Maximum Entropy On-Policy Actor-Critic via Entropy Advantage Estimation

Jean Seong Bjorn Choe
School of Electrical Engineering
Korea University
garangg@korea.ac.kr

Jong-Kook Kim
School of Electrical Engineering
Korea University
jongkook@korea.ac.kr

Abstract

Entropy Regularisation is a widely adopted technique that enhances policy optimisation performance and stability. A notable form of entropy regularisation is augmenting the objective with an entropy term, thereby simultaneously optimising the expected return and the entropy. This framework, known as maximum entropy reinforcement learning (MaxEnt RL), has shown theoretical and empirical successes. However, its practical application in straightforward on-policy actor-critic settings remains surprisingly underexplored. We hypothesise that this is due to the difficulty of managing the entropy reward in practice. This paper proposes a simple method of separating the entropy objective from the MaxEnt RL objective, which facilitates the implementation of MaxEnt RL in on-policy settings. Our empirical evaluations demonstrate that extending Proximal Policy Optimisation (PPO) and Trust Region Policy Optimisation (TRPO) within the MaxEnt framework improves policy optimisation performance in both MuJoCo and Procgen tasks. Additionally, our results highlight MaxEnt RL's capacity to enhance generalisation.

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 is to regularise the policy evaluation step by augmenting the conventional RL task objective with an entropy term, thereby directing policies toward areas of higher expected trajectory entropy. This scheme is often called Maximum Entropy RL (MaxEnt RL) [Ziebart, 2010, Haarnoja et al., 2018, Levine, 2018]. MaxEnt RL formulation is known to improve the exploration and robustness of policies by promoting stochasticity [Eysenbach and Levine, 2019, 2021]. In practice, MaxEnt RL can simply be implemented by adding an entropy reward to the original task reward.

Recent theoretical advancements inspired by the Mirror Descent theory have developed a unified view of these approaches [Vieillard et al., 2020, Geist et al., 2019, Tomar et al., 2020], suggesting that their combination could lead to faster convergence to the solution of the regularised objective [Shani et al., 2020]. Furthermore, the latest studies on policy gradient (PG) methods have shown the effectiveness of the MaxEnt RL in accelerating the convergence of PG algorithms [Mei et al., 2020, Agarwal et al., 2021, Cen et al., 2022]. However, despite the enticing theoretical support, its practical application remains underexplored, particularly in stochastic policy gradient methods in on-policy settings.

17th European Workshop on Reinforcement Learning (EWRL 2024).

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 Soft Actor-Critic (SAC) [Haarnoja et al., 2018], an off-policy MaxEnt algorithm. 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.

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. This work extends the well-established PPO [Schulman et al., 2017b] and TRPO [Schulman et al., 2015a].

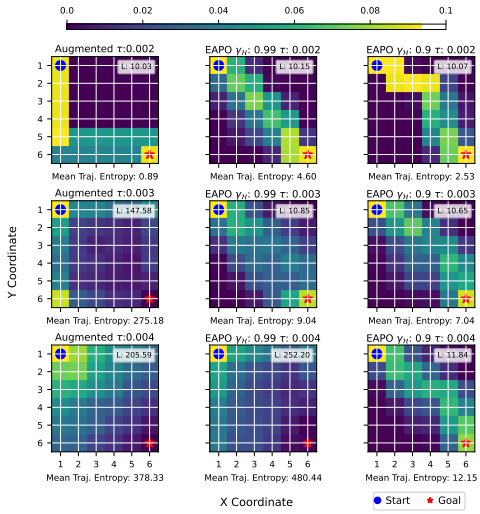

Figure 1: The normalised state visitation counts from 100 rollouts with policies trained on the modified `MiniGrid-Empty-8x8` task using a naive MaxEnt algorithm (PPO with the augmented entropy reward) and EAPO using 2 different discount factors $\gamma_{\mathcal{H}} \in (0.9, 0.99)$ and TD(0) for entropy estimation. We compare 3 different temperatures $\tau \in (0.002, 0.003, 0.004)$. A discount factor $\gamma_V = 0.99$ is used for the task reward. L is the mean length of trajectories, with agents aiming to minimise it (10 is optimal). See Appendix B.1 for more details.

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 empirically demonstrate that EAPO allows the development of a practical MaxEnt on-policy actor-critic algorithm. We test the efficacy of EAPO within deterministic environments with the discrete action space to align with existing theories on the MaxEnt formulation [Levine, 2018] and the softmax policy gradient methods [Mei et al., 2020]. Specifically, we evaluate the general training performance on 4 discretised [Tang and Agrawal, 2020] Mujoco continuous control tasks, including those with an infinite horizon [Todorov et al., 2012]. We also test the robustness in 16 Procgen episodic environments [Cobbe et al., 2020]. Additionally, we examine the usefulness of a MaxEnt policy in MiniGrid DoorKey environment [Chevalier-Boisvert et al., 2023].

