ALEATORIC IMPULSE DAMPING VIA DIRECT EVIDENTIAL REGRESSION

Anonymous authors
Paper under double-blind review

ABSTRACT

Off-policy reinforcement learning can suffer from aleatoric impulse, a transient surge in value overestimation when stochastic exploration interacts with limited data. Suppressing this impulse requires aleatoric value uncertainty, a higher-order uncertainty of the learned value estimate itself. However, conventional distributional reinforcement learning models only lower-order aleatoric return uncertainty arising from the variability of return samples. Existing methods either overlook this distinction or heuristically construct value uncertainty from return uncertainty through scaling and annealing, leading to hyperparameter sensitivity and loss of state–action dependence. To address this issue, we propose **D**irect **E**videnti**A**l **R**egression (**DEAR**), a general framework for directly estimating the magnitude of higher-order uncertainty in parametric estimators. DEAR leverages temporal prediction variation and an exponential moving average anchor to learn this uncertainty. Building upon DEAR, we develop the **A**leatoric **I**mpulse **D**amping via Direct **E**vidential **R**egression (**AIDER**) algorithm, which augments a distributional critic with an evidential head to directly estimate value estimation uncertainty. Experiments on continuous-control benchmarks show that AIDER achieves state-of-the-art performance.

1 INTRODUCTION

Deep reinforcement learning has achieved remarkable success across a wide range of complex sequential decision-making tasks (Silver et al., 2017; Vinyals et al., 2019; Guan et al., 2021; Rudin et al., 2022; Duan et al., 2024; Guan et al., 2026). However, addressing the severe overestimation bias of action-values remains a fundamental challenge for stable off-policy optimization (Thrun & Schwartz, 2014). The root cause of this vulnerability lies in the intrinsic uncertainty associated with empirical value estimation. While modern ensemble-based algorithms have proven highly effective at mitigating epistemic value uncertainty, which arises from structural model inaccuracies, they fundamentally struggle with the aleatoric value uncertainty arising from environmental and policy stochasticity under finite data (Hüllermeier & Waegeman, 2021). During the early stages of training, the resonance of high policy stochasticity with severe data scarcity acutely amplifies this intrinsic noise. This amplification triggers a transient but destructive overestimation surge, defined as the *aleatoric impulse*, which directly misleads the agent toward suboptimal actions (Yu et al., 2026).

Learning a return distribution alone does not resolve the estimation problem underlying the aleatoric impulse. The central challenge lies in the mismatch between uncertainty levels: aleatoric value uncertainty is a higher-order uncertainty associated with the learned value estimate itself, whereas conventional distributional reinforcement learning directly models only aleatoric return uncertainty, a lower-order uncertainty arising from the intrinsic variability of return samples. Methods such as DSAC-T (Duan et al., 2025) and IV-RL (Mai et al., 2022) operate only at the return-uncertainty level, using it for gradient modulation. DoubleGum (Hui et al., 2023) and AID (Yu et al., 2026) consider value uncertainty, but construct it indirectly from the learned return uncertainty for explicit penalization. This proxy construction relies on fixed or scheduled global scaling coefficients, leading to hyperparameter sensitivity and failing to fully preserve the fine-grained dependence of value uncertainty on individual state–action pairs.

Fundamentally resolving these structural bottlenecks necessitates a paradigm shift toward directly estimating higher-order aleatoric value uncertainty. Evidential learning is naturally suited to this

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.

task because it models a distribution over predictive-distribution parameters, enabling a network to quantify the uncertainty of its own estimates (Gao et al., 2025). However, existing evidential learning frameworks are predominantly designed for discrete spaces, such as classification tasks (Sensoy et al., 2018; Charpentier et al., 2020). Their extensions to continuous regression rely on heuristic formulations with well-recognized structural deficiencies, preventing the resulting uncertainty from being calibrated on an absolute scale (Amini et al., 2020; Meinert et al., 2023; Wu et al., 2024). Consequently, they cannot accurately estimate the magnitude of higher-order aleatoric uncertainty.

To transcend these limitations, we propose **D**irect **E**videnti**A**l **R**egression (**DEAR**), a general framework for directly estimating the magnitude of higher-order uncertainty in parametric estimators. DEAR leverages temporal prediction variation and an exponential moving average (EMA) anchor to learn this uncertainty, avoiding heuristic evidence regularization. By seamlessly embedding the DEAR operator into AID, we further propose the **AIDER** (**A**leatoric **I**mpulse **D**amping via Direct **E**vidential **R**egression) mechanism. Within this fully integrated framework, the directly learned aleatoric uncertainty is added to the epistemic uncertainty to obtain the total value estimation uncertainty. This uncertainty is then utilized for overestimation penalization and exploration encouragement, completely eliminating the reliance on complex heuristic annealing.

The main contributions of this work are twofold: (1) We propose a generalized **DEAR** operator. To the best of our knowledge, DEAR is the first method to directly and accurately estimate the absolute magnitude of higher-order aleatoric uncertainty for arbitrary parametric estimators, free of heuristic approximations. (2) Building upon this generalized operator, we develop the **AIDER** mechanism. By directly learning the aleatoric value uncertainty, AIDER gracefully tames the aleatoric impulse, achieving state-of-the-art performance while reducing its reliance on heuristic uncertainty-scaling hyperparameters.

2 PRELIMINARIES

2.1 DEEP EVIDENTIAL REGRESSION

Deep Evidential Regression (DER) places a higher-order prior over the parameters of a lower-order likelihood, allowing a single network forward pass to represent uncertainty at both levels (Amini et al., 2020). For a scalar regression target, DER assumes a lower-order Gaussian likelihood $y \sim \mathcal{N}(\mu, \sigma^2)$ and models its unknown parameters with a higher-order Normal–Inverse-Gamma (NIG) evidential distribution $(\mu, \sigma^2) \sim \mathrm{NIG}(\gamma, \nu, \alpha, \beta)$. Here, γ is the predictive mean, while ν and 2α act as virtual evidence counts for the mean and variance, respectively. The total evidence is therefore $e = \nu + 2\alpha$. The predictive mean, lower-order uncertainty, and higher-order uncertainty are

$$\hat{y} = \mathbb{E}[\mu] = \gamma, \qquad \sigma^2_{\text{low}} = \mathbb{E}[\sigma^2] = \frac{\beta}{\alpha - 1}, \qquad \sigma^2_{\text{high}} = \mathrm{Var}[\mu] = \frac{\beta}{\nu(\alpha - 1)}. \tag{1}$$

Here, higher-order uncertainty characterizes how the prediction at a fixed input varies across identically configured neural networks trained on different finite datasets under the same training procedure. The network predicts $(\gamma, \nu, \alpha, \beta)$, and DER is trained by minimizing the negative log marginal likelihood with an evidence regularizer. The complete DER objective is

$$\mathcal{L}_{\text{DER}} = \mathcal{L}_{\text{NLL}}(\gamma, \nu, \alpha, \beta) + \lambda \, |y - \gamma| \, e, \tag{2}$$

where λ controls the penalty for assigning excessive evidence to inaccurate predictions. This regularizer is heuristic and its coefficient affects the absolute scale of the resulting uncertainty.

2.2 UNCERTAINTY IN REINFORCEMENT LEARNING

2.2.1 PROBLEM FORMULATION

Our setting is an entropy-regularized MDP with tuple $(\mathcal{S}, \mathcal{A}, p, r, \gamma)$ (Li, 2023). At state $s \in \mathcal{S}$, the agent draws an action $a \in \mathcal{A}$ from $\pi(a|s)$; rewards and successor states are specified by $r(s, a)$ and $p(s'|s, a)$, respectively. Policy learning optimizes $J(\pi) = \mathbb{E}[\sum_{t=0}^{\infty} \gamma^t (r_t + \alpha \mathcal{H}(\pi(\cdot|s_t)))]$. Here, γ discounts future outcomes, and the temperature α weights policy entropy relative to reward.

2.2.2 Uncertainty Decomposition

In off-policy reinforcement learning, the true action-value is the expectation of the return distribution, $\tilde{Q}(s,a) = \mathbb{E}[Z(s,a)]$, but its empirical estimate remains uncertain under finite data and function approximation. The total value uncertainty can be decomposed following AID (Yu et al., 2026) as

$$\mathrm{Var}[Q(s,a)] \approx \underbrace{\mathrm{Var}_\theta[Q_\theta(s,a)]}_{\text{Epistemic}} + \underbrace{\frac{1}{n(s,a)} \underbrace{\mathbb{E}_\theta[\mathrm{Var}[Z_\theta(s,a)]]}_{\text{Aleatoric uncertainty of } Z}}_{\text{Aleatoric uncertainty of } Q}, \tag{3}$$

Here, n essentially characterizes the evidence count, reflecting the effective number of visits to the specific state-action pair (s,a). The first term captures epistemic uncertainty from critic disagreement, while $\mathrm{Var}(Z_\theta)$ captures the lower-order aleatoric return uncertainty induced by environmental and policy stochasticity. The factor $1/n(s,a)$ bridges this lower-order uncertainty to the higher-order aleatoric uncertainty of the learned value.

2.2.3 Higher-order Uncertainty Construction

Based on this decomposition, AID (Yu et al., 2026) models each return distribution as a Gaussian, $Z_{\theta_i}(s,a) \sim \mathcal{N}(Q_{\theta_i}(s,a), \sigma^2_{\mathrm{ale-z},\theta_i}(s,a))$, and defines the value estimate as the ensemble mean $Q(s,a) = m^{-1}\sum_i Q_{\theta_i}(s,a)$. AID then constructs the higher-order aleatoric uncertainty by scaling the lower-order return uncertainty with a coefficient β:

$$\tilde{\sigma}^2_{\mathrm{ale-q}}(s,a) = \beta\, \mathbb{E}_{\theta_i}\left[\sigma^2_{\mathrm{ale-z},\theta_i}(s,a)\right]. \tag{4}$$

The coefficient β is adjusted in a closed loop according to the observed value-estimation bias. It captures the global trend of the inverse evidence count, but cannot preserve its state–action dependence. The resulting total value uncertainty is

$$\sigma_Q(s,a) = \sqrt{\sigma^2_{\mathrm{epi}}(s,a) + \tilde{\sigma}^2_{\mathrm{ale-q}}(s,a)}, \tag{5}$$

$$\sigma^2_{\mathrm{epi}}(s,a) = \frac{1}{m-1}\sum_{i=1}^{m}\left(Q_{\theta_i}(s,a) - Q(s,a)\right)^2. \tag{6}$$

This uncertainty is subsequently used to suppress overestimation and encourage exploration.

