Distributional Soft Actor-Critic with Adaptive Entropy Regularization

Meysam Fozi
Amirkabir University of Technology
Tehran, Iran
meysam.fozi@aut.ac.ir

Zahra Ghorrati
Purdue University
West Lafayette, IN, USA
zghorrat@purdue.edu

Ahmad Esmaeili
Wichita State University
Wichita, KS, USA
ahmad.esmaeili@wichita.edu

Mohammad Mehdi Ebadzadeh
Amirkabir University of Technology
Tehran, Iran
ebadzadeh@aut.ac.ir

ABSTRACT

Soft Actor-Critic (SAC) and its distributional extensions achieve strong performance by combining entropy regularization with off-policy learning. However, existing automatic temperature tuning mechanisms rely on fixed target entropy formulations, entirely ignoring the rich empirical variance information observed during training. In this paper, we propose a variance-adaptive entropy regularization framework for Distributional SAC (DSAC). Our approach dynamically adjusts the entropy temperature as a function of the empirical variance of the model's loss or rewards. By introducing linear and exponential adaptation schemes, we directly couple exploration strength with these variance metrics, utilizing them as a practical proxy for training instability. Evaluated on continuous control tasks from the MuJoCo suite, our method demonstrates improved stability and generalization compared to standard SAC and DSAC-T. Ultimately, this variance-adaptive strategy indirectly mitigates overestimation and provides a more efficient solution to the exploration-exploitation dilemma in continuous reinforcement learning.

KEYWORDS

Reinforcement Learning, Continuous Control, Soft Actor-Critic, Distributional Soft Actor-Critic, Entropy Regularization

1 INTRODUCTION

Reinforcement Learning (RL) has shown remarkable success across various domains, from game playing [21, 22] to robotics [7, 12] and control systems [15]. Among the numerous RL algorithms, actor-critic methods have become standard due to their ability to handle continuous action spaces and their stable convergence properties [3, 10]. Algorithms such as Deep Deterministic Policy Gradient (DDPG) [13], Twin Delayed DDPG (TD3) [6], and Soft Actor-Critic (SAC) [9] represent pivotal advancements in this family, leveraging deterministic and stochastic policy updates to address sample efficiency and stability challenges.

However, standard actor-critic formulations, including SAC, are inherently limited by their representation of the value function, as they compute a single expected value for each state-action pair. This scalar expectation fails to capture the intrinsic uncertainty and variance associated with future returns. To address this limitation, distributional RL [1, 2] was introduced to model the entire distribution of returns for each state-action pair. Building on this, Distributional Soft Actor-Critic (DSAC) [16] incorporates the distributional perspective into the maximum entropy framework. By learning a probability distribution over possible returns rather than just the mean, DSAC agents obtain a richer representation of the return landscape, allowing for more informed decision-making and robust performance.

Despite these representational advantages, distributional Q-learning algorithms still contend with overestimation bias. Recent extensions, such as the Distributional Soft Policy Iteration (DSPI) framework [4], attempt to address this by refining how distributional deep neural networks are trained in the actor-critic setting. Furthermore, recent literature has demonstrated the value of adaptive optimization mechanisms in RL [17], including critic-aware refinements [18]. However, existing automatic entropy tuning mechanisms in SAC and DSAC still rely on fixed target entropy formulations or generic scheduling, failing to dynamically adapt to the empirical variance observed during the learning process..

To bridge this gap, we propose a principled integration of variance-aware entropy adaptation into the DSAC framework. Rather than relying on fixed entropy tuning approaches, our method provides a feedback-driven exploration mechanism. While motivated by the uncertainty modeling perspective of distributional RL, we utilize the empirical variance of recent training signals (such as episodic rewards or critic loss) as a practical proxy for training instability. This dynamically adjusts the temperature parameter based on the learning process's actual fluctuations.

The main contributions of this paper are summarized as follows:

- We introduce an adaptive entropy regulation scheme grounded in the empirical variance of training metrics.
- We propose a mechanism to decouple the actor and critic updates to improve convergence and the overall robustness of the learning process.
- We provide evaluations on standard continuous control benchmarks, including challenging MuJoCo environments such as Humanoid-v2[1] and Swimmer-v2[2]. Our results demonstrate that Adaptive DSAC significantly outperforms standard SAC and TD3 in both sample efficiency and final performance.

Proc. of the Adaptive and Learning Agents Workshop (ALA 2026), Aydeniz, Delgrange, Mohammedalamen, Yang (eds.), May 25 – 26, 2026, Paphos, Cyprus, https://alaworkshop2026.github.io/. 2026.