2 Background

2.1 Preliminaries

This work considers a finite discounted deterministic Markov Decision Process (MDP) $\langle \mathcal{S}, \mathcal{A}, r, \rho, \mathcal{T}, \gamma_V, \gamma_{\mathcal{H}} \rangle$, 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 deterministic transition function $\mathcal{T} : \mathcal{S} \times \mathcal{A} \mapsto \mathcal{S}$, and r is the reward function $r : \mathcal{S} \times \mathcal{A} \mapsto \mathbb{R}$. γ_V and $\gamma_{\mathcal{H}}$ are the discount factors. We define the value func-

tion of state s under the policy π as $V^\pi(s) \coloneqq \mathbb{E}_{s_0=s, a_t\sim\pi(\cdot|s_t), s_{t+1}=\mathcal{T}(s_t,a_t)}\left[\sum_{t=0}^{\infty}\gamma_V^t r(s_t,a_t)\right]$. Also, the action-value of performing action a at state s under the policy π is $Q^\pi(s,a) \coloneqq \mathbb{E}_{s_0=s, a_0=a, a_{t>0}\sim\pi(\cdot|s_t), s_{t+1}=\mathcal{T}(s_t,a_t)}\left[\sum_{t=0}^{\infty}\gamma_V^t r(s_t,a_t)\right]$. 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 discounted policy-induced trajectory entropy, or the entropy rate of state s under policy π as

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

This trajectory entropy represents the Shannon entropy of the possible future trajectories' distribution in an MDP with deterministic dynamics [Levine, 2018, Tiapkin et al., 2023]. 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:

$$\begin{aligned} J(\pi) &= \mathbb{E}_{\substack{s_0\sim\rho, a_t\sim\pi,\\ s_{t+1}=\mathcal{T}(s_t,a_t)}}\left[\sum_{t=0}^{\infty}\gamma_V^t r(s_t,a_t) - \gamma_{\mathcal{H}}^t\tau\log\pi(a_t|s_t)\right] &(2)\\ &= \mathbb{E}_{s_0\sim\rho}\left[V^\pi(s_0) + \tau V_{\mathcal{H}}^\pi(s_0)\right], &(3)\end{aligned}$$

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.

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:

$$Q_{\mathcal{H}}^\pi(s_t,a_t) \coloneqq \gamma_{\mathcal{H}} V_{\mathcal{H}}^\pi(\mathcal{T}(s_t,a_t)) = \gamma_{\mathcal{H}} V_{\mathcal{H}}^\pi(s_{t+1}). \tag{4}$$

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 discounted trajectory entropy of the next state determined by the deterministic transition function.

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

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

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

$$\begin{aligned} A_{\mathcal{H}}^\pi(s_t,a_t) &\coloneqq Q_{\mathcal{H}}^\pi(s_t,a_t) - \mathbb{E}_{a\sim\pi(\cdot|s_t)}\left[Q_{\mathcal{H}}^\pi(s_t,a)\right]\\ &= Q_{\mathcal{H}}^\pi(s_t,a_t) - V_{\mathcal{H}}^\pi(s_t) + \mathbb{E}_{a\sim\pi(\cdot|s_t)}\left[-\log\pi(a|s_t)\right]. &(6)\end{aligned}$$

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

$$\begin{aligned} \tilde{A}^\pi(s_t,a_t) &\coloneqq A^\pi(s_t,a_t) + \tau A_{\mathcal{H}}^\pi(s_t,a_t) &(7)\\ &= Q^\pi(s_t,a_t) - V^\pi(s_t) + \tau(Q_{\mathcal{H}}^\pi(s_t,a_t) - V_{\mathcal{H}}^\pi(s_t) + \mathbb{E}_{a\sim\pi(\cdot|s_t)}\left[-\log\pi(a|s_t)\right])\\ &= \tilde{Q}^\pi(s_t,a_t) - \tilde{V}^\pi(s_t) + \tau\mathbb{E}_{a\sim\pi(\cdot|s_t)}\left[-\log\pi(a|s_t)\right]. &(8)\end{aligned}$$

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,*

$$\nabla_\theta J(\pi_\theta) = \mathbb{E}_{\substack{s_0\sim\rho,\\ a_t\sim\pi,\\ s_{t+1}=\mathcal{T}(s_t,a_t)}}\left[(\gamma_V^t A(s_t,a_t) + \gamma_{\mathcal{H}}^t\tau A_{\mathcal{H}}^\pi(s_t,a_t))\nabla_\theta\log\pi_\theta(a_t|s_t)\right]. \tag{9}$$

We provide the proof in Appendix A. 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:

$$\nabla_\theta J(\pi_\theta) \approx \mathbb{E}_{\substack{s_0\sim\rho,\\ a_t\sim\pi,\\ s_{t+1}=\mathcal{T}(s_t,a_t)}} \left[\tilde{A}^\pi(s_t,a_t)\nabla_\theta \log \pi_\theta(a_t|s_t)\right]. \tag{10}$$

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

One of the most prominent aspects of the MaxEnt RL formulation that has been studied is the ability to connect policy gradient methods and off-policy value-based methods that learn the soft Q-function [Haarnoja et al., 2018, Nachum et al., 2017, O'Donoghue et al., 2016, Schulman et al., 2017a]. However, this work focuses on the soft policy gradient using the soft advantage estimation in an on-policy setting.

Shi et al. [2019] explored the soft policy gradient method, emphasising its inherent simplicity. While they employed the soft Q-function to guide the policy gradient, linking their method to off-policy techniques, EAPO leverages the variance-reduced estimation of the entropy advantage function to formulate the soft advantage function suitable for on-policy algorithms. Moreover, Shi et al. [2019] introduced additional techniques to mitigate the challenging task of estimating the soft Q-function. However, EAPO can seamlessly integrate with existing techniques, such as value function normalisation, and GAE, due to its structural equivalence between its method for estimating the entropy advantage function and the conventional advantage function.

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 the entropy bonus term seeks to maximise policy entropy at visited states, MaxEnt RL 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]. Despite its empirical success, this method remains a heuristic approach without solid theoretical understanding[Ahmed et al., 2019]. In this work, we show that the use of MaxEnt RL can replace the traditional entropy cost term.

The algorithms studied in this work can be seen as an instance of algorithm that both KL regularisation and entropy regularisation are applied [Vieillard et al., 2020, Geist et al., 2019]. However, this work does not analyse how these two types of regularisations interact.

Yu et al. [2022] studied the harmful effect of having entropy regularisation on the policy evaluation step and has motivated this research. Unlike their work, we draw a positive conclusion for the use of entropy rewards. In our experiments comparing the entropy cost method, using the MaxEnt framework instead of the entropy cost consistently improves the performance of on-policy algorithms.

Recent studies have investigated the theoretical properties of policy gradient methods [Agarwal et al., 2021, Mei et al., 2020, Cen et al., 2022]. Notably, the authors have shown that combining Natural Policy Gradient (NPG) methods [Kakade, 2001] and the entropy-regularised MDPs can speed up the convergence. Although some authors [Khodadadian et al., 2021, Shani et al., 2020] have drawn the connection between PPO and NPG methods rigorously in theoretical settings, it remains unclear whether PPO-based EAPO, which extends PPO to the Regularised MDP setting, can benefit from the theoretical guarantees.

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] for a variance-reduced estimation of the entropy advantage:

$$\hat{A}^{\mathcal{H},\mathrm{GAE}(\lambda_{\mathcal{H}},\gamma_{\mathcal{H}})}(s_t,a_t) \coloneqq \sum_{l=0}^{\infty}(\lambda_{\mathcal{H}}\gamma_{\mathcal{H}})^l \delta^{\mathcal{H}}_{t+l}, \tag{11}$$

where $\delta^{\mathcal{H}}_t \coloneqq -\log\pi(a_t|s_t) + \gamma_{\mathcal{H}} V^{\pi}_{\mathcal{H}}(s_{t+1}) - V^{\pi}_{\mathcal{H}}(s_t)$, 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 $V^{\pi}_{\mathcal{H}}$, and it is trained by minimising the mean squared error $L^{\mathcal{H}}(\omega) \coloneqq \hat{\mathbb{E}}_t\left[\frac{1}{2}\left(V^{\pi}_{\mathcal{H}}(s_t;\omega) - \hat{V}^{\pi}_{\mathcal{H}}(s_t)\right)^2\right]$, where the trajectory entropy estimate $\hat{V}^{\pi}_{\mathcal{H}}(s_t)$ is calculated using TD($\lambda_{\mathcal{H}}$): $\hat{V}^{\pi}_{\mathcal{H}}(s_t) = \hat{A}^{\mathcal{H},\mathrm{GAE}(\lambda_{\mathcal{H}},\gamma_{\mathcal{H}})}(s_t,a_t) + V^{\pi}_{\mathcal{H}}(s_t;\omega)$. Throughout the conducted experiments, we implemented the entropy critic network to share its parameters with the return value critic V^{V}_{ϕ}, 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 $\hat{A}^{\pi}_{V}$, 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

$$\tilde{A}^{\pi}(s_t,a_t) = \hat{A}^{V,\mathrm{GAE}(\lambda_V,\gamma_V)}(s_t,a_t) + \tau\hat{A}^{\mathcal{H},\mathrm{GAE}(\lambda_{\mathcal{H}},\gamma_{\mathcal{H}})}(s_t,a_t), \tag{12}$$

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:

$$\begin{aligned} L(\theta,\phi,\omega) = \hat{\mathbb{E}}_t\Big[&\min(r_t(\theta)\tilde{A}^{\pi}(s_t,a_t), \mathrm{clip}(r_t(\theta),1-\epsilon,1+\epsilon)\tilde{A}^{\pi}(s_t,a_t))\Big] \\ &+ c_1(L^{V}(\phi) + c_2 L^{\mathcal{H}}(\omega)), \end{aligned} \tag{13}$$

where $r_t(\theta)$ 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:

$$\max_{\theta\in\Theta} \hat{\mathbb{E}}_t\Big[r_t(\theta)\tilde{A}^\pi_t(s,a)\Big], \;\; s.t. \;\; \hat{\mathbb{E}}_t[\mathrm{KL}(\pi_{\theta_{\text{old}}}||\pi_\theta)] \leq \delta, \tag{14}$$

where δ is a hyperparameter.

