Preprint – IJCAI 2025: This is the accepted version made available for conference attendees.
Do not cite. The final version will appear in the IJCAI 2025 proceedings.

Maximum Entropy Softmax Policy Gradient via Entropy Advantage Estimation

Jean Seong Bjorn Choe and **Jong-kook Kim**
Korea University
{garangg, jongkook}@korea.ac.kr

Abstract

Entropy Regularisation is a widely adopted technique that enhances policy optimisation performance and stability. Maximum entropy reinforcement learning (MaxEnt RL) regularises policy evaluation by augmenting the objective with an entropy term, showing theoretical benefits in policy optimisation. However, its practical application in straightforward direct policy gradient settings remains surprisingly underexplored. We hypothesise that this is due to the difficulty of managing the entropy reward in practice. This paper proposes Entropy Advantage Policy Optimisation (EAPO), a simple method that facilitates MaxEnt RL implementation by separately estimating task and entropy objectives. Our empirical evaluations demonstrate that extending Proximal Policy Optimisation (PPO) and Trust Region Policy Optimisation (TRPO) within the MaxEnt framework improves optimisation performance, generalisation, and exploration in various environments. Moreover, our method provides a stable and performant MaxEnt RL algorithm for discrete action spaces.

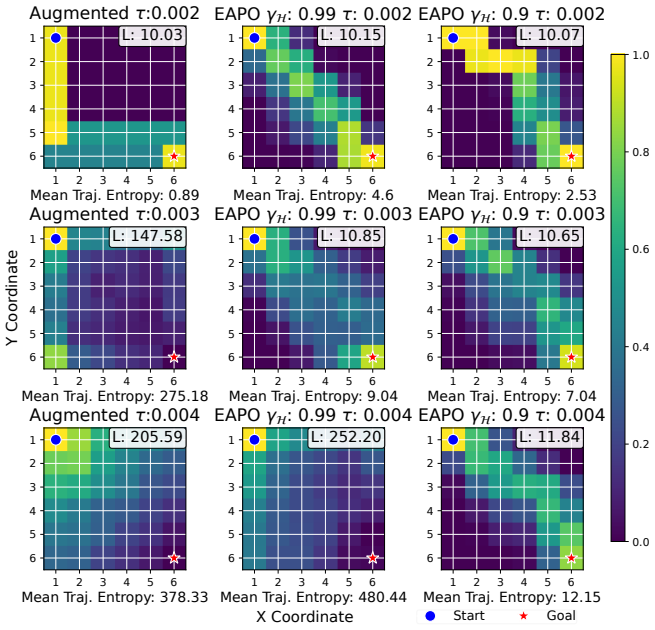

Figure 1: Visitation frequencies of policies trained on the modified `MiniGrid-Empty-8x8` task using naive MaxEnt (PPO with the augmented entropy reward) and EAPO with discount factors $\gamma_{\mathcal{H}} \in (0.9, 0.99)$ and TD(0) for entropy estimation. We compare 3 temperatures $\tau \in (0.002, 0.003, 0.004)$, using $\gamma_V = 0.99$. L shows the mean length of trajectories, where agents minimise toward the optimal value of 10. See the appendix[1] for details.

1 Introduction

Entropy regularisation is pivotal to many practical deep reinforcement learning (RL) algorithms. Practical algorithms such as Trust Region Policy Optimization (TRPO) [Schulman *et al.*, 2015a] penalise the policy improvement or greedy step using Kullback-Leibler (KL) divergence (also called as relative entropy) to regularise the deviations between consecutive policies. This method, often termed KL regularisation, has been the foundational approach for contemporary deep RL algorithms [Vieillard *et al.*, 2020; Geist *et al.*, 2019].

Another critical approach, Maximum Entropy RL (MaxEnt RL) [Ziebart, 2010; Haarnoja *et al.*, 2018; Levine, 2018; Marino *et al.*, 2021; Cetin and Celiktutan, 2022], augments the conventional RL task objective with an entropy term to directing policies toward areas of higher expected trajectory entropy. It is known to improve the exploration and robustness of policies by promoting stochasticity [Eysenbach and Levine, 2019; Eysenbach and Levine, 2021]. In practice, this can be implemented by adding an entropy reward to the original task reward.

Recent theoretical advancements have shown the effectiveness of MaxEnt RL in accelerating the convergence of policy gradient (PG) methods [Mei *et al.*, 2020; Agarwal *et al.*, 2021; Cen *et al.*, 2022]. However, despite the enticing theoretical support, a significant gap exists between theory and practice. While off-policy methods like Soft Actor-Critic [Haarnoja *et al.*, 2018] have demonstrated practical success with MaxEnt RL, its application in simpler direct Softmax policy gradient settings remains surprisingly underexplored.

[1]Technical appendix and codes available at: https://github.com/rnilva/eapo-ijcai25

Preprint – IJCAI 2025: This is the accepted version made available for conference attendees.
Do not cite. The final version will appear in the IJCAI 2025 proceedings.

We hypothesise that this research gap is potentially attributed to the difficulty of handling the entropy reward in practice. [Yu *et al.*, 2022] empirically analysed the problematic nature of the entropy reward using SAC. Authors pointed out that in an episodic setting, the entropy return is largely correlated to the episode's length, thereby rendering the policy overly optimistic or pessimistic, and even in infinite-horizon settings, the entropy reward can still obscure the task reward. This raises a fundamental question: can MaxEnt RL provide practical benefits in straightforward stochastic Softmax policy gradient settings?

Inspired by this observation, we proposed a simple but practical approach to control the impact of the entropy reward. In this paper, we introduce Entropy Advantage Policy Optimisation (EAPO), a method that estimates the task and entropy objectives of the regularised (soft) objective separately. By employing a dedicated discount factor for the entropy reward and utilising Generalised Advantage Estimation (GAE) [Schulman *et al.*, 2015b] on each objective separately, EAPO controls the effective horizon of the entropy return estimation and the entropy regularisation on policy evaluation. EAPO's simplicity requires only minor modifications to existing advantage actor-critic algorithms, providing a clear demonstration of impact of MaxEnt RL framework in basic settings. Our empirical evaluation not only extend the well-established PPO [Schulman *et al.*, 2017b] and TRPO [Schulman *et al.*, 2015a], but also demonstrate superior stability in discrete action spaces compared to TD-SAC [Zhou *et al.*, 2024], a discrete version of SAC with improved stability.

Figure 1 illustrates the challenge of learning the MaxEnt policy for an episodic task using a naive implementation that simply augments the task reward with an entropy reward. In this task, the agent is required to reach the goal state while performing the minimum number of actions. The naive MaxEnt agent fails to learn the optimal stochastic policy, resulting in two failure modes: acting almost deterministically when the temperature τ is low or wandering around indefinitely when τ is high. In contrast, EAPO successfully achieves the near-optimal stochastic policy by utilising TD(0) learning [Sutton and Barto, 2018] (i.e., set GAE λ to 0) for the entropy objective. Additionally, the example demonstrates that lowering the discount factor for the entropy estimation $\gamma_{\mathcal{H}}$ helps prevent the inflation of the entropy reward [Yu *et al.*, 2022] and reduces sensitivity to the temperature.

