AUT Journal of Modeling and Simulation

AUT J. Model. Simul., 57(2) (2025) 215-228
DOI: 10.22060/miscj.2026.23574.5387

Distributional Soft Actor-Critic with Adaptive Entropy Regularization

Meysam Fozi, Mohammad Mehdi Ebadzadeh *

Department of Computer Engineering, Amirkabir University of Technology, Tehran, Iran

ABSTRACT: Soft Actor-Critic (SAC) and its distributional extensions have demonstrated strong performance by combining entropy regularization with off-policy learning. However, existing automatic entropy tuning mechanisms primarily rely on fixed target entropy formulations and do not explicitly exploit the uncertainty information available in distributional critics. In this paper, we propose a variance-adaptive entropy regularization method for Distributional Soft Actor-Critic (DSAC), in which the entropy temperature is dynamically adjusted as a function of the predicted return distribution variance. Linear and exponential adaptation schemes are introduced to couple exploration strength with epistemic uncertainty estimated by the distributional critic. Unlike prior entropy scheduling or automatic temperature tuning methods, the proposed approach integrates uncertainty-aware adaptation directly into the distributional reinforcement learning framework. Experimental results on OpenAI's MuJoCo continuous-control tasks, demonstrate that the proposed method improves stability and generalization compared to DSAC-T and standard SAC with automatic entropy adjustment. This performance improvement highlights the importance of adaptive entropy regularization strategies in reinforcement learning, particularly for tasks requiring fine-tuned control in continuous environments. The findings suggest that the proposed adaptive DSAC algorithm not only enhances learning stability by reducing overestimation but also offers a more efficient solution to the exploration-exploitation dilemma, providing a promising direction for future research in reinforcement learning for continuous control settings.

Review History:

Received: Oct. 06, 2024
Revised: Jan. 04, 2025
Accepted: Dec. 27, 2025
Available Online: Dec. 31, 2025

Keywords:

Reinforcement Learning
Continuous Control
Distributional Soft Actor-Critic (DSAC)
Entropy Regularization

1- Introduction

Reinforcement Learning (RL) has shown remarkable success across various domains, from game playing [1,2] to robotics [3,4] and control systems [5]. Among the numerous RL algorithms, actor-critic methods have gained significant importance due to their ability to handle continuous action spaces and achieve stable convergence properties [6,7]. Deep Deterministic Policy Gradient (DDPG) [8], Twin Delayed DDPG (TD3) [9], and Soft Actor-Critic (SAC) [10] represent pivotal advancements in this family, incorporating deterministic and stochastic policy updates to address sample efficiency and stability challenges.

However, traditional SAC suffers from a limitation in its representation of value functions, often assuming a single expected value for each state-action pair. This assumption fails to capture the inherent uncertainty and variability associated with estimating value functions accurately. To address this limitation, a recent advancement in RL research introduced the concept of distributional RL [11,12] which aims to model the entire distribution of returns for each state-action pair.

In this paper, we delve into the realm of Distributional Soft Actor-Critic (DSAC) [13] an extension of SAC that incorporates the distributional perspective into its learning framework. DSAC leverages the power of distributional RL to estimate value distributions, enabling agents to better understand the uncertainty and variability of future rewards.

The core idea behind DSAC lies in learning a distributional value function that estimates the probability distribution over possible returns for each state-action pair. By doing so, DSAC not only captures the expected value but also provides valuable insights into the full range of possible outcomes, including their probabilities. This distributional perspective allows agents to make more informed decisions and adapt their policies accordingly, leading to enhanced performance and robustness.

A downside of all Q-learning algorithms is that they suffer overestimation. In a recent extension of DSAC [14], a distributional framework called DSPI uses a distributional soft policy iteration algorithm to train the corresponding deep neural networks in the actor-critic setting, such to address overestimation.

Throughout this paper, we will explore the theoretical foundations of DSAC, highlighting its key components and algorithmic details. We will discuss how DSAC leverages distributional RL techniques, such as Categorical

*Corresponding author's email: ebadzadeh@aut.ac.ir

Distributional RL [15] or Quantile Regression DQN [16] to estimate value distributions accurately. Also, we take into account adaptive optimization methods incorporated in RL problems from different perspectives such as in [17], as well as its critic-aware refined version [18]. Furthermore, we will examine the empirical results and performance improvements achieved by DSAC in various benchmark environments, including OpenAI MuJoCo Humanoid[1], showcasing its potential as a state-of-the-art RL algorithm.

In summary, the contributions of this paper are as follows: 1) We introduce an adaptive entropy regulation scheme grounded in policy uncertainty metrics. 2) We decouple actor and critic updates to improve convergence and learning robustness. 3) We provide extensive evaluations on standard control benchmarks to demonstrate the superiority of Adaptive DSAC over SAC and TD3, particularly in terms of sample efficiency and final performance.

The main contribution of this work is not a generic entropy scheduling strategy, but a principled integration of uncertainty-aware entropy adaptation into the Distributional Soft Actor-Critic framework. By leveraging the variance of the learned return distribution, the proposed method provides a feedback-driven exploration mechanism that is fundamentally different from fixed-target or heuristic entropy tuning approaches.

2- Related Work

2- 1- Standard RL

A rigorous mathematical foundation for RL is a Markov Decision Process (MDP), defined by the tuple $(\mathcal{S},\mathcal{A},\mathcal{R},p)$. Generally, the state space $\mathcal{S}$ and action space $\mathcal{A}$ are assumed to be continuous, and $R(r_t|s_t,a_t):\mathcal{S}\times\mathcal{A}\to\mathcal{P}(r_t)$ is a stochastic reward function mapping a state-action pair (s_t,a_t) to a distribution over a set of bounded rewards. The unknown state transition probability $p(s_{t+1}|s_t,a_t):\mathcal{S}\times\mathcal{A}\to\mathcal{P}(s_{t+1})$ maps a given (s_t,a_t) to the probability distribution over s_{t+1}, and the agent's behavior is defined by a stochastic policy $\pi(a_t|s_t):\mathcal{S}\to\mathcal{P}(a_t)$. The state distribution induced by policy is denoted by $\rho_\pi(s)$. In standard RL, the goal is to learn a policy which maximizes the expected future accumulated return:

$$\max_\pi J(\pi)=\mathbb{E}_{\substack{(s_{i\geq t},a_{i\geq t})\sim\rho_\pi\\ r_{i\geq t}\sim R(\cdot|s_i,a_i)}}\left[\sum_{i=t}^{\infty}\gamma^{i-t}r_i\right], \tag{1}$$

where $\gamma\in(0,1)$ is the discount factor. This classical formulation is risk-neutral and focuses on maximizing reward expectation only. However, it often suffers from premature convergence due to insufficient exploration.

2- 2- Entropy Regularization in RL: Historical Origins

Entropy regularization was first introduced in policy-gradient RL to mitigate the issue of premature convergence by encouraging stochasticity in policies. The key idea is that policies should not collapse to deterministic choices too early during learning; instead, a positive entropy term encourages exploration and prevents overfitting to suboptimal modes.