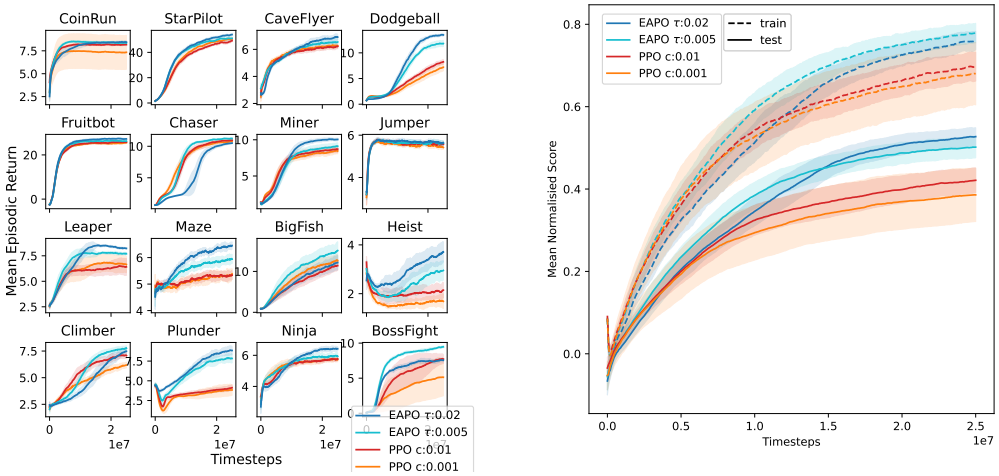

Figure 2: **Left**: Generalisation test results of EAPO agents with $\gamma_{\mathcal{H}} = 0.8$, $\lambda_{\mathcal{H}} = 0.95$, and two different temperatures $\tau = 0.02$ and $\tau = 0.005$ against PPO agents with entropy coefficients of 0.001 and 0.01 on 16 Procgen [Cobbe et al., 2020] benchmark environments. Agents are evaluated on 100 levels unseen during the training. EAPO consistently outperforms or at least matches PPO in all environments. Results are averaged over 10 seeds, and the shaded area indicates the 95% confidence interval. **Right**: The mean normalised score for both test and training, computed according to [Cobbe et al., 2020].

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 environments. Moreover, we investigate EAPO's efficacy on continuing control tasks using 4 discretised popular MuJoCo [Todorov et al., 2012] environments, and we analyse the impact of hyperparameters τ, $\gamma_{\mathcal{H}}$ and $\lambda_{\mathcal{H}}$. Finally, we include `MiniGrid-DoorKey-8x8` environment [Chevalier-Boisvert et al., 2023] to examine if EAPO can help solve the hard exploration task.

We implemented EAPO using Stable-baseline3 [Raffin et al., 2019] and conducted experiments on environments provided by the Envpool [Weng et al., 2022] library. All empirical results are averaged over from 10 random seeds, with a 95% confidence interval 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 Appendix B.

5.1 Procgen benchmark environments

We evaluate the performance and the generalisation capability of PPO-based EAPO on the 16 Procgen benchmark environments [Cobbe et al., 2020]. These environments have a discrete action space and use raw images as observations. Procgen suite includes episodic tasks with both positive (e.g.,

Table 1: Mean episodic return at the final timestep of tests on 100 unseen levels on 16 Procgen environments. We report the mean and the 95% confidence interval from 10 different seeds.

Env.	EAPO $\gamma_{\mathcal{H}}$:0.8 $\lambda_{\mathcal{H}}$:0.95		EAPO $\gamma_{\mathcal{H}}$:0.9 $\lambda_{\mathcal{H}}$:0.0		PPO	
	τ:0.02	τ:0.005	τ:0.02	τ:0.005	c:0.01	c:0.001
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
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
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
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
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
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
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
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
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
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
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
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
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
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
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
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
Norm.	**0.54±0.06**	0.5±0.05	0.52±0.07	0.51±0.06	0.42±0.07	0.38±0.11

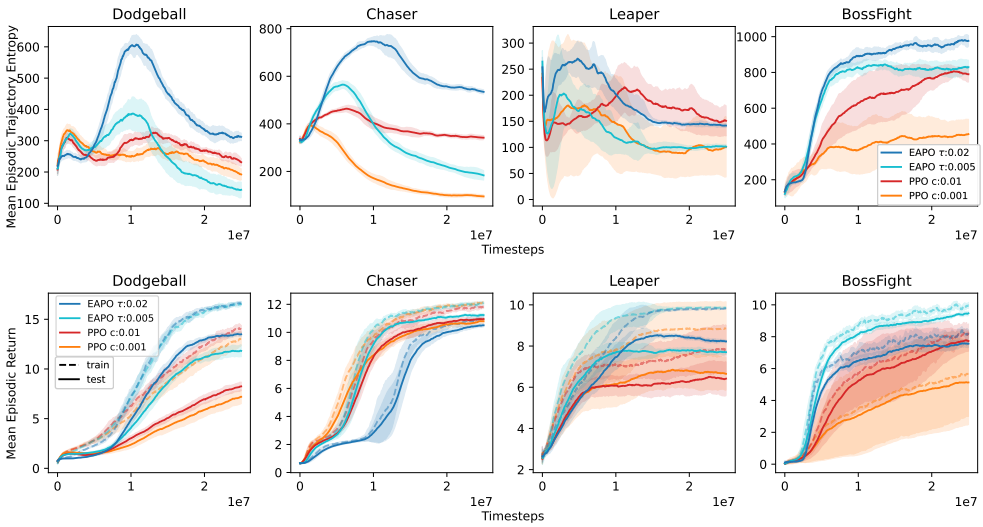

Figure 3: **Top**: Mean episodic trajectory entropy of EAPO ($\gamma_{\mathcal{H}} = 0.8$, $\lambda_{\mathcal{H}} = 0.95$) and PPO with entropy cofficients $c \in (0.01, 0.001)$, on a subset of Procgen environments during the test. The trajectory entropy of an episode is calculated as the sum of the negative log probability of the actions taken in the episode. **Bottom**: Mean episodic return of the selected environments during the test and the training. The higher entropy policy ($\tau = 0.02$) outperforms the lower entropy policy ($\tau = 0.005$) during the test while achieving matching performance during the training (Dodgeball, Leaper) and exhibits a smaller generalisation gap (Dodgeball, Chaser, Leaper).