In this work, we primarily focus on empirical examinations of the Softmax MaxEnt Policy Gradient method [Levine, 2018; Mei *et al.*, 2020] across diverse environments: 4 discretised [Tang and Agrawal, 2020] MuJoCo continuous control tasks, 16 Procgen episodic environments [Cobbe *et al.*, 2020], and the MiniGrid DoorKey environment [Chevalier-Boisvert *et al.*, 2023]. Our results demonstrate the efficacy of MaxEnt RL policies in improving optimisation performance, generalization, and potential exploration benefits.

2 Background

2.1 Preliminaries

This work considers a finite undiscounted Markov Decision Process (MDP) $\langle \mathcal{S}, \mathcal{A}, r, \rho, \mathcal{T} \rangle$, where $\mathcal{S}$ is the set of states s and $\mathcal{A}$ is the set of actions a, and ρ is the initial state distribution. $\mathcal{T}$ is the transition function $\mathcal{T} : \mathcal{S} \times \mathcal{A} \mapsto \Delta(\mathcal{S})$, where $\Delta(\mathcal{S})$ is the probability simplex over $\mathcal{S}$, and r is the reward function $r : \mathcal{S} \times \mathcal{A} \mapsto \mathbb{R}$. We introduce discount factors γ_V and $\gamma_{\mathcal{H}}$ as variance reduction parameters as in [Schulman *et al.*, 2015b]. We define the value function of state s under the policy π as $V^\pi(s) \coloneqq \mathbb{E}_{s_0=s, a_t\sim\pi(\cdot|s_t), s_{t+1}\sim\mathcal{T}(s_t,a_t)}\left[\sum_{t=0}^{\infty} \gamma_V^t r(s_t, a_t)\right]$. Also, the action-value of performing action a at state s under the policy π is $Q^\pi(s,a) \coloneqq \mathbb{E}_{s_0=s, a_0=a, a_{t>0}\sim\pi(\cdot|s_t), s_{t+1}\sim\mathcal{T}(s_t,a_t)}\left[\sum_{t=0}^{\infty} \gamma_V^t r(s_t, a_t)\right]$. And define the advantage function A^π as $A^\pi(s,a) \coloneqq Q^\pi(s,a) - \mathbb{E}_{\pi(\cdot|s)}\left[Q^\pi(s,\cdot)\right] = Q^\pi(s,a) - V^\pi(s)$. We also define the cumulative discounted entropy return of state s under policy π as

$$V_{\mathcal{H}}^\pi(s) \coloneqq \mathop{\mathbb{E}}_{\substack{s_0=s, a_t\sim\pi(\cdot|s_t),\\ s_{t+1}\sim\mathcal{T}(s_t,a_t)}} \left[\sum_{t=0}^{\infty} -\gamma_{\mathcal{H}}^t \log \pi(a_t|s_t)\right]. \quad (1)$$

In deterministic MDPs, the cumulative discounted entropy return $V_{\mathcal{H}}^\pi(s)$ represents the Shannon entropy of the possible future trajectories' distribution [Levine, 2018; Tiapkin *et al.*, 2023], and we refer to it as trajectory entropy throughout this work for brevity.

The objective of Maximum Entropy Reinforcement Learning (MaxEnt RL), or often Regularised MDPs [Geist *et al.*, 2019; Neu *et al.*, 2017] is to maximise the expectation of the sum of the value and the trajectory entropy with respect to the initial state distribution:

$$J(\pi) = \mathop{\mathbb{E}}_{\substack{s_0\sim\rho\\ a_t\sim\pi,\\ s_{t+1}\sim\mathcal{T}}} \left[\sum_{t=0}^{\infty} \gamma_V^t r(s_t, a_t) - \gamma_{\mathcal{H}}^t \tau \log \pi(a_t|s_t)\right] \quad (2)$$

$$= \mathop{\mathbb{E}}_{s_0\sim\rho} \left[V^\pi(s_0) + \tau V_{\mathcal{H}}^\pi(s_0)\right], \quad (3)$$

where the temperature parameter $\tau \geq 0$ is a hyperparameter to be controlled to balance the significance between these two objectives, and we introduce the distinct discount factors. We ensure the objective $J(\pi)$ is finite for all policies by assuming either an episodic MDP or an MDP with an absorbing state that yields zero reward.

2.2 Soft Advantage Function

Analogous to the definition of the action-value function Q^π as the expected cumulative rewards after selecting an action a [Sutton and Barto, 2018], we define $Q_{\mathcal{H}}^\pi$ as the expected future trajectory entropy after selecting an action:

$$Q_{\mathcal{H}}^\pi(s_t, a_t) \coloneqq \mathop{\mathbb{E}}_{s_{t+1}\sim\mathcal{T}(s_t,a_t)} \left[\gamma_{\mathcal{H}} V_{\mathcal{H}}^\pi(s_{t+1})\right]. \quad (4)$$

The definition arises naturally from the consideration that uncertainty exists due to the stochastic policy at the current state, which has settled by the time an action is performed. Consequently, the $Q_{\mathcal{H}}^\pi$ is simply the expected discounted future trajectory entropy.

From the recursive relation of trajectory entropy from (1) and the defintion (4), the following relation is derived:

$$V_{\mathcal{H}}^\pi(s_t) = \mathop{\mathbb{E}}_{a_t\sim\pi(\cdot|s_t)} \left[-\log \pi(a_t|s_t) + Q_{\mathcal{H}}^\pi(s_t, a_t)\right]. \quad (5)$$

Preprint – IJCAI 2025: This is the accepted version made available for conference attendees.
Do not cite. The final version will appear in the IJCAI 2025 proceedings.

We now define the entropy advantage function $A^\pi_{\mathcal{H}}$ analogous to the conventional advantage function:

$$A^\pi_{\mathcal{H}}(s_t, a_t) \coloneqq Q^\pi_{\mathcal{H}}(s_t, a_t) - \mathop{\mathbb{E}}_{a\sim\pi}[Q^\pi_{\mathcal{H}}(s_t, a)]$$
$$= Q^\pi_{\mathcal{H}}(s_t, a_t) - V^\pi_{\mathcal{H}}(s_t) + \mathop{\mathbb{E}}_{a\sim\pi}[-\log\pi(a|s_t)]\,. \quad (6)$$

We let $\tilde{V}^\pi(s) \coloneqq V^\pi(s) + \tau V^\pi_{\mathcal{H}}(s)$ as the soft value function, and let $\tilde{Q}^\pi(s,a) \coloneqq Q^\pi(s) + \tau Q^\pi_{\mathcal{H}}(s,a)$ as the soft Q-function. Finally, we define the soft advantage function:

$$\tilde{A}^\pi(s_t, a_t) \coloneqq A^\pi(s_t, a_t) + \tau A^\pi_{\mathcal{H}}(s_t, a_t) \quad (7)$$
$$= Q^\pi(s_t, a_t) - V^\pi(s_t)$$
$$+ \tau(Q^\pi_{\mathcal{H}}(s_t, a_t) - V^\pi_{\mathcal{H}}(s_t) + \mathop{\mathbb{E}}_{a\sim\pi}[-\log\pi(a|s_t)])$$
$$= \tilde{Q}^\pi(s_t, a_t) - \tilde{V}^\pi(s_t) + \tau \mathop{\mathbb{E}}_{a\sim\pi}[-\log\pi(a|s_t)]\,. \quad (8)$$

2.3 Soft Policy Gradient Theorem

[Shi *et al.*, 2019] showed that it is possible to optimise the soft objective using direct policy gradient from samples. Thus, we can use the soft advantage function to find the policy that maximises the MaxEnt RL objective.