3 Direct Evidential Regression

3.1 Temporal Surrogate Theory

To bypass the heuristic evidence regularization used in conventional evidential regression, we propose **D**irect **E**videnti**A**l **R**egression (**DEAR**). Consider an estimator $y_{\mathcal{D}_n}(x)$ obtained by applying a fixed training procedure to a dataset $\mathcal{D}_n$ of size n. At a fixed input x, its higher-order uncertainty is defined as

$$\sigma^2_{\mathrm{high}}(x;n) = \mathrm{Var}_{\mathcal{D}_n}\left[y_{\mathcal{D}_n}(x)\right], \tag{7}$$

where the variance is taken over datasets independently drawn from the same distribution and used to train identically configured estimators. Estimating Eq. 7 directly is expensive because it requires additional datasets and repeated training. Variation within one mini-batch is not a valid substitute either: its samples generally correspond to different inputs, so it does not reveal the amount of evidence local to x.

DEAR instead exploits the repeated local evidence carried by stochastic updates. Different random mini-batch sequences generate different estimator realizations under the same sampling law as repeatedly drawn finite datasets; local stationarity and ergodicity then allow their ensemble moments to be estimated from one training trajectory. Let θ_t denote the estimator parameters after update t. The *temporal uncertainty surrogate* is defined directly as

$$\sigma^2_{\mathrm{temp}}(x) := \mathrm{Var}_t\left[y_{\theta_t}(x)\right] = \mathbb{E}_t\left[\left(y_{\theta_t}(x) - \mathbb{E}_t[y_{\theta_t}(x)]\right)^2\right]. \tag{8}$$

Here, $\mathbb{E}_t$ and Var_t take moments over the locally stationary training dynamics. The definition therefore uses repeated predictions at the same input and preserves the input dependence that cannot be recovered from heterogeneous samples within a mini-batch.

To connect this temporal variation to standard optimization quantities, let η be the SGD learning rate, B the mini-batch size, and $h(x)$ the local effective curvature of the loss along the prediction direction at x. A local linearization of SGD contracts the previous prediction error by $1 - \eta h(x)$. We therefore define the temporal forgetting rate as $\rho(x) := \eta h(x)$ and write it as ρ when x is fixed. Thus, ρ is induced jointly by the learning rate and local loss geometry rather than introduced as an additional hyperparameter. Let $v(x)$ denote the variance of the prediction contribution from one randomly sampled observation at x. The following result bounds the error between the temporal surrogate and the higher-order uncertainty after t updates.

Theorem 1 (Bounded error of the temporal uncertainty surrogate)**.** *Assume local stationarity and ergodicity, and suppose that the prediction contributions of individual observations at x are independent with variance $v(x)$. If training on all Bt observations is locally equivalent to uniform averaging, whereas the local SGD dynamics satisfy $\rho = \eta h(x) \in (0, 1)$ and exponentially forget previous mini-batches at this rate, then, after removing the initialization contribution and normalizing the observed-data weights,*

$$0 \le \sigma^2_{\mathrm{temp}}(x) - \sigma^2_{\mathrm{high}}(x; Bt) \le \frac{\rho}{2-\rho}\frac{v(x)}{B}. \tag{9}$$

Moreover, the discrepancy is $\mathcal{O}\big(v(x)\rho^2 t/B\big)$ when $\rho t \ll 1$, and approaches the upper bound as $t \to \infty$.

Theorem 1 shows that the trajectory and cross-dataset variances are nearly identical while evidence is accumulating; after the exponential memory saturates, their absolute discrepancy remains uniformly controlled by the update rate. The proof is provided in Appendix A.

3.2 EMA-Based Temporal Uncertainty Learning

DEAR realizes the temporal surrogate with two coupled components: an exponential-moving-average (EMA) target estimates the local temporal mean, while an auxiliary uncertainty head learns the corresponding residual second moment. The EMA target is updated by

$$\bar{\theta}_t = \tau\theta_t + (1-\tau)\bar{\theta}_{t-1}, \qquad \tau \in (0, 1). \tag{10}$$

For a general estimator y_θ, an auxiliary head $\sigma^2_{y,\theta}$ predicts the conditional variance of the temporal residual and is trained with the Gaussian negative log-likelihood

$$\mathcal{L}^{\mathrm{DEAR}}(\theta) = \frac{\big(\mathrm{sg}[y_\theta(x)] - y_{\bar{\theta}}(x)\big)^2}{2\sigma^2_{y,\theta}(x)} + \frac{1}{2}\log\sigma^2_{y,\theta}(x), \tag{11}$$

where $\mathrm{sg}[\cdot]$ stops gradients through the online prediction. When the EMA horizon covers the local mixing time, its prediction approximates the local trajectory mean, i.e., $y_{\bar{\theta}}(x) \approx \mathbb{E}_t[y_{\theta_t}(x)]$. Consequently,

$$\mathbb{E}_t\Big[\big(y_{\theta_t}(x) - y_{\bar{\theta}_t}(x)\big)^2 \mid x\Big] \approx \sigma^2_{\mathrm{temp}}(x), \tag{12}$$

which provides a direct training target for higher-order uncertainty without collecting additional datasets or training additional networks. The approximation error introduced by the EMA anchor is analyzed in Appendix B.

4 Aleatoric Impulse Damping via Direct Evidential Regression

While aleatoric impulse varies drastically across the state-action space depending on local visitations, existing damping mechanisms rely on global proxy variables, failing to capture this microscopic heterogeneity. To resolve this bottleneck, we propose **A**leatoric **I**mpulse **D**amping via Direct **E**vidential **R**egression (**AIDER**) by integrating DEAR.

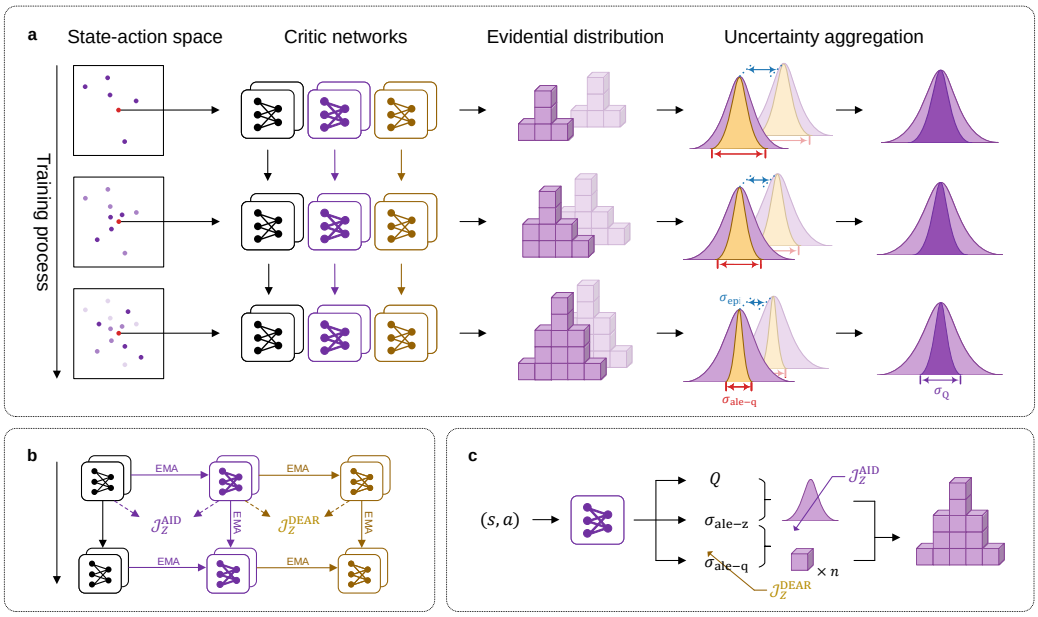


Figure 1: **The critic framework of AIDER. (a) Temporal evolution:** Over training iterations (top to bottom), the accumulation of state-action visitations expands the evidence count n, driving the convergence of the discrete evidential distribution. Consequently, both epistemic uncertainty (σ_{epi}) and aleatoric value uncertainty ($\sigma_{\text{ale-q}}$) naturally diminish, narrowing the total value uncertainty. **(b) Optimization pipeline:** The underlying training workflow of the critic networks in (a). **(c) Single-pass parameterization:** The microscopic implementation of the network forward pass in (a).

4.1 Evidential Critic Augmentation

Within the distributional RL architecture, we restructure the action-value function into an evidential neural network. Alongside the conventional modeling of the return distribution $Z \sim \mathcal{N}(Q_{\theta_i}, \sigma^2_{\text{ale-z},\theta_i})$, the i-th critic network is augmented with an uncertainty head. We consider two parameterizations of aleatoric value uncertainty. The evidence-ratio parameterization, following the DER design, predicts an evidence count $n_{\theta_i}(s,a)$ and defines:

$$\sigma^2_{\text{ale-q},\theta_i}(s,a) = \text{sg}\left[\sigma^2_{\text{ale-z},\theta_i}(s,a)\right] / n_{\theta_i}(s,a). \tag{13}$$

Since direct prediction performs better overall in the ablation study (Table 2), AIDER adopts it as the default parameterization. One possible explanation is that the evidence ratio couples higher-order value uncertainty to the lower-order return variance, whose training fluctuations may therefore propagate into the uncertainty estimate. Given that standard AID uses target networks $\bar{\theta}_i$ for stable backups, $\sigma^2_{\text{ale-q},\theta_i}$ quantifies uncertainty in $Q_{\bar{\theta}_i}$. Following DEAR, we maintain a higher-order EMA target $\hat{\theta}_i$, giving $Q_{\bar{\theta}_i}(s,a) \sim \mathcal{N}(Q_{\hat{\theta}_i}(s,a), \sigma^2_{\text{ale-q},\theta_i}(s,a))$.