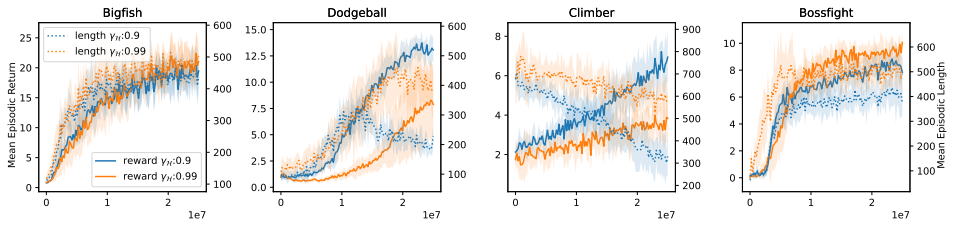

Figure 4: Comparison of Mean episodic returns and mean episodic lengths of EAPO using different discount factors $\gamma_{\mathcal{H}} \in (0.9, 0.99)$ for entropy return on 4 Procgen environments during the test. Results are averaged over 5 random seeds.

BigFish) and negative (e.g., Climber) correlations between the episode length and the return, making them suitable for testing MaxEnt RL algorithms. Following the evaluation procedure in [Cobbe et al., 2020], we trained agents on 200 procedurally generated levels and tested the performance as the mean episodic return on 100 unseen levels for each environment. We use the *easy* difficulty setting.

We employed a single set of hyperparameters for all environments. For hyperparameter tuning, we used three representative environments: BigFish, Climber and Dodgeball. Each represents different levels of correlation between episode length and the return. BigFish requires the agent to survive as long as possible, Climber allows indefinite exploration, and in Dodgeball, the agent must survive during the first phase and reach the goal quickly in the second stage.

Figure 2 and Table 1 summarise the generalisation test results of EAPO and baseline PPO agents with varying τ and the entropy bonus coefficient. The mean normalised score is computed according to [Cobbe et al., 2020]. EAPO, with a lower discount factor of $\gamma_{\mathcal{H}} = 0.8$ and $\lambda_{\mathcal{H}} = 0.95$ surpasses the baseline PPO agent in most of the environments. Additionally, EAPO also outperforms the baseline during the training phase, demonstrating its efficiency in policy optimisation. We also investigate the impact of the GAE $\lambda_{\mathcal{H}}$ hyperparameter for the entropy advantage estimation. The result shows that $\lambda_{\mathcal{H}}$ does not affect the performance significantly, suggesting that adjusting $\gamma_{\mathcal{H}}$ and τ would be sufficient for most cases.

Figure 2 (Right) shows the improved generalisation capability of high entropy policies. The policy trained with higher temperature τ favours high entropy trajectories (see Figure 3) and performs similar or worse than the one with lower τ during the training but achieves better during the test. This result is in coherence with the previous study of [Eysenbach and Levine, 2021] that a MaxEnt policy is robust to the distributional shift in environments.

Figure 4 demonstrates that lowering the 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 environment 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 coeffcient 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]. Experiments using continuous action space are provided in Appendix C. 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 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

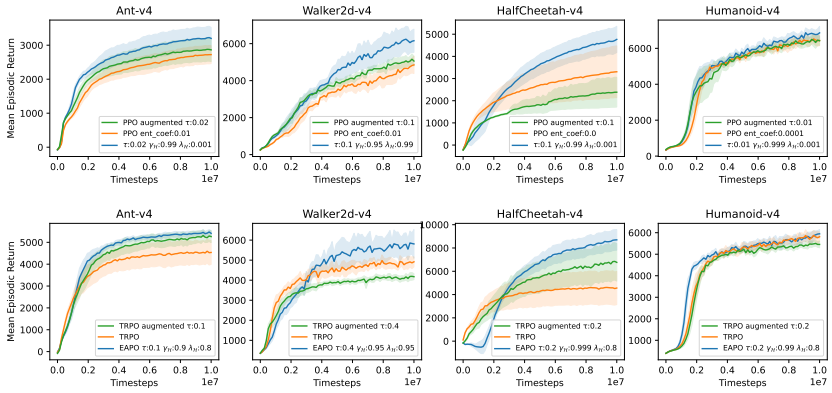

Figure 5: Performance comparison on 4 MuJoCo tasks. We measured the mean episodic return of the stochastic policy periodically over 100 episodes during the training. Results are averaged from 10 random seeds, and the shaded area indicates the 95% confidence interval. **Top**: EAPO-PPO. We compare EAPO to the PPO agent with the best-performing entropy coefficient, and with the entropy reward augmented PPO. **Bottom**: EAPO-TRPO. We also compare with the entropy reward augmented TRPO.