Theorem 1 (Soft Policy Gradient)**.** *Let $J(\pi)$ the MaxEnt RL objective defined in 2. And $\pi_\theta(a|s)$ be a parameterised policy. Then,*

$$\tilde{A}^\pi_\gamma(s_t, a_t) \coloneqq \gamma^t_V A(s_t, a_t) + \gamma^t_{\mathcal{H}} \tau A^\pi_{\mathcal{H}}(s_t, a_t),$$
$$\nabla_\theta J(\pi_\theta) = \mathop{\mathbb{E}}_{\substack{s_0\sim\rho,\\ a_t\sim\pi,\\ s_{t+1}\sim\mathcal{T}}}\left[\tilde{A}^\pi_\gamma(s_t, a_t)\nabla_\theta \log\pi_\theta(a_t|s_t)\right]. \quad (9)$$

We provide the proof in the appendix. While the exact soft policy gradient theorem requires the corresponding exponential discount term for each advantage estimate, we use the approximate policy gradient in this work by replacing $\tilde{A}^\pi_\gamma(s_t, a_t)$ with $\tilde{A}^\pi(s_t, a_t)$. It is worth noting that when the exact gradient is known, [Mei *et al.*, 2020] proved that the soft policy gradient has the global convergence property and may converge faster than the policy gradient without entropy regularisation despite the objective being biased. However, in our practical setup, this is not guaranteed.

3 Related Works

3.1 MaxEnt RL in Discrete Action Spaces

Off-policy methods like Soft Q-Learning and SAC [Haarnoja *et al.*, 2017; Schulman *et al.*, 2017a] learn soft Q-functions for discrete action spaces but often suffer from instability due to the coupled challenges of temperature adjustment, off-policy soft Q-function estimation, and policy updates [Christodoulou, 2019; Zhou *et al.*, 2024; Xu *et al.*, 2021]. In contrast, EAPO utilises direct policy gradient and approximates the soft value function $\tilde{V}^\pi(s)$ using GAE from on-policy samples, making it less prone to the estimation error at the cost of sample efficiency. This work also demonstrates the better policy optimisation stability and performance of EAPO over SD-SAC [Zhou *et al.*, 2024] across procgen environments.

3.2 Soft Policy Gradient Method

A more directly related work is [Shi *et al.*, 2019], which explored a soft policy gradient method emphasising its inherent simplicity and proved the soft policy gradient theorem. Their method also involves estimating $\tilde{Q}^\pi$ from off-policy samples and introduces additional techniques for mitigating the estimation issues. On the other hand, EAPO uses the soft advantage function $\tilde{A}^\pi$ for policy gradient estimator with additional hyperparameters to reduce variance and seamlessly integrates with existing techniques, such as value function normalisation, due to its structural equivalence between its method for estimating the entropy advantage function and the conventional advantage function.

3.3 Reward Inflation Problem

[Yu *et al.*, 2022] showed that the entropy reward of MaxEnt RL can cause reward inflation from indefinite exploration in episodic settings as it is given at every time step before termination. We show that having a lower discount factor for entropy objective can mitigate the problem and establish positive implications for the use of entropy rewards in policy optimisation.

3.4 Entropy Cost vs. MaxEnt RL

A more common approach to applying entropy regularisation to PG methods is to add an entropy cost term to the sample-based policy gradient estimator to maximise the policy entropy at each sampled state, retaining the stochasticity of the policy during optimisation process [Mnih *et al.*, 2016; Schulman *et al.*, 2017b]. While this entropy bonus term remains a heuristic approach despite its practical success [Ahmed *et al.*, 2019], MaxEnt RL provides a theoretically grounded framework that directs a policy toward regions of higher expected trajectory entropy, albeit at the cost of bias imposed on the objective [Levine, 2018; Schulman *et al.*, 2017a]. This work demonstrates the performance improvements of the MaxEnt approach of EAPO over PPO's entropy bonus term.

4 Proposed Method

4.1 Overview

In this section, we develop our Entropy Advantage Policy Optimisation (EAPO) method. At its core, EAPO independently estimates both the value advantage function and the entropy advantage function and combines them to derive the soft advantage function. EAPO adopts a separate prediction head to the conventional value critic to approximate the trajectory entropy of a state, which is then used for entropy advantage estimation. We extend the PPO [Schulman *et al.*, 2017b] and TRPO [Schulman *et al.*, 2015a] by substituting the advantage estimate with the soft advantage estimate and omitting the entropy bonus term.

4.2 Entropy Advantage Estimation

The entropy advantage $A^\pi_{\mathcal{H}}$ is estimated from the sampled log probabilities of the behaviour policy. We utilise the Generalised Advantage Estimation (GAE) [Schulman *et al.*, 2015b]

Preprint – IJCAI 2025: This is the accepted version made available for conference attendees.
Do not cite. The final version will appear in the IJCAI 2025 proceedings.

for a variance-reduced estimation of the entropy advantage:

$$\hat{A}^{\mathcal{H},\mathrm{GAE}(\lambda_{\mathcal{H}},\gamma_{\mathcal{H}})}(s_t,a_t) \coloneqq \sum_{l=0}^{\infty}(\lambda_{\mathcal{H}}\gamma_{\mathcal{H}})^l \delta_{t+l}^{\mathcal{H}}, \quad (10)$$

where $\delta_t^{\mathcal{H}} \coloneqq -\log\pi(a_t|s_t)+\gamma_{\mathcal{H}}V_{\mathcal{H}}^{\pi}(s_{t+1})-V_{\mathcal{H}}^{\pi}(s_t)$, and $\gamma_{\mathcal{H}}$ and $\lambda_{\mathcal{H}}$ are the discount factor and GAE lambda for entropy advantage estimation, respectively. Note that the equation is the same as the GAE for the conventional advantage, except the reward term is replaced by the negative log probability. This simplicity is also consistent with the remark that the only modification required for the MaxEnt policy gradient is to add the negative log probability term to the reward at each time step [Levine, 2018].