The policy-gradient with entropy bonus [19] formulation augments the reward with a fixed-weight entropy term:

$$J(\pi)=\mathbb{E}\left[\sum_{i=t}^{\infty}\gamma^{i-t}(r_i+\beta\mathcal{H}(\pi(\cdot|s_i)))\right], \tag{2}$$

where $\mathcal{H}(\pi(\cdot|s))=-\int\pi(a|s)\log\pi(a|s)da$ is the Shannon entropy of the policy at state s, and $\beta>0$ is a fixed coefficient. The entropy encourages broad action distributions, particularly useful in high-dimensional or multimodal tasks. The improvement over standard RL is better exploration and avoidance of poor local optima. However, the fixed coefficient β must be tuned manually and is highly problem-dependent.

A major theoretical leap was the maximum entropy RL (MaxEnt RL) framework [20], which formalized entropy regularization as part of the optimal control objective [21]. The MaxEnt formulation modifies the expected return to:

$$J_\pi=\mathbb{E}\left[\sum_{i=t}^{\infty}\gamma^{i-t}(r_i+\alpha\mathcal{H}(\pi(\cdot|s_i)))\right], \tag{3}$$

where $\alpha>0$ is a temperature parameter controlling the trade-off between reward maximization and entropy maximization. Compared to the early entropy-bonus heuristic, MaxEnt RL frames entropy as an integral part of the return. This results in policies that prefer both high return and high randomness, yielding robustness to noise and improved exploration.

The soft Bellman operator for MaxEnt RL is defined as:

$$\mathcal{T}^\pi Q(s,a)=\mathbb{E}[r]+\gamma\mathbb{E}_{\substack{s'\sim p\\ a'\sim\pi}}[Q(s',a')-\alpha\log\pi(a'|s')]. \tag{4}$$

Soft policy iteration alternates between evaluating Q^π under this operator and improving the policy via:

$$\pi_{new}=\arg\max_\pi\mathbb{E}_{\substack{s\sim\rho_\pi\\ a\sim\pi}}[Q^{\pi_{old}}(s,a)-\alpha\log\pi(a|s)]. \tag{5}$$

This framework inspired modern algorithms such as Soft Actor-Critic (SAC) [10], which introduced automatic temperature tuning based on a fixed target entropy. While effective, this mechanism does not adapt entropy in response to learning uncertainty.

While MaxEnt RL considers entropy of entire trajectories, causal entropy regularization [22] emphasizes entropy of actions conditioned on past states:

1 https://gymnasium.farama.org/environments/mujoco/humanoid/

$$\mathcal{H}_{causal}(\pi) = \mathbb{E}_{\pi}\left[-\sum_{t} \log \pi(a_t|s_t)\right]. \quad (6)$$

The causal entropy ensures that exploration decisions are conditioned only on available information, consistent with causality. This concept underpins Maximum Causal Entropy Inverse RL (MaxEnt IRL) and has been influential in imitation learning.

The Soft Actor-Critic (SAC) algorithm [10] operationalized MaxEnt RL into a practical off-policy actor-critic method. SAC introduced:

- A stochastic actor $\pi_\theta(a|s)$ trained to maximize a soft policy objective.
- A critic $Q_\phi(s,a)$ estimating soft Q-values under entropy-augmented returns.
- An automatic entropy-temperature tuning mechanism: the temperature α is optimized online via a dual objective enforcing a target entropy $\bar{\mathcal{H}}$:

$$L(\alpha) = \mathbb{E}_{a_t\sim\pi_\theta}[-\alpha \log \pi_\theta(a_t|s_t) - \alpha\bar{\mathcal{H}}]. \quad (7)$$

This automatic adjustment eliminated manual tuning of α, a key practical improvement. The adaptive temperature balances exploitation and exploration dynamically.

Some works proposed scheduled entropy regularization, where α decays according to an annealing schedule, e.g., exponential decay:

$$\alpha_t = \alpha_0 \cdot \exp(-kt), \quad (8)$$

with hyper parameters α_0 (initial value) and k (decay rate). This method encourages broad exploration initially and more deterministic behavior as learning progresses. Though heuristic, it reflects an intuitive curriculum: explore widely early, exploit later. These methods, however, are typically heuristic and independent of the critic's uncertainty estimates.

In contrast, fixed entropy regularization methods set α to a constant chosen manually. This corresponds to the original entropy-bonus approach but lacks adaptability. Fixed entropy can perform well in stable domains but struggles in environments with shifting exploration requirements.

The earliest entropy-regularized policy gradient algorithms [23] had the following objective:

$$J(\theta) = \mathbb{E}_{\pi_\theta}\left[\sum_{t} \log \pi_\theta(a_t|s_t)A^{\pi}(s_t,a_t) + \beta\mathcal{H}(\pi_\theta(\cdot|s_t))\right], \quad (9)$$

where A^{π} is the advantage function. The entropy bonus coefficient β is manually tuned. These methods demonstrated the feasibility of stabilizing policy learning with entropy.

Alternative entropy measures such as Tsallis entropy [24] have been proposed:

$$\mathcal{H}_q(\pi(\cdot|s)) = \frac{1}{q-1}\left(1 - \int \pi(a|s)^q da\right), \quad (10)$$

where $q > 0$ is the entropic index. For $q \to 1$, Tsallis entropy reduces to Shannon entropy. The Tsallis family allows control over exploration sparsity: larger q induces sparser, more deterministic distributions. Algorithms such as Tsallis Actor-Critic (TAC) use this to balance exploration and exploitation in a tunable manner.

Another branch of entropy regularization enforces conservative updates via KL-divergence penalties. Trust Region Policy Optimization (TRPO) [25] solves:

$$\max_{\theta} \mathbb{E}_{\pi_\theta}[A^{\pi}(s,a)] \quad (11)$$

$$s.t. \quad \mathbb{E}_s\left[D_{KL}\left(\pi_\theta(\cdot|s) \,\|\, \pi_{\theta_{old}}(\cdot|s)\right)\right] \le \delta, \quad (12)$$

where δ is a trust-region bound. Proximal Policy Optimization (PPO) [26] approximates this constraint with a clipped surrogate objective. KL-based regularization improves stability and sample efficiency, though it regulates divergence rather than entropy directly.

2- 3- Distributional Soft Actor-Critic

The distributional perspective in RL extends entropy-augmented RL into the space of return distributions. Distributional Soft Policy Iteration (DSPI) [14] defines the random return distribution:

$$Z^{\pi}(s_t,a_t) = r_t + \gamma G_{t+1}, \quad (13)$$

$$G_{t+1} = \sum_{i=t+1}^{\infty} \gamma^{i-t-1}(r_i - \alpha \log \pi(a_i|s_i)). \quad (14)$$

The distributional Bellman operator with entropy has a contraction property, ensuring convergence. This distributional framework underlies Distributional Soft Actor-Critic (DSAC).

Classical Q-learning suffers from overestimation due to the max operator. DSAC integrates entropy regularization with distributional critics to mitigate this bias. DSAC introduces twin critics, variance regularization, and distributional targets to produce more accurate estimates.

DSAC-T [27] introduced refinements: (1) expected-value substitution for more stable targets, (2) twin distributional critics to reduce bias, and (3) variance-based gradient adjustments for stability. These advances highlight the

Table 1. Summary of entropy regularization techniques in reinforcement learning.