[1] https://gymnasium.farama.org/environments/mujoco/humanoid/
[2] https://gymnasium.farama.org/environments/mujoco/swimmer/

The remainder of this paper is organized as follows. Section 2 provides the preliminary background on Soft Actor-Critic and entropy regularization. Section 3 presents the core methodology of our proposed adaptive DSAC framework. Section 4 discusses the empirical evaluation of the model across various continuous control settings. Finally, Section 5 concludes the paper and outlines directions for future research.

2 PRELIMINARIES AND RELATED WORK

2.1 Standard Reinforcement Learning

The rigorous mathematical foundation for reinforcement learning (RL) is the Markov Decision Process (MDP), defined by the tuple $(\mathcal{S}, \mathcal{A}, \mathcal{R}, p, \gamma)$. Generally, the state space $\mathcal{S}$ and action space $\mathcal{A}$ are assumed to be continuous. The stochastic reward function, $R(r_t|s_t, a_t) : \mathcal{S} \times \mathcal{A} \to \mathcal{P}(r_t)$, maps 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 a probability distribution over the subsequent state s_{t+1}.

The agent's behavior is defined by a stochastic policy $\pi(a_t|s_t) : \mathcal{S} \to \mathcal{P}(a_t)$, and the state distribution induced by this policy is denoted by $\rho_\pi(s)$. In standard RL, the objective is to learn a policy that maximizes the expected future accumulated return:

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

where $\gamma \in (0, 1)$ is the discount factor. This classical formulation is inherently risk-neutral and focuses solely on maximizing the reward expectation. However, in complex continuous control tasks, this often leads to premature convergence to suboptimal policies due to insufficient exploration.

2.2 Entropy Regularization Methods

To mitigate the issue of premature convergence, entropy regularization was introduced to encourage stochasticity in policies. The foundational premise is that policies should not collapse to deterministic choices too early during the learning process; instead, a positive entropy term promotes exploration and prevents overfitting to suboptimal modes.

Early policy-gradient algorithms [23] incorporated entropy by augmenting the objective function:

$$J(\theta) = \mathbb{E}_{\pi_\theta} \left[\sum_t \log \pi_\theta(a_t|s_t) A^\pi(s_t, a_t) + \beta \mathcal{H}(\pi_\theta(\cdot|s_t)) \right], \tag{2}$$

where A^π is the advantage function. In this context, the entropy-bonus policy-gradient formulation [14] augments the expected return with a fixed-weight entropy term, replacing the expected reward r_i with $r_i + \beta\mathcal{H}(\pi(\cdot|s_i))$ 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. While this approach effectively encourages broader action distributions, the fixed coefficient β is highly problem-dependent and requires extensive manual tuning.

A major theoretical leap was the Maximum Entropy RL (MaxEnt RL) framework [8], which formalized entropy regularization as an intrinsic component of the optimal control objective [25]. The MaxEnt formulation utilizes the same structure, but formalizes the weight as a temperature parameter $\alpha > 0$:

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

where α controls the trade-off between reward maximization and entropy maximization. By framing entropy as an integral part of the return, MaxEnt RL yields policies that are inherently robust to noise and exhibit vastly improved exploration. Defining the soft action-value function as $Q(s, a)$, the corresponding soft Bellman operator for MaxEnt RL is defined as:

$$\begin{aligned} \mathcal{T}^\pi Q(s_t, a_t) = \mathbb{E}[r] \\ + \gamma \mathbb{E}_{\substack{s_{t+1}\sim p \\ a_{t+1}\sim\pi}} \left[Q(s_{t+1}, a_{t+1}) - \alpha \log \pi(a_{t+1}|s_{t+1}) \right]. \end{aligned} \tag{4}$$

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

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

Beyond Shannon entropy, other regularization strategies have been widely explored. Causal entropy regularization [24] emphasizes the entropy of actions conditioned strictly on past states:

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

This concept underpins Maximum Causal Entropy Inverse RL, ensuring that exploration decisions respect causality. Similarly, alternative entropy measures such as Tsallis entropy [11] have been proposed to control exploration sparsity:

$$\mathcal{H}_q(\pi(\cdot|s_t)) = \frac{1}{q-1} \left(1 - \int \pi(a_t|s_t)^q da_t \right), \tag{7}$$

where $q > 0$ is the entropic index. Another branch of regularization enforces conservative updates via KL-divergence penalties. For instance, Trust Region Policy Optimization (TRPO) [19] solves:

$$\max_\theta \quad \mathbb{E}_{\pi_\theta}[A^\pi(s_t, a_t)] \tag{8}$$

$$\text{s.t.} \quad \mathbb{E}_s \left[D_{\text{KL}}(\pi_\theta(\cdot|s_t) \| \pi_{\theta_{\text{old}}}(\cdot|s_t)) \right] \leq \delta, \tag{9}$$

where δ is a trust-region bound, an approach later approximated by Proximal Policy Optimization (PPO) [20].