4.3 Entropy Critic

An entropy critic network, parameterised by ω, approximates the trajectory entropy $V_{\mathcal{H}}^{\pi}(\cdot;\omega)$. To train the network, we construct the TD($\lambda_{\mathcal{H}}$) bootstrapped target $\hat{V}_{\mathcal{H}}^{\pi}(s_t;\omega)$ from sampled trajectories. We estimate the entropy advantages using the entropy critic, and the target is derived as $\hat{V}_{\mathcal{H}}^{\pi}(s_t;\omega) = \hat{A}^{\mathcal{H},\mathrm{GAE}(\lambda_{\mathcal{H}},\gamma_{\mathcal{H}})}(s_t,a_t;\omega) + V_{\mathcal{H}}^{\pi}(s_t;\omega)$. Then the entropy critic is trained by minimising the mean squared error using the semi-gradient method: $L^{\mathcal{H}}(\omega) \coloneqq \hat{\mathbb{E}}_t\left[\frac{1}{2}\left(V_{\mathcal{H}}^{\pi}(s_t;\omega) - \hat{V}_{\mathcal{H}}^{\pi}(s_t;\omega')\right)^2\right]$, where ω' indicates it is treated as a constant. Note that this process is exactly the same procedure of training the standard critic, except we use negative log probabilities instead of rewards.

Throughout the conducted experiments, we implemented the entropy critic network to share its parameters with the return value critic V_{ϕ}^{V}, with only the final linear layers for outputting its prediction distinct. This form of parameter sharing allows minimal computational overhead to implement EAPO.

Further, we employ the PopArt normalisation [van Hasselt *et al.*, 2016] to address the scale difference of entropy and return estimates. It is important to note that the negative log probability $-\log\pi(a_t|s_t)$ is collected for every timestep. In contrast, the reward can be sparse, leading to significant magnitude variations based on the dynamics of the environment [Hessel *et al.*, 2019]. This discrepancy can pose challenges, especially when using a shared architecture. Thus, utilising the value normalisation technique like PopArt is pivotal for the practical implementation of EAPO.

4.4 Entropy Advantage Policy Optimisation

Subsequently, we integrate the entropy advantage with the standard advantage estimate $\hat{A}_V^{\pi}$, also computed using GAE and return value critic parameterised by ϕ, analogously to the entropy advantage estimation process we describe above. Then the soft advantage function $\tilde{A}^{\pi}$ is

$$\tilde{A}^{\pi}(s_t,a_t) = \hat{A}^{V,\mathrm{GAE}(\lambda_V,\gamma_V)}(s_t,a_t) + \tau\hat{A}^{\mathcal{H},\mathrm{GAE}(\lambda_{\mathcal{H}},\gamma_{\mathcal{H}})}(s_t,a_t), \quad (11)$$

where $\hat{A}^{V,\mathrm{GAE}(\lambda_V,\gamma_V)}$ is the value advantage estimation using GAE. Finally, we substitute the estimated conventional advantage function in the policy objective of PPO and TRPO with $\tilde{A}^{\pi}$. The PPO objective function becomes:

$$L(\theta,\phi,\omega) = \hat{\mathbb{E}}_t\Big[\min(r_t^{\theta}\tilde{A}^{\pi}(s_t,a_t), \mathrm{clip}(r_t^{\theta},1-\epsilon,1+\epsilon)\tilde{A}^{\pi}(s_t,a_t))\Big] + c_1(L^V(\phi) + c_2 L^{\mathcal{H}}(\omega)), \quad (12)$$

where r_t^{θ} is the probability ratio between the behaviour policy $\pi_{\theta_{\text{old}}}(a_t|s_t)$ and the current policy $\pi_{\theta}(a_t|s_t)$, and c_1, c_2 and ϵ are hyperparameters to be adjusted. The value critic loss L^V is also defined by the mean square error, $L^V(\phi) = \hat{\mathbb{E}}_t\left[\frac{1}{2}\left(V(s_t;\phi) - \hat{V}^{\pi}(s_t)\right)^2\right]$ where $\hat{V}^{\pi}$ is the return value estimate.

Similarly, the optimisation problem of TPRO becomes:

$$\max_{\theta\in\Theta}\hat{\mathbb{E}}_t\Big[r_t^{\theta}\tilde{A}_t^{\pi}(s,a)\Big], \;\; s.t. \;\; \hat{\mathbb{E}}_t[\mathrm{KL}(\pi_{\theta_{\text{old}}}||\pi_{\theta})] \le \delta, \quad (13)$$

where δ is a hyperparameter.

4.5 Combining KL and Entropy Regularisation

EAPO, by extending TRPO, combines KL regulasization on policy updates and entropy regularisation on policy evaluation. Recent works [Vieillard *et al.*, 2020; Geist *et al.*, 2019; Shani *et al.*, 2020] suggest such combination could lead to improved convergence in regularised MDPs. Exploring the practical implications of this combined regularisation remains an interesting direction for future work.

5 Experiments

In this section, we evaluate the policy optimisation performance of EAPO against the corresponding baseline on-policy algorithms, PPO and TRPO. Specifically, we assess the optimisation efficiency for episodic tasks and the generalisation capability of EAPO on 16 Procgen [Cobbe *et al.*, 2020] benchmark tasks. Moreover, we investigate EAPO's efficacy on continuing control tasks using 4 discretised popular MuJoCo [Todorov *et al.*, 2012] tasks, and we analyse the impact of hyperparameters τ, $\gamma_{\mathcal{H}}$ and $\lambda_{\mathcal{H}}$. Finally, we include `MiniGrid-DoorKey-8x8` task [Chevalier-Boisvert *et al.*, 2023] to examine if EAPO can help solve the hard exploration task.

We implemented EAPO using Stable-baselines3 [Raffin *et al.*, 2019] and conducted experiments on environments provided by the Envpool [Weng *et al.*, 2022] library. All empirical results are averaged over 10 random seeds, with 95% confidence intervals indicated.

For the hyperparameter selection, we conducted a brief search for baseline algorithm hyperparameters that perform reasonably well, tuning only the EAPO-specific hyperparameters such as $\gamma_{\mathcal{H}}$ to ensure fair comparisons. Implementation details and hyperparameters are reported in the appendix.

5.1 Procgen Benchmark Environments

We evaluate PPO-based EAPO on the 16 Procgen benchmark environments [Cobbe *et al.*, 2020], which feature discrete actions and image-based observations. These environments include tasks with varied correlations between the episode

Preprint – IJCAI 2025: This is the accepted version made available for conference attendees.
Do not cite. The final version will appear in the IJCAI 2025 proceedings.