4.2 Evidential Critic Learning

Following the proposed DEAR paradigm, we directly optimize this evidential parameterization by formulating it as a Negative Log-Likelihood (NLL) minimization problem. To empirically stabilize the training process and prevent potential gradient explosion or variance collapse, the objective is scaled by the expectation of the learned variance, $\mathbb{E}[\sigma^2_{\text{ale-q},\theta_i}]$. The new evidential loss for the i-th critic network is thus explicitly defined as:

$$\mathcal{J}_Z^{\text{DEAR}}(\theta_i) = \text{sg}\left[\mathbb{E}_{\mathcal{D}}[\sigma^2_{\text{ale-q},\theta_i}]\right] \cdot \mathbb{E}_{\mathcal{D}}\left[\left(\frac{(Q_{\bar{\theta}_i} - Q_{\hat{\theta}_i})^2}{2\sigma^2_{\text{ale-q},\theta_i}} + \frac{1}{2}\log \sigma^2_{\text{ale-q},\theta_i}\right)\right] \tag{14}$$

This pure end-to-end mechanism drives the critic to autonomously calibrate the aleatoric value uncertainty $\sigma^2_{\text{ale-q},\theta_i}(s,a)$ based strictly on the temporal consistency of its predictions. Consequently,

this evidential objective seamlessly integrates with the base distributional reinforcement learning framework. The total loss for optimizing the critic network is simply formulated as the summation of the original value regression loss and the newly introduced evidential loss:

$$\mathcal{J}_Z^{\text{AIDER}}(\theta_i) = \mathcal{J}_Z^{\text{AID}}(\theta_i) + \mathcal{J}_Z^{\text{DEAR}}(\theta_i) \tag{15}$$

4.3 Uncertainty Utilization

Once the explicitly parameterized aleatoric value uncertainty is learned, we first compute the ensemble averages of the value-level and return-level aleatoric variances:

$$\sigma^2_{\text{ale-q}}(s,a) = \mathbb{E}_{\theta_i}[\sigma^2_{\text{ale-q},\theta_i}(s,a)], \qquad \sigma^2_{\text{ale-z}}(s,a) = \mathbb{E}_{\theta_i}[\sigma^2_{\text{ale-z},\theta_i}(s,a)]. \tag{16}$$

Although DEAR directly estimates $\sigma_{\text{ale-q}}$, an individual critic can occasionally produce an anomalously large uncertainty estimate in sparsely visited regions. We therefore apply a state-action-wise clipping rule before combining it with epistemic uncertainty:

$$\tilde{\sigma}_{\text{ale-q}}(s,a) = \min(\sigma_{\text{ale-q}}(s,a),\, \beta\sigma_{\text{ale-z}}(s,a))\,. \tag{17}$$

Here, β is a fixed clipping coefficient, set to 0.3 throughout our experiments. Empirically, the clipping bound is reached only during approximately the first 5,000 steps (about 0.25% of training), limiting excessively large early uncertainty estimates while leaving the learned uncertainty unconstrained for the remainder of training.

The total value uncertainty is then constructed by combining the clipped aleatoric component with the epistemic uncertainty derived from ensemble disagreement:

$$\sigma_Q(s,a) = \sqrt{\sigma^2_{\text{epi}}(s,a) + \tilde{\sigma}^2_{\text{ale-q}}(s,a)}. \tag{18}$$

With $\sigma_Q(s,a)$ derived in this way, its subsequent integration follows the original AID mechanism: it serves as the core penalty for the pessimistic target and as the exploration bonus for the decoupled behavior policy. The clipping rule preserves the robust overestimation suppression of AID while replacing its global annealing schedule with a fixed, architecture-dependent safeguard against abnormal uncertainty values.

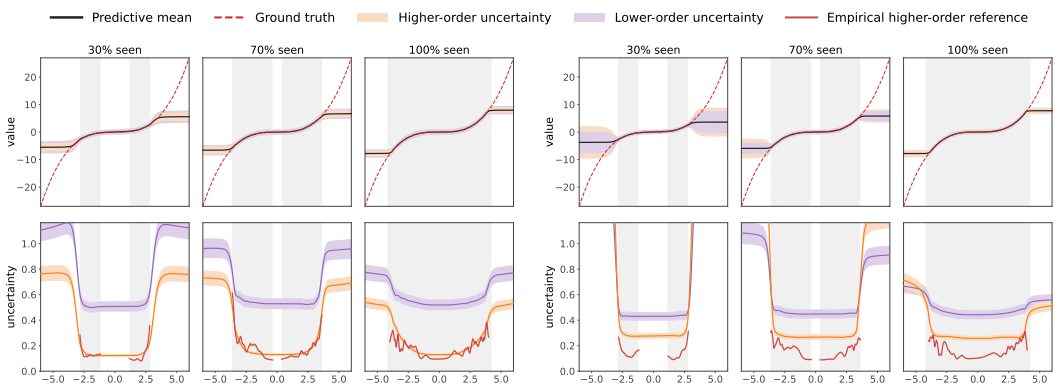
5 Experiments

5.1 Toy Experiments: Validating DEAR

The toy experiment is designed to isolate whether DEAR can recover uncertainty arising from finite and spatially nonuniform evidence, without confounding this question with ensemble disagreement or reinforcement-learning target construction.

Experimental Setup. We use a one-dimensional regression task whose deterministic target is the cubic function $f(x) = x^3/8$. Each observed sample is generated as $y = f(x) + \epsilon$, where $\epsilon \sim \mathcal{N}(0, 0.5^2)$; the observation noise is therefore homoscedastic across the training domain. At the beginning of training, visible sample locations are concentrated around the two points $x = \pm 2$, and the observation process progressively expands outward from these two centers. We train on $x \in [-4, 4]$ and evaluate on the extended domain $x \in [-6, 6]$, allowing unsupported and out-of-distribution (OOD) regions to be inspected separately. To remove the effect of epistemic uncertainty from the comparison, neither method uses an ensemble. We compare DEAR with conventional deep evidential regression (DER), using the regularization coefficient $\lambda = 0.001$ for DER. We train one network for each of 20 data seeds. A data seed changes only the random observation stream and does not change the network initialization. The empirical higher-order uncertainty reference is defined as the standard deviation of the predicted means across these 20 data seeds, computed only at locations with observation support.

Results. As shown in Fig. 2, DEAR's higher-order uncertainty closely tracks the empirical reference at every snapshot where that reference is available, including intermediate stages before the evidence has fully accumulated. It also assigns visibly larger uncertainty to unsupported and OOD regions, thereby detecting the lack of evidence outside the training support. In contrast, DER's



(a) DEAR (ours) (b) DER

Figure 2: **Toy regression comparison between DEAR and DER.** Each column shows a training snapshot as the observed support expands from the two initial centers. The upper row displays the predictive mean and uncertainty bands; the lower row displays the learned uncertainty together with the empirical higher-order reference where observations are available. Gray regions indicate substantial observation support, while white regions are unsupported or OOD and therefore do not provide an empirical reference.

heuristic uncertainty construction also produces elevated uncertainty in OOD regions, so it can signal distributional shift, but its absolute magnitude is not consistently aligned with the empirical reference. Its uncertainty profile also develops an unexpected left-right asymmetry despite the symmetric observation-noise setting. This qualitative mismatch shows that OOD detection alone does not guarantee an uncertainty estimate whose scale agrees with the underlying finite-sample variability.

5.2 Main Experiments: Evaluating AIDER

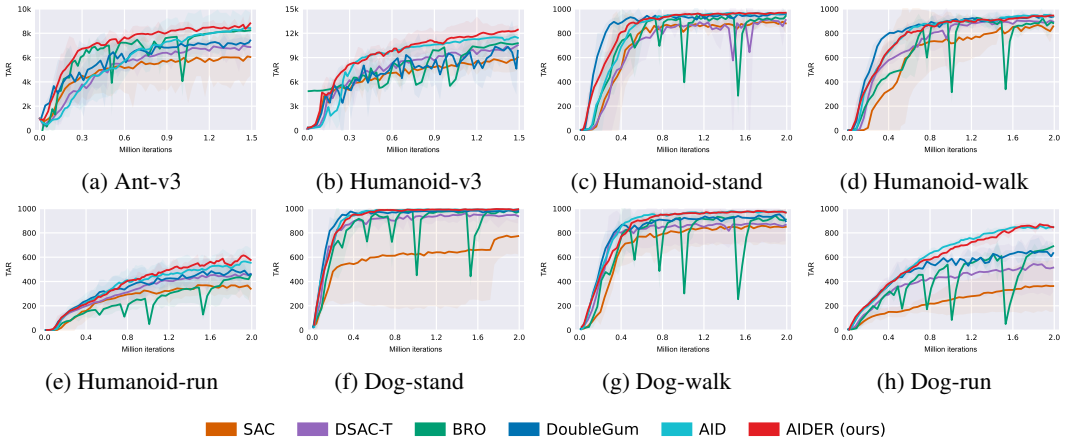
5.2.1 Experimental Setup

Baselines. We compare AIDER with five representative model-free baselines that differ in how they handle value-estimation uncertainty. DSAC-T (Duan et al., 2025) and BRO (Nauman et al., 2024) mitigate estimation instability through distributional learning or network regularization, without explicitly estimating higher-order aleatoric value uncertainty. DoubleGum (Hui et al., 2023) constructs a higher-order uncertainty surrogate using a fixed coefficient, whereas AID (Yu et al., 2026) uses an adaptive coefficient for the same purpose. Detailed implementation and hyperparameter settings are provided in Appendix C.2

Benchmarks. Our control evaluation contains eight locomotion tasks: Ant-v3 and Humanoid-v3 from OpenAI Gym's MuJoCo environments (Brockman et al., 2016; Todorov et al., 2012), together with the stand, walk, and run tasks for both dog and humanoid in DMC (Tassa et al., 2018). Across these agents, successful control requires coordinated continuous actions to maintain posture and generate movement. Appendix C.1 specifies the task spaces and rewards.

5.2.2 Experimental Results

Table 1 shows that AIDER achieves the best final TAR on all reported tasks, while AID consistently ranks second, despite AIDER using fewer and more uniform hyperparameter settings. AIDER obtains the highest normalized average return on both Gym-MuJoCo (0.862) and DMC (0.904), improving over AID by **5.9%** and **1.0%**, respectively. The improvement is more pronounced on the challenging Humanoid-v3 and Humanoid-run tasks, where AIDER outperforms AID by **8.6%** and **5.8%**, respectively. The advantage stems from DEAR's ability to characterize the dependence between higher-order value uncertainty and the state-action pair, enabling finer-grained uncertainty modulation than AID's task-level heuristic parameterization. These results indicate that directly learning higher-order aleatoric value uncertainty improves the robustness of the AID framework,



(a) Ant-v3 (b) Humanoid-v3 (c) Humanoid-stand (d) Humanoid-walk

(e) Humanoid-run (f) Dog-stand (g) Dog-walk (h) Dog-run

Figure 3: **Training curves.** Each task panel summarizes three training seeds: the curve gives the mean return and the surrounding band shows its confidence interval. BRO collects one environment transition per step with an update-to-data (UTD) ratio of 10; the remaining methods collect 20 transitions per step and use a UTD ratio of 1.

Table 1: Total Average Return (TAR) of model-free algorithms on DMC Suite and Gym-MuJoCo tasks. Mean ± Std over 3 seeds $(12345, 22345, 32345)$. Gym-MuJoCo values are in **k** (10^3). Darker Blue = best, Lighter Blue = second-best. The average scores of individual benchmarks (†) are normalized by task-specific thresholds before averaging: Ant (10k) and Humanoid (15k) for Gym-MuJoCo; all DMC Suite tasks are normalized by 1k.

Task	**SAC**	**DSAC-T**	**BRO**	**DoubleGum**	**AID**	**AIDER (ours)**
Ant-v3 (k)	6.4 ±0.8	7.1 ±0.3	8.3 ±0.1	7.5 ±0.0	8.5 ±0.7	8.9 ±0.5
Humanoid-v3 (k)	9.4 ±0.6	10.8 ±0.2	11.0 ±0.3	10.8 ±0.9	11.6 ±0.2	12.6 ±0.3
AVG. Gym-MuJoCo†	0.635 ±0.065	0.715 ±0.023	0.778 ±0.050	0.735 ±0.042	0.814 ±0.066	0.862 ±0.046
Dog-stand	790.4 ±77.3	966.5 ±25.4	993.6 ±4.0	992.7 ±1.8	996.9 ±0.2	998.0 ±0.4
Dog-walk	869.4 ±67.2	889.1 ±81.3	964.2 ±1.7	965.4 ±7.9	984.8 ±2.0	985.2 ±2.5
Dog-run	372.4 ±108.0	558.0 ±124.9	687.4 ±15.9	699.6 ±37.1	868.2 ±6.0	881.9 ±5.6
Humanoid-stand	907.5 ±15.9	933.3 ±2.0	961.8 ±4.7	966.7 ±3.2	973.0 ±1.3	973.4 ±2.8
Humanoid-walk	881.5 ±2.5	920.1 ±0.7	940.3 ±6.7	951.1 ±5.7	954.9 ±2.9	958.6 ±2.6
Humanoid-run	394.9 ±46.9	473.0 ±49.0	472.4 ±66.3	523.6 ±42.1	595.0 ±16.3	629.7 ±23.0
AVG. DMC Suite†	0.703 ±0.237	0.790 ±0.207	0.837 ±0.191	0.850 ±0.178	0.895 ±0.141	0.904 ±0.128

particularly in high-dimensional control problems, while reducing the burden of manual hyperparameter tuning.

5.2.3 Ablation Study

We ablate the parameterization of aleatoric value uncertainty and the clipping coefficient β on Dog-run. Direct parameterization avoids coupling the estimated value uncertainty to fluctuations in the lower-order return variance, yielding a more stable uncertainty estimate. The clipping coefficient must balance two competing effects: an overly small β may truncate valid uncertainty estimates and induce excessive clipping, whereas an overly large β may fail to suppress anomalously high estimates. As shown in Table 2, direct parameterization with $\beta = 0.3$ achieves the highest TAR, supporting the configuration used by AIDER.

6 Related Work

This section reviews the literature from two synergistic perspectives: aleatoric uncertainty quantification and value overestimation suppression.

Table 2: Ablation of aleatoric value uncertainty parameterization and clipping on Dog-run.

Parameterization	Dog-run TAR					
	$\beta = 0.1$	$\beta = 0.2$	$\beta = 0.3$	$\beta = 0.4$	$\beta = 0.5$	$\beta = 1.0$
Evidence ratio	868.3 ± 7.6	–	862.8 ± 3.0	–	–	830.5 ± 21.9
Direct prediction	846.7 ± 12.8	866.7 ± 9.6	$\mathbf{881.9 \pm 5.6}$	865.9 ± 6.3	867.0 ± 3.9	865.7 ± 9.3

Aleatoric uncertainty quantification. Existing methods follow two primary paradigms: (1) *Distributional regression.* This paradigm models the full return distribution, including quantile-based approaches (Dabney et al., 2018; Kuznetsov et al., 2020) that approximate arbitrary distributions via discrete quantiles, and Gaussian-based approaches (Duan et al., 2025; Yu et al., 2026) that parameterize distributions via mean and variance for analytic efficiency. While these methods explicitly represent return uncertainty, they treat distribution parameters as point estimates, failing to quantify higher-order estimation uncertainty in learning. (2) *Evidential regression.* Evidential regression (Amini et al., 2020) parameterizes higher-order distributions in a single sampling-free forward pass. However, traditional frameworks rely on heuristic regularizing losses to penalize evidence, which often misrepresents true uncertainty in non-stationary prediction. Our DEAR operator addresses this by adopting a target network as an empirical anchor, replacing heuristic evidence-curbing with a more grounded, precise estimation of higher-order value uncertainty.

Value overestimation suppression. Approaches for mitigating value overestimation in off-policy RL fall into two paradigms: network regularization and objective regularization. (1) *Network regularization.* This paradigm stabilizes approximation via architectural constraints, such as LayerNorm (Ba et al., 2016) in BRO (Nauman et al., 2024), GroupNorm (Wu & He, 2018) in DoubleGum (Hui et al., 2023), and hyperspherical normalization in Simba (Lee et al., 2025). Despite stability benefits, these static structural priors impose constant regularization overhead that may restrict model expressivity in low-risk regions. (2) *Objective regularization.* This paradigm mitigates overestimation by augmenting the learning objective with uncertainty constraints or penalties. Implicit approaches leverage structural heuristics or distributional properties without direct penalty terms, such as TD3 (Fujimoto et al., 2018) and DSAC-T (Duan et al., 2025). Explicit approaches construct quantifiable penalties from specific uncertainty sources. While traditional methods target isolated sources (Li et al., 2023; Cetin & Celiktutan, 2023; Clements et al., 2019), recent frameworks like DoubleGum (Hui et al., 2023) and AID (Yu et al., 2026) incorporate both epistemic and aleatoric components. However, such frameworks rely on indirect estimation of higher-order aleatoric uncertainty.

7 Conclusion

In this work, we proposed Direct Evidential Regression (DEAR), a target-anchored framework for directly estimating higher-order uncertainty in parametric estimators. Building upon DEAR, we developed AIDER, which learns aleatoric value-estimation uncertainty and uses it with epistemic uncertainty for pessimistic target correction and optimistic exploration. Experiments on continuous-control benchmarks show that AIDER achieves state-of-the-art results. Future work will explore extending DEAR to on-policy reinforcement learning, as well as large language model applications.

Beyond reinforcement learning, DEAR can be applied to other parametric prediction problems that require direct modeling of higher-order uncertainty. This perspective may inspire new methods for calibrating, propagating, and exploiting higher-order uncertainty across a broader range of learning systems.

AI use statement

In this work, we used generative AI tools for language polishing and academic expression refinement of the manuscript to improve the readability and fluency of the text. We have not used generative AI tools for core research processes including research design, data collection, data analysis, result interpretation and original academic content creation, and the rest of the required disclosure tasks for generative AI usage are not applicable to this work. Additionally, we have not used generative AI

tools for any tasks with recommended disclosure other than the language polishing stated above. We have reviewed all AI-assisted work thoroughly. Specifically, all text segments revised by generative AI tools were manually checked and validated by the authors to ensure academic accuracy, logical coherence and full consistency with the research conclusions of this work. We take full responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This work develops and evaluates machine-learning methods using a synthetic regression task and publicly available simulated-control benchmarks from MuJoCo and the DeepMind Control Suite. It does not involve human participants, animal subjects, personal data, or sensitive attributes, and therefore does not raise concerns related to informed consent, privacy, or demographic bias in the reported experiments. All experiments are performed in simulation; no claims are made regarding safety in real-world or safety-critical deployments.

REPRODUCIBILITY STATEMENT

All results presented in this work are fully reproducible. The appendix provides complete derivation proofs, training pseudocode, benchmark details, network architectures, and full hyperparameter configurations. We include our full source code and runnable experiment scripts as supplementary material, and will open-source the entire codebase upon paper acceptance.

REFERENCES

Alexander Amini, Wilko Schwarting, Ava Soleimany, and Daniela Rus. Deep evidential regression. *Advances in neural information processing systems*, 33:14927–14937, 2020.