Despite these various approaches, the Soft Actor-Critic algorithm [9] remains the standard for operationalizing MaxEnt RL in off-policy continuous control. SAC introduced an automatic temperature tuning mechanism, optimizing α online via a dual objective to enforce a target entropy $\overline{\mathcal{H}}$:

$$L(\alpha) = \mathbb{E}_{a_t\sim\pi_\theta} \left[-\alpha \log \pi_\theta(a_t|s_t) - \alpha\overline{\mathcal{H}} \right]. \tag{10}$$

While this automatic adjustment eliminated the need to manually tune α, it still relies on a fixed target entropy. Alternative works have proposed heuristic scheduled entropy regularization, where α decays according to an annealing schedule (e.g., $\alpha_t = \alpha_0 \cdot \exp(-kt)$), intuitively shifting the agent from exploration to exploitation. However, these methods remain independent of the critic's actual uncertainty estimates.

2.3 Distributional Soft Actor-Critic

The distributional perspective in RL extends the entropy-augmented framework by modeling the full distribution of returns rather than just their expected values. Distributional Soft Actor-Critic [4] defines the random return distribution as:

$$Z^{\pi}(s_t, a_t) \overset{D}{=} r_t + \gamma G_{t+1}, \tag{11}$$

$$G_{t+1} = \sum_{i=t+1}^{\infty} \gamma^{i-t-1}\big(r_i - \alpha \log \pi(a_i|s_i)\big). \tag{12}$$

The distributional Bellman operator with entropy exhibits a contraction property that ensures theoretical convergence. Because classical Q-learning suffers from overestimation bias due to the max operator, DSAC integrates entropy regularization with distributional critics to mitigate this bias.

Recent refinements, such as DSAC-T [5], introduced expected-value substitution for more stable targets, twin distributional critics to reduce bias, and variance-based gradient adjustments. These advances highlight the powerful interplay between distributional RL and adaptive entropy regularization.

Table 1 summarizes this evolution of entropy regularization in RL. Each methodological step has improved exploration efficiency, robustness, or stability. However, while modern distributional RL methods (like DSAC and DSAC-T) successfully model the full return distribution, they do not explicitly exploit the variance of this distribution for adaptive entropy control. Our work directly builds upon this trajectory by introducing a novel, uncertainty-aware adaptive entropy regularization scheme integrated seamlessly into the distributional Bellman framework.

3 PROPOSED METHOD

In this section, we propose an adaptive entropy regularization technique within the DSAC framework. The primary objective of this method is to dynamically maintain a balanced exploration-exploitation profile, which indirectly mitigates premature convergence to overestimated Q-values and improves sample efficiency and performance in continuous control tasks. The core innovation of our approach is the dynamic adjustment of the entropy regularization coefficient α, which explicitly governs the exploration-exploitation trade-off. Instead of relying on a fixed temperature or a static target entropy, our method dynamically adapts α based on the variance observed in the model's loss or reward distributions during training.

3.1 Entropy Regularization in DSAC

The DSAC algorithm optimizes a stochastic policy $\pi_\theta(a|s)$ and augments the standard RL objective with an entropy term to encourage exploration. The objective function for the policy (actor) with entropy regularization is given by:

$$J_\pi(\theta) = \mathbb{E}_{s\sim\mathcal{D}, a\sim\pi_\theta}[Q_\phi(s,a) - \alpha \log \pi_\theta(a|s)], \tag{13}$$

where α is the entropy regularization coefficient; $Q_\phi(s,a)$ is the Q-value function parameterized by ϕ; $\log \pi_\theta(a|s)$ is the log-probability of action a given state s under the current policy; and $\mathcal{D}$ represents the replay buffer. A larger entropy coefficient α produces broader action distributions, which averages value estimates over more actions and reduces the impact of individual overestimated Q-values. Conversely, when learning stabilizes and variance decreases, reducing α allows the policy to concentrate around high-value actions. By dynamically adjusting policy stochasticity in response to training instability, our method indirectly mitigates overestimation.