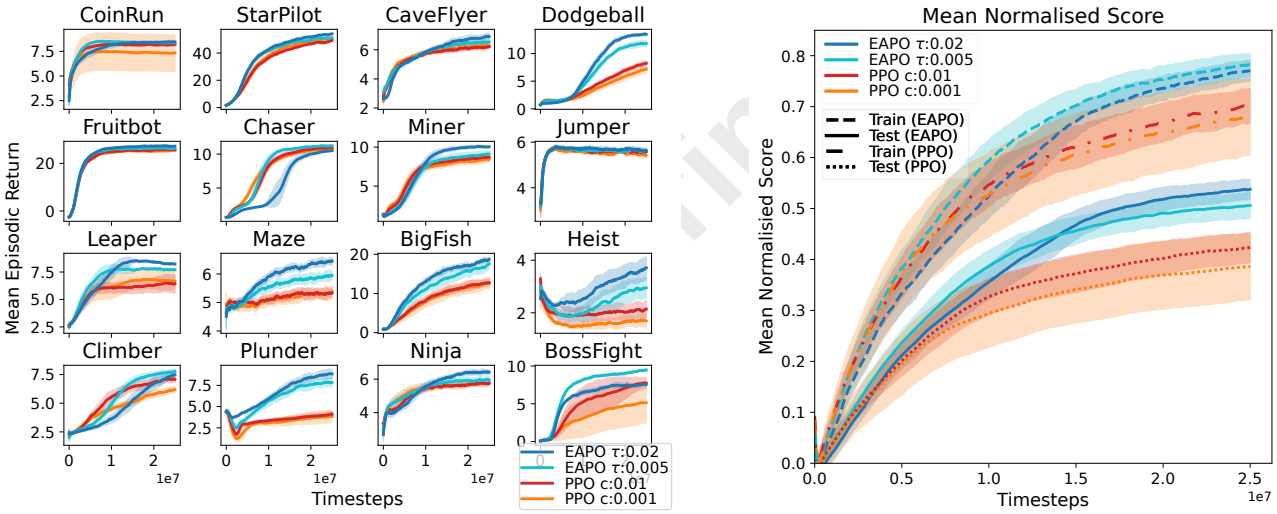

Figure 2: **Left**: Procgen test results of EAPO with $\gamma_{\mathcal{H}} = 0.8$, $\lambda_{\mathcal{H}} = 0.95$, and $\tau \in (0.02, 0.005)$ against PPO with entropy coefficients $c \in (0.001, 0.01)$. **Right**: The mean normalised score for both test and training.

length and the return, making them suitable for testing MaxEnt RL algorithms. Following [Cobbe *et al.*, 2020], we train on 200 procedurally generated levels and test on 100 unseen levels using *easy* difficulty setting. The same set of hyperparameters is used across all environments. We also compare against SD-SAC [Zhou *et al.*, 2024], a discrete version of SAC [Haarnoja *et al.*, 2018] that adopts a clipping mechanism for Q-function update and introduces entropy penalty to mitigate policy entropy collapsing. For hyperparameter tuning, we selected three representative environments: BigFish (long survival), Climber (indefinite exploration) and Dodgeball (survival followed by quick goal-reaching). These tasks represent different episode length-return correlations.

Figure 2 and Table 1 summarise the generalisation test results of EAPO, baseline PPO agents with varying τ and entropy bonus coefficients, and SD-SAC. The mean normalised score is computed as in [Cobbe *et al.*, 2020]. EAPO with $\gamma_{\mathcal{H}} = 0.8$ and $\lambda_{\mathcal{H}} = 0.95$ significantly surpasses the baseline PPO, achieving a 29% higher mean normalised score (0.54 ± 0.06 vs 0.42 ± 0.07) and consistently outperforming across most environments, with particularly large improvements in Plunder (+116%), BigFish (+55%), and Dodgeball (+60%). These results demonstrate EAPO's effectiveness in both test (generalisation) and training (optimisation) phases. Notably, SD-SAC significantly underperforms in these environments, highlighting the difficulty of stabilising discrete SAC. SD-SAC particularly struggles with the reward inflation problem, as shown by the its pattern of performance: it achieves reasonable results only in environments with positive length-return correlation (StarPilot, Fruitbot and BigFish) where extended episodes naturally lead to higher returns, while failing in environments allowing indefinite actions without reward (e.g., Plunder, Leaper). We provide the used hyperparameters and the full learning curves of SD-SAC in the appendix.

Moreover, we investigated the impact of the GAE $\lambda_{\mathcal{H}}$ for the entropy advantage estimation, finding it does not significantly affect performance. This suggests adjusting $\gamma_{\mathcal{H}}$ and τ is usually sufficient in episodic tasks given a small $\lambda_{\mathcal{H}}$.

Figure 2 (Right) shows the improved generalisation of high-entropy policies. Higher temperature τ favours high-entropy trajectories (see Figure 3), performing similarly or worse during the training but better in testing. This aligns with [Eysenbach and Levine, 2021], showing MaxEnt policies' robustness to distributional shifts.

Figure 4 demonstrates that the lower discount factor $\gamma_{\mathcal{H}}$ mitigates the reward inflation problem [Yu *et al.*, 2022]. A small $\gamma_{\mathcal{H}}$ significantly improves the performance in environments where agents can traverse without meaningful reward gain (Dodgeball and Climber) while maintaining performance in environments inherently requiring longer episodes (Bigfish, Bossfight).

5.2 Discretised Continuous Control Tasks

We measure the performance of EAPO extending PPO (EAPO-PPO) and TRPO (EAPO-TRPO) on continuing control tasks in 4 MuJoCo environments, comparing them against their corresponding baselines. For the PPO baselines, we searched for the best entropy coefficient within the set $c \in (0.0001, 0.001, 0.01)$. Additionally, we tested the PPO and TRPO agent with the reward augmented by the entropy reward $-\tau \log \pi(a_t|s_t)$ to evaluate the impact of separating the MaxEnt objective. Note that the entropy reward-augmented baseline is effectively regarded as EAPO with $\gamma_{\mathcal{H}} = \gamma$ and $\lambda_{\mathcal{H}} = \lambda$, but without the entropy critic. We discretise the continuous action space using the method proposed by [Tang and Agrawal, 2020]. Results using the original continuous action space are provided in the appendix. We measured the mean episodic return of the stochastic policy periodically over 100 episodes during the training. We compare EAPO to the PPO agent with the best-performing entropy coefficient, and with the entropy reward augmented PPO.

The training curves are presented in Figure 5. The result shows that by adjusting $\gamma_{\mathcal{H}}$ and $\lambda_{\mathcal{H}}$, we can configure EAPO

Preprint – IJCAI 2025: This is the accepted version made available for conference attendees.
Do not cite. The final version will appear in the IJCAI 2025 proceedings.