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.

Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. Openai gym. *arXiv preprint arXiv:1606.01540*, 2016.

Edoardo Cetin and Oya Celiktutan. Learning pessimism for reinforcement learning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 37, pp. 6971–6979, 2023.

Bertrand Charpentier, Daniel Zügner, and Stephan Günnemann. Posterior network: Uncertainty estimation without ood samples via density-based pseudo-counts. *Advances in neural information processing systems*, 33:1356–1367, 2020.

William R Clements, Bastien Van Delft, Benoît-Marie Robaglia, Reda Bahi Slaoui, and Sébastien Toth. Estimating risk and uncertainty in deep reinforcement learning. *arXiv preprint arXiv:1905.09638*, 2019.

Will Dabney, Georg Ostrovski, David Silver, and Rémi Munos. Implicit quantile networks for distributional reinforcement learning. In *International conference on machine learning*, pp. 1096–1105. PMLR, 2018.

Jingliang Duan, Yangang Ren, Fawang Zhang, Jie Li, Shengbo Eben Li, Yang Guan, and Keqiang Li. Encoding distributional soft actor-critic for autonomous driving in multi-lane scenarios [research frontier]. *IEEE Computational Intelligence Magazine*, 19(2):96–112, 2024.

Jingliang Duan, Wenxuan Wang, Liming Xiao, Jiaxin Gao, Shengbo Eben Li, Chang Liu, Ya-Qin Zhang, Bo Cheng, and Keqiang Li. Distributional soft actor-critic with three refinements. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2025.

Scott Fujimoto, Herke Hoof, and David Meger. Addressing function approximation error in actor-critic methods. In *International conference on machine learning*, pp. 1587–1596. PMLR, 2018.

Junyu Gao, Mengyuan Chen, Liangyu Xiang, and Changsheng Xu. A comprehensive survey on evidential deep learning and its applications. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2025.

Yang Guan, Shengbo Eben Li, Jingliang Duan, Jie Li, Yangang Ren, Qi Sun, and Bo Cheng. Direct and indirect reinforcement learning. *International Journal of Intelligent Systems*, 36(8):4439–4467, 2021.

Yang Guan, Liye Tang, Yao Lyu, Shengbo Eben Li, Yujie Yang, Kehua Sheng, Bo Zhang, Lei He, and Keqiang Li. Enhanced integrated decision and control for high-level automated vehicles and its experiment verification. *IEEE Transactions on Automation Science and Engineering*, 23: 11578–11595, 2026.

David Yu-Tung Hui, Aaron C Courville, and Pierre-Luc Bacon. Double gumbel q-learning. *Advances in neural information processing systems*, 36:2580–2616, 2023.

Eyke Hüllermeier and Willem Waegeman. Aleatoric and epistemic uncertainty in machine learning: An introduction to concepts and methods. *Machine learning*, 110(3):457–506, 2021.

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

Hojoon Lee, Youngdo Lee, Takuma Seno, Donghu Kim, Peter Stone, and Jaegul Choo. Hyperspherical normalization for scalable deep reinforcement learning. *arXiv preprint arXiv:2502.15280*, 2025.

Shengbo Eben Li. *Reinforcement learning for sequential decision and optimal control*. Springer, 2023.

Sicen Li, Qinyun Tang, Yiming Pang, Xinmeng Ma, and Gang Wang. Realistic actor-critic: A framework for balance between value overestimation and underestimation. *Frontiers in Neurorobotics*, 16:1081242, 2023.

Yao Lyu, Xiangteng Zhang, Shengbo Eben Li, Jingliang Duan, Letian Tao, Qing Xu, Lei He, and Keqiang Li. Conformal symplectic optimization for stable reinforcement learning. *IEEE Transactions on Neural Networks and Learning Systems*, 36(6):11049–11063, 2025.

Vincent Mai, Kaustubh Mani, and Liam Paull. Sample efficient deep reinforcement learning via uncertainty estimation. In *International conference on learning representations*, pp. 10283–10306, 2022.

Nis Meinert, Jakob Gawlikowski, and Alexander Lavin. The unreasonable effectiveness of deep evidential regression. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pp. 9134–9142, 2023.

Michal Nauman, Mateusz Ostaszewski, Krzysztof Jankowski, Piotr Miłoś, and Marek Cygan. Bigger, regularized, optimistic: scaling for compute and sample efficient continuous control. *Advances in neural information processing systems*, 37:113038–113071, 2024.

Zhixin Ren, Yao Lyu, Congrong Li, Liping Zhang, and Shengbo Eben Li. Momentum as residual-driven multiplier correction for deep learning optimization. *arXiv preprint arXiv:2608.12925*, 2026.

Nikita Rudin, David Hoeller, Philipp Reist, and Marco Hutter. Learning to walk in minutes using massively parallel deep reinforcement learning. In *Conference on robot learning*, pp. 91–100. PMLR, 2022.

Andrew M Saxe, James L McClelland, and Surya Ganguli. Exact solutions to the nonlinear dynamics of learning in deep linear neural networks. *arXiv preprint arXiv:1312.6120*, 2013.

Murat Sensoy, Lance Kaplan, and Melih Kandemir. Evidential deep learning to quantify classification uncertainty. *Advances in neural information processing systems*, 31, 2018.

David Silver, Julian Schrittwieser, Karen Simonyan, Ioannis Antonoglou, Aja Huang, Arthur Guez, Thomas Hubert, Lucas Baker, Matthew Lai, Adrian Bolton, et al. Mastering the game of go without human knowledge. *Nature*, 550(7676):354–359, 2017.

Yuval Tassa, Yotam Doron, Alistair Muldal, Tom Erez, Yazhe Li, Diego de Las Casas, David Budden, Abbas Abdolmaleki, Josh Merel, Andrew 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 1993 connectionist models summer school*, pp. 255–263. Psychology Press, 2014.

Emanuel Todorov, Tom Erez, and Yuval Tassa. Mujoco: A physics engine for model-based control. In *2012 IEEE/RSJ international conference on intelligent robots and systems*, pp. 5026–5033. IEEE, 2012.

Oriol Vinyals, Igor Babuschkin, Wojciech M Czarnecki, Michaël Mathieu, Andrew Dudzik, Junyoung Chung, David H Choi, Richard Powell, Timo Ewalds, Petko Georgiev, et al. Grandmaster level in starcraft ii using multi-agent reinforcement learning. *Nature*, 575(7782):350–354, 2019.

Wenxuan Wang, Yuhang Zhang, Jiaxin Gao, Yuxuan Jiang, Yujie Yang, Zhilong Zheng, Wenjun Zou, Jie Li, Congsheng Zhang, Wenhan Cao, et al. Gops: A general optimal control problem solver for autonomous driving and industrial control applications. *Communications in Transportation Research*, 3:100096, 2023.

Yuefei Wu, Bin Shi, Bo Dong, Qinghua Zheng, and Hua Wei. The evidence contraction issue in deep evidential regression: Discussion and solution. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 21726–21734, 2024.

Yuxin Wu and Kaiming He. Group normalization. In *Proceedings of the European conference on computer vision (ECCV)*, pp. 3–19, 2018.

Jingjing Xu, Xu Sun, Zhiyuan Zhang, Guangxiang Zhao, and Junyang Lin. Understanding and improving layer normalization. *Advances in neural information processing systems*, 32, 2019.

Zhouyang Yu, Guojian Zhan, Yang Guan, Jingliang Duan, Letian Tao, and Shengbo Eben Li. Taming aleatoric impulse in off-policy reinforcement learning. In *International conference on machine learning*, 2026.

A Proof of the Temporal Surrogate Bound

Let $z_{k,b} = (x_{k,b}, y_{k,b})$ be observation $b \in \{1, \dots, B\}$ in the mini-batch received at update $k \in \{1, \dots, t\}$. For a fixed query input x, denote the prediction obtained with infinitely many observations by $y^*(x)$. Under the standard local asymptotic linearity condition, the prediction learned from a finite dataset $\mathcal{D}_n$ satisfies

$$y_{\mathcal{D}_n}(x) - y^*(x) \approx \frac{1}{n} \sum_{z_j \in \mathcal{D}_n} \psi(z_j; x), \qquad \mathbb{E}_z[\psi(z; x)] = 0, \quad \mathrm{Var}_z[\psi(z; x)] = v(x). \tag{19}$$

Here, $\psi(z_j; x)$ is the first-order change in the prediction at the query input x induced by training observation z_j. Thus, $v(x)$ measures how strongly one randomly sampled observation can perturb that prediction.

Applying Eq. 19 to the Bt observations received in t mini-batches gives

$$\sigma^2_{\mathrm{high}}(x; Bt) = \mathrm{Var}\left[\frac{1}{Bt} \sum_{k=1}^{t} \sum_{b=1}^{B} \psi(z_{k,b}; x)\right] = \frac{v(x)}{Bt}. \tag{20}$$

For each update, define the average prediction perturbation of its mini-batch as

$$\bar{\psi}_k(x) := \frac{1}{B} \sum_{b=1}^{B} \psi(z_{k,b}; x), \qquad \mathrm{Var}[\bar{\psi}_k(x)] = \frac{v(x)}{B}. \tag{21}$$

To relate the temporal weights to standard optimization, consider the mini-batch SGD update

$$\theta_k = \theta_{k-1} - \eta \frac{1}{B} \sum_{b=1}^{B} \nabla_\theta \ell(\theta_{k-1}; z_{k,b}), \tag{22}$$

where η is the learning rate. Let $e_k(x) = y_{\theta_k}(x) - y^*(x)$ be the prediction error after update k. To define the curvature seen by this scalar prediction, write $L(\theta) = \mathbb{E}_z[\ell(\theta; z)]$ and $g(x) = \nabla_\theta y_{\theta^*}(x)$. Near the local solution θ^*, we use the one-dimensional projection

$$g(x)^\top \nabla_\theta L(\theta) \approx h(x) \left(y_\theta(x) - y^*(x)\right), \tag{23}$$

which defines $h(x)$ as the effective local curvature along the prediction direction. Linearizing Eq. 22 and applying this projection gives

$$e_k(x) \approx (1 - \eta h(x))\, e_{k-1}(x) + \eta h(x) \bar{\psi}_k(x), \tag{24}$$

Defining

$$\rho(x) := \eta h(x) \tag{25}$$

shows that a previous mini-batch contribution is multiplied by $1 - \rho(x)$ after each subsequent SGD update. Thus, the forgetting rate increases with both the learning rate and the local curvature. For a fixed x, we write $\rho(x)$ as ρ below.

The part of the prediction error attributable to the observed mini-batches is isolated and bias-corrected by

$$\tilde{e}_k(x) := \frac{e_k(x) - (1 - \rho)^k e_0(x)}{1 - (1 - \rho)^k}. \tag{26}$$

Substituting Eq. 24 yields the normalized update

$$\tilde{e}_k(x) = (1 - a_k)\tilde{e}_{k-1}(x) + a_k \bar{\psi}_k(x), \qquad a_k = \frac{\rho}{1 - (1 - \rho)^k}, \quad k \geq 2, \tag{27}$$

with $\tilde{e}_1(x) = \bar{\psi}_1(x)$. Thus, a_k is the fraction assigned to the current mini-batch at update k, while the weights of two consecutive historical mini-batches have ratio $1 - \rho$. Unrolling Eq. 27 gives

$$w_{k,t} = \frac{\rho(1 - \rho)^{t-k}}{1 - (1 - \rho)^t}, \qquad \sum_{k=1}^{t} w_{k,t} = 1. \tag{28}$$

Across random mini-batch sequences, the variance of the corresponding weighted estimate is therefore

$$\sigma^2_{\text{temp}}(x) = \frac{v(x)}{B}\sum_{k=1}^{t} w_{k,t}^2 \tag{29}$$

$$= \frac{v(x)}{B}\frac{\rho}{2-\rho}\frac{1+(1-\rho)^t}{1-(1-\rho)^t} = \frac{v(x)}{n_{\text{eff}}(t)}, \tag{30}$$

where

$$n_{\text{eff}}(t) = \frac{B}{\sum_{k=1}^{t} w_{k,t}^2} = B\frac{2-\rho}{\rho}\frac{1-(1-\rho)^t}{1+(1-\rho)^t}. \tag{31}$$

Under local stationarity and ergodicity, this variance across random mini-batch sequences equals the temporal variance in Eq. 8. Thus, predictions observed along one training trajectory recover the same moment without repeated training runs. Because the weights sum to one, the Cauchy–Schwarz inequality gives $\sum_k w_{k,t}^2 \geq 1/t$, or equivalently $n_{\text{eff}}(t) \leq Bt$. Combining this result with Eq. 20 proves the lower bound in Eq. 9.

For the upper bound, write $q = 1-\rho$. The exact discrepancy is

$$\Delta_t(x) := \sigma^2_{\text{temp}}(x) - \sigma^2_{\text{high}}(x; Bt) = \frac{v(x)}{B}\left[\frac{\rho}{2-\rho}\frac{1+q^t}{1-q^t} - \frac{1}{t}\right]. \tag{32}$$

Using $1-q^t = (1-q)\sum_{j=0}^{t-1} q^j$ and $(1+q)\sum_{j=0}^{t-1} q^j \geq 2tq^t$ yields

$$\frac{2\rho q^t}{(2-\rho)(1-q^t)} \leq \frac{1}{t}. \tag{33}$$

Substituting this inequality into Eq. 32 gives

$$\Delta_t(x) \leq \frac{\rho}{2-\rho}\frac{v(x)}{B}, \tag{34}$$

which completes the proof.

The two limiting regimes further explain the tightness of the surrogate. A Taylor expansion for $\rho t \ll 1$ gives

$$\Delta_t(x) = \frac{v(x)}{B}\left[\frac{t^2-1}{12t}\rho^2 + \mathcal{O}(\rho^3 t^2)\right], \tag{35}$$

whereas $q^t \to 0$ and $1/t \to 0$ imply

$$\lim_{t\to\infty} \Delta_t(x) = \frac{\rho}{2-\rho}\frac{v(x)}{B}. \tag{36}$$

Thus, early in training the effective evidence satisfies $n_{\text{eff}}(t) \approx Bt$; after the exponential memory saturates, the absolute variance error remains bounded and is approximately $\rho v(x)/(2B)$ for small ρ.

B Error Analysis of EMA-Anchored Temporal Variance

DEAR estimates temporal variance through the residual between a parametric estimator and its exponential-moving-average (EMA) anchor. This section first justifies analyzing the EMA in prediction space although it is implemented in parameter space. We then characterize the resulting variance error in two regimes: a locally stationary and temporally independent estimator, and the locally dynamic value estimator used by AIDER.

B.1 From Parameter-Space to Prediction-Space EMA

Let $f_x(\theta) := y_\theta(x)$ denote the prediction at a fixed input x and consider the parameter-space update

$$\bar{\theta}_t = \tau\theta_t + (1-\tau)\bar{\theta}_{t-1}, \qquad \tau \in (0,1). \tag{37}$$

For comparison, define the corresponding output-space EMA

$$m_t(x) = \tau f_x(\theta_t) + (1-\tau)m_{t-1}(x), \qquad m_0(x) = f_x(\bar{\theta}_0). \tag{38}$$

These two operations are identical for a predictor that is affine in its parameters, but not for a general neural network. Suppose that f_x is twice differentiable in the local region traversed by the two networks and

$$\left\|\nabla_\theta^2 f_x(\theta)\right\|_{\mathrm{op}} \le L_x. \tag{39}$$

The second-order interpolation remainder gives

$$r_t^{\mathrm{step}}(x) := f_x(\bar{\theta}_t) - \tau f_x(\theta_t) - (1-\tau) f_x(\bar{\theta}_{t-1}), \tag{40}$$

$$\left|r_t^{\mathrm{step}}(x)\right| \le \frac{L_x}{2}\tau(1-\tau)\left\|\theta_t - \bar{\theta}_{t-1}\right\|^2. \tag{41}$$

Let $r_t^{\mathrm{par}}(x) := f_x(\bar{\theta}_t) - m_t(x)$. Equations 37–38 imply

$$r_t^{\mathrm{par}}(x) = (1-\tau)r_{t-1}^{\mathrm{par}}(x) + r_t^{\mathrm{step}}(x). \tag{42}$$

Consequently, if $\|\theta_t - \bar{\theta}_{t-1}\| \le D_x$ locally and the two EMAs share the same initialization, then

$$|r_t^{\mathrm{par}}(x)| \le \frac{L_x}{2}(1-\tau)D_x^2. \tag{43}$$

Thus, output-space EMA analysis is a first-order description of the implemented parameter-space EMA, with an accumulated discrepancy that is second order in the local parameter distance. In the following two subsections, we adopt this output-space description and omit the bounded second-order remainder to focus on the variance error induced by temporal filtering and local drift.

B.2 Locally Stationary and Temporally Independent Case

Consider a locally stationary prediction sequence

$$Y_t(x) = \mu(x) + \varepsilon_t(x), \qquad \mathbb{E}[\varepsilon_t(x)] = 0, \qquad \mathrm{Var}[\varepsilon_t(x)] = V(x), \tag{44}$$

where $\{\varepsilon_t(x)\}$ are temporally independent. The true temporal variance is therefore $V(x)$. Let $M_t(x)$ be the output-space EMA of $Y_t(x)$ with coefficient τ. At stationarity,

$$\mathrm{Var}[M_t(x)] = \tau^2 V(x) + (1-\tau)^2\,\mathrm{Var}[M_{t-1}(x)], \tag{45}$$

and hence

$$\mathrm{Var}[M_t(x)] = \frac{\tau}{2-\tau}V(x). \tag{46}$$

Moreover,

$$Y_t(x) - M_t(x) = (1-\tau)\left(Y_t(x) - M_{t-1}(x)\right). \tag{47}$$

Temporal independence makes $Y_t(x)$ independent of $M_{t-1}(x)$. Therefore, the ideal EMA-anchored variance is

$$V_{\mathrm{EMA}}(x) := \mathbb{E}\left[(Y_t(x) - M_t(x))^2\right] \tag{48}$$

$$= (1-\tau)^2\left(V(x) + \frac{\tau}{2-\tau}V(x)\right) \tag{49}$$

$$= \kappa(\tau)V(x), \qquad \kappa(\tau) := \frac{2(1-\tau)^2}{2-\tau}. \tag{50}$$

It follows that

$$0 \le V(x) - V_{\mathrm{EMA}}(x) = \frac{\tau(3-2\tau)}{2-\tau}V(x). \tag{51}$$

For the coefficient used in AIDER, $\tau = 0.005$, $\kappa(\tau) \approx 0.9925$; hence, a single output-space EMA anchor underestimates the temporal variance of an independent stationary prediction sequence by approximately 0.75%. Equation 51 therefore gives the complete filtering error in the locally stationary and temporally independent case.

B.3 Locally Dynamic Case: AIDER

AIDER applies DEAR to a critic that already uses a Polyak target. For a fixed state–action pair $x = (s, a)$, define

$$q_t(x) = Q_{\theta_t}(x), \qquad \bar{q}_t(x) = Q_{\bar{\theta}_t}(x), \qquad \hat{q}_t(x) = Q_{\hat{\theta}_t}(x). \tag{52}$$

The corresponding parameter updates are

$$\bar{\theta}_t = \tau\theta_t + (1-\tau)\bar{\theta}_{t-1}, \qquad \hat{\theta}_t = \tau\bar{\theta}_t + (1-\tau)\hat{\theta}_{t-1}. \tag{53}$$

Following the output-space approximation established above, the AIDER residual is

$$d_t(x) := \bar{q}_t(x) - \hat{q}_t(x). \tag{54}$$

Because $\bar{q}_t$ is already a lower-order EMA of the online prediction, short-term variation is smoothed before comparison with the higher-order anchor $\hat{q}_t$. This filtering acts on the complete residual, regardless of whether the variation originates from finite-sample randomness, optimization, or Bellman-target evolution. We focus below on the remaining error caused by the locally dynamic value mean.

Let $\mu_t(x) := \mathbb{E}[q_t(x)]$, with $\bar{\mu}_t(x)$ and $\hat{\mu}_t(x)$ denoting the corresponding means of the two target predictions. Under the output-space description, they satisfy $\bar{\mu}_t = \tau\mu_t + (1-\tau)\bar{\mu}_{t-1}$ and $\hat{\mu}_t = \tau\bar{\mu}_t + (1-\tau)\hat{\mu}_{t-1}$. Suppose that $Q_\theta(x)$ is locally $G_Q(x)$-Lipschitz in its parameters and the critic gradient is bounded by G_θ. An SGD update with learning rate η then gives

$$\begin{aligned} |\mu_t(x) - \mu_{t-1}(x)| &\le G_Q(x)\|\theta_t - \theta_{t-1}\| \\ &\le \eta G_Q(x) G_\theta =: \delta_Q(x). \end{aligned} \tag{55}$$

The first EMA preserves this increment bound:

$$|\bar{\mu}_t(x) - \bar{\mu}_{t-1}(x)| \le \delta_Q(x). \tag{56}$$

This follows because $\bar{\mu}_t - \bar{\mu}_{t-1}$ is the same convex EMA of the increments $\mu_{t-j} - \mu_{t-j-1}$. Writing $m_t^d(x) := \mathbb{E}[d_t(x)] = \bar{\mu}_t(x) - \hat{\mu}_t(x)$ gives

$$m_t^d(x) = (1-\tau)m_{t-1}^d(x) + (1-\tau)\left(\bar{\mu}_t(x) - \bar{\mu}_{t-1}(x)\right). \tag{57}$$

Consequently,

$$|m_t^d(x)| \le (1-\tau)^t |m_0^d(x)| + \frac{1-\tau}{\tau}\delta_Q(x). \tag{58}$$

After the initialization transient, the tracking bias therefore satisfies

$$|m_t^d(x)| \le \frac{1-\tau}{\tau}\eta G_Q(x) G_\theta. \tag{59}$$

Hence, although a non-stationary Bellman target introduces tracking lag, its contribution is controlled by the critic learning rate and the EMA time scale.

Finally, the bounded tracking signal integrates naturally with AIDER's uncertainty-utilization mechanism. A larger residual highlights state–action regions whose value estimates are evolving rapidly. AIDER responds with a stronger pessimistic correction in the Bellman target to stabilize value propagation, while its optimistic behavior policy directs further visitation and evidence acquisition toward the same regions. Together, these mechanisms form a feedback loop that substantially reduces the impact of uncertainty estimation errors.

C Experimental Details

C.1 Benchmark Description

DeepMind Control Suite. The DMC evaluation uses stand, walk, and run for each of two morphologies, dog and humanoid (Tassa et al., 2018). Dog observations have 223 dimensions and its actions have 38; humanoid uses 67 observation dimensions and 24 action dimensions. Standing evaluates balance, whereas walking and running additionally require tracking a prescribed forward speed. In the moving tasks, posture and speed contributions are combined multiplicatively: ***Reward*** = (*Standing reward*) × (*Forward velocity reward*).



Figure 4: Dog

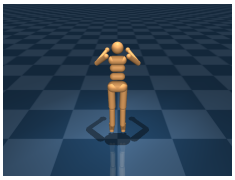


Figure 5: Humanoid

***Standing reward*:** Measures whether the agent keeps its body upright.

***Forward velocity reward*:** Evaluates speed tracking. The walking target is 1 m/s for both agents; the running targets are 9 m/s for dog and 10 m/s for humanoid.

MuJoCo Continuous Control. The Gym-MuJoCo portion complements DMC with Ant-v3 and Humanoid-v3 (Brockman et al., 2016; Todorov et al., 2012), testing quadrupedal and bipedal locomotion under their respective reward definitions.

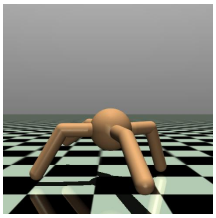


Figure 6: Ant

Ant-v3: An 8-dimensional action controls the quadruped using a 111-dimensional state representation. Forward progress requires a coordinated gait that also preserves balance.

***Reward*:** $r = v + r_{\text{healthy}} - C_{\text{ctrl}} - C_{\text{contact}}$.
Positive reward terms account for forward speed (v) and healthy posture (r_{healthy}). The deductions C_{ctrl} and C_{contact} discourage excessive actuation and contact forces, respectively.



Figure 7: Humanoid

Humanoid-v3: This biped receives a 376-dimensional state and issues a 17-dimensional action. It must sustain an upright gait while advancing.
Reward: $r = 1.25v + r_{\text{healthy}} - C_{\text{ctrl}} - C_{\text{contact}}$.

The coefficient on v emphasizes forward motion. The healthy-posture term r_{healthy} rewards balance, while C_{ctrl} and C_{contact} subtract actuation and external-contact costs.

C.2 Implementation Details

Implementation framework. Our implementation extends the released AID codebase (Yu et al., 2026), which uses the PyTorch-based GOPS solver (Wang et al., 2023). GOPS also provides implementations of SAC, PPO, and DSAC-T. For DoubleGum and BRO, we start from the authors' public repositories and apply the configurations described below. Table 3 lists the repository locations.

Table 3: Summary of Implementation Frameworks and Code Repositories.

Algorithm / Framework	Source Repository
GOPS (Wang et al., 2023)	`https://github.com/Intelligent-Driving-Laboratory/GOPS`
AID (Yu et al., 2026)	`https://github.com/yzy-yuzhouyang/DSAC-AID`
DoubleGum (Hui et al., 2023)	`https://github.com/dyth/doublegum`
BRO (Nauman et al., 2024)	`https://github.com/naumix/BiggerRegularizedOptimistic`

Network architecture. The SAC and DSAC-T baselines use MLPs. Following the AID implementation (Yu et al., 2026), AIDER and AID use the DoubleGum-style MLP design, with orthogonal weight initialization (Saxe et al., 2013) and GroupNorm (Wu & He, 2018); the normalization omits learnable shift and scale (Xu et al., 2019). Humanoid-v3 is the exception, for which normalization is disabled. BRO (Nauman et al., 2024) keeps the larger architecture specified by its original implementation. Table 4 reports network sizes and other architectural settings.

Hyperparameters. Table 4 collects the training settings for every method. We share configurations where appropriate and use RAD for all parameter updates (Lyu et al., 2025; Ren et al., 2026). For DoubleGum, network size and the number of collected samples per step are matched to the compact SAC and DSAC-T baselines to control for these configuration differences. Its pessimism

Table 4: General hyperparameters.

Hyperparameter	AIDER (ours)	AID	DSAC-T	SAC	DoubleGum	BRO
Common						
Discount Factor (γ)	0.99	0.99	0.99	0.99	0.99	0.99
Reward Scale (MuJoCo / DMC)	0.1 / 50	0.1 / 50	0.1 / 50	1 / 1	1 / 1	1 / 1
Replay Buffer & Batch Settings						
Replay Batch Size	256	256	256	256	256	256
Sample Batch Size	20	20	20	20	20	1
Replay Buffer Size	10^6	10^6	10^6	10^6	10^6	10^6
Warmup Steps	10,000	10,000	10,000	10,000	10,000	10,000
Replay Ratio	1	1	1	1	1	10
Polyak (τ)	0.005	0.005	0.005	0.005	0.005	0.005
Neural Network						
Critic Ensemble (Online/Target)	2 / 4	2 / 2	2 / 2	2 / 2	2 / 2	2 / 2
Critic Hidden (Depth $\times$ Dim)	3×256	3×256	3×256	3×256	3×256	5×512
Actor Ensemble (Online/Target)	2 / 1	2 / 1	1 / 1	1 / -	1 / 1	2 / -
Actor Hidden (Depth $\times$ Dim)	3×256	3×256	3×256	3×256	3×256	1×256
Activation (Humanoid-v3 / Others)	GeLU / ReLU	GeLU / ReLU	GeLU	GeLU	ReLU	ReLU
Normalization (Humanoid-v3 / Others)	- / GroupNorm	- / GroupNorm	-	-	GroupNorm	LayerNorm
Learning Rate (Humanoid-v3 / Others)	1×10^{-4} / 3×10^{-4}	1×10^{-4} / 3×10^{-4}	1×10^{-4}	1×10^{-4}	3×10^{-4}	3×10^{-4}
α Learning Rate	3×10^{-4}	3×10^{-4}	3×10^{-4}	3×10^{-4}	-	3×10^{-4}
Total Parameters	1.42M	1.08M	0.93M	0.78M	0.93M	4.93M

Table 5: Task-specific hyperparameters for AID.

Benchmark	Task	Confidence Bound Coefficients		Aleatoric Pessimism Coefficient (β)			
		λ_{LB} (Lower)	λ_{UB} (Upper)	Initial	Annealing Rate	Update Interval	K
MuJoCo	Ant-v3	$1/\sqrt{2}$	$1/\sqrt{2}$	0.03	1.5×10^{-4}	2500	200
	Humanoid-v3	$1/\sqrt{2}$	$1/\sqrt{2}$	0.20	5.0×10^{-4}	2500	200
DMC	Dog tasks	0.5	0.5	0.10	5.0×10^{-3}	10000	200
	Humanoid-stand	0.02	0.02	0.10	5.0×10^{-3}	10000	200
	Humanoid-walk	0.02	0.02	0.10	2.0×10^{-3}	10000	200
	Humanoid-run	0.02	0.02	0.10	2.0×10^{-3}	10000	200

setting is $c = -0.1$, following the recommended configuration. BRO is trained for the iteration budget used in this evaluation, but keeps its native sampling setup. We retain that setup because preliminary runs with the common sampling batch size performed substantially worse. The AID baseline retains the initialization and update settings for its aleatoric coefficient from AID (Yu et al., 2026): the initial β, annealing rate, update interval, and bias-estimation sample count K are listed in Table 5. AIDER directly learns aleatoric value uncertainty and therefore does not require these dynamic calibration parameters. It uses the same λ_{LB} and λ_{UB} settings as AID, while its clipping coefficient is fixed at $\beta = 0.3$ for all tasks. The higher-order EMA target $\hat{\theta}_i$ uses the same Polyak coefficient $\tau = 0.005$ as the lower-order target critic $\bar{\theta}_i$.

D Comparison of Higher-Order Uncertainty Estimation

Figure 8 compares the effects of different higher-order uncertainty estimation schemes during Humanoid-v3 training. In the top panel, the ratio between higher-order and lower-order aleatoric uncertainty serves as a proxy for effective evidence, i.e., the number of effective state–action visitations. AID captures only the global trend of this quantity, whereas AIDER preserves its state–action dependence, as reflected by the shaded variation. The middle panel shows that AIDER suppresses the aleatoric impulse while avoiding excessive conservatism and over-suppression. The bottom panel shows the resulting improvement in training performance.

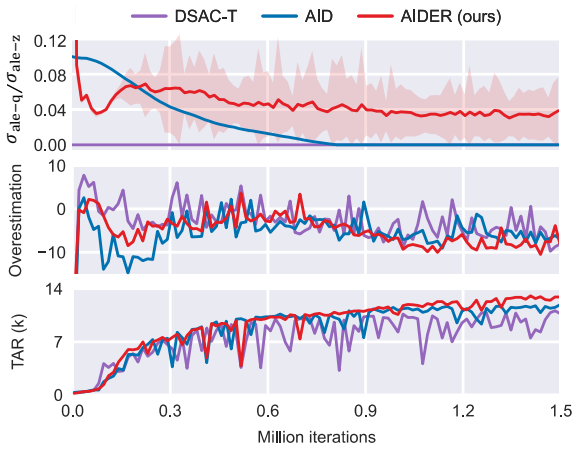


Figure 8: Comparison of higher-order uncertainty estimation schemes on Humanoid-v3.

E Computational Overhead

To assess the practical computational overhead of AIDER, Table 6 reports the wall-clock training time on the Humanoid-run task. All algorithms are trained for 2×10^6 environment steps on a single NVIDIA GeForce RTX 4090 GPU. As shown, AIDER requires 29.52 h, i.e., $1.36\times$ DSAC-T and only $1.15\times$ AID. This modest overhead stems from the higher-order EMA anchor. This extra anchor only incurs a lightweight EMA weight update and a small number of additional forward passes, so its computational footprint is negligible.

Table 6: Comparison of wall-clock training time on Humanoid-run.

	SAC	**DSAC-T**	**BRO**	**DoubleGum**	**AID**	**AIDER (ours)**
Time (Hours)	15.55	21.70	24.13	47.78	25.72	29.52
Relative to DSAC-T	0.72	1.00	1.11	2.20	1.19	1.36

F Complete Pseudocode

We present the complete training procedure for AIDER in Algorithm 1. Changes with respect to standard SAC that are inherited from AID are shown in blue, whereas the additional components introduced by AIDER are shown in red.

Algorithm 1 AIDER training procedure, adapted from AID (Yu et al., 2026). Inherited AID components are blue; AIDER additions are red.

Require: $\{\theta_i\}_{i=1}^m$ - online critic parameters, $\{\bar{\theta}_i\}_{i=1}^m$ - target critic parameters, $\{\hat{\theta}_i\}_{i=1}^m$ - higher-order target parameters, ϕ_T - target policy parameters, ϕ_B - behavior policy parameters, λ_{LB} - lower confidence bound coefficient, λ_{UB} - upper confidence bound coefficient, α_{init} - temperature, β - clipping coefficient, l_z - critic learning rate, l_π - actor learning rate, l_α - temperature learning rate, τ - Polyak coefficient

Initialize target networks: $\bar{\theta}_i \leftarrow \theta_i$, $\hat{\theta}_i \leftarrow \theta_i$, $\bar{\phi}_T \leftarrow \phi_T$
for each iteration **do**
  for each sampling step **do**
    Calculate action using the optimistic behavior policy
      $a \sim \pi_B(a|s)$
    Get reward r and new state s'
    Store samples (s, a, r, s') in replay buffer $\mathcal{B}$
  end for
  for each update step **do**
    Sample a minibatch $\mathcal{D}$ from $\mathcal{B}$
    Calculate next action using the pessimistic target policy
      $a' \sim \pi_T(a'|s')$
    Evaluate each evidential target critic at (s', a') and compute
      $\sigma^2_{\text{ale}-\text{q}}(s', a') = \frac{1}{m}\sum_{i=1}^m \sigma^2_{\text{ale}-\text{q},\theta_i}(s', a'), \quad \sigma^2_{\text{ale}-\text{z}}(s', a') = \frac{1}{m}\sum_{i=1}^m \sigma^2_{\text{ale}-\text{z},\theta_i}(s', a')$
    Clip the aleatoric value uncertainty before combining it with epistemic uncertainty
      $\tilde{\sigma}_{\text{ale}-\text{q}}(s', a') = \min\left(\sqrt{\sigma^2_{\text{ale}-\text{q}}(s', a')},\ \beta\sqrt{\sigma^2_{\text{ale}-\text{z}}(s', a')}\right)$
    Calculate the epistemic uncertainty of the ensemble value estimate
      $\sigma^2_{\text{epi}}(s', a') = \frac{1}{m-1}\sum_{i=1}^m \left[Q_{\bar{\theta}_i}(s', a') - Q(s', a')\right]^2, \quad Q = \frac{1}{m}\sum_{i=1}^m Q_{\bar{\theta}_i}$
    Calculate the total value uncertainty using the learned aleatoric component
      $\sigma_Q(s', a') = \sqrt{\sigma^2_{\text{epi}}(s', a') + \tilde{\sigma}^2_{\text{ale}-\text{q}}(s', a')}$
    Calculate the pessimistic Bellman target
      $y_q = r + \gamma\left[Q(s', a') - \lambda_{\text{LB}}\sigma_Q(s', a') - \alpha \log \pi_{\bar{\phi}_T}(a'|s')\right]$
    Calculate sampled return targets
      $y_{z_i} = r + \gamma\left[z'_{\bar{\theta}_i} - \alpha \log \pi_{\bar{\phi}_T}(a'|s')\right],\ z'_{\bar{\theta}_i} \sim \mathcal{Z}_{\bar{\theta}_i}(\cdot|s', a')$
    Compute the DEAR loss for each critic using its selected parameterization:
      $\mathcal{J}_Z^{\text{DEAR}}(\theta_i) = \text{sg}\left[\mathbb{E}_{\mathcal{D}}\sigma^2_{\text{ale}-\text{q},\theta_i}\right]\mathbb{E}_{\mathcal{D}}\left[\frac{(Q_{\bar{\theta}_i} - Q_{\hat{\theta}_i})^2}{2\sigma^2_{\text{ale}-\text{q},\theta_i}} + \frac{1}{2}\log\sigma^2_{\text{ale}-\text{q},\theta_i}\right]$
    Define the original AID distributional critic loss
      $\mathcal{J}_Z^{\text{AID}}(\theta_i) = \mathbb{E}_{\mathcal{D}}\left[-\frac{\mathbb{E}_{\mathcal{D}}[\sigma^2_{\theta_i} + \sigma^2_{\text{epi}}]}{\sigma^2_{\theta_i} + \sigma^2_{\text{epi}}}(y_q - Q_{\theta_i})\nabla Q_{\theta_i} - \frac{\mathbb{E}_{\mathcal{D}}[\sigma^2_{\theta_i}]}{\sigma^2_{\theta_i}}\frac{(y_{z_i} - Q_{\theta_i})^2 - \sigma^2_{\theta_i}}{\sigma_{\theta_i}}\nabla\sigma_{\theta_i}\right]$
    Update evidential critics using the original AID loss plus DEAR:
      $\theta_i \leftarrow \theta_i - l_z\nabla_{\theta_i}\left(\mathcal{J}_Z^{\text{AID}}(\theta_i) + \mathcal{J}_Z^{\text{DEAR}}(\theta_i)\right)$
    Update the pessimistic target policy
      $\phi_T \leftarrow \phi_T + l_\pi\mathbb{E}_{s\sim\mathcal{D}, a\sim\pi_T}\left[Q(s, a) - \lambda_{\text{LB}}\sigma_{\text{epi}}(s, a) - \alpha \log \pi_{\phi_T}(a|s)\right]$
    Update the optimistic behavior policy
      $\phi_B \leftarrow \phi_B + l_\pi\mathbb{E}_{s\sim\mathcal{D}, a\sim\pi_B}\left[Q(s, a) + \lambda_{\text{UB}}\sigma_Q(s, a) - \alpha \log \pi_{\phi_B}(a|s)\right]$
    Update entropy temperature
      $\alpha \leftarrow \alpha - l_\alpha\mathbb{E}_{a\sim\pi_{\phi_T}}[-\log \pi_{\phi_T}(a|s) - \bar{\mathcal{H}}]$
    Update target networks
      $\bar{\theta}_i \leftarrow \tau\theta_i + (1-\tau)\bar{\theta}_i,\ \bar{\phi}_T \leftarrow \tau\phi_T + (1-\tau)\bar{\phi}_T$
    Update higher-order target networks with EMA:
      $\hat{\theta}_i \leftarrow \tau\bar{\theta}_i + (1-\tau)\hat{\theta}_i$
  end for
  if evaluation phase **then**
    Collect evaluation trajectories $\{(s_t, a_t, r_t)\}_{t=1}^T$ using the pessimistic target policy
    Calculate total average return (TAR)
  end if
end for