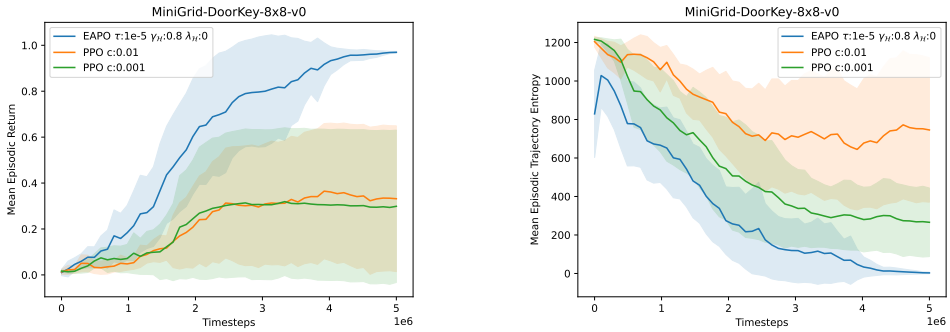

Figure 6: The return and trajectory entropy comparison results of EAPO with $\tau = 1e-5$, $\gamma_{\mathcal{H}} = 0.8$ and $\lambda_{\mathcal{H}} = 0$ and PPO with entropy coefficent 0.01 and 0.001. Results are averaged from 10 random seeds, and the shaded area indicates the 95% confidence interval.

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 Appendix C.

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, with the given hyperparameters, can solve this hard exploration task within 5,000,000 frames for all 10 seeds, whereas the baseline PPO only achieves the goal for 3 seeds. However, unlike other tasks presented in this work, this task was highly sensitive to hyperparameters. As noted by [Mei et al., 2020], entropy regularization may not effectively mitigate epistemic uncertainty. Our results show inconclusive evidence for better exploration using MaxEnt RL in this context.

6 Conclusion

We have introduced EAPO, a model-free on-policy actor-critic algorithm based on the maximum entropy reinforcement learning framework. Our approach shows that a straightforward extension of existing mechanisms for standard value learning in on-policy actor-critic algorithms to the trajectory entropy objective can facilitate the practical implementation of MaxEnt RL. Through empirical evaluations, our method has been shown to replace the conventional entropy regularisation method and that a more principled entropy maximisation method enhances generalisation. While this paper focuses on PPO and TRPO, one can seamlessly adapt EAPO to other advantage actor-critic algorithms such as A3C [Mnih et al., 2016]. This adaptability lays the groundwork for deeper investigations into the interactions between the MaxEnt RL framework and various components of reinforcement learning algorithms. We anticipate that the inherent simplicity of on-policy algorithms and EAPO will encourage broader applications of the MaxEnt RL algorithm to promising areas like competitive reinforcement learning and robust reinforcement learning.

References

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.

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.

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.

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.

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.

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

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

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.

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.

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.

Sham M Kakade. A natural policy gradient. *Advances in neural information processing systems*, 14, 2001.

Sajad Khodadadian, Prakirt Raj Jhunjhunwala, Sushil Mahavir Varma, and Siva Theja Maguluri. On the linear convergence of natural policy gradient algorithm. In *2021 60th IEEE Conference on Decision and Control (CDC)*, pages 3794–3799. IEEE, 2021.

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

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.

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.

Ofir Nachum, Mohammad Norouzi, Kelvin Xu, and Dale Schuurmans. Bridging the gap between value and policy based reinforcement learning. *Advances in neural information processing systems*, 30, 2017.

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

Brendan O'Donoghue, Remi Munos, Koray Kavukcuoglu, and Volodymyr Mnih. Combining policy gradient and q-learning. *arXiv preprint arXiv:1611.01626*, 2016.

Antonin Raffin, Ashley Hill, Maximilian Ernestus, Adam Gleave, Anssi Kanervisto, and Noah Dormann. Stable baselines3, 2019.

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, 2015a.

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

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

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

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.

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

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

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.

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.

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.

Manan Tomar, Lior Shani, Yonathan Efroni, and Mohammad Ghavamzadeh. Mirror descent policy optimization. *arXiv preprint arXiv:2005.09814*, 2020.

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.

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.

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.

Lucas Willems. Rl starter files. `https://github.com/lcswillems/rl-starter-files`, 2023.

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

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

A Proof of the soft policy gradient theorem

We begin with the proof of Shi et al. [2019]. Let us denote the discounted state distributions induced by the policy π as $\rho^\pi(s)$ for the discount factor γ_V, and as $\rho^\pi_{\mathcal{H}}(s)$ for the discount factor $\gamma_{\mathcal{H}}$, respectively. Then,

$$
\begin{aligned}
\nabla_\theta J(\pi_\theta) &= \mathbb{E}_{\substack{s_0\sim\rho,\\ a_t\sim\pi,\\ s_{t+1}=\mathcal{T}(s_t,a_t)}} \Big[(\tilde{Q}^\pi(s_t,a_t) - \tau\log\pi(a_t|s_t) - 1)\nabla_\theta \log\pi(a_t|s_t)\Big] \\
&= \sum_s \Bigg[\rho^\pi(s)[\nabla_\theta \sum_a \pi(s|a)[Q^\pi(s,a) - V^\pi(s)]] \\
&\qquad + \tau\rho^\pi_{\mathcal{H}}(s)[\nabla_\theta \sum_a \pi(s|a)(Q^\pi_{\mathcal{H}}(s,a) - \log\pi(a|s))]\Bigg] \\
&= \mathbb{E}_{s\sim\rho^\pi, a_t\sim\pi}\left[\nabla_\theta \log\pi_\theta(a_t|s_t) A^\pi(s_t,a_t)\right] + \tau\mathbb{E}_{s_t\sim\rho^\pi_{\mathcal{H}}, a_t\sim\pi}\left[\nabla_\theta\log\pi_\theta(a_t|s_t)A^\pi_{\mathcal{H}}(s_t,a_t)\right] \\
&= \mathbb{E}_{\substack{s_0\sim\rho,\\ a_t\sim\pi,\\ s_{t+1}=\mathcal{T}(s_t,a_t)}} \left[\nabla_\theta \log\pi_\theta(a_t|s_t)\gamma_V^t A^\pi(s_t,a_t)\right] \\
&\quad + \tau\mathbb{E}_{\substack{s_0\sim\rho,\\ a_t\sim\pi,\\ s_{t+1}=\mathcal{T}(s_t,a_t)}} \left[\nabla_\theta \log\pi_\theta(a_t|s_t)\gamma_{\mathcal{H}}^t A^\pi_{\mathcal{H}}(s_t,a_t)\right] \\
&= \mathbb{E}_{\substack{s_0\sim\rho,\\ a_t\sim\pi,\\ s_{t+1}=\mathcal{T}(s_t,a_t)}} \left[(\gamma_V^t A(s_t,a_t) + \gamma_{\mathcal{H}}^t\tau A^\pi_{\mathcal{H}}(s_t,a_t))\nabla_\theta\log\pi_\theta(a_t|s_t)\right]. \qquad \square
\end{aligned}
$$

B Hyperparameters and implementation details

B.1 Details of the `MiniGrid-Empty-8x8-v0` example

The action space of `MiniGrid-Empty-8x8-v0` [Chevalier-Boisvert et al., 2023] consists of 7 discrete actions: 2 actions for turning left or right, 1 action for moving forward, and 4 no-op actions. The reward, calculated as $1 - 0.9t/T$ is given only when the agent reaches the goal state, where t is the number of steps taken and T is the maximum number of steps allowed. For T, the default value (256) is used. We modified the turning actions so that they also move the agent forward to the corresponding directions (i.e., no extra step is required to maneuver), enabling multiple optimal trajectories. This modification reduces the optimal number of steps from 11 to 10. The observation used is the full $8 \times 8 \times 3$ image without partial observability. We employ a simple CNN architecture from [Willems, 2023] as a shared feature extractor. The state visitation plots are generated from 100 rollouts using policies trained over 4M frames. The mean trajectory entropy and the mean steps are averaged from 10 different random seeds. We select the representative heatmap for each setup from the run with the trajectory entropy closest to the average trajectory entropy across all seeds. The normalised state visitation frequency is calculated by dividing the number of visits to each state divided by the total number of state visits in the trajectories.

B.2 Hyperparameters

We use the default values in stable-baseline3 [Raffin et al., 2019] and envpool [Weng et al., 2022] libraries for the settings not specified in the table 2. EAPO-specific hyperparameters are reported in Table 3. The parameters for the MuJoCo tasks are found in a coarse hyperparameter search.

B.3 Network architecture

For discretised MuJoCo tasks, we used simple $\tanh$ networks for the policy and the critics with hidden layers of depth [64, 64] and [128, 128], respectively. We implemented the entropy critic as the independent output layer that shares the hidden layers with the value network. For the continuous MuJoCo tasks we use [256, 256] and [512, 512] for policy and critic networks and the state- and action-dependent σ output for the Gaussian policy with the Softplus output layer.

In Procgen benchmark experiments, we adopt the same IMPALA CNN architecture used in Cobbe et al. [2020]. The entropy critic is again a single final output layer that shares the CNN feature extractor.

Table 2: Common hyperparameters.

Parameter	MuJoCo (PPO)	MuJoCo (TRPO)	Procgen	Empty	DoorKey
Timesteps	10M	10M	25M	4M	5M
Num. Envs	64	64	64	16	64
Num. Steps	128	64	256	128	128
Learning Rate	5×10^{-4}	5×10^{-4}	5×10^{-4}	5×10^{-4}	1×10^{-3}
Batch Size	2048	1024 (critic)	2048	1024	1024
Epochs	4	4 (critic)	3	4	4
Discount Factor γ_V	0.99	0.99	0.995	0.99	0.995
GAE λ_V	0.95	0.95	0.8	0.95	0.995
Clip Range ϵ	0.2	0.2	0.1	0.2	0.2
Max Grad. Norm.	3.5	3.5	0.5	0.5	0.5
Adv. Norm.	True	True	True	True	True
Obs. Norm.	True	True	False	False	False
Rew. Norm.	False	False	False	False	False
PopArt β	0.03	0.03	0.03	0.03	0.03
Value Loss Coeff. c	0.25	0.25	0.5	0.5	0.5
Entropy Coeff.	(1e-2, 1e-3, 1e-4)	None	(1e-2, 1e-3)	None	[1e-5, 1e-4]
Target KL	None	0.07	None	None	None

Table 3: EAPO specific hyperparameters.

Parameter	Procgen	Empty	DoorKey		MuJoCo (PPO)	MuJoCo (TRPO)
$\gamma_{\mathcal{H}}$	(0.8, 0.9)	(0.9, 0.99)	0.8	Ant	0.99	0.9
				Walker2d	0.95	0.95
				HalfCheetah	0.99	0.999
				Humanoid	0.999	0.99
$\lambda_{\mathcal{H}}$	(0.95, 0.0)	0.0	0.0	Ant	0.001	0.8
				Walker2d	0.99	0.95
				HalfCheetah	0.001	0.8
				Humanoid	0.001	0.8
τ	(0.2, 0.005)	(0.002, 0.003, 0.004)	1e-5	Ant	0.02	0.1
				Walker2d	0.1	0.4
				HalfCheetah	0.1	0.2
				Humanoid	0.01	0.2
c_2	0.5	1.0	1.0		1.0	1.0

C Additional experiments

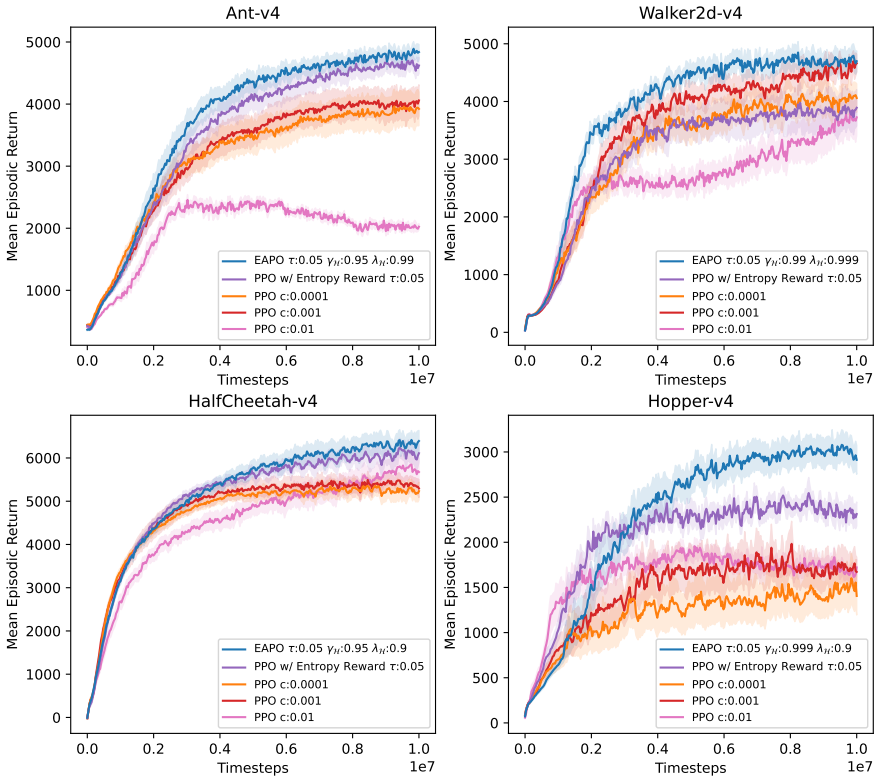

Figure 7: Performance comparison on 4 MuJoCo continuous locomotion tasks.

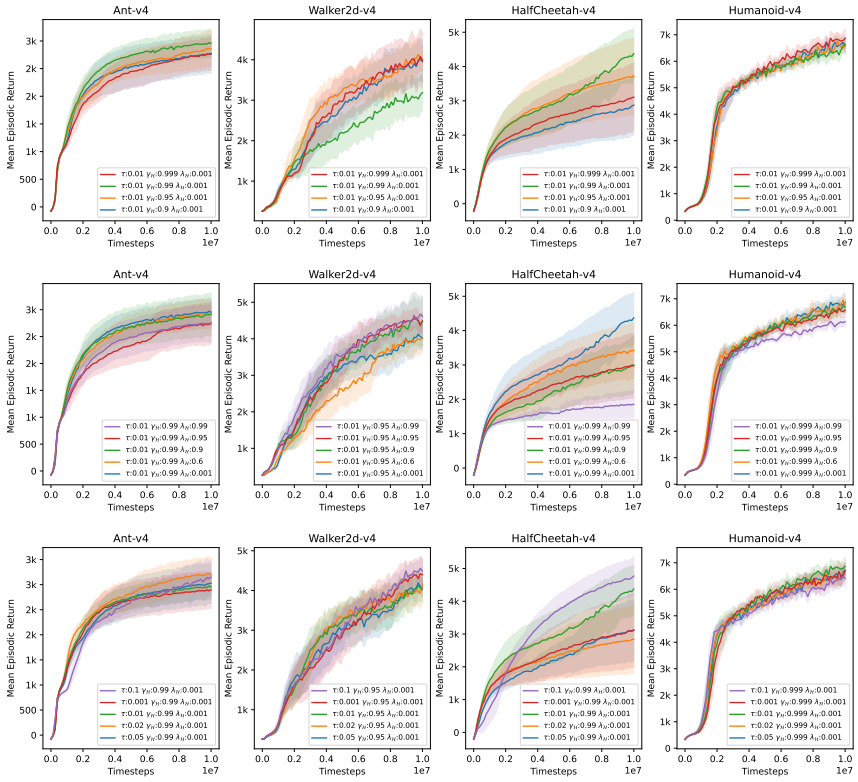

Figure 8: The ablation results on $\gamma_{\mathcal{H}}$, $\lambda_{\mathcal{H}}$, and τ for PPO-based EAPO.