Env.	EAPO $\gamma_{\mathcal{H}}$:0.8 $\lambda_{\mathcal{H}}$:0.95		EAPO $\gamma_{\mathcal{H}}$:0.9 $\lambda_{\mathcal{H}}$:0.0		PPO		SD-SAC
	τ:0.02	τ:0.005	τ:0.02	τ:0.005	c:0.01	c:0.001	α:0.02
CoinRun	8.34±0.24	8.31±0.22	7.59±0.51	8.33±0.36	8.13±0.22	7.38±2.48	4.08±0.88
StarPilot	**54.33±3.41**	52.12±1.95	53.28±2.57	54.24±3.13	49.4±3.72	50.57±2.4	32.35±5.34
CaveFlyer	7.03±0.32	6.48±0.57	**7.17±0.3**	6.62±0.56	6.32±0.7	6.3±0.63	3.19±0.97
Dodgeball	**13.64±0.68**	11.59±0.86	13.23±0.63	12.34±0.85	8.51±0.98	7.2±1.14	0.90±0.33
Fruitbot	**27.28±0.74**	26.5±0.9	27.19±1.14	26.57±1.32	25.91±1.26	25.34±1.18	24.00±1.17
Chaser	10.5±0.46	11.12±0.3	10.52±0.38	11.05±0.38	11.12±0.45	10.64±0.4	1.32±0.56
Miner	**10.13±0.38**	8.9±0.84	10.13±0.5	9.08±0.78	8.6±0.76	8.06±0.94	1.33±0.32
Jumper	5.7±0.46	5.76±0.27	**5.84±0.54**	5.45±0.48	5.21±0.45	5.17±0.44	5.00±0.72
Leaper	**8.24±0.52**	7.75±0.33	7.51±1.17	7.88±0.67	6.54±0.9	6.61±1.06	2.56±1.11
Maze	**6.5±0.48**	5.7±0.27	5.75±0.52	5.56±0.47	5.38±0.56	5.45±0.31	4.78±0.89
BigFish	**19.89±2.14**	17.88±1.46	19.83±2.4	18.73±2.87	12.84±1.54	12.56±2.86	15.80±5.28
Heist	3.75±0.66	2.97±0.67	**4.22±0.58**	3.06±0.49	2.11±0.54	1.68±0.5	3.36±1.00
Climber	7.43±0.87	**8.0±0.5**	6.29±0.74	7.22±0.52	6.78±0.68	6.22±0.48	2.60±0.68
Plunder	9.0±0.71	7.66±1.15	**9.55±0.9**	8.35±1.38	4.16±0.55	3.94±1.23	1.35±0.45
Ninja	6.49±0.47	6.07±0.46	6.43±0.41	6.21±0.37	6.09±0.63	5.87±0.58	2.72±0.91
BossFight	7.49±0.77	**9.58±0.66**	7.82±0.71	9.38±0.59	7.61±0.76	5.15±3.32	5.40±1.23
Norm.	**0.54±0.06**	0.51±0.05	0.52±0.07	0.51±0.06	0.42±0.07	0.38±0.11	0.11±0.10

Table 1: Mean episodic return and 95% CI from 10 seeds at the final timestep of tests on 100 unseen levels on 16 Procgen environments (EAPO and PPO). For SD-SAC, we report the maximum mean episodic return and 95% CI from 5 seeds, since the value at the final timestep is not representative due to its instability.

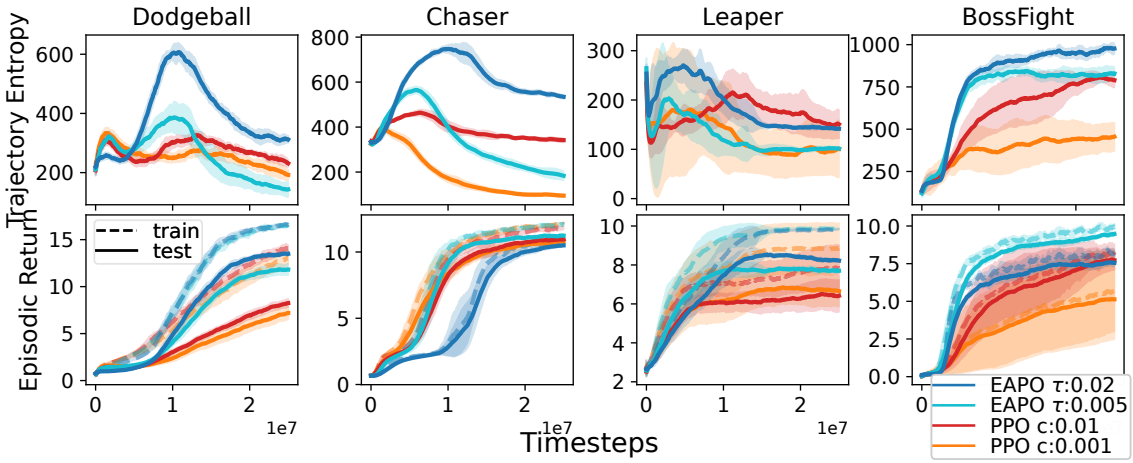

Figure 3: **Top**: Mean episodic trajectory entropy of EAPO ($\gamma_{\mathcal{H}} = 0.8$, $\lambda_{\mathcal{H}} = 0.95$) and PPO with entropy cofficients $c \in (0.01, 0.001)$, in a subset of Procgen environments during the test. **Bottom**: Mean episodic return during the test and the training. The high-entropy policy outperforms the low one during the test while achieving matching performance during the training in Dodgeball and Leaper, and exhibits a smaller generalisation gap in Dodgeball, Chaser and Leaper.

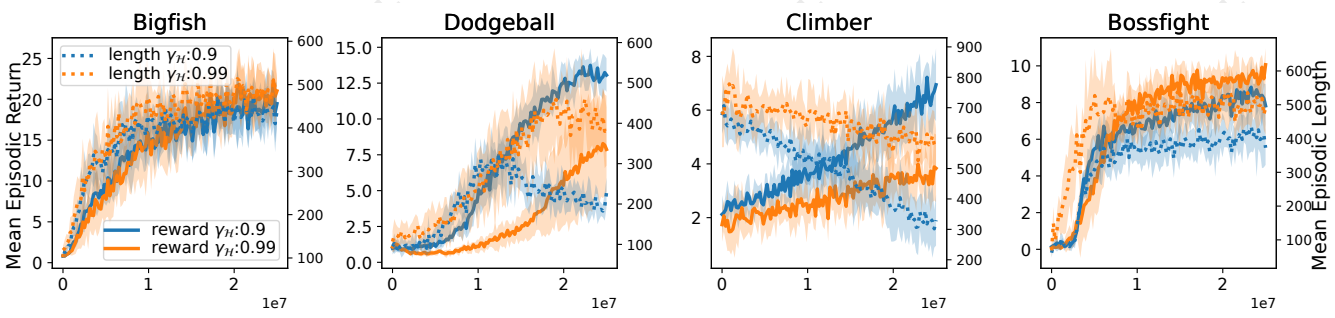

Figure 4: Comparison of Mean episodic returns and mean episodic lengths of EAPO using different discount factors $\gamma_{\mathcal{H}} \in (0.9, 0.99)$ for entropy return on 4 Procgen environments during the test. Results are averaged over 5 random seeds.

Preprint – IJCAI 2025: This is the accepted version made available for conference attendees.
Do not cite. The final version will appear in the IJCAI 2025 proceedings.

