On Principled Entropy Exploration in Policy Optimization

Jincheng Mei[1*], **Chenjun Xiao**[1*], **Ruitong Huang**[2], **Dale Schuurmans**[1] and **Martin Müller**[1]
[1]University of Alberta
[2]Borealis AI Lab
{jmei2, chenjun}@ualberta.ca, ruitong.huang@borealisai.com, {daes, mmueller}@ualberta.ca

Abstract

In this paper, we investigate Exploratory Conservative Policy Optimization (ECPO), a policy optimization strategy that improves exploration behavior while assuring monotonic progress in a principled objective. ECPO conducts maximum entropy exploration within a mirror descent framework, but updates policies using reversed KL projection. This formulation bypasses undesirable mode seeking behavior and avoids premature convergence to suboptimal policies, while still supporting strong theoretical properties such as guaranteed policy improvement. Experimental evaluations demonstrate that the proposed method significantly improves practical exploration and surpasses the empirical performance of state-of-the art policy optimization methods in a set of benchmark tasks.

1 Introduction

Deep reinforcement learning (RL) has recently shown to be remarkably effective in solving challenging sequential decision making problems [Schulman *et al.*, 2015; Mnih *et al.*, 2015; Silver *et al.*, 2016]. A central method of deep RL is *policy optimization*, which is based on formulating the problem as the optimization of a stochastic objective (expected return) of the underlying policy parameters [Williams and Peng, 1991; Williams, 1992; Sutton *et al.*, 1998]. Unlike standard optimization, policy optimization requires the objective and gradient to be *estimated* from data, typically gathered from a process depending on current parameters, simultaneously with parameter updates. Such an interaction between estimation and updating complicates the optimization process, and often necessitates the introduction of variance reduction methods, leading to algorithms with subtle hyperparameter sensitivity. Joint estimation and updating can also create poor local optima whenever sampling neglect of some region can lead to further entrenchment of a current solution. For example, a non-exploring policy might fail to sample from high reward trajectories, preventing any further improvement since no useful signal is observed. In practice, it is well known that successful application of deep RL techniques requires a combination of extensive hyperparameter tuning, and a large, if not impractical, number of sampled trajectories. It remains a major challenge to develop methods that can reliably perform policy improvement while maintaining sufficient exploration and avoiding poor local optima, yet do so quickly.

Several ideas for improving policy optimization have been proposed, generally focusing on the goals of improving stability and data efficiency [Peters *et al.*, 2010; Van Hoof *et al.*, 2015; Fox *et al.*, 2015; Schulman *et al.*, 2015; Montgomery and Levine, 2016; Nachum *et al.*, 2017b; Abdolmaleki *et al.*, 2018; Haarnoja *et al.*, 2018]. Unfortunately, a notable gap remains between empirically successful methods and their underlying theoretical support. Current analyses typically assume a simplified setting that either ignores the policy parametrization or only considers linear models. These assumptions are hard to justify when current practice relies on complex function approximators, such as deep neural networks, that are highly nonlinear in their underlying parameters. This gulf between theory and practice is a barrier to wider adoption of model-free policy gradient methods.

In this paper, we consider the maximum entropy reward objective, which has recently re-emerged as a foundation for state-of-the-art RL methods [Fox *et al.*, 2015; Schulman *et al.*, 2017a; Nachum *et al.*, 2017b; Haarnoja *et al.*, 2017; Neu *et al.*, 2017; Levine, 2018; Deisenroth *et al.*, 2013; Daniel *et al.*, 2012]. We first reformulate the maximization of this objective as a lift-and-project procedure, following Mirror Descent [Nemirovskii *et al.*, 1983; Beck and Teboulle, 2003]. We establish a monotonic improvement guarantee and the fixed point properties of this setup. The reformulation also has practical algorithmic consequences, suggesting that multiple gradient updates should be performed in the projection. These considerations lead to the Policy Mirror Descent (PMD) algorithm, which first lifts the policy to the simplex, ignoring the parametrization constraint, then approximately solves the projection by gradient updates in the parameter space.