Method	Improvement	Limitations
Policy Gradient with Entropy Bonus [19]	Reduces premature convergence	Weak impact in high-dimensional tasks
Causal Entropy Regularization [22]	Handles sequential/causal decision making	Limited application, less explored empirically
Tsallis Entropy Regularization [24]	Flexible exploration, robustness	Parameter q difficult to tune
KL-Divergence Regularization [25, 26]	Smooth updates, stability in trust-region methods	Extra computation, requires reference policy
Maximum Entropy RL [20]	Theoretical foundation, encourages exploration	May cause overly stochastic policies
SAC Entropy Regularization [10]	Stable training, adaptive exploration	Sensitive to α tuning

interplay between distributional RL and adaptive entropy regularization.

2- 4- Summary

In summary, entropy regularization in RL evolved as in Table 1. Each step improved exploration efficiency, robustness, or stability. Distributional RL methods, including DSAC and DSAC-T, model the full return distribution to reduce overestimation bias. Existing DSAC variants do not explicitly exploit distributional variance for adaptive entropy control. Our work builds on this trajectory by coupling distributional critics with a novel adaptive entropy regularization scheme under the distributional Bellman framework, offering both theoretical guarantees and empirical improvements.

3- Proposed method

In this study, we propose an adaptive entropy regularization technique within the Distributional Soft Actor-Critic (DSAC) framework to mitigate Q-value overestimation and improve performance in continuous control tasks. The primary innovation of our method is the dynamic adjustment of the entropy regularization coefficient α, which governs the exploration-exploitation balance. Instead of using a fixed value of α, as in the original DSAC, we adapt α based on the variance observed in the model's loss and reward distributions during training.

3- 1- Entropy Regularization in DSAC

The DSAC algorithm uses a stochastic policy $\pi(a|s)$ and adds an entropy term to the objective function to encourage exploration. The objective function with entropy regularization is given by:

$$\mathcal{H}_q(\pi(\cdot\,|s)) = \frac{1}{q-1}\left(1 - \int \pi(a|s)^q da\right), \tag{15}$$

where:

α is the entropy regularization coefficient.

$Q_\theta(s,a)$ is the Q-value function.

$\log \pi_\theta(a|s)$ is the log probability of action a given state s under the current policy π.

In the original DSAC, α is a fixed value, leading to a static exploration-exploitation trade-off. However, our experiments on the OpenAI MuJoCo Humanoid task showed that variance in the model's loss and reward distributions fluctuates during training, making a static α suboptimal. To address this, we adapt α dynamically in response to these fluctuations, using the following procedure.

3- 2- Adaptive Entropy Regularization

Our method adjusts the entropy regularization coefficient α based on the observed variance in the model. The variance, $\mathbb{V}(X)$, is computed from the reward or loss distributions during training. We define a target variance interval $[V_{low}, V_{high}] = [0.05, 0.5]$, and adjust α accordingly to keep the variance within this range.

Let V_i be the variance of the reward or loss at the i-th epoch. The adjustment of α is determined by the following conditions:

1) If $V_i < V_{low}$ (i.e., variance is too low, indicating insufficient exploration):

$$\alpha_{i+1} = \alpha_i + \beta(V_{low} - V_i), \tag{16}$$

In this case, we increase α to encourage greater exploration by increasing the variance, where β is a positive scaling factor controlling the rate of change.

2) If $V_i > V_{high}$ (i.e., variance is too high, indicating excessive exploration):

$$\alpha_{i+1} = \alpha_i - \beta\left(V_i - V_{high}\right), \tag{17}$$

Here, α is reduced to decrease variance, thereby encouraging more exploitation and stabilizing the learning process.

By keeping the variance within the range $\left[V_{low}, V_{high}\right]$ the model dynamically adjusts its behavior to maintain an appropriate balance between exploration and exploitation.

3- 3- Variants of Adaptive Entropy Regularization

We explore four variations of this adaptive entropy regularization technique, each applying different schedules or mechanisms for adjusting α :

1) Linear Decay:

$$\alpha_{i+1} = \alpha_i - \gamma, \tag{18}$$

where γ is a constant decay factor, reducing α linearly over time. This method decreases exploration gradually but does not take variance into account directly.

2) Exponential decay:

$$\alpha_{i+1} = \alpha_i \exp(-\lambda i), \tag{19}$$

where γ is the decay rate. In this method, α decays exponentially over time, reducing exploration more quickly in the early stages of training.

3) Linear Adaptive: In the linear adaptive method, α is adjusted based on variance as described above. If the variance drops below the target interval, α is increased linearly:

$$\alpha_{i+1} = \alpha_i + \beta(V_{low} - V_i). \tag{20}$$

This encourages more exploration when variance is low.

4) Exponential Adaptive: In the exponential adaptive method, α is adjusted more aggressively based on variance:

$$V_i < V_{low} \Longrightarrow \alpha_{i+1} = \alpha_i \exp\left(\beta(V_{low} - V_i)\right), \tag{21}$$

$$V_i > V_{high} \Longrightarrow \alpha_{i+1} = \alpha_i \exp\left(-\beta\left(V_i - V_{high}\right)\right). \tag{22}$$

This allows the entropy regularization to change rapidly in response to large deviations in variance, either increasing or decreasing exploration as needed.

4- Theoretical Analysis

This section provides a theoretical interpretation of the proposed adaptive entropy regularization mechanism. We first formalize the mapping from critic uncertainty to the entropy coefficient, then analyze the boundedness and local stability of the resulting update, and finally discuss how the proposed mechanism can mitigate overestimation bias in distributional actor-critic learning.

4- 1- Uncertainty-to-Entropy Mapping

In the proposed method, the entropy coefficient is no longer treated as a fixed hyperparameter. Instead, it is dynamically adjusted according to the uncertainty of the critic. Let $X_i = \{X_{i,1}, X_{i,2}, \ldots, X_{i,n}\}$ denote a set of observations collected at training iteration i , where $X_{i,j}$ may correspond to critic losses, predicted returns, temporal-difference errors, or samples from the learned return distribution. The empirical uncertainty at iteration i is estimated as

$$V_i = \frac{1}{n}\sum_{j=1}^{n}\left(X_{i,j} - \overline{X_i}\right)^2, \tag{23}$$

where

$$\overline{X_i} = \frac{1}{n}\sum_{j=1}^{n} X_{i,j}. \tag{24}$$

The adaptive entropy regularization coefficient is then defined as a function of this uncertainty measure, $\alpha_i = f\left(V_i\right)$, where $f\left(\cdot\right)$ denotes an uncertainty-to-entropy mapping. The purpose of this mapping is to regulate the exploration-exploitation trade-off according to the critic's confidence. When the critic exhibits high uncertainty, a larger entropy coefficient encourages a more stochastic policy and prevents premature exploitation of unreliable value estimates. Conversely, when the critic uncertainty is low, the entropy coefficient can be reduced to allow the policy to exploit the learned value function more aggressively.

In this work, the uncertainty-to-entropy adaptation is controlled by two thresholds, V_{low} and V_{high}. These thresholds define an uncertainty interval in which the current entropy coefficient is considered adequate. If the estimated variance falls outside this interval, the entropy coefficient is adjusted accordingly:

$$\alpha_{i+1} = \begin{cases} \alpha_i \cdot \beta_{inc}, & V_i < V_{low} \\ \alpha_i \cdot \beta_{decay}, & V_i > V_{high} \\ \alpha_i, & V_{low} \le V_i \le V_{high} \end{cases} \tag{25}$$

where $\beta_{inc} > 1$ increases the entropy coefficient and $0 < \beta_{decay} < 1$ decreases it. To avoid degenerate deterministic behavior, the coefficient is projected onto a feasible interval:

$$\alpha_{i+1} \leftarrow \max(\alpha_{i+1}, \alpha_{min}) \tag{26}$$

where $\alpha_{min} > 0$ is the minimum allowable entropy coefficient.

For completeness, the additive adaptive variant can be written as

$$\alpha_{i+1} = \begin{cases} \alpha_i + \beta(V_{low} - V_i), & V_i < V_{low} \\ \alpha_i - \beta(V_i - V_{high}), & V_i > V_{high} \\ \alpha_i, & V_{low} \le V_i \le V_{high} \end{cases} \quad (27)$$

where $\beta > 0$ controls the sensitivity of the update. Both multiplicative and additive forms implement the same principle: the entropy coefficient is adapted according to the deviation of critic uncertainty from a desired range.

4- 2- Assumptions

To analyze the behavior of the proposed mechanism, we consider the following mild assumptions.

Assumption 1 (Bounded uncertainty estimate). The empirical uncertainty estimate V_i is bounded for all iterations:

$$0 \le V_i \le V_{max} < \infty. \quad (28)$$

This assumption is reasonable in practical implementations where rewards, critic losses, or distributional returns are either normalized or observed over finite mini-batches.

Assumption 2. (Bounded entropy coefficient). The entropy coefficient is lower bounded by $\alpha_{min} > 0$. Optionally, an upper bound α_{max} may also be enforced:

$$\alpha_i \in [\alpha_{min}, \alpha_{max}]. \quad (29)$$

Although the lower bound is sufficient to avoid collapse to a deterministic policy, an upper bound can be useful in practice to avoid excessive exploration.

Assumption 3. (Smooth policy parameterization). The policy $\pi_\theta(a|s)$ is differentiable with respect to its parameters θ, and the gradient of the entropy-regularized objective is locally Lipschitz continuous.

Assumption 4. (Bounded critic error). The estimation error of the critic,

$$\epsilon(s,a) = \hat{Q}(s,a) - Q^{\pi}(s,a), \quad (30)$$

has bounded variance:

$$\mathbb{V}[\epsilon(s,a)] \le \sigma_Q^2. \quad (31)$$

These assumptions are standard in theoretical analyses of actor-critic algorithms with function approximation and are used only to establish stability and interpretability of the adaptive entropy mechanism, rather than to claim global convergence in arbitrary nonlinear settings.

4- 3- Boundedness of the Adaptive Entropy Coefficient

The first property of the proposed update is that the entropy coefficient remains strictly positive throughout training.

Lemma 1 (Lower boundedness of α). Let $\alpha_0 \ge \alpha_{min} > 0$. Under the update rule in Eq. (25) followed by the projection in Eq. (26), the entropy coefficient satisfies

$$\alpha_i \ge \alpha_{min}, \quad \forall i \ge 0. \quad (32)$$

Proof. The update in Eq. (25) may either increase, decrease, or keep the entropy coefficient unchanged. Regardless of the intermediate value produced by this update, the projection step in Eq. (26) enforces

$$\alpha_{i+1} = \max(\alpha_{i+1}, \alpha_{min}). \quad (33)$$

Therefore, if $\alpha_i \ge \alpha_{min}$, then $\alpha_{i+1} \ge \alpha_{min}$. Since $\alpha_0 \ge \alpha_{min}$ by assumption, the result follows by induction.

If an upper bound is also enforced, the update becomes

$$\alpha_{i+1} \leftarrow \min(\max(\alpha_{i+1}, \alpha_{min}), \alpha_{max}). \quad (34)$$

This projection guarantees that $\alpha_i \in [\alpha_{min}, \alpha_{max}]$ for all iterations.

Lemma 2 (Boundedness under interval projection). If Eq. (34) is applied after each update and $\alpha_0 \in [\alpha_{min}, \alpha_{max}]$, then

$$\alpha_i \in [\alpha_{min}, \alpha_{max}], \quad \forall i \ge 0. \quad (35)$$

Proof. The proof follows directly from the projection operator in Eq. (34), which maps any real-valued update back to the compact interval $[\alpha_{min}, \alpha_{max}]$.

The boundedness property is important because it prevents two undesirable regimes: *entropy collapse*, where α becomes too small and the policy prematurely converges to near-deterministic behavior, and *excessive randomization*, where α becomes too large and prevents value-based exploitation.

4- 4- Local Stability of Adaptive Entropy Regularization

We now analyze the local stability of the adaptive entropy mechanism. Intuitively, the proposed method forms a feedback loop between critic uncertainty and policy entropy. The uncertainty estimate V_i determines α_i, and α_i influences subsequent exploration and data collection, which in turn affects the critic uncertainty.

Let $V^{\star} \in \left[V_{low}, V_{high}\right]$ denote a desired uncertainty level or equilibrium region. When V_i lies inside this interval, no update is applied to α_i. When V_i moves outside this interval, the entropy coefficient is adjusted to steer the learning process back toward a more stable regime.

Theorem 1 (Local stability of the uncertainty-to-entropy feedback). Suppose that the uncertainty sequence $\{V_i\}$ remains in a bounded neighborhood of a stationary value $V^{\star}$, i.e.,

$$|V_i - V^{\star}| \le \delta, \qquad \forall i, \tag{36}$$

for some $\delta > 0$. Suppose further that the mapping $f(\cdot)$ from uncertainty to entropy is Lipschitz continuous in this neighborhood with constant $L_f > 0$. Then the entropy sequence $\{\alpha_i\}$ remains bounded around $\alpha^{\star} = f(V^{\star})$:

$$|\alpha_i - \alpha^{\star}| \le L_f \delta. \tag{37}$$

Proof. Since f is Lipschitz continuous, for any V_i and $V^{\star}$ in the local neighborhood,

$$|f(V_i) - f(V^{\star})| \le L_f |V_i - V^{\star}|. \tag{38}$$

Using $\alpha_i = f(V_i)$ and $\alpha^{\star} = f(V^{\star})$, we obtain

$$|\alpha_i - \alpha^{\star}| = |f(V_i) - f(V^{\star})| \le L_f |V_i - V^{\star}|. \tag{39}$$

By Eq. (36), finally we obtain

$$|\alpha_i - \alpha^{\star}| \le L_f \delta. \tag{40}$$

This proves that bounded fluctuations in critic uncertainty induce bounded fluctuations in the entropy coefficient.

This theorem indicates that the proposed adaptive entropy mechanism is locally stable as long as the estimated critic uncertainty does not diverge. In practical terms, if the critic variance fluctuates within a bounded region, the entropy coefficient will also remain within a controlled neighborhood. This is particularly important for stabilizing actor updates, since large oscillations in α may cause unstable policy improvement.

4- 5- Effect on Actor Update Stability

The entropy-regularized actor objective used in the proposed method can be written as

$$J_{\pi}(\theta) = \mathbb{E}_{s\sim\mathcal{D}, a\sim\pi_{\theta}}\left[\alpha_i \log \pi_{\theta}(a|s) - Q_{\phi}(s,a)\right], \tag{41}$$

where $Q_{\phi}(s,a)$ is the critic estimate and α_i is the adaptive entropy coefficient at iteration i. The gradient of this objective is affected by α_i through the entropy term:

$$\nabla_{\theta} J_{\pi}(\theta) = \mathbb{E}\left[\nabla_{\theta}\left(\alpha_i \log \pi_{\theta}(a|s) - Q_{\phi}(s,a)\right)\right]. \tag{42}$$

When α_i is bounded, the entropy contribution to the actor gradient is also controlled. Under Assumption 3, if $\nabla_{\theta} \log \pi_{\theta}(a|s) \le G_{\pi}$ for some constant G_{π}, then

$$\|\alpha_i \nabla_{\theta} \log \pi_{\theta}(a|s)\| \le \alpha_{max} G_{\pi}. \tag{43}$$

Therefore, enforcing bounds on α_i prevents the entropy term from dominating the policy update. This contributes to stable training while preserving the ability to increase exploration when critic uncertainty is high.

4- 6- Overestimation Bias Analysis

A central motivation of the proposed method is to reduce overestimation bias in actor-critic learning. Overestimation occurs when the policy update over-exploits noisy critic estimates, especially when the critic assigns spuriously high values to some actions. Let the critic estimate be decomposed as

$$\hat{Q}(s,a) = Q^{\pi}(s,a) + \epsilon(s,a), \tag{44}$$

where $Q^{\pi}(s,a)$ is the true action-value function under policy π and $\dot{\mathbf{o}}(s,a)$ is the estimation error. For a low-entropy or nearly deterministic policy, the actor tends to select actions that maximize $\hat{Q}(s,a)$. In such a case, the selected action may correspond to a positive realization of the critic error, leading to

$$\mathbb{E}\left[\max_a \hat{Q}(s,a)\right] \ge \max_a Q^{\pi}(s,a). \tag{45}$$

The gap between the two sides is the overestimation bias.

Entropy regularization mitigates this effect by smoothing the policy distribution over actions. Instead of assigning all probability mass to the action with the largest noisy estimate, the policy optimizes an entropy-regularized objective. The corresponding soft value can be expressed as

$$V_{\alpha}(s) = \alpha \log \int_{\mathcal{A}} \exp\left(\frac{Q(s,a)}{\alpha}\right) da. \tag{46}$$

For discrete actions, the integral is replaced by a

summation. The parameter α controls the smoothness of the maximum operator. As $\alpha \to 0$, the soft value approaches the hard maximum:

$$\lim_{\alpha \to 0} \alpha \log \sum_{a} \exp\left(\frac{Q(s,a)}{\alpha}\right) = \max_{a} Q(s,a). \quad (47)$$

For larger α, the operator becomes smoother and less sensitive to isolated noisy overestimates. This observation motivates the following proposition.

Proposition 1 (Entropy smoothing reduces sensitivity to critic noise). Let $\hat{Q}(s,a) = Q^{\pi}(s,a) + \in(s,a)$, where $\in(s,a)$ is zero-mean estimation noise with bounded variance. The entropy-regularized soft value operator in Eq. (46) is smoother than the hard maximum operator and reduces the sensitivity of value estimation to isolated positive critic errors. Increasing α increases this smoothing effect.

Proof sketch. The hard maximum operator selects the action with the largest estimated value and is therefore highly sensitive to positive outliers in $\hat{Q}(s,a)$. By contrast, the soft value operator aggregates action values through a log-sum-exp transformation. The gradient of the soft value with respect to $\hat{Q}(s,a)$ is a Boltzmann distribution:

$$\frac{\partial V_{\alpha}(s)}{\partial \hat{Q}(s,a)} = \frac{\exp\left(\hat{Q}(s,a)/\alpha\right)}{\sum_{a'} \exp\left(\hat{Q}(s,a')/\alpha\right)}. \quad (48)$$

For small α, this distribution concentrates on the largest estimated action value. For larger α, the distribution becomes smoother and assigns non-negligible probability to multiple actions. Therefore, the influence of a single overestimated action is reduced as α increases.

In the proposed method, α_i is increased or maintained in response to critic uncertainty. Therefore, when the critic estimates are noisy, the actor update becomes less aggressive with respect to potentially overestimated values. This creates a variance-aware regularization effect.

4- 7- Computational Complexity

The proposed adaptive entropy mechanism introduces minimal computational overhead. The additional operation is the computation of the empirical variance in Eq. (23). Given n samples, this computation has time complexity $\mathcal{O}(n)$. The update of α itself is constant-time $\mathcal{O}(1)$. Let $\mathcal{C}_{dsac}$ denote the computational cost of one standard DSAC update, including critic update, actor update, and target network update. The proposed method changes this cost to

$$\mathcal{C}_{ours} = \mathcal{C}_{dsac} + \mathcal{O}(n) + \mathcal{O}(1). \quad (49)$$

Since n is typically the mini-batch size or the number of recent observations used for variance estimation, and since this variance computation does not require additional neural network forward or backward passes beyond those already used by DSAC, the asymptotic complexity of the proposed method remains the same as that of the base DSAC algorithm.

Similarly, the memory overhead is negligible. The method only requires storing recent scalar statistics or computing the variance from values already available during training. Therefore, the proposed adaptive entropy regularization improves exploration control and training stability without significantly increasing computational or memory complexity.

4- 8- Discussion of Theoretical Scope

The preceding analysis is intended to justify the stability and regularization behavior of the proposed adaptive entropy mechanism. It does not claim global convergence of deep distributional actor-critic learning with nonlinear function approximation, which remains difficult to guarantee in general. Instead, the analysis establishes three practically relevant properties:

1) The entropy coefficient remains positive and bounded under the proposed projection rule.
2) Bounded fluctuations in critic uncertainty lead to bounded fluctuations in the entropy coefficient.
3) Increasing the entropy coefficient in high-uncertainty regimes smooth the actor update and reduces sensitivity to overestimated critic values.

These properties explain why coupling critic uncertainty with entropy regularization can improve training stability and reduce over-exploitation of unreliable value estimates.

5- Empirical Evaluation

5- 1- Baseline

We compare our algorithms against Distributional Soft Actor-Critic (DSAC) [14] which have been extensively verified and applied in a variety of challenging domains specially continuous control tasks. Using this algorithm as baseline, the performance of the proposed extensions of the DSAC algorithm can be evaluated objectively. This paper improves the overestimation mitigation by clipping variance into a smaller valid range.

5- 2- Experimental Setup

All the off-policy algorithms mentioned above are implemented in PyTorch. All algorithms adopt almost the same neural network architecture and hyperparameters. The hyperparameters associated with the experiments are depicted in Table 2. These experiments show that it causes subtle improvements in the overestimation originated in the Q-learning setting. We evaluate our proposed methods on two benchmarks to elaborate the effect of entropy term and the effect of bounding variance to the overestimation mitigation process.

5- 3- Benchmark suite

To evaluate the performance of the proposed Adaptive DSAC algorithm, we utilize a suite of standard continuous

Algorithm 1 Variance-Adaptive Entropy Regularization for DSAC

Initialize policy parameters θ, critic parameters ϕ, target parameters $\bar{\phi}$
Initialize replay buffer $\mathcal{D}$
Initialize entropy coefficient $\alpha \leftarrow \alpha_0$
Set thresholds V_{low}, V_{high}
Set adaptation factors $\beta_{\text{inc}} > 1$, $0 < \beta_{\text{decay}} < 1$
Set bounds $\alpha_{\min}$ and optionally $\alpha_{\max}$
for each training iteration **do**
  Collect transitions using policy π_θ and store them in $\mathcal{D}$
  Sample a mini-batch from $\mathcal{D}$
  Estimate critic uncertainty V_i using Eq. (1)
  if $V_i > V_{\text{high}}$ **then**
    $\alpha \leftarrow \alpha \cdot \beta_{\text{inc}}$
  else if $V_i < V_{\text{low}}$ **then**
    $\alpha \leftarrow \alpha \cdot \beta_{\text{decay}}$
  else
    $\alpha \leftarrow \alpha$
  end if
  $\alpha \leftarrow \max(\alpha, \alpha_{\min})$
  Optionally, $\alpha \leftarrow \min(\alpha, \alpha_{\max})$
  Update critic parameters ϕ
  Update actor parameters θ using the adaptive entropy coefficient α
  Update target network parameters $\bar{\phi}$
end for

Table 2. Hyperparameters used in experiments.

Hyperparameters	Value
Optimizer	Adam
Number of hidden layers	5
Number of hidden units per layer	256
Nonlinearity of hidden layer	GELU
Replay buffer size	5×10^5
Batch size	256
Discount factor (γ)	0.99
Update interval (m)	2
Target smoothing coefficient (τ)	0.001
Reward scale	0.2
Number of actor processes	6
Number of learner processes	4
Number of buffer processes	3
Bounds of variance	[0.05,0.5]
Clipping boundary	$b = 10$

control environments from the MuJoCo physics simulator via the OpenAI Gym interface. The environments include HalfCheetah-v2, Hopper-v2, Walker2d-v2, and Ant-v2, and Humanoid-v2, which are widely adopted in the literature for benchmarking deep RL algorithms. These tasks vary in terms of complexity, dynamics, and degrees of freedom, thereby providing a comprehensive testbed for policy evaluation.

Each environment is characterized by a continuous action space and high-dimensional state representations, requiring agents to learn stable locomotion strategies. For instance, HalfCheetah-v2 involves a 2D bipedal robot aiming to maximize forward velocity, while Ant-v2 features a quadruped with eight controllable joints navigating through terrain. The agents are trained for 1 million timesteps using the same hyperparameter configurations across all methods to ensure fair comparisons. We report the average episodic return over five independent runs with different random seeds, and results are smoothed using a moving average window of size 10. This benchmark setup aligns with existing work [9,10] and enables direct performance comparisons with SAC and TD3.

5- 4- MuJoCo Humanoid Benchmark

To evaluate the proposed method, we conduct experiments on the Humanoid-v2 benchmark from the MuJoCo suite, a widely used high-dimensional continuous control task available through the OpenAI Gym interface.

Table 2. Humanoid-v2 Observation (State) Space Components.

Category	Description	Dim.
Joint positions	Relative joint angles and positions	66
Joint velocities	Angular and linear joint velocities	66
Inertial data	Mass, center of mass, velocities	17
Contact forces	External force vectors on body parts	260
Total		376

The Humanoid-v2 environment simulates a 3D bipedal humanoid robot with articulated limbs and a torso, where the objective is to maintain balance and move forward as quickly as possible without falling.

The humanoid model includes 17 actuated joints and multiple body segments connected through a physical simulator, making it one of the most challenging tasks due to its high dimensionality, complex dynamics, and sensitivity to control inputs. The task requires precise control to achieve stable locomotion and resilience against perturbations.

Observation Space: The observation (state) space is a 376-dimensional continuous vector, which includes:

- Joint positions (excluding the global x, y, z position): 66 dimensions
- Joint velocities: 66 dimensions
- Inertial measurements (e.g., center of mass inertia, mass, velocity): 17 dimensions
- Actuator forces, external contact forces: 260 dimensions

A breakdown of the state space is summarized in Table 3.

Action Space: The action space is a 17-dimensional continuous vector, where each element corresponds to the torque applied at one of the humanoid's motor joints. The agent outputs a vector $a \in [-1,1]^{17}$ at each timestep, which is scaled and applied as control input by MuJoCo.

Reward Structure and Termination: The reward function for Humanoid-v2 encourages forward progress and includes penalties for control effort and contact forces. Specifically, the reward includes:

- A positive reward proportional to the forward velocity of the center of mass.
- A penalty proportional to the square of the control input to discourage excessive torque.
- A small penalty for contact forces (excluding feet) to encourage natural movement.

The episode terminates when the humanoid falls (i.e., its torso's height drops below a threshold) or when the maximum episode length is reached (default: 1000 steps).

Table 4. Humanoid-v2 Action Space.

Joint Group	Control Dimension(s)
Abdomen	Pitch, Yaw, Roll (3)
Hip (Left/Right)	Pitch, Yaw, Roll (6)
Knee (Left/Right)	Pitch (2)
Ankle (Left/Right)	Pitch, Roll (4)
Shoulder (Left/Right)	Pitch, Roll (2)
Total	17

This complex, high-variance environment serves as a stress test for RL algorithms, particularly with respect to stability, exploration-exploitation balance, and resilience to overestimation bias in Q-values.

5- 5- MuJoCo Swimmer Benchmark

To evaluate the proposed method, we also consider the Swimmer-v2 benchmark from the MuJoCo suite, a commonly used low-dimensional continuous control task available through the OpenAI Gym interface. The Swimmer-v2 environment simulates a simplified 2D three-link snake-like agent in a viscous fluid, where the objective is to generate forward locomotion by producing coordinated joint torques.

Unlike high-dimensional tasks such as Humanoid-v2, the Swimmer benchmark is relatively low-dimensional and is often used as a testbed for validating the stability and sample

Table 5. Swimmer-v2 Observation (State) Space Components.

Category	Description	Dim.
Joint angles	Relative joint orientations	2
Joint velocities	Angular velocities of joints	2
Torso orientation	Cosine and sine of global torso angle	2
Torso velocities	Linear velocity (x, y) of the torso	2
Total		8

efficiency of reinforcement learning algorithms. Its dynamics are easier to model, but successful locomotion still requires rhythmic coordination of the joints.

Observation Space: The observation (state) space is an 8-dimensional continuous vector, which includes:

- Joint angles (excluding global position): 2 dimensions
- Joint angular velocities: 2 dimensions
- Torso orientation (sine and cosine of angle): 2 dimensions
- Linear velocities of the torso (x and y directions): 2 dimensions

A breakdown of the state space is summarized in Table 5.

Action Space: The action space is a 2-dimensional continuous vector, where each element corresponds to the torque applied at one of the swimmer's two actuated joints. The agent outputs a vector $a \in [-1,1]^2$ at each timestep, which is scaled and applied as control input by MuJoCo.

Reward Structure and Termination: The reward function for Swimmer-v2 primarily encourages forward progress in the x-direction and includes a penalty for large control magnitudes. Specifically, the reward includes:

- A positive reward equal to the forward velocity of the swimmer's torso in the x-direction.
- A penalty proportional to the squared magnitude of the control input, discouraging inefficient swimming motions.

Episodes terminate only when the maximum horizon length is reached (default: 1000 steps). There is no termination conditions based on falling or failure, unlike in the humanoid task.

Despite its simplicity, the Swimmer benchmark provides valuable insights into how well reinforcement learning algorithms can discover coordinated, rhythmic policies for locomotion in low-dimensional continuous domains.

5- 6- Results

In this section, the results of the experimental evaluation of the proposed methods are illustrated and justified. In our benchmark, the MuJoCo humanoid continuous control task, we have run 5 models each of which has four seeds, each one trained the algorithm for 1,500,000 iterations.

Table 6. Swimmer-v2 Action Space.

Joint	Control Dimension
Joint 1	Torque (1)
Joint 2	Torque (1)
Total	2

As shown in Figure 1, it appears that the loss function is strictly decreasing, but the variance is increasing due to adaptivity in order to handle exploration-exploitation dilemma. Also, we see that the exponential adaptive version slightly outperforms the baseline and other methods in the case of the expected accumulated return, which is depicted in Figure 2.

In addition to the humanoid experiments, we also conducted evaluations on the Swimmer-v2 benchmark, which provides a simpler and lower-dimensional continuous control task. Unlike the high-dimensional humanoid robot, the swimmer is a three-link agent in a viscous 2D environment with only two actuated joints and an 8-dimensional observation space. The objective is to generate forward locomotion by producing coordinated joint torques in a rhythmic manner.

As shown in Figure 3, the swimmer task demonstrates much lower variance across different seeds, and the learning curves converge more stably. Nevertheless, we observe that the exponential adaptive method continues to outperform the baseline, achieving higher forward progress with less control effort, which highlights the robustness of our approach even in relatively low-dimensional control settings.

6- Conclusion

The proposed variance-adaptive DSAC method extends

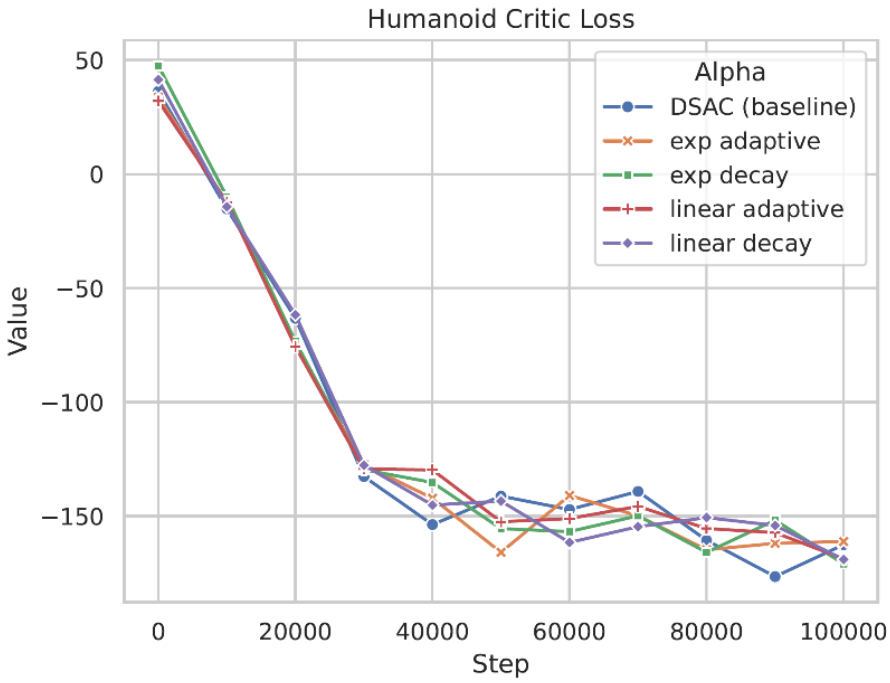

Fig. 1. Humanoid-v2 critic loss.

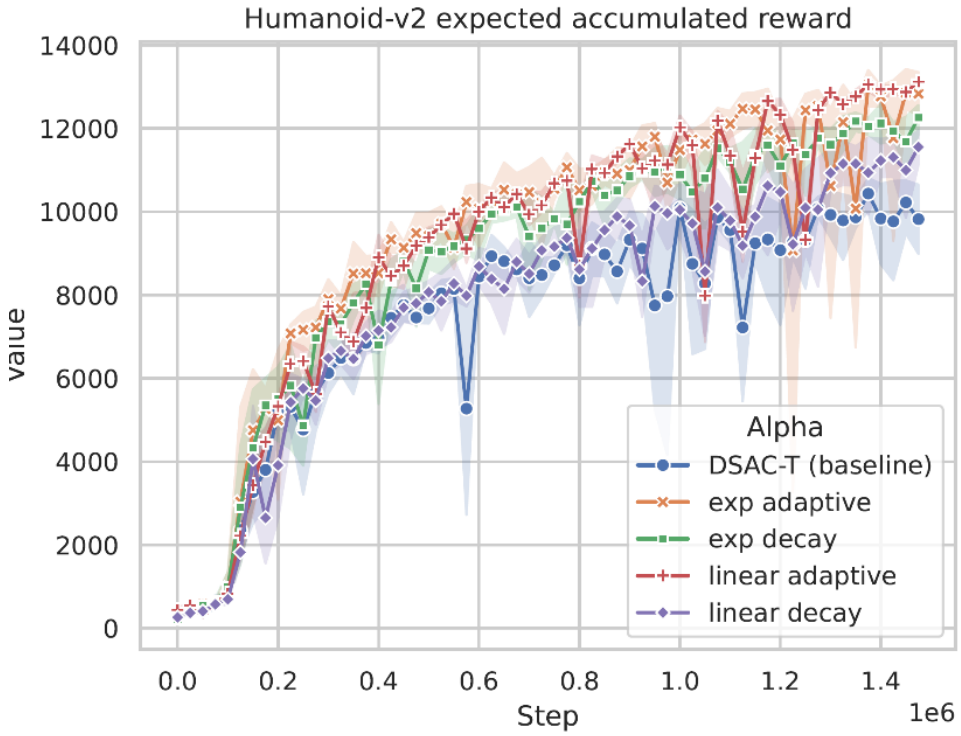

Fig. 2. Humanoid-v2 expected accumulated return.

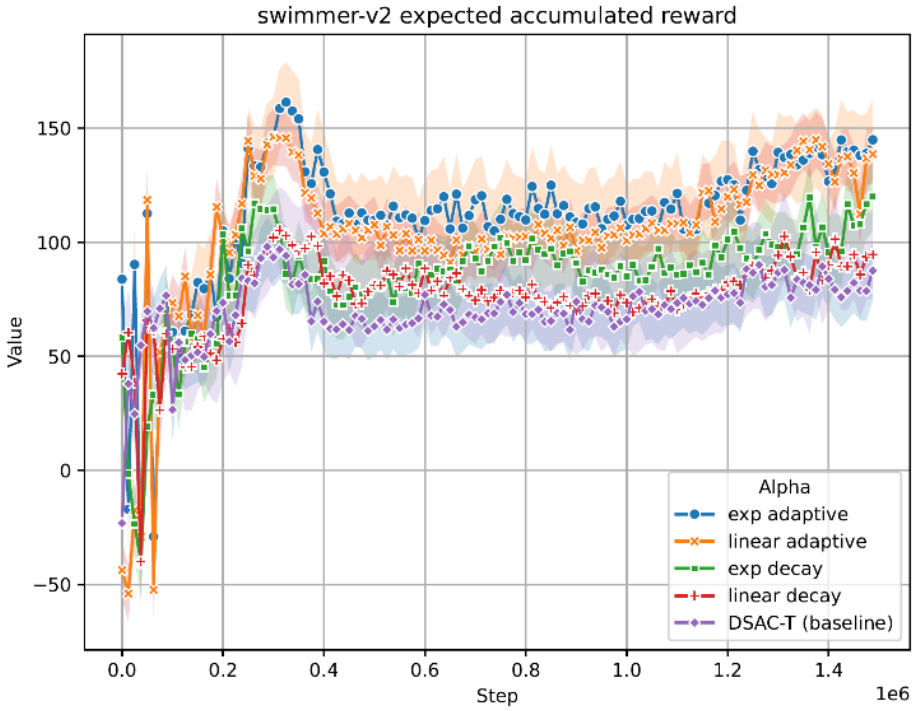

Fig. 3. Swimmer-v2 expected accumulated return.

entropy-regularized reinforcement learning by incorporating uncertainty-aware entropy control within a distributional setting. This process keeps the variance of the state-action returns within reasonable range to address exploding and vanishing gradient problems. We evaluate our proposed methods on the suite of OpenAI control tasks. Experimental results confirm improved stability and better generalization compared to existing SAC and DSAC variants.

References

[1] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. Van Den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot et al., "Mastering the game of go with deep neural networks and tree search," nature, vol. 529, no. 7587, pp. 484–489, 2016.

[2] O. Vinyals, I. Babuschkin, W. M. Czarnecki, M. Mathieu, A. Dudzik, J. Chung, D. H. Choi, R. Powell, T. Ewalds, P. Georgiev et al., "Grandmaster level in StarCraft II using multiagent reinforcement learning," nature, vol. 575, no. 7782, pp. 350–354, 2019.

[3] S. Levine, C. Finn, T. Darrell, and P. Abbeel, "End-to-end training of deep visuomotor policies," Journal of Machine Learning Research, vol. 17, no. 39, pp. 1–40, 2016.

[4] S. Gu, E. Holly, T. P. Lillicrap, and S. Levine, "Deep reinforcement learning for robotic manipulation," arXiv preprint arXiv:1610.00633, vol. 1, no. 1, 2016.

[5] F.-M. Luo, T. Xu, H. Lai, X.-H. Chen, W. Zhang, and Y. Yu, "A survey on model-based reinforcement learning," Science China Information Sciences, vol. 67, no. 2, p. 121101, 2024.

[6] V. Konda and J. Tsitsiklis, "Actor-critic algorithms," Advances in neural information processing systems, vol. 12, 1999.

[7] T. Degris, P. M. Pilarski, and R. S. Sutton, "Model-free reinforcement learning with continuous action in practice," in 2012 American control conference (ACC). IEEE, 2012, pp. 2177–2182.

[8] 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," arXiv preprint arXiv:1509.02971, 2015.

[9] S. Fujimoto, H. Hoof, and D. Meger, "Addressing function approximation error in actor-critic methods," in International conference on machine learning. PMLR, 2018, pp. 1587–1596.

[10] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine, "Soft Actor-Critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor," pp. 1861–1870, 2018.

[11] M. G. Bellemare, W. Dabney, and R. Munos, "A distributional perspective on reinforcement learning," in International conference on machine learning. PMLR, 2017, pp. 449–458.

[12] M. G. Bellemare, W. Dabney, and M. Rowland, Distributional Reinforcement Learning. MIT Press, 2023, http://www.distributional-rl.org.

[13] X. Ma, L. Xia, Z. Zhou, J. Yang, and Q. Zhao, "Dsac: Distributional soft actor-critic for risk-sensitive reinforcement learning," arXiv preprint arXiv:2004.14547, 2020.

[14] J. Duan, Y. Guan, S. E. Li, Y. Ren, Q. Sun, and B. Cheng, "Distributional soft actor-critic: Off-policy reinforcement learning for addressing value estimation errors," IEEE Transactions on Neural Networks and Learning Systems, vol. 33, no. 11, pp. 6584–6598, 2022.

[15] M. Rowland, M. Bellemare, W. Dabney, R. Munos, and Y. W. Teh, "An analysis of categorical distributional reinforcement learning," in International Conference on Artificial Intelligence and Statistics. PMLR, 2018, pp. 29–37.

[16] W. Dabney, M. Rowland, M. Bellemare, and R. Munos, "Distributional reinforcement learning with quantile regression," in Proceedings of the AAAI Conference on Artificial Intelligence, vol. 32, no. 1, 2018.

[17] N. Mohammadpour, M. Fozi, M. M. Ebadzadeh, A. Azimi, and A. K. Iglie, "Proximal policy optimization with adaptive generalized advantage estimate," in Proceedings of the First International Conference on Machine Learning and Knowledge Discovery (MLKD 2024), 2024, pp. 453–458. [Online]. Available: https://mlkd.aut.ac.ir/proceedings/2024/paper/4B.7.pdf

[18] N. Mohammadpour, M. Fozi, M. M. Ebadzadeh, A. Azimi, and A. Kamalie Iglie, "Proximal policy optimization with adaptive generalized advantage estimate: critic-aware refinements," Journal of Mathematical Modeling, 2025.

[19] J. Liu, X. Gu, and S. Liu, "Policy optimization reinforcement learning with entropy regularization," arXiv preprint arXiv:1912.01557, 2019.

[20] T. Haarnoja, H. Tang, P. Abbeel, and S. Levine, "Reinforcement learning with deep energy-based policies," in International conference on machine learning. PMLR, 2017, pp. 1352–1361.

[21] B. D. Ziebart, A. L. Maas, J. A. Bagnell, A. K. Dey et al., "Maximum entropy inverse reinforcement learning." in AAAI, vol. 8. Chicago, IL, USA, 2008, pp. 1433–1438.

[22] B. D. Ziebart, J. A. Bagnell, and A. K. Dey, "Modeling interaction via the principle of maximum causal entropy," 2010.

[23] R. J. Williams, "Simple statistical gradient-following algorithms for connectionist reinforcement learning," Machine learning, vol. 8, no. 3, pp. 229–256, 1992.

[24] K. Lee, S. Kim, S. Lim, S. Choi, and S. Oh, "Tsallis reinforcement learning: A unified framework for maximum entropy reinforcement learning," arXiv preprint arXiv:1902.00137, 2019.

[25] J. Schulman, S. Levine, P. Abbeel, M. Jordan, and P. Moritz, "Trust region policy optimization," in International conference on machine learning. PMLR, 2015, pp. 1889–1897.

[26] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," arXiv preprint arXiv:1707.06347, 2017.

[27] J. Duan, W. Wang, L. Xiao, J. Gao, and S. E. Li, "DSAC-T: Distributional soft actor-critic with three refinements," arXiv preprint arXiv:2310.05858, 2023.

HOW TO CITE THIS ARTICLE

M. Fozi, M. M. Ebadzadeh, Distributional Soft Actor-Critic with Adaptive Entropy Regularization, AUT J. Model. Simul., 57(2) (2025) 215-228.

DOI: 10.22060/miscj.2026.23574.5387