In baseline configurations of DSAC, α is either fixed or tuned toward a static target entropy, leading to a rigid exploration-exploitation profile. However, our preliminary experiments on the MuJoCo Humanoid task revealed that the variance in the model's loss and reward distributions fluctuates significantly during training, rendering a static exploration target suboptimal. To address this, we introduce a procedure to dynamically adapt α in response to these statistical fluctuations.

3.2 Adaptive Entropy Regularization

Our method adjusts the entropy regularization coefficient α based on the empirical variance of the learning process. The variance, denoted as V, is computed from the reward or loss distributions over recent training episodes. We define a target variance interval $[V_{\text{low}}, V_{\text{high}}] = [0.05, 0.5]$ and adjust α at each epoch i to maintain the variance within this bounded range. This interval was chosen empirically rather than derived theoretically; our primary goal is to test whether variance-aware entropy adaptation improves stability, rather than to derive a fully principled bound from distributional RL theory. Preliminary experiments indicated these values provide a stable exploration regime across the evaluated tasks.

Let V_i denote the measured variance at the i-th epoch. The adjustment of α follows a piecewise logic designed to correct out-of-bound variance: if the variance drops below the lower threshold ($V_i < V_{\text{low}}$), it indicates that the policy is becoming excessively deterministic, prompting an increase in α to inject stochasticity. Conversely, if the variance exceeds the upper threshold ($V_i > V_{\text{high}}$), the policy is exploring too broadly, prompting a decrease in α to encourage exploitation.

By bounding the variance within $[V_{\text{low}}, V_{\text{high}}]$, the agent dynamically modulates its behavior, maintaining an appropriate balance between exploring new states and exploiting known, high-reward trajectories.

3.3 Variants of Adaptive Entropy Regularization

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

(1) **Linear Decay:**

$$\alpha_{i+1} = \max(\alpha_i - \gamma, \alpha_{\min}), \tag{14}$$

where γ is a constant decay factor. This method reduces exploration gradually over time but operates entirely independently of the observed variance.

(2) **Exponential Decay:**

$$\alpha_{i+1} = \alpha_i \exp(-\lambda i), \tag{15}$$

where $\lambda > 0$ is the decay rate. This mechanism decays α exponentially, reducing exploration aggressively in the early stages of training.

(3) **Linear Adaptive:** This method scales α linearly with respect to the variance error. If the variance falls outside the target

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

Method	Improvement	Limitations
Policy Gradient with Entropy Bonus [14]	Reduces premature convergence	Weak impact in high-dimensional tasks
Causal Entropy Regularization [24]	Handles sequential/causal decision making	Limited application, less explored empirically
Tsallis Entropy Regularization [11]	Flexible exploration, robustness	Parameter q difficult to tune
KL-Divergence Regularization [19, 20]	Smooth updates, stability in trust-region methods	Extra computation, requires reference policy
Maximum Entropy RL [8]	Theoretical foundation, encourages exploration	May cause overly stochastic policies
SAC Entropy Regularization [9]	Stable training, adaptive exploration	Sensitive to target α tuning

interval, α is adjusted linearly using a scaling factor $\beta > 0$:

$$\alpha_{i+1} = \begin{cases} \alpha_i + \beta(V_{\text{low}} - V_i) & \text{if } V_i < V_{\text{low}}, \\ \alpha_i - \beta(V_i - V_{\text{high}}) & \text{if } V_i > V_{\text{high}}. \end{cases} \tag{16}$$

(4) **Exponential Adaptive:** In this variant, α is adjusted aggressively via an exponential function of the variance error, allowing the coefficient to react rapidly to sudden, large deviations in variance:

$$\alpha_{i+1} = \begin{cases} \alpha_i \exp\big(\beta(V_{\text{low}} - V_i)\big) & \text{if } V_i < V_{\text{low}}, \\ \alpha_i \exp\big(-\beta(V_i - V_{\text{high}})\big) & \text{if } V_i > V_{\text{high}}. \end{cases} \tag{17}$$

To ensure a baseline level of exploration is maintained across all variants, α is strictly bounded from below by a minimum value, $\alpha_{\min}$. These four variants represent the specific ablations evaluated in our empirical study. Among these, the Exponential Adaptive (Variant 4) is our primary proposed method, as it demonstrated the most robust empirical performance and is the variant formally presented in Algorithm 1. While this approach introduces new hyperparameters, their roles are relatively simple and consistent across tasks: V_{low} and V_{high} act as bounds defining acceptable training stability; $\alpha_{\min}$ prevents complete entropy collapse; and β controls the adaptation rate.

3.4 Real-time Variance Calculation

To operationalize these adjustments dynamically, we calculate the empirical variance of the targeted metric (loss or reward) in real-time. At each epoch, following the policy update, the variance V_i is computed over n recent episodes or batch samples:

$$V_i = \frac{1}{n} \sum_{j=1}^{n} (X_j - \overline{X})^2, \tag{18}$$

where X_j represents the specific observation (reward or loss) at step j, and $\overline{X}$ is the empirical mean of these observations. It is important to clarify that V_t represents the empirical variance of recent training signals, rather than the variance of the predicted return distribution $Z^\pi(s_t, a_t)$ directly computed by the distributional critic. We utilize these empirical signals because they serve as a reliable, practical proxy for training instability, which strongly correlates with the agent's uncertainty during the learning process.

Our complete procedure is summarized in Algorithm 1. By enforcing variance constraints, this framework robustly balances exploration and exploitation. Subsequent experiments on the MuJoCo Humanoid task verify that the exponentially adaptive variant outperforms both baseline DSAC and other scheduling mechanisms, facilitating highly efficient learning and superior continuous control.

Algorithm 1 Adaptive Entropy Regularization for DSAC

Require: Initial entropy coefficient α_0, variance thresholds V_{high} and V_{low}, adaptation rate β, minimum $\alpha_{\min}$

```
Initialize α ← α_0
while training do
    for each episode do
        Compute variance V_i from the reward or loss distribution
        if V_i > V_high then
            α ← α exp(−β(V_t − V_high))          ▷ Decrease exploration when variance is too high
        else if V_i < V_low then
            α ← α exp(β(V_low − V_t))            ▷ Increase exploration when variance is too low
        end if
        α ← max(α, α_min)                        ▷ Ensure α does not fall below a minimum value
        Update the actor policy π_θ using the adjusted entropy coefficient α
    end for
end while
```

4 EMPIRICAL EVALUATION

We compare our proposed algorithms against the DSAC [4], a method that has been extensively verified across a variety of challenging continuous control tasks. By employing this algorithm as our baseline, the performance of the proposed extensions can be evaluated objectively. Specifically, this paper improves overestimation mitigation by clipping the variance into a smaller, valid range.

All off-policy algorithms evaluated in this work are implemented in PyTorch, adopting almost identical neural network architectures and hyperparameters to ensure fair comparison (detailed in Table 2). Our experiments demonstrate that these modifications yield subtle but important reductions in the overestimation bias originating in the Q-learning setting.

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
Minimum entropy coefficient (α_{min})	0.0001
Adaptation rate(β)	0.01
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	$b = 10$

4.1 Evaluation Environments

To assess the proposed Adaptive DSAC algorithm, we utilize standard continuous control tasks from the MuJoCo physics simulator. While the broader MuJoCo suite includes diverse environments (such as `HalfCheetah`, `Hopper`, `Walker2d`, and `Ant`), we deliberately focus our empirical analysis on two structurally distinct extremes to isolate the effects of adaptive entropy and variance bounding: the highly complex, high-dimensional `Humanoid-v2` and the lower-dimensional, rhythmic `Swimmer-v2`.

Following established protocols [6, 9], agents are trained for 1.5×10^6 timesteps using identical configurations to enable direct, fair performance comparisons with the baseline variants. We report the average episodic return over four independent runs with different random seeds, with results smoothed using a moving average window of size 10 to clearly visualize performance trends and standard deviations.

4.1.1 MuJoCo Humanoid Benchmark. The `Humanoid-v2` environment simulates a 3D bipedal robot with articulated limbs and a torso. The objective is to maintain balance and move forward as quickly as possible without falling. Featuring 17 actuated joints and complex multi-segment dynamics, it represents one of the most challenging tasks in continuous control due to its high dimensionality and extreme sensitivity to control inputs. It requires highly precise control to achieve stable locomotion and resilience against perturbations.

Observation Space. The observation (state) space is a 376-dimensional continuous vector, encompassing joint positions (66 dims), joint velocities (66 dims), inertial measurements (17 dims), and external contact forces (260 dims). A summary is provided in Table 3.

Action Space. The action space is a 17-dimensional continuous vector $a \in [-1, 1]^{17}$, corresponding to the torque applied at the motor joints (detailed in Table 4).

Table 3: `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**

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**

Reward Structure and Termination. The reward function encourages forward progress (proportional to the center of mass velocity) while strictly penalizing control effort and unnatural contact forces. Episodes terminate when the torso's height drops below a critical threshold or upon reaching the 1000-step limit. This high-variance environment serves as a rigorous stress test for the exploration-exploitation balance and the algorithm's resilience to Q-value overestimation.

4.1.2 MuJoCo Swimmer Benchmark. Conversely, `Swimmer-v2` provides a low-dimensional testbed commonly used to validate algorithmic stability and sample efficiency. The environment simulates a simplified 2D three-link snake-like agent in a viscous fluid, where forward locomotion is generated via coordinated, rhythmic joint torques.

Observation Space. The 8-dimensional continuous observation space includes joint angles (2 dims), joint angular velocities (2 dims), torso orientation (2 dims), and linear velocities of the torso (2 dims), summarized in Table 5.

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**

Action Space. The action space is a 2-dimensional continuous vector $a \in [-1, 1]^2$, dictating the torque applied to the two actuated joints (Table 6).

Reward Structure and Termination. The reward primarily encourages forward velocity in the x-direction, penalized by the squared magnitude of the control input to discourage inefficient motion.

Table 6: `Swimmer-v2` Action Space.

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

Episodes terminate exclusively at the maximum horizon of 1000 steps, with no failure conditions for falling.

For the empirical evaluations in Section 4.2, the adaptive variants computed the variance V_t based on empirical training signals. Specifically, the high-dimensional Humanoid-v2 task utilized the variance of recent critic loss values, while the Swimmer-v2 task utilized the variance of episodic rewards. Importantly, to evaluate robustness, the exact same variance thresholds and hyperparameters were utilized across both structurally distinct environments without any task-specific tuning. To ensure a fair comparison, the DSAC-T baseline utilizes a fixed entropy coefficient, following the configuration and recommended values reported in the original DSAC-T reference implementation.

4.2 Results and Analysis

To evaluate our proposed method, we benchmark 5 distinct algorithmic variations: the DSAC-T baseline, exponential adaptive, exponential decay, linear adaptive, and linear decay. Each model configuration is executed across 4 independent random seeds and trained for a total of 1.5×10^6 environmental interactions.

Figure 1 illustrates the critic loss function during the initial crucial learning phase (the first 100,000 steps) on the high-dimensional `Humanoid-v2` environment. As observed, the loss functions for all variants exhibit a strict, rapid downward trend, confirming that the value function approximation remains highly stable and converges reliably across both the baseline and our proposed adaptive mechanisms.

The significant algorithmic improvements are most apparent when examining the expected accumulated return. As depicted in Figure 2, the learning curves on the `Humanoid-v2` task show that the adaptive exploration methods (specifically the exponential adaptive and linear adaptive variants) achieve a noticeably higher asymptotic performance compared to both the decay-based methods and the DSAC-T baseline. The shaded regions, denoting the standard deviation across seeds, highlight that while the task inherently possesses high variance due to its complex dimensionality, our adaptive methods reliably navigate the exploration-exploitation dilemma to yield superior locomotion policies.

Furthermore, we evaluate the algorithmic robustness on the `Swimmer-v2` task (Figure 3), which requires rhythmic rather than purely reactive control. In this setting, the performance gap is even more pronounced. The exponential adaptive method consistently dominates the learning process, reaching the highest forward progress with expected returns near 150. In stark contrast, the DSAC-T baseline struggles significantly, remaining at the bottom of the performance tier with returns hovering near 75. This demonstrates that our proposed modifications require less control effort and find optimal policies much faster, even in lower-dimensional control settings.

These empirical results validate the core theoretical contribution of this work: an adaptive entropy regularization mechanism uniquely enabled by the distributional critic in DSAC. Unlike standard SAC's automatic entropy tuning, which enforces a rigidly fixed target entropy, our approach dynamically adjusts exploration based directly on the aleatoric uncertainty and variance captured in the value predictions.

When compared to DSAC-T, which relies heavily on temperature tuning primarily to manage overestimation, our exponential and linear adaptive methods establish an explicit, mathematical coupling between the distributional variance of the return and the entropy strength. Because the algorithm transitions smoothly and naturally from aggressive exploration to fine-tuned exploitation precisely as the return distribution sharpens, it consistently reaches higher performance asymptotes (as seen in the Humanoid task) and avoids suboptimal local minima (as seen in the Swimmer task). This integration represents a substantive algorithmic extension to distributional RL frameworks, moving beyond simple parameter-scheduling heuristics to achieve genuinely adaptive, uncertainty-aware continuous control.

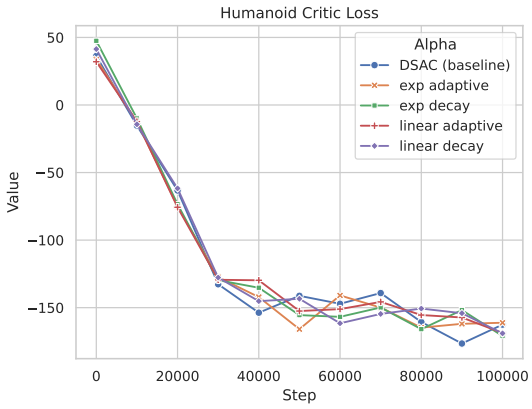

Figure 1: `Humanoid-v2` critic loss during the initial 100,000 steps of the training process. All methods show stable convergence.

5 CONCLUSION

In this paper, we proposed an empirical variance-adaptive entropy regularization framework integrated within the Distributional Soft Actor-Critic (DSAC) architecture. Unlike standard maximum entropy algorithms that rely on static target entropy or heuristic decay schedules, our approach explicitly couples the temperature parameter (α) with the empirical variance of recent training signals, utilizing them as a practical proxy for training instability. By dynamically constraining this variance within a designated bound, the agent gracefully transitions from broad exploration, i.e. when uncertainty is high, to stable exploitation as the return distribution sharpens. Ultimately, this feedback mechanism directly mitigates Q-value overestimation bias and prevents premature convergence. Empirical evaluations on MuJoCo continuous control benchmarks of the high-dimensional `Humanoid-v2` and the rhythm-dependent

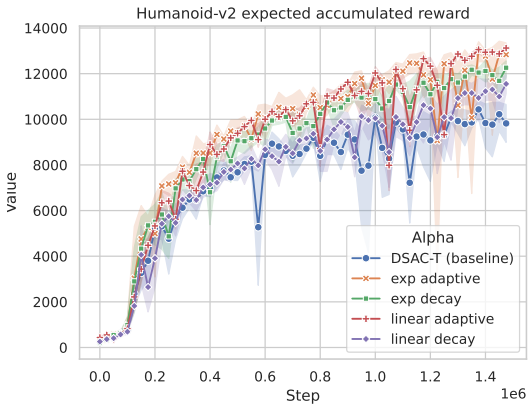

Figure 2: `Humanoid-v2` expected accumulated return over 1.5×10^6 steps. Shaded regions represent the standard deviation across 4 independent seeds.

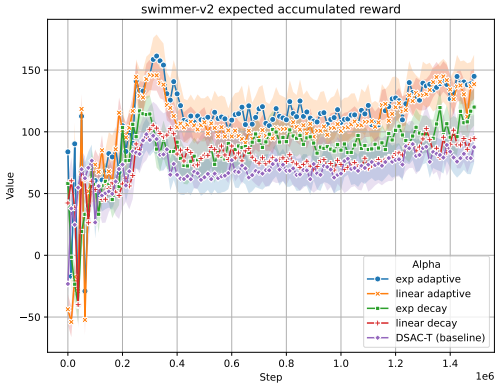

Figure 3: `Swimmer-v2` expected accumulated return over 1.5×10^6 steps.

`Swimmer-v2` environments confirm improved stability and better generalization compared to existing SAC and DSAC variants. While the target variance interval $[V_{\text{low}}, V_{\text{high}}]$ was chosen empirically based on observations of early training stability, it successfully establishes a closed-loop feedback mechanism without requiring per-task hyperparameter tuning. Future work will focus on systematically ablating these variance thresholds and investigating methods to derive them mathematically from the distributional framework itself. Additionally, future research could apply this adaptive framework to discrete action spaces or deploying it in real-world robotic locomotion tasks for validating its broader generalizability.

REFERENCES

[1] Marc G Bellemare, Will Dabney, and Rémi Munos. 2017. A distributional perspective on reinforcement learning. In *International conference on machine learning*. PMLR, 449–458.

[2] Marc G. Bellemare, Will Dabney, and Mark Rowland. 2023. *Distributional Reinforcement Learning*. MIT Press. http://www.distributional-rl.org.

[3] Thomas Degris, Patrick M Pilarski, and Richard S Sutton. 2012. Model-free reinforcement learning with continuous action in practice. In *2012 American control conference (ACC)*. IEEE, 2177–2182.

[4] Jingliang Duan, Yang Guan, Shengbo Eben Li, Yangang Ren, Qi Sun, and Bo Cheng. 2022. Distributional Soft Actor-Critic: Off-Policy Reinforcement Learning for Addressing Value Estimation Errors. *IEEE Transactions on Neural Networks and Learning Systems* 33, 11 (2022), 6584–6598.

[5] Jingliang Duan, Wenxuan Wang, Liming Xiao, Jiaxin Gao, and Shengbo Eben Li. 2023. DSAC-T: Distributional soft actor-critic with three refinements. *arXiv preprint arXiv:2310.05858* (2023).

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

[7] Shixiang Gu, Ethan Holly, Timothy P Lillicrap, and Sergey Levine. 2016. Deep reinforcement learning for robotic manipulation. *arXiv preprint arXiv:1610.00633* 1, 1 (2016).

[8] Tuomas Haarnoja, Haoran Tang, Pieter Abbeel, and Sergey Levine. 2017. Reinforcement learning with deep energy-based policies. In *International conference on machine learning*. PMLR, 1352–1361.

[9] Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. 2018. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. (2018), 1861–1870.

[10] Vijay Konda and John Tsitsiklis. 1999. Actor-critic algorithms. *Advances in neural information processing systems* 12 (1999).

[11] Kyungjae Lee, Sungyub Kim, Sungbin Lim, Sungjoon Choi, and Songhwai Oh. 2019. Tsallis reinforcement learning: A unified framework for maximum entropy reinforcement learning. *arXiv preprint arXiv:1902.00137* (2019).

[12] Sergey Levine, Chelsea Finn, Trevor Darrell, and Pieter Abbeel. 2016. End-to-end training of deep visuomotor policies. *Journal of Machine Learning Research* 17, 39 (2016), 1–40.

[13] Timothy P Lillicrap, Jonathan J Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. 2015. Continuous control with deep reinforcement learning. *arXiv preprint arXiv:1509.02971* (2015).

[14] Jingbin Liu, Xinyang Gu, and Shuai Liu. 2019. Policy optimization reinforcement learning with entropy regularization. *arXiv preprint arXiv:1912.01557* (2019).

[15] Fan-Ming Luo, Tian Xu, Hang Lai, Xiong-Hui Chen, Weinan Zhang, and Yang Yu. 2024. A survey on model-based reinforcement learning. *Science China Information Sciences* 67, 2 (2024), 121101.

[16] Xiaoteng Ma, Li Xia, Zhengyuan Zhou, Jun Yang, and Qianchuan Zhao. 2020. DSAC: Distributional Soft Actor Critic for Risk-Sensitive Reinforcement Learning. *arXiv preprint arXiv:2004.14547* (2020).

[17] Naemeh Mohammadpour, Meysam Fozi, Mohammad Mehdi Ebadzadeh, Ali Azimi, and Ali Kamalie Iglie. 2024. Proximal Policy Optimization with Adaptive Generalized Advantage Estimate. In *Proceedings of the First International Conference on Machine Learning and Knowledge Discovery (MLKD 2024)*. 453–458. https://mlkd.aut.ac.ir/proceedings/2024/paper/4B.7.pdf

[18] Naemeh Mohammadpour, Meysam Fozi, Mohammad Mehdi Ebadzadeh, Ali Azimi, and Ali Kamalie Iglie. 2025. Proximal policy optimization with adaptive generalized advantage estimate: critic-aware refinements. *Journal of Mathematical Modeling* (2025).

[19] John Schulman, Sergey Levine, Pieter Abbeel, Michael Jordan, and Philipp Moritz. 2015. Trust region policy optimization. In *International conference on machine learning*. PMLR, 1889–1897.

[20] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347* (2017).

[21] David Silver, Aja Huang, Chris J Maddison, Arthur Guez, Laurent Sifre, George Van Den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, et al. 2016. Mastering the game of Go with deep neural networks and tree search. *nature* 529, 7587 (2016), 484–489.

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

[23] Ronald J Williams. 1992. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning* 8, 3 (1992), 229–256.

[24] Brian D Ziebart, J Andrew Bagnell, and Anind K Dey. 2010. Modeling interaction via the principle of maximum causal entropy. (2010).

[25] Brian D Ziebart, Andrew L Maas, J Andrew Bagnell, Anind K Dey, et al. 2008. Maximum entropy inverse reinforcement learning.. In *Aaai*, Vol. 8. Chicago, IL, USA, 1433–1438.