We then investigate additional improvements to mitigate the potential deficiencies of PMD. The main algorithm we propose, Exploratory Conservative Policy Optimization (ECPO), incorporates both an entropy and relative entropy regularizer, and uses the mean seeking KL divergence for projection, which helps avoids poor deterministic policies. The projection can be efficiently solved to global optimality in certain non-convex cases, such as one-layer softmax networks. The

[*]Equal contribution

entropy exploration is principled. Firstly, in the convex subset setting, the algorithm enjoys sublinear regret. Secondly, we prove monotonic guarantees for ECPO with respect to a surrogate objective $\text{SR}(\pi)$. We further study the properties of $\text{SR}(\pi)$ and provide theoretical and empirical evidence that SR can effectively guide good policy search. Finally, we also extend this algorithm using value function approximations, and develop an actor-critic version that is effective in practice.

1.1 Notation and Problem Setting

We consider episodic settings with finite state and action spaces. The agent is modelled by a policy $\pi(\cdot|s)$ that specifies a probability distribution overs actions given state s. At each step t, the agent takes an action a_t by sampling from $\pi(\cdot|s_t)$. The environment then returns a reward $r_t = r(s_t, a_t)$ and the next state $s_{t+1} = f(s_t, a_t)$, where f is the transition not revealed to the agent. Given a trajectory, a sequence of states and actions $\rho = (s_1, a_1, \ldots, a_{T-1}, s_T)$, the policy probability and the total reward of ρ are defined as $\pi(\rho) = \prod_{t=1}^{T-1} \pi(a_t|s_t)$ and $r(\rho) = \sum_{t=1}^{T-1} r(s_t, a_t)$. Given a set of parametrized policy functions $\pi_\theta \in \Pi$, policy optimization aims to find the optimal policy π_θ^* by maximizing the expected reward,

$$\pi_\theta^* \in \arg\max_{\pi_\theta \in \Pi} \mathop{\mathbb{E}}_{\rho\sim\pi_\theta} r(\rho), \tag{1}$$

We use $\Delta \triangleq \{\pi | \sum_\rho \pi(\rho) = 1, \pi(\rho) \geq 0, \forall \rho\}$ to refer to the probability simplex over all trajectories. Without loss of generality, we assume that the state transition is deterministic, and the discount factor $\gamma = 1$. All theoretical results for the general stochastic environment are presented in the appendix.

2 Policy Mirror Descent

We first introduce the Policy Mirror Descent (PMD) strategy, which forms the basis for our algorithms. Consider the following optimization problem: given a *reference policy* $\bar{\pi}$ (usually the current policy), maximize the proximal regularized expected reward, using relative entropy as the regularizer:

$$\pi_\theta = \arg\max_{\pi_\theta \in \Pi} \mathop{\mathbb{E}}_{\rho\sim\pi_\theta} r(\rho) - \tau D_{\text{KL}}(\pi_\theta \| \bar{\pi}). \tag{2}$$

Relative entropy has been widely studied in online learning and optimization [Nemirovskii *et al.*, 1983; Beck and Teboulle, 2003], primarily as a component of the mirror descent method. This regularization makes the policy update in a conservative fashion, by searching policies within the neighbours of the current policy. In practice π_θ is usually parametrized as a function of $\theta \in \mathbb{R}^d$ and Π is generally a non-convex set. Therefore, Eq. (2) is a difficult constrained optimization problem.

One useful way to decompose Eq. (2) is to consider an alternating lift-and-project procedure.

$$\begin{aligned} &\textbf{(Project)} \quad \arg\min_{\pi_\theta \in \Pi} D_{\text{KL}}(\pi_\theta \| \bar{\pi}_\tau^*), \\ &\textbf{(Lift)} \quad \text{where } \bar{\pi}_\tau^* = \arg\max_{\pi \in \Delta} \mathop{\mathbb{E}}_{\rho\sim\pi} r(\rho) - \tau D_{\text{KL}}(\pi \| \bar{\pi}). \end{aligned} \tag{3}$$

Crucially, Eq. (3) remains equivalent to Eq. (2), in that it preserves the same solution, as established in Proposition 1.

Proposition 1. *Given a* reference policy $\bar{\pi}$*,*

$$\arg\max_{\pi_\theta \in \Pi} \mathop{\mathbb{E}}_{\rho\sim\pi_\theta} r(\rho) - \tau D_{\text{KL}}(\pi_\theta \| \bar{\pi}) = \arg\min_{\pi_\theta \in \Pi} D_{\text{KL}}(\pi_\theta \| \bar{\pi}_\tau^*).$$

Note this result holds even for the non-convex setting. Eq. (3) immediately leads to the PMD algorithm: Lift the current policy π_{θ_t} to $\bar{\pi}_\tau^*$, then perform multiple steps of gradient descent in the Project step to update $\pi_{\theta_{t+1}}$.[1]

When Π is convex, PMD converges to the optimal policy [Nemirovskii *et al.*, 1983; Beck and Teboulle, 2003]. For general Π, PMD still enjoys desirable properties.

Proposition 2. *PMD satisfies the following properties for an arbitrary parametrization* Π*.*

1. **(Monotonic Improvement)** *If the Project step* $\min_{\pi_\theta \in \Pi} D_{\text{KL}}(\pi_\theta \| \bar{\pi}_\tau^*)$ *can be globally solved, then*

$$\mathbb{E}_{\rho\sim\pi_{\theta_{t+1}}} r(\rho) - \mathbb{E}_{\rho\sim\pi_{\theta_t}} r(\rho) \geq 0.$$

2. **(Fixed Points)** *If the Project step is optimized by gradient descent, then the fixed points of PMD are stationary points of* $\mathbb{E}_{\rho\sim\pi_\theta} r(\rho)$*.*

Proposition 2 relies on the condition that the Project step in PMD is solved to global optimality. It is usually not practical to achieve such a stringent requirement when Π is not convex, limiting the applicability of Proposition 2.

Another shortcoming is that PMD typically gets trapped in poor local optima. The regularizer helps prevent large policy updates, it also tends to limit exploration. Moreover, minimizing $D_{\text{KL}}(\pi_\theta \| \bar{\pi}_\tau^*)$ is known to be *mode seeking* [Murphy, 2012], which can lead to mode collapse during learning. Once a policy has lost important modes, learning can easily become trapped at a sub-optimal policy. Unfortunately, at such points, the regularizer does not encourage further exploration.

3 Exploratory Conservative Policy Optimization

We propose two modifications to PMD that overcome the aforementioned deficiencies. These two modifications lead to our proposed algorithm, Exploratory Conservative Policy Optimization (ECPO), which retains desirable theoretical properties while achieving superior performance to PMD in practice.

The first modification is to add an additional entropy regularizer in the Lift step, to improve the exploration. The second modification is to use a reversed, *mean seeking* direction of KL divergence in the Project step. In particular, the ECPO algorithm solves the following alternating optimization problems:

$$\begin{aligned} &\textbf{(Project)} \quad \arg\min_{\pi_\theta \in \Pi} D_{\text{KL}}(\bar{\boldsymbol{\pi}}^{\boldsymbol{*}}_{\boldsymbol{\tau,\tau'}} \| \boldsymbol{\pi_\theta}), \\ &\textbf{(Lift)} \quad \text{where } \bar{\pi}^*_{\tau,\tau'} = \arg\max_{\pi \in \Delta} \mathop{\mathbb{E}}_{\rho\sim\pi} r(\rho) - \tau D_{\text{KL}}(\pi \| \pi_{\theta_t}) + \boldsymbol{\tