULC). Section E then examines the learned calibration trajectories and the corresponding ablations.

D.1 Interactive Learning

Tables 5 to 7 report final return, interquartile mean, and area under the learning curve, respectively, on MuJoCo and DMC under the interactive protocol. AEA is competitive with the strongest baselines in this setting, although the best-performing method varies across tasks and metrics.

Table 5: Final return on continuous-control environments with 2 critics, a UTD ratio of 1, and an interaction budget of 1M. Values are averaged over evaluation repetitions and 10 seeds. ± denotes standard deviation over seeds. Highest mean is bold.

	AEA	REDQ	SAC	DRND	DSAC
MuJoCo-v5	(ours)	(Chen et al., 2021)	(Haarnoja et al., 2018a)	(Yang et al., 2024)	(Ma et al., 2025)
Ant	**5355**±1126	4083±1477	4396±1911	4769±1139	4792±2016
HalfCheetah	10595±947	11437±264	10947±630	10574±887	**11574**±584
Hopper	**3425**±379	2344±842	3060±885	3241±692	2034±837
Humanoid	4780±1295	4646±1420	4761±1225	5034±451	**6062**±593
Walker2d	4270±1350	4182±1352	4396±245	4207±647	**5207**±416
DMC					
Cheetah-run	802±30	**814**±105	784±28	**814**±44	655±99
Hopper-hop	219±44	**242**±86	114±89	30±60	165±96
Hopper-stand	789±305	**921**±99	753±311	781±192	727±241
Humanoid-run	159±36	**180**±28	159±18	1±0	158±21
Humanoid-stand	638±265	728±156	644±151	661±141	**893**±30
Humanoid-walk	517±92	557±22	564±89	113±198	**629**±148
Quadruped-run	823±67	**878**±59	852±88	843±53	792±30
Quadruped-walk	914±164	**955**±36	848±236	936±41	952±12
Walker-run	756±33	**779**±26	735±49	732±38	710±57

Table 6: Interquartile mean (IQM) of final return (Agarwal et al., 2021) on continuous-control environments with 2 critics, a UTD ratio of 1, and an interaction budget of 1M. Results are based on 10 independent runs. ± denotes standard deviation over seeds. Highest mean is bold.

	AEA	REDQ	SAC	DRND	DSAC
MuJoCo-v5	(ours)	(Chen et al., 2021)	(Haarnoja et al., 2018a)	(Yang et al., 2024)	(Ma et al., 2025)
Ant	**5708**±200	4595±451	5164±502	5067±1008	4773±2097
HalfCheetah	11007±70	11400±80	11074±138	10572±894	**11572**±587
Hopper	**3531**±22	2254±538	3458±80	3233±717	2053±869
Humanoid	5286±35	5271±360	5281±102	5283±176	**6468**±228
Walker2d	4674±155	4378±246	4396±108	4292±462	**5217**±406
DMC					
Cheetah-run	797±13	**844**±15	779±14	814±43	655±99
Hopper-hop	228±8	**242**±17	80±76	26±61	149±119
Hopper-stand	925±16	**940**±9	896±50	787±265	704±327
Humanoid-run	164±9	**178**±13	159±7	1±0	161±20
Humanoid-stand	686±129	749±89	616±49	664±142	**904**±29
Humanoid-walk	529±14	558±10	563±19	113±199	**632**±150
Quadruped-run	831±28	**891**±21	865±33	847±46	794±29
Quadruped-walk	945±15	945±11	941±17	946±24	**956**±15
Walker-run	764±8	**781**±11	740±19	732±38	710±57

Table 7: Area under the learning curve (AULC) on continuous-control environments with 2 critics, a UTD ratio of 1, and an interaction budget of 1M. Values are averaged over evaluation repetitions and 10 seeds. ± denotes standard deviation over seeds. Highest mean is bold.

	AEA	REDQ	SAC	DRND	DSAC
MuJoCo-v5	(ours)	(Chen et al., 2021)	(Haarnoja et al., 2018a)	(Yang et al., 2024)	(Ma et al., 2025)
Ant	**3294**±542	2161±445	3059±524	3010±666	2903±1020
HalfCheetah	8486±845	8852±434	8594±824	8327±740	**9029**±436
Hopper	**2924**±145	2115±360	2500±422	2589±355	1794±389
Humanoid	**3713**±286	2740±185	3363±372	3502±307	3663±328
Walker2d	3295±521	2762±395	2977±385	3046±421	**3801**±311
DMC					
Cheetah-run	628±21	651±23	617±24	**663**±34	508±73
Hopper-hop	**150**±41	149±46	64±27	14±30	98±54
Hopper-stand	606±189	**735**±54	429±215	524±236	550±218
Humanoid-run	105±12	**109**±16	77±16	1±0	80±17
Humanoid-stand	372±106	383±103	312±102	291±105	**535**±88
Humanoid-walk	325±26	**339**±15	296±87	37±71	322±93
Quadruped-run	634±77	**656**±78	640±74	627±67	647±38
Quadruped-walk	782±58	761±39	718±126	664±84	**813**±31
Walker-run	641±37	**655**±40	608±49	629±40	611±65

D.2 Sample-Efficient Learning

Tables 8 to 10 report the three performance metrics on MuJoCo and DMC under the sample-efficient protocol. The IQM table repeats the results in Table 1 so that all three metrics can be read together here. AEA achieves the strongest result on a majority of tasks under each metric, while individual baselines remain strongest on selected tasks.

Table 8: Final return on continuous-control environments with 10 critics for AEA, REDQ, and SAC-N, 2 critics for DRND and DSAC, a UTD ratio of 20, and an interaction budget of 300K. Values are averaged over evaluation repetitions and 10 seeds. ± denotes standard deviation over seeds. Highest mean is bold.

MuJoCo-v5	AEA (ours)	REDQ (Chen et al., 2021)	SAC-N (An et al., 2021)	DRND (Yang et al., 2024)	DSAC (Ma et al., 2025)
Ant	**4485**±1044	4473±1142	1±4	819±494	731±201
HalfCheetah	**9721**±817	8692±1380	5586±662	7893±1181	9059±1194
Hopper	**3326**±583	3169±592	1992±1278	3097±725	2680±681
Humanoid	**5365**±446	4718±1544	4903±1376	2726±1313	1051±225
Walker2d	4579±815	4697±599	1581±1384	5053±416	**5363**±348
DMC					
Cheetah-run	823±58	851±48	596±134	**871**±24	746±30
Hopper-hop	**130**±102	105±113	1±3	9±19	79±17
Hopper-stand	**779**±333	318±431	5±16	374±310	273±70
Humanoid-run	**145**±15	117±43	1±1	1±0	17±32
Humanoid-stand	**678**±145	539±204	52±117	204±137	271±183
Humanoid-walk	**478**±57	416±147	2±1	158±149	2±0
Quadruped-run	**823**±124	728±260	209±338	234±372	520±55
Quadruped-walk	**935**±58	878±188	284±385	200±276	570±325
Walker-run	**705**±100	631±129	270±130	688±93	650±86

Table 9: Interquartile mean (IQM) of final return (Agarwal et al., 2021) on continuous-control environments with 10 critics for AEA, REDQ, and SAC-N, 2 critics for DRND and DSAC, a UTD ratio of 20, and an interaction budget of 300K. Results are based on 10 independent runs. ± denotes standard deviation over seeds. Highest mean is bold. This table reproduces the results in Table 1.

MuJoCo-v5	AEA (ours)	REDQ (Chen et al., 2021)	SAC-N (An et al., 2021)	DRND (Yang et al., 2024)	DSAC (Ma et al., 2025)
Ant	4744±260	**4828**±398	1±1	797±560	776±177
HalfCheetah	**9661**±395	8678±813	5425±191	7892±1180	9059±1191
Hopper	**3555**±32	3415±153	2063±1034	3015±840	2640±783
Humanoid	5376±87	**5404**±61	5341±124	2853±1684	908±171
Walker2d	4652±267	4749±170	1035±830	5054±414	**5362**±349
DMC					
Cheetah-run	827±9	866±6	635±26	**873**±21	753±37
Hopper-hop	**86**±75	54±81	0±1	3±7	54±58
Hopper-stand	**935**±16	109±268	0±1	251±339	39±58
Humanoid-run	**144**±5	131±8	1±1	1±0	17±32
Humanoid-stand	**665**±55	559±31	9±1	213±144	284±190
Humanoid-walk	**484**±14	454±35	2±1	166±160	2±0
Quadruped-run	**856**±19	847±24	43±52	224±379	536±80
Quadruped-walk	**947**±14	932±16	112±90	184±285	555±371
Walker-run	**741**±27	649±56	258±84	689±93	658±86

Table 10: Area under the learning curve (AULC) on continuous-control environments with 10 critics for AEA, REDQ, and SAC-N, 2 critics for DRND and DSAC, a UTD ratio of 20, and an interaction budget of 300K. Values are averaged over evaluation repetitions and 10 seeds. $\pm$ denotes standard deviation over seeds. Highest mean is bold.

MuJoCo-v5	AEA (ours)	REDQ (Chen et al., 2021)	SAC-N (An et al., 2021)	DRND (Yang et al., 2024)	DSAC (Ma et al., 2025)
Ant	**2343**$\pm$616	2303$\pm$431	33$\pm$11	324$\pm$163	407$\pm$62
HalfCheetah	**7214**$\pm$459	6722$\pm$721	4791$\pm$440	5705$\pm$594	6869$\pm$823
Hopper	2723$\pm$322	**2843**$\pm$232	1318$\pm$912	2544$\pm$283	2334$\pm$459
Humanoid	2954$\pm$523	**3031**$\pm$569	2357$\pm$914	872$\pm$223	750$\pm$72
Walker2d	2992$\pm$620	**3389**$\pm$210	975$\pm$690	2073$\pm$220	3010$\pm$59
DMC					
Cheetah-run	627$\pm$42	**670**$\pm$56	466$\pm$102	623$\pm$50	544$\pm$26
Hopper-hop	**74**$\pm$62	63$\pm$47	0$\pm$0	5$\pm$11	33$\pm$11
Hopper-stand	**567**$\pm$166	257$\pm$226	4$\pm$1	215$\pm$138	181$\pm$64
Humanoid-run	**80**$\pm$16	57$\pm$27	1$\pm$0	1$\pm$0	2$\pm$3
Humanoid-stand	**315**$\pm$65	270$\pm$98	27$\pm$41	80$\pm$36	89$\pm$62
Humanoid-walk	**239**$\pm$51	196$\pm$97	2$\pm$0	53$\pm$64	3$\pm$2
Quadruped-run	**620**$\pm$148	547$\pm$170	191$\pm$60	182$\pm$138	341$\pm$121
Quadruped-walk	**744**$\pm$59	653$\pm$154	231$\pm$42	147$\pm$119	374$\pm$96
Walker-run	**570**$\pm$80	496$\pm$113	188$\pm$75	516$\pm$84	502$\pm$67

D.3 Learning with External Data

Tables 11 to 13 report the three performance metrics for each available MuJoCo task and external-dataset condition. The IQM table includes the four tasks shown in Table 2 as well as the remaining tasks. Learning with external data is particularly challenging because replay combines a fixed behavior distribution with experience generated by an evolving online policy. Despite these heterogeneous data sources, AEA uses the same coefficient update as in the interactive and sample-efficient settings and remains competitive across tasks and external datasets. The gains are especially consistent on HalfCheetah, where AEA leads across all three external datasets and all three metrics. On Humanoid, AEA achieves substantially higher performance with medium and expert external data, while the simple-data condition is mixed across metrics.

Table 11: Final return on continuous-control environments with external data. All experiments have an interaction budget of 1M. Values are averaged over evaluation repetitions and 10 seeds. ± denotes the standard deviation across seeds. Highest mean is bold.

MuJoCo-v5	AEA (ours)	RLPD (Ball et al., 2023)	Cal-QL (Nakamoto et al., 2023)	SACfD (Ball et al., 2023)
ant-expert	7015±310	**7349**±237	4336±3359	6118±328
ant-medium	7064±70	**7068**±231	−1170±2131	6062±292
ant-simple	6900±191	**7129**±251	4634±239	5159±418
halfcheetah-expert	**17120**±101	16738±62	8773±1204	2112±1927
halfcheetah-medium	**15779**±91	14360±127	12735±1297	13195±245
halfcheetah-simple	**13505**±295	12299±112	8095±134	11163±170
hopper-expert	**3754**±384	3455±768	1844±565	2070±396
hopper-medium	3283±569	**3549**±614	3392±270	2928±475
hopper-simple	2736±710	3096±660	**3133**±87	2931±646
humanoid-expert	**9832**±424	5374±39	6671±1311	193±107
humanoid-medium	**8753**±510	5438±86	6733±1308	324±95
humanoid-simple	6831±1227	**7023**±229	5796±108	5688±94
pusher-expert	**−24**±1	−25±1	−31±4	−25±1
pusher-medium	**−23**±1	−24±1	−32±3	−25±1
reacher-expert	**−3**±0	−3±0	−3±0	−3±0
reacher-medium	**−3**±0	−3±0	−4±0	−3±0
swimmer-expert	**235**±97	135±6	152±32	120±93
swimmer-medium	**163**±63	136±3	123±49	76±13
walker2d-expert	**6913**±70	6872±43	4182±544	2316±1572
walker2d-medium	**6324**±92	6315±67	5716±365	5148±1657
walker2d-simple	5460±259	**5733**±297	449±664	4580±71

Table 12: Interquartile mean (IQM) of final return (Agarwal et al., 2021) on continuous-control environments with external data. All experiments have an interaction budget of 1M. Results are based on 10 independent runs. ± denotes the standard deviation across seeds. Highest mean is bold. Selected results are presented in Table 2.

MuJoCo-v5	AEA (ours)	RLPD (Ball et al., 2023)	Cal-QL (Nakamoto et al., 2023)	SACfD (Ball et al., 2023)
ant-expert	7116±137	**7417**±36	5791±1106	6221±111
ant-medium	7080±23	**7162**±55	−1784±366	6137±142
ant-simple	6970±67	**7214**±43	4630±71	5202±103
halfcheetah-expert	**17127**±40	16725±27	9009±302	1630±469
halfcheetah-medium	**15769**±44	14330±47	12695±416	13184±95
halfcheetah-simple	**13467**±116	12292±47	8108±49	11153±66
hopper-expert	**3819**±180	3640±242	1789±229	2069±148
hopper-medium	3258±243	**3754**±181	3474±94	2858±190
hopper-simple	2793±333	**3354**±257	3154±18	3197±142
humanoid-expert	**9933**±153	5375±16	6762±597	191±57
humanoid-medium	**8746**±218	5454±30	7149±237	346±26
humanoid-simple	**7249**±426	6988±82	5780±46	5671±24
pusher-expert	**−24**±0	−25±0	−32±1	−25±0
pusher-medium	**−23**±0	−24±0	−32±1	−25±0
reacher-expert	**−3**±0	−3±0	−3±0	−3±0
reacher-medium	**−3**±0	−3±0	−4±0	−3±0
swimmer-expert	**227**±51	137±1	155±11	81±28
swimmer-medium	**144**±11	135±1	119±20	75±5
walker2d-expert	**6922**±24	6882±9	4336±218	2089±762
walker2d-medium	**6340**±42	6317±25	5837±114	5670±267
walker2d-simple	5383±103	**5756**±134	176±229	4585±33

Table 13: Area under the learning curve (AULC) on continuous-control environments with external data. All experiments have an interaction budget of 1M. Values are averaged over evaluation repetitions and 10 seeds. ± denotes the standard deviation across seeds. Highest mean is bold.

MuJoCo-v5	AEA (ours)	RLPD (Ball et al., 2023)	Cal-QL (Nakamoto et al., 2023)	SACfD (Ball et al., 2023)
ant-expert	6161±184	**6904**±36	3649±2650	3847±77
ant-medium	6262±117	**6582**±31	−1357±1947	3981±62
ant-simple	6477±79	**6743**±53	4390±83	3742±57
halfcheetah-expert	**15762**±95	14713±762	5925±389	1090±524
halfcheetah-medium	**14747**±73	12951±1474	8633±392	7269±618
halfcheetah-simple	**11988**±77	11144±92	7849±83	8551±68
hopper-expert	**3306**±158	3247±199	1745±212	1722±131
hopper-medium	3051±278	**3235**±100	3159±312	2029±136
hopper-simple	2770±479	**3007**±338	2904±116	2173±80
humanoid-expert	**8026**±189	3206±408	3588±290	163±61
humanoid-medium	**6730**±233	3511±348	5992±299	225±68
humanoid-simple	5396±490	**6252**±153	5574±131	4281±122
pusher-expert	**−26**±0	**−26**±0	−38±4	−28±0
pusher-medium	**−25**±0	**−25**±0	−37±2	−29±0
reacher-expert	−4±1	−4±0	**−3**±0	−4±0
reacher-medium	**−3**±0	−4±0	−4±0	−4±0
swimmer-expert	**260**±76	128±12	161±22	98±61
swimmer-medium	**154**±62	123±6	118±46	70±8
walker2d-expert	6238±79	**6297**±134	3223±527	1158±650
walker2d-medium	5711±281	**5997**±57	4726±232	3123±1049
walker2d-simple	4859±149	**5223**±279	245±321	3415±73

E CALIBRATION ANALYSIS AND ABLATIONS

We examine three aspects of AEA's calibration: how the critic-reference and actor-reference coefficients evolve across learning settings (Section E.1), whether summaries of these trajectories remain useful when deployed as fixed coefficients in independent runs (Section E.2), and how sensitive learning is to the initialization of the critic calibration (Section E.3). The first analysis concerns calibration within an AEA run, whereas the second separates calibration from deployment.

E.1 CALIBRATION TRAJECTORIES ACROSS LEARNING SETTINGS

We report the trajectories of the critic-reference coefficient $\bar{\kappa}$ and actor-reference coefficient κ produced by the same AEA update across the three learning settings. The plots show the coefficient values and moving averages for each seed. These trajectories describe calibration within an AEA run; Section E.2 separately tests whether the resulting coefficient information remains useful when fixed in independent runs.

Interactive learning. Figures 2 and 3 show the coefficient trajectories on MuJoCo and DMC under $(N = 2, G = 1)$. On most tasks, $\bar{\kappa}$ initially becomes less negative before shifting to a more negative late-training value. The timing of this shift differs across tasks and seeds, but many tasks end with closely grouped critic coefficients. The actor coefficients vary less across seeds, while their magnitudes remain task dependent: for example, the upward trend in κ is more pronounced on HalfCheetah than on Hopper.

Sample-efficient learning. Figures 4 and 5 show the coefficient trajectories on MuJoCo and DMC under $(N = 10, G = 20)$. The critic coefficients generally remain closer to zero than the late-training values in the interactive setting. They also show continuing changes and more variation across seeds over the 300K interactions, particularly on HalfCheetah, Walker2d, and Hopper-stand. The actor trajectories remain more tightly grouped, although their magnitudes also evolve during training: κ gradually increases on HalfCheetah, whereas most other tasks show comparatively small changes. Thus, the same AEA update produces different degrees of aggregation and different training trajectories across the two settings.

Learning with external data. Figure 6 shows the coefficient trajectories for Ant, HalfCheetah, Hopper, and Humanoid with simple, medium, and expert external datasets under $(N = 10, G = 20)$. The patterns differ from those in the interactive and sample-efficient settings, with clearer separation between dataset conditions and critic-coefficient changes on some tasks. On Ant, expert data correspond to a critic coefficient closer to zero and a larger actor coefficient than simple or medium data. In HalfCheetah and Humanoid, the critic coefficients continue to move and differ across seeds well into training, even when the actor coefficients are comparatively steady. Interestingly, these continuing critic-coefficient changes occur on tasks where AEA shows particularly strong gains over the baselines for HalfCheetah and Humanoid (Tables 11 to 13). Hopper provides a contrasting pattern: both coefficients follow similar trajectories across all three datasets.

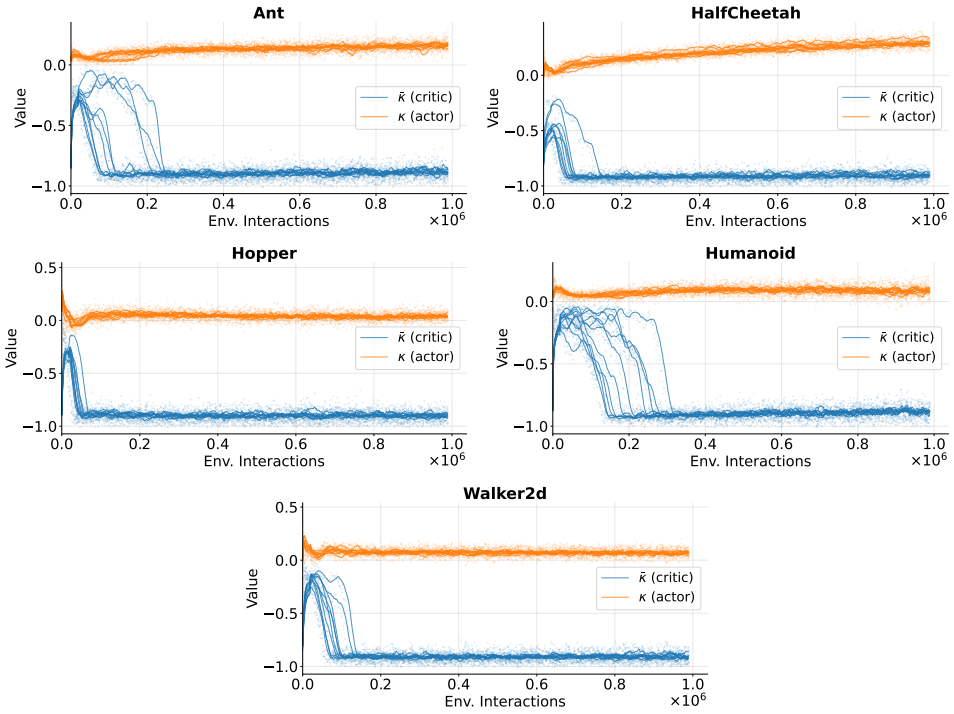

Figure 2: Trajectories of the critic-reference coefficient $\bar{\kappa}$ and actor-reference coefficient κ across seeds on MuJoCo under $(N = 2, G = 1)$. We show the coefficient values and moving averages for each seed.

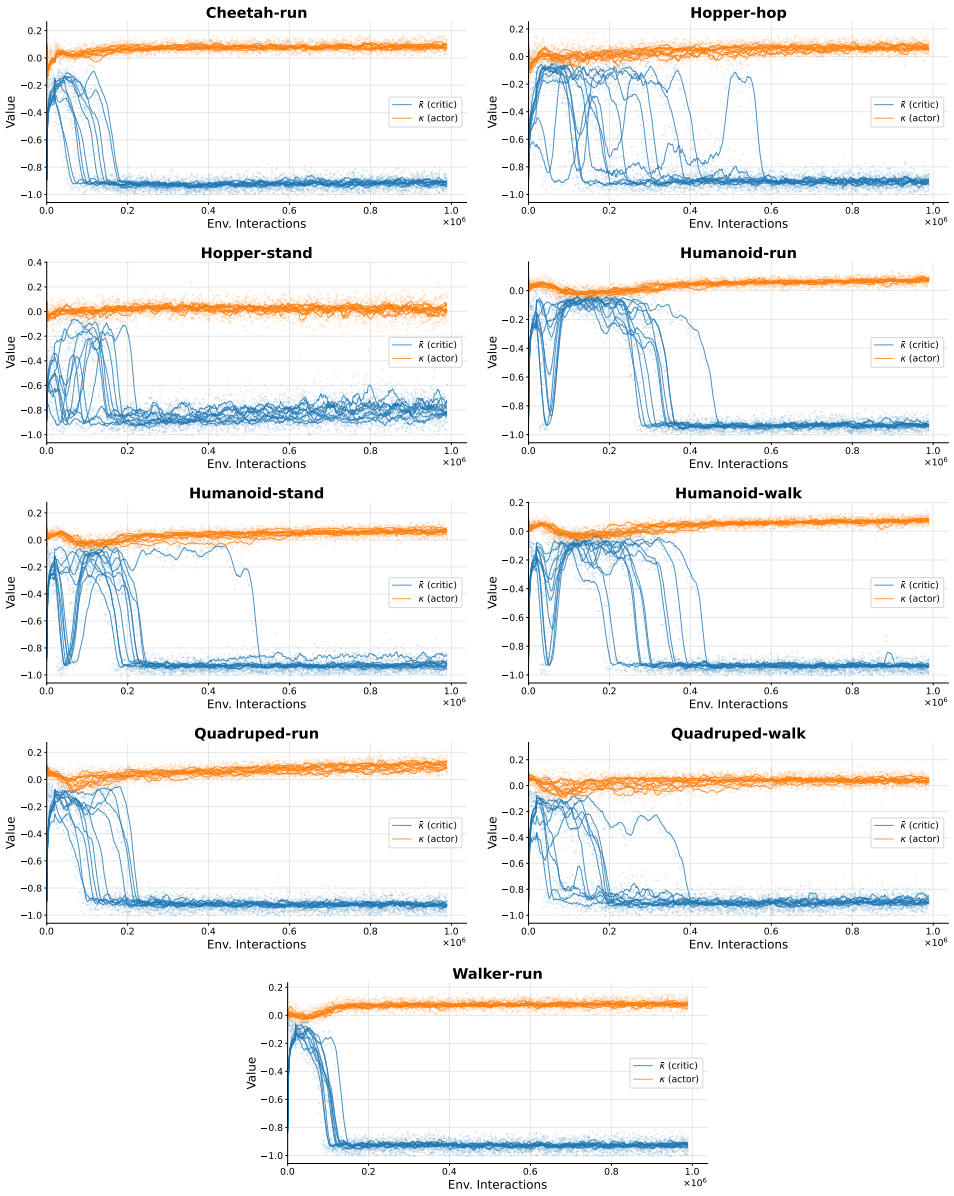

Figure 3: Trajectories of the critic-reference coefficient $\bar{\kappa}$ and actor-reference coefficient κ across seeds on DMC under $(N = 2, G = 1)$. We show the coefficient values and moving averages for each seed.

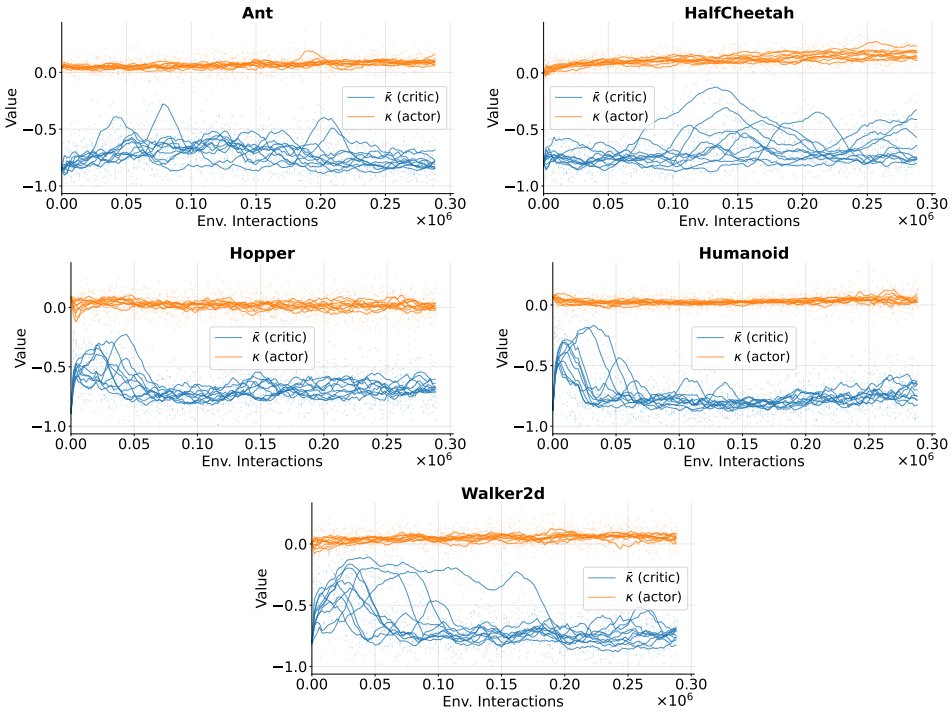

Figure 4: Trajectories of the critic-reference coefficient $\bar{\kappa}$ and actor-reference coefficient κ across seeds on MuJoCo under $(N = 10, G = 20)$. We show the coefficient values and moving averages for each seed.

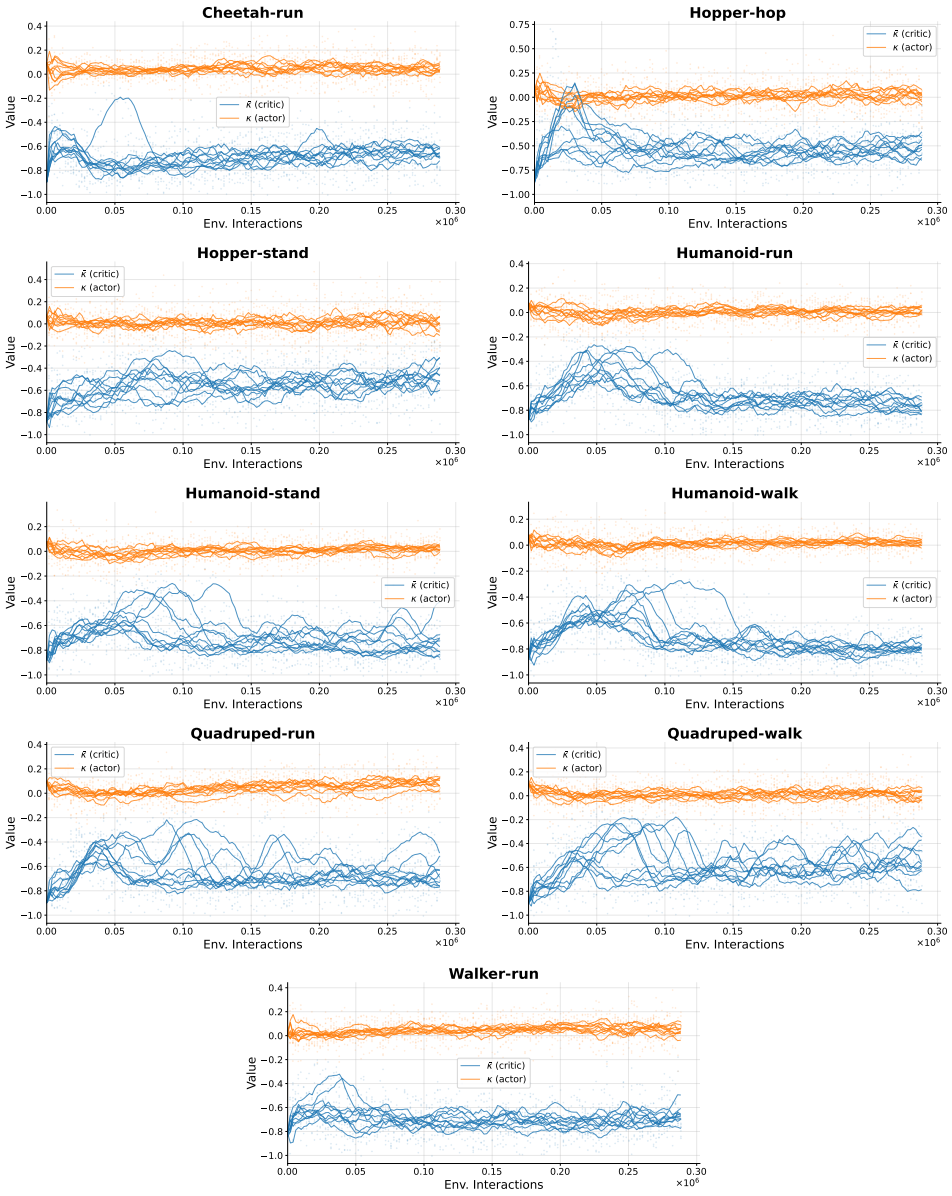

Figure 5: Trajectories of the critic-reference coefficient $\bar{\kappa}$ and actor-reference coefficient κ across seeds on DMC under $(N = 10, G = 20)$. We show the coefficient values and moving averages for each seed.

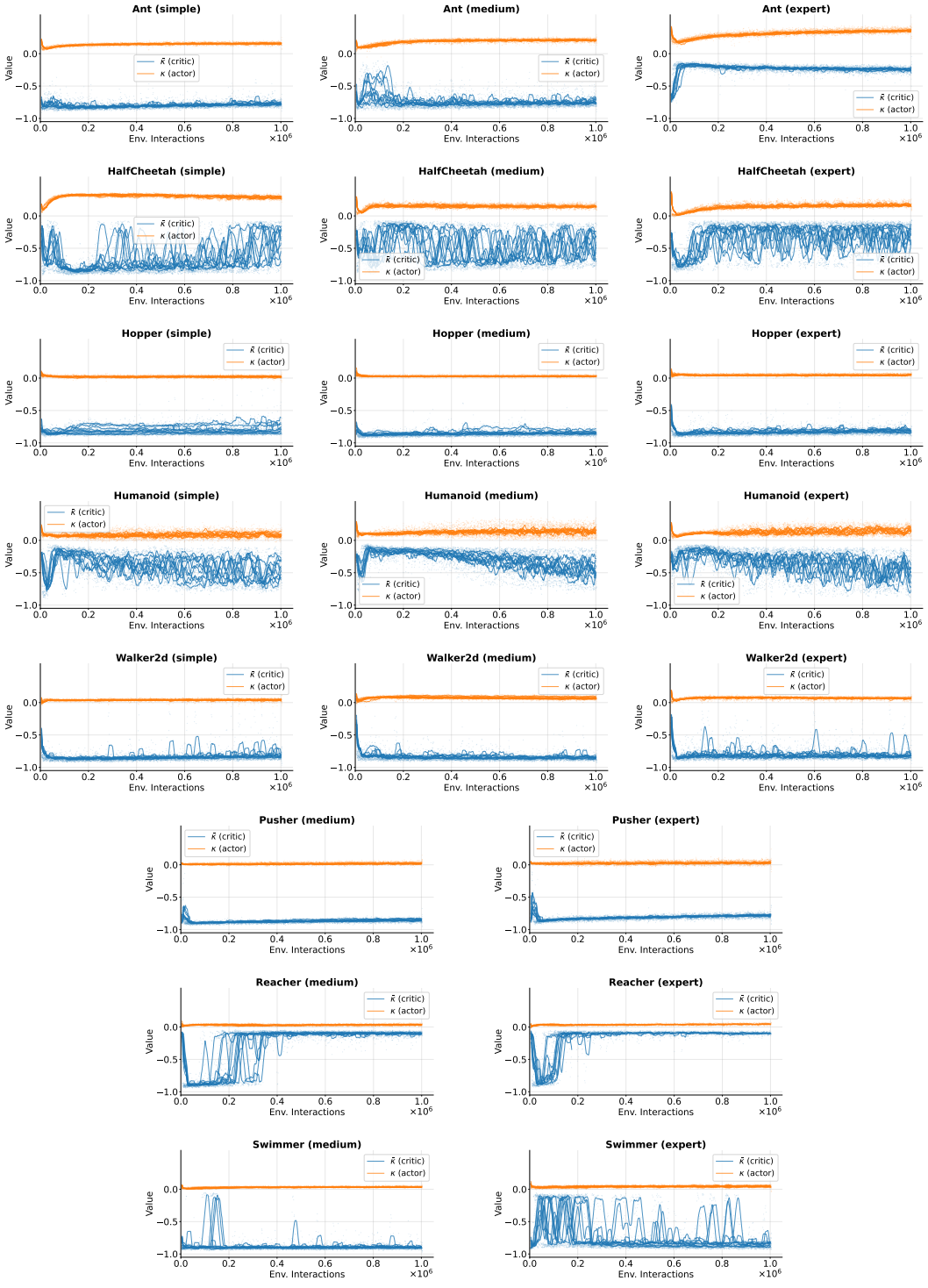

Figure 6: Trajectories of the critic-reference coefficient $\bar{\kappa}$ and actor-reference coefficient κ across seeds on MuJoCo with external data under $(N = 10, G = 20)$. Rows correspond to Ant, HalfCheetah, Hopper, Humanoid, Walker2d, Pusher, Reacher, and Swimmer (top to bottom). The top five rows show simple, medium, and expert external datasets (left to right). The dataset collection does not provide a simple variant for Pusher, Reacher, or Swimmer, so their corresponding panels show only the medium and expert datasets centered on the row. We show the coefficient values and moving averages for each seed.

E.2 Performance with AEA-Informed Fixed Coefficients

We examine two questions raised by AEA: do we need to keep adapting the coefficients during training, and can AEA's calibration mechanism find useful fixed coefficient pairs? To fully answer the first question, we would need to compare AEA with the best fixed pair for each task. Finding such a pair is expensive: even the coarse 5×3 grid used in USAC requires training 15 configurations per task (Tasdighi et al., 2025), and a finer search would require many more runs.

Our comparisons with SAC, SAC-N, and REDQ already provide reference points because these methods prescribe their aggregation rules. In particular, Proposition 1 identifies $(-1, -1)$ with twin-critic SAC's references and $(-1, 0)$ with REDQ's conditional expected critic reference and exact actor reference when $M = 2$. SAC-N uses a minimum over its full ensemble, a prescribed rule that does not generally correspond to a single constant pair in our parameterization for $N > 2$. We now add one more fixed pair for each task condition, chosen in hindsight from AEA's coefficient trajectories, and examine whether it remains useful in independent training runs.

Coefficient selection. For each task condition, we obtain an AEA-informed fixed pair by averaging the corresponding AEA coefficient trajectories over their final 300,000 recorded points. When the coefficients have nearly settled, this average is close to the values they have settled around. When they continue to fluctuate, it summarizes their average level over this window. We use the resulting pair from the beginning of independent runs under the corresponding experimental setting and keep both coefficients fixed throughout. The pairs are listed alongside their performance in Tables 14 to 16. This tests whether the values discovered by AEA remain useful when held fixed; it does not search for the best possible constant pair.

Interactive learning. The coefficient trajectories are generally close to their settled levels late in interactive training, so averaging the final 300,000 recorded points yields fixed pairs near those levels (Figures 2 and 3). In independent runs, these pairs achieve higher final returns than SAC and REDQ on several MuJoCo tasks (Tables 5 and 14). This shows that AEA can identify useful constants, although fixing them does not preserve adaptive performance across the full benchmark.

Table 14: Final return, IQM, and AULC in interactive learning with 2 critics, a UTD ratio of 1, and an interaction budget of 1M. Adaptive denotes AEA; Fixed denotes an AEA-chosen pair held fixed throughout training. ± denotes the standard deviation across seeds. Highest mean between Adaptive and Fixed is bold.

		Final return		IQM		AULC	
Task	$(\bar{\kappa}, \kappa)$	Adaptive	Fixed	Adaptive	Fixed	Adaptive	Fixed
MuJoCo-v5							
Ant	(-0.89, 0.16)	5355±1126	**5491**±1325	5708±200	**6206**±846	3294±542	**4028**±900
HalfCheetah	(-0.91, 0.27)	**10595**±947	9057±1255	**11007**±70	9632±799	**8486**±845	7215±825
Hopper	(-0.90, 0.03)	**3425**±379	3242±41	**3531**±22	3232±28	**2924**±145	1944±481
Humanoid	(-0.88, 0.10)	4780±1295	**5092**±214	**5286**±35	5188±138	**3713**±286	3640±271
Walker2d	(-0.90, 0.06)	4270±1350	**4430**±170	**4674**±155	4443±104	3295±521	**3310**±133
DMC							
Cheetah-run	(-0.92, 0.08)	**802**±30	664±22	**797**±13	657±15	**628**±21	541±5
Hopper-hop	(-0.88, 0.05)	**219**±44	111±28	**228**±8	98±18	**150**±41	70±23
Hopper-stand	(-0.82, 0.02)	**789**±305	741±211	**925**±16	797±152	**606**±189	505±262
Humanoid-run	(-0.91, 0.06)	**159**±36	55±54	**164**±9	51±42	**105**±12	13±13
Humanoid-walk	(-0.91, 0.06)	**517**±92	248±161	**529**±14	275±106	**325**±26	66±70
Quadruped-run	(-0.92, 0.08)	**823**±67	677±222	**831**±28	790±145	**634**±77	547±132
Quadruped-walk	(-0.89, 0.04)	914±164	**915**±63	**945**±15	943±41	**782**±58	766±78
Walker-run	(-0.93, 0.08)	**756**±33	663±69	**764**±8	693±44	**641**±37	573±68

Sample-efficient learning. The critic coefficients often continue to change over the 300K-interaction horizon (Figures 4 and 5). Adaptive AEA achieves higher reported final return than REDQ and SAC-N on most tasks, and higher final return and IQM than the AEA-informed Fixed variant on most tasks (Tables 8, 9 and 15). The selected fixed pairs can outperform the prescribed

aggregation baselines, but usually do not reproduce adaptive AEA's final performance. This pattern is consistent with a benefit from adjusting aggregation during training. However, this does not rule out the possibility of another fixed pair that could perform well though.

Table 15: Final return, IQM, and AULC in sample-efficient learning with 10 critics, a UTD ratio of 20, and an interaction budget of 300K. Adaptive denotes AEA; Fixed denotes an AEA-chosen pair held fixed throughout training. ± denotes the standard deviation across seeds. Highest mean between Adaptive and Fixed is bold.

		Final return		IQM		AULC	
Task	$(\bar{\kappa}, \kappa)$	Adaptive	Fixed	Adaptive	Fixed	Adaptive	Fixed
MuJoCo-v5							
Ant	(-0.72, 0.07)	**4485**$\pm$1044	3823$\pm$1446	**4744**$\pm$260	3959$\pm$980	**2343**$\pm$616	1702$\pm$790
HalfCheetah	(-0.68, 0.12)	**9721**$\pm$817	9506$\pm$1275	**9661**$\pm$395	8885$\pm$805	7214$\pm$459	**7479**$\pm$797
Hopper	(-0.68, 0.02)	**3326**$\pm$583	1992$\pm$784	**3555**$\pm$32	1912$\pm$547	**2723**$\pm$322	1936$\pm$472
Humanoid	(-0.75, 0.03)	**5365**$\pm$446	3349$\pm$1384	**5376**$\pm$87	3490$\pm$957	**2954**$\pm$523	2230$\pm$147
Walker2d	(-0.66, 0.05)	**4579**$\pm$815	4226$\pm$1695	4652$\pm$267	**4805**$\pm$1103	**2992**$\pm$620	2186$\pm$891
DMC							
Cheetah-run	(-0.69, 0.04)	**823**$\pm$58	556$\pm$51	**827**$\pm$9	547$\pm$32	**627**$\pm$42	432$\pm$45
Hopper-hop	(-0.52, 0.01)	130$\pm$102	**264**$\pm$35	86$\pm$75	**262**$\pm$23	74$\pm$62	**174**$\pm$34
Hopper-stand	(-0.54, 0.01)	779$\pm$333	**925**$\pm$18	**935**$\pm$16	925$\pm$13	567$\pm$166	**807**$\pm$19
Humanoid-run	(-0.67, 0.00)	**145**$\pm$15	98$\pm$54	**144**$\pm$5	116$\pm$34	**80**$\pm$16	37$\pm$28
Humanoid-walk	(-0.71, 0.01)	**478**$\pm$57	440$\pm$36	**484**$\pm$14	440$\pm$24	**239**$\pm$51	200$\pm$42
Quadruped-run	(-0.64, 0.04)	823$\pm$124	**824**$\pm$19	**856**$\pm$19	824$\pm$12	620$\pm$148	**695**$\pm$60
Quadruped-walk	(-0.58, 0.01)	**935**$\pm$58	**935**$\pm$13	**947**$\pm$14	935$\pm$9	744$\pm$59	**833**$\pm$43
Walker-run	(-0.69, 0.05)	**705**$\pm$100	675$\pm$80	**741**$\pm$27	709$\pm$54	570$\pm$80	**578**$\pm$73

Learning with external data. Both adaptive AEA and the AEA-informed Fixed pairs achieve higher reported final return and IQM than the strongest baseline on a majority of task-dataset conditions (Tables 11, 12 and 16). Their comparison is mixed: Fixed has higher reported final return more often, while adaptive AEA remains ahead in other conditions. Each external dataset yields a different coefficient trajectory and hence a different hindsight pair; some trajectories approach steady levels, while others continue to move (Figure 6). This variation may help explain why a fixed pair works well in some conditions but not others, although apparent trajectory stability alone does not predict which variant performs better.

Table 16: Final return, IQM, and AULC in learning with external data using 10 critics, a UTD ratio of 20, and an interaction budget of 1M. Adaptive denotes AEA; Fixed denotes an AEA-chosen pair held fixed throughout training. ± denotes the standard deviation across seeds. Highest mean between Adaptive and Fixed is bold.

		Final return		IQM		AULC	
Task	$(\bar{\kappa}, \kappa)$	Adaptive	Fixed	Adaptive	Fixed	Adaptive	Fixed
MuJoCo-v5							
ant-expert	(-0.24, 0.35)	**7015**±310	6499±462	**7116**±137	6436±285	**6161**±184	6136±188
ant-medium	(-0.75, 0.21)	**7064**±70	6830±250	**7080**±23	6908±168	6262±117	**6307**±88
ant-simple	(-0.78, 0.16)	6900±191	**7065**±79	6970±67	**7062**±59	6477±79	**6542**±60
halfcheetah-expert	(-0.31, 0.16)	**17120**±101	17036±19	**17127**±40	17038±13	**15762**±95	15670±56
halfcheetah-medium	(-0.47, 0.14)	**15779**±91	15515±49	**15769**±44	15510±34	**14747**±73	14550±61
halfcheetah-simple	(-0.54, 0.29)	**13505**±295	13335±233	**13467**±116	13384±168	11988±77	11988±188
hopper-expert	(-0.81, 0.04)	3754±384	**4069**±99	3819±180	**4108**±64	3306±158	**3447**±88
hopper-medium	(-0.84, 0.03)	3283±569	**3299**±439	3258±243	**3295**±283	**3051**±278	2845±100
hopper-simple	(-0.79, 0.02)	2736±710	**3100**±735	2793±333	**3397**±463	**2770**±479	2709±421
humanoid-expert	(-0.45, 0.15)	9832±424	**10243**±106	9933±153	**10202**±68	8026±189	**8741**±61
humanoid-medium	(-0.41, 0.14)	8753±510	**9267**±157	8746±218	**9331**±101	6730±233	**7640**±209
humanoid-simple	(-0.48, 0.09)	6831±1227	**8153**±400	7249±426	**8194**±300	5396±490	**6549**±233
pusher-expert	(-0.79, 0.03)	−24±1	**−19**±1	−24±0	**−19**±1	−26±0	**−22**±0
pusher-medium	(-0.85, 0.02)	−23±1	**−20**±1	−23±0	**−21**±0	−25±0	**−23**±0
reacher-expert	(-0.26, 0.04)	−3±0	−3±0	−3±0	−3±0	−3±0	−3±0
reacher-medium	(-0.10, 0.03)	−3±0	−3±0	−3±0	−3±0	**−3**±0	−4±0
swimmer-expert	(-0.81, 0.05)	235±97	**300**±89	227±51	**351**±58	260±76	**327**±37
swimmer-medium	(-0.89, 0.03)	**163**±63	143±4	**144**±11	142±2	**154**±62	126±10
walker2d-expert	(-0.81, 0.07)	6913±70	**6964**±40	6922±24	**6977**±25	6238±79	**6254**±87
walker2d-medium	(-0.84, 0.07)	6324±92	**6336**±73	**6340**±42	6321±50	5711±281	**5779**±223
walker2d-simple	(-0.82, 0.04)	**5460**±259	5275±381	**5383**±103	5308±249	**4859**±149	4584±441

Interpretation. These experiments compare one AEA-informed fixed pair per condition with adaptive AEA, rather than searching for the best constant pair. AEA's trajectories sometimes provide effective fixed choices, while the selected pairs do not consistently reproduce adaptive performance. When the coefficients continue to move, a late-training average cannot represent the different aggregation choices made earlier in training. Conversely, holding an effective average fixed may avoid residual fluctuations. These are possible explanations of the mixed results, not causal conclusions; the comparisons do not establish that adaptation is necessary or rule out other fixed pairs.

E.3 Initialization Sensitivity of the Critic Calibration

We next study sensitivity to the initialization of the critic calibration, focusing on the parameter $\overline{\kappa}$. We vary the initial value of $\overline{\kappa}$ while keeping the actor initialization fixed at $\kappa = 0$, which is neutral with respect to conservatism or optimism. This isolates the effect of the critic initialization and avoids confounding interactions between the two calibrations at the start of training.

All experiments use the protocols and hyperparameters in Tables 3 and 4 and are reported for seed 1 on MuJoCo for illustrative purposes. Figure 7 shows results for Ant, Hopper, Humanoid, and Walker2d in the interactive and sample-efficient settings, with $\overline{\kappa}$ initialized to -0.5, -0.6, -0.7, -0.8, or -0.9.

Effective learning is not restricted to the default initialization $\overline{\kappa} = -0.8$: AEA improves returns across all tested initializations, and alternative initial values can match or exceed the default on individual tasks. Even when the initial calibration is not well matched to a task, the learned update adjusts it during training, and many initializations achieve performance competitive with, and in several cases better than, SAC, DRND, DSAC, and REDQ under the same setting. Initialization still affects learning speed and final return, with differences varying across tasks and settings.

If initialization sensitivity is a practical concern in a particular setting, there are two possible mitigations. One can choose a more suitable initialization for $\overline{\kappa}$ based on a small pilot run. Alternatively, increasing the minibatch size used for updating κ and $\overline{\kappa}$, for example from 256 to 512, may reduce gradient noise in these scalar updates and yield smoother calibration trajectories.

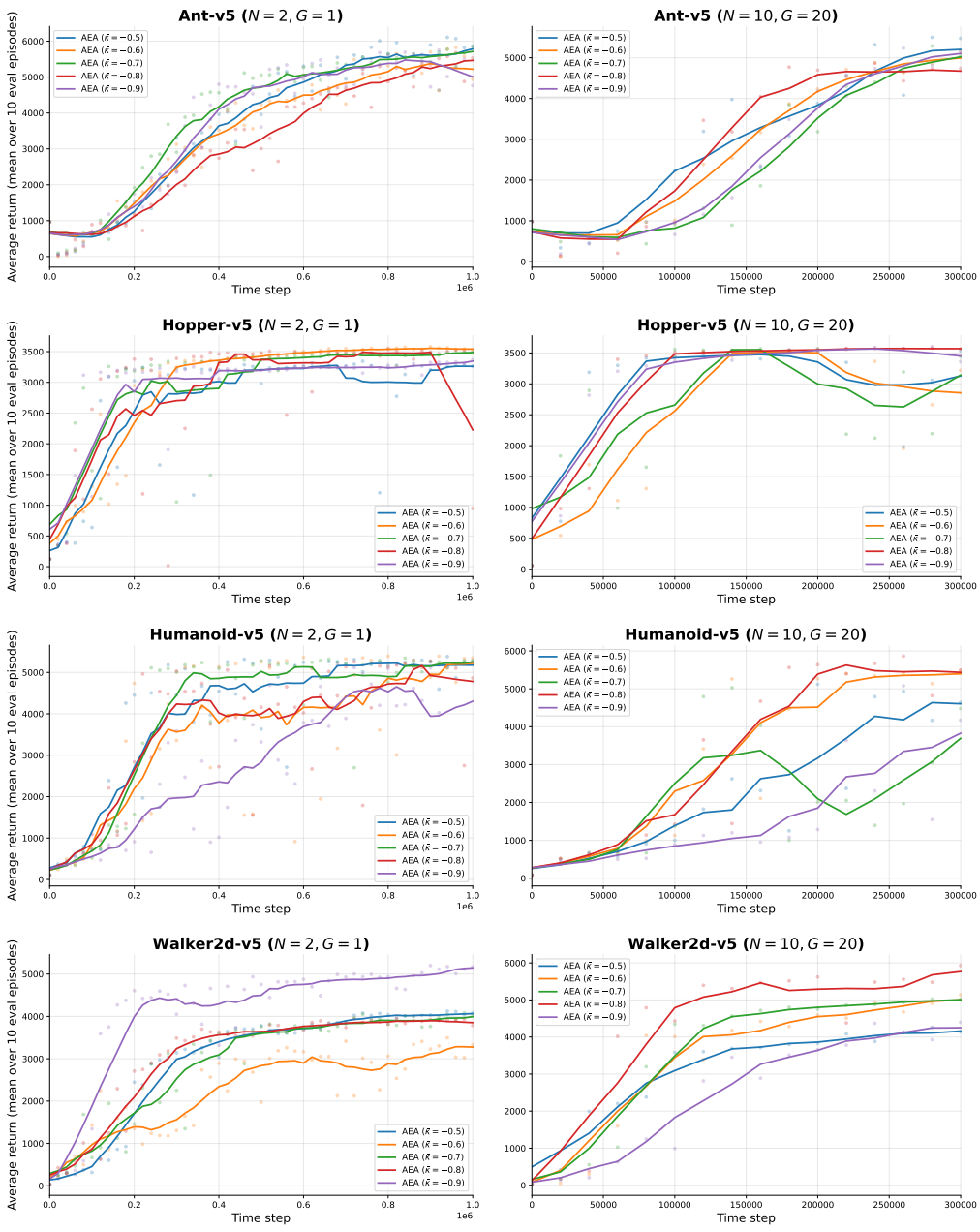

Figure 7: Effect of initializing the critic-side ensemble aggregation parameter $\overline{\kappa}$ on AEA's performance across MuJoCo environments. The left column shows results under $(N = 2, G = 1)$, while the right column corresponds to $(N = 10, G = 20)$. Each line is a moving average over the returns of a single seed.