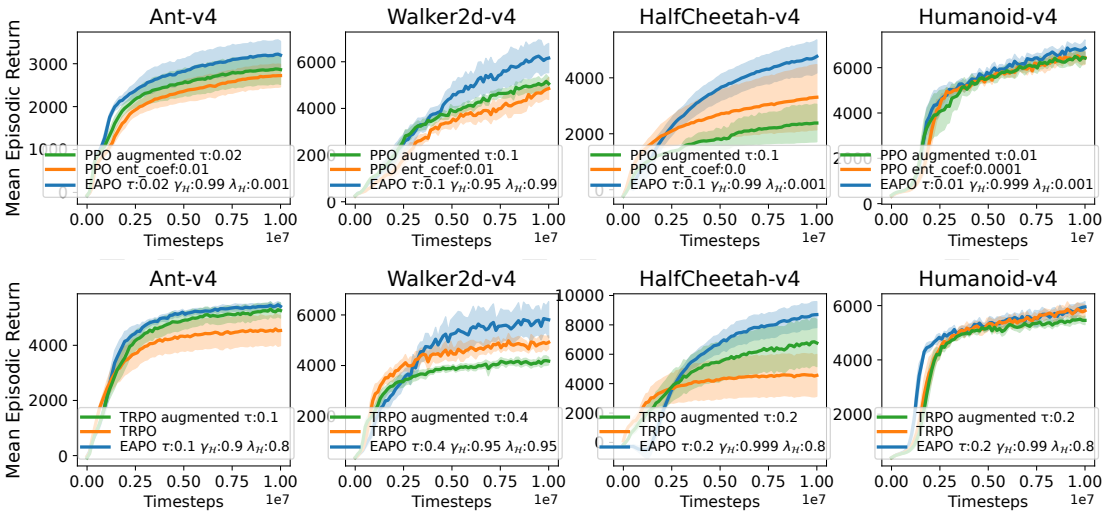

Figure 5: Performance results on 4 MuJoCo tasks. **Top**: EAPO-PPO. **Bottom**: EAPO-TRPO.

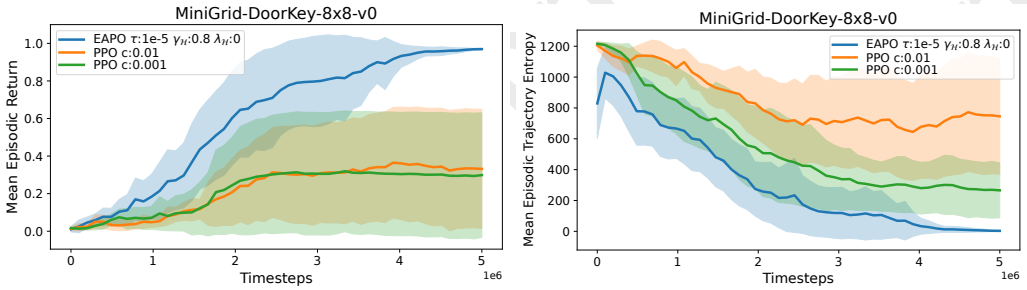

Figure 6: The return and trajectory entropy results of EAPO with $\tau = 1 \times 10^{-5}$, $\gamma_{\mathcal{H}} = 0.8$ and $\lambda_{\mathcal{H}} = 0$ and PPO.

to outperform or match the conventional entropy regularisation method throughout all environments. We found that the best-performing values of $\gamma_{\mathcal{H}}$ and $\lambda_{\mathcal{H}}$ vary depending on the characteristics of the environment, similar to their value counterparts γ and λ, respectively. Although EAPO demonstrates more stable performance compared to the entropy bonus, this relatively modest performance gain suggests that EAPO may be less efficient for continuing tasks. Figure 5 also demonstrates that the adjustability adopted by EAPO improves the naive implementation of the MaxEnt policy that augments the entropy reward. We also provide ablation experiments on $\gamma_{\mathcal{H}}$ and $\lambda_{\mathcal{H}}$ using PPO-based EAPO in the appendix.

5.3 `MiniGrid-DoorKey-8x8` Environment

Finally, we evaluate the exploration performance of PPO-based EAPO on the MiniGrid-DoorKey-8x8 environment [Chevalier-Boisvert *et al.*, 2023]. Figure 6 shows that EAPO achieves consistent success, solving this hard exploration task within 5M frames across all 10 seeds, while the baseline PPO succeeds in only 3 seeds. However, EAPO did not improve the stability since both EAPO and PPO were highly sensitive to their common hyperparameters (such as γ_V and λ_V). Moreover, EAPO's improvement in success rate appears to contradict to the theory that entropy regularisation may not effectively mitigate epistemic uncertainty [Mei *et al.*, 2020]. One possible explanation is that EAPO's trajectory entropy estimation, based on log probabilities from on-policy samples, tends to underestimate the entropy of previously visited trajectories. This underestimation induces an implicit bias toward exploring new trajectories.

6 Conclusion

We have introduced EAPO, a model-free on-policy actor-critic algorithm based on the maximum entropy reinforcement learning framework. Our approach facilitates a practical MaxEnt RL algorithm by taking advantage of existing mechanisms for standard value learning in actor-critic algorithms to the entropy objective. Through empirical evaluations, EAPO has been shown to replace the entropy cost method and that a more principled entropy maximisation method enhances policy optimisation. Moreover, by providing a stable MaxEnt method for discrete action spaces, EAPO enables deeper investigation of the MaxEnt RL in emerging applications such as GFlowNets [Mohammadpour *et al.*, 2024; Tiapkin *et al.*, 2024] and RLHF [Christiano *et al.*, 2017]. We anticipate that the method's simplicity will encourage broader adoption of MaxEnt RL and inference-based methods [Cetin and Celiktutan, 2022; Marino *et al.*, 2021; Ward *et al.*, 2019] in promising areas like competitive RL and robust RL.

Preprint – IJCAI 2025: This is the accepted version made available for conference attendees.
Do not cite. The final version will appear in the IJCAI 2025 proceedings.

References

[Agarwal *et al.*, 2021] Alekh Agarwal, Sham M Kakade, Jason D Lee, and Gaurav Mahajan. On the theory of policy gradient methods: Optimality, approximation, and distribution shift. *The Journal of Machine Learning Research*, 22(1):4431–4506, 2021.

[Ahmed *et al.*, 2019] Zafarali Ahmed, Nicolas Le Roux, Mohammad Norouzi, and Dale Schuurmans. Understanding the impact of entropy on policy optimization. In *International conference on machine learning*, pages 151–160. PMLR, 2019.

[Cen *et al.*, 2022] Shicong Cen, Chen Cheng, Yuxin Chen, Yuting Wei, and Yuejie Chi. Fast global convergence of natural policy gradient methods with entropy regularization. *Operations Research*, 70(4):2563–2578, 2022.

[Cetin and Celiktutan, 2022] Edoardo Cetin and Oya Celiktutan. Policy gradient with serial markov chain reasoning. *Advances in Neural Information Processing Systems*, 35:8824–8839, 2022.

[Chevalier-Boisvert *et al.*, 2023] Maxime Chevalier-Boisvert, Bolun Dai, Mark Towers, Rodrigo de Lazcano, Lucas Willems, Salem Lahlou, Suman Pal, Pablo Samuel Castro, and Jordan Terry. Minigrid & miniworld: Modular & customizable reinforcement learning environments for goal-oriented tasks. *CoRR*, abs/2306.13831, 2023.

[Christiano *et al.*, 2017] Paul F Christiano, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei. Deep reinforcement learning from human preferences. *Advances in neural information processing systems*, 30, 2017.

[Christodoulou, 2019] Petros Christodoulou. Soft actor-critic for discrete action settings. *arXiv preprint arXiv:1910.07207*, 2019.

[Cobbe *et al.*, 2020] Karl Cobbe, Chris Hesse, Jacob Hilton, and John Schulman. Leveraging procedural generation to benchmark reinforcement learning. In *International conference on machine learning*, pages 2048–2056. PMLR, 2020.

[Eysenbach and Levine, 2019] Benjamin Eysenbach and Sergey Levine. If maxent rl is the answer, what is the question? *arXiv preprint arXiv:1910.01913*, 2019.

[Eysenbach and Levine, 2021] Benjamin Eysenbach and Sergey Levine. Maximum entropy rl (provably) solves some robust rl problems. *arXiv preprint arXiv:2103.06257*, 2021.

[Geist *et al.*, 2019] Matthieu Geist, Bruno Scherrer, and Olivier Pietquin. A theory of regularized markov decision processes. In *International Conference on Machine Learning*, pages 2160–2169. PMLR, 2019.

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

[Haarnoja *et al.*, 2018] Tuomas Haarnoja, Aurick Zhou, Kristian Hartikainen, George Tucker, Sehoon Ha, Jie Tan, Vikash Kumar, Henry Zhu, Abhishek Gupta, Pieter Abbeel, et al. Soft actor-critic algorithms and applications. *arXiv preprint arXiv:1812.05905*, 2018.

[Hessel *et al.*, 2019] Matteo Hessel, Hubert Soyer, Lasse Espeholt, Wojciech Czarnecki, Simon Schmitt, and Hado van Hasselt. Multi-task deep reinforcement learning with popart. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 3796–3803, 2019.

[Levine, 2018] Sergey Levine. Reinforcement learning and control as probabilistic inference: Tutorial and review. *arXiv preprint arXiv:1805.00909*, 2018.

[Marino *et al.*, 2021] Joseph Marino, Alexandre Piché, Alessandro Davide Ialongo, and Yisong Yue. Iterative amortized policy optimization. *Advances in Neural Information Processing Systems*, 34:15667–15681, 2021.

[Mei *et al.*, 2020] Jincheng Mei, Chenjun Xiao, Csaba Szepesvari, and Dale Schuurmans. On the global convergence rates of softmax policy gradient methods. In *International Conference on Machine Learning*, pages 6820–6829. PMLR, 2020.

[Mnih *et al.*, 2016] Volodymyr Mnih, Adria Puigdomenech Badia, Mehdi Mirza, Alex Graves, Timothy Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu. Asynchronous methods for deep reinforcement learning. In *International conference on machine learning*, pages 1928–1937. PMLR, 2016.

[Mohammadpour *et al.*, 2024] Sobhan Mohammadpour, Emmanuel Bengio, Emma Frejinger, and Pierre-Luc Bacon. Maximum entropy gflownets with soft q-learning. In *International Conference on Artificial Intelligence and Statistics*, pages 2593–2601. PMLR, 2024.

[Neu *et al.*, 2017] Gergely Neu, Anders Jonsson, and Vicenç Gómez. A unified view of entropy-regularized markov decision processes. *arXiv preprint arXiv:1705.07798*, 2017.

[Raffin *et al.*, 2019] Antonin Raffin, Ashley Hill, Maximilian Ernestus, Adam Gleave, Anssi Kanervisto, and Noah Dormann. Stable baselines3, 2019.

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

[Schulman *et al.*, 2015b] John Schulman, Philipp Moritz, Sergey Levine, Michael Jordan, and Pieter Abbeel. High-dimensional continuous control using generalized advantage estimation. *arXiv preprint arXiv:1506.02438*, 2015.

[Schulman *et al.*, 2017a] John Schulman, Xi Chen, and Pieter Abbeel. Equivalence between policy gradients and soft q-learning. *arXiv preprint arXiv:1704.06440*, 2017.

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

Preprint – IJCAI 2025: This is the accepted version made available for conference attendees.
Do not cite. The final version will appear in the IJCAI 2025 proceedings.

[Shani *et al.*, 2020] Lior Shani, Yonathan Efroni, and Shie Mannor. Adaptive trust region policy optimization: Global convergence and faster rates for regularized mdps. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 5668–5675, 2020.

[Shi *et al.*, 2019] Wenjie Shi, Shiji Song, and Cheng Wu. Soft policy gradient method for maximum entropy deep reinforcement learning. *arXiv preprint arXiv:1909.03198*, 2019.

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

[Tang and Agrawal, 2020] Yunhao Tang and Shipra Agrawal. Discretizing continuous action space for on-policy optimization. In *Proceedings of the aaai conference on artificial intelligence*, volume 34, pages 5981–5988, 2020.

[Tiapkin *et al.*, 2023] Daniil Tiapkin, Denis Belomestny, Daniele Calandriello, Eric Moulines, Remi Munos, Alexey Naumov, Pierre Perrault, Yunhao Tang, Michal Valko, and Pierre Menard. Fast rates for maximum entropy exploration. *arXiv preprint arXiv:2303.08059*, 2023.

[Tiapkin *et al.*, 2024] Daniil Tiapkin, Nikita Morozov, Alexey Naumov, and Dmitry P Vetrov. Generative flow networks as entropy-regularized rl. In *International Conference on Artificial Intelligence and Statistics*, pages 4213–4221. PMLR, 2024.

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

[van Hasselt *et al.*, 2016] Hado P van Hasselt, Arthur Guez, Matteo Hessel, Volodymyr Mnih, and David Silver. Learning values across many orders of magnitude. *Advances in neural information processing systems*, 29, 2016.

[Vieillard *et al.*, 2020] Nino Vieillard, Tadashi Kozuno, Bruno Scherrer, Olivier Pietquin, Rémi Munos, and Matthieu Geist. Leverage the average: an analysis of kl regularization in reinforcement learning. *Advances in Neural Information Processing Systems*, 33:12163–12174, 2020.

[Ward *et al.*, 2019] Patrick Nadeem Ward, Ariella Smofsky, and Avishek Joey Bose. Improving exploration in soft-actor-critic with normalizing flows policies. *arXiv preprint arXiv:1906.02771*, 2019.

[Weng *et al.*, 2022] Jiayi Weng, Min Lin, Shengyi Huang, Bo Liu, Denys Makoviichuk, Viktor Makoviychuk, Zichen Liu, Yufan Song, Ting Luo, Yukun Jiang, et al. Envpool: A highly parallel reinforcement learning environment execution engine. *Advances in Neural Information Processing Systems*, 35:22409–22421, 2022.

[Xu *et al.*, 2021] Yaosheng Xu, Dailin Hu, Litian Liang, Stephen McAleer, Pieter Abbeel, and Roy Fox. Target entropy annealing for discrete soft actor-critic. *arXiv preprint arXiv:2112.02852*, 2021.

[Yu *et al.*, 2022] Haonan Yu, Haichao Zhang, and Wei Xu. Do you need the entropy reward (in practice)? *arXiv preprint arXiv:2201.12434*, 2022.

[Zhou *et al.*, 2024] Haibin Zhou, Tong Wei, Zichuan Lin, junyou li, Junliang Xing, Yuanchun Shi, Li Shen, Chao Yu, and Deheng Ye. Revisiting discrete soft actor-critic. *Transactions on Machine Learning Research*, 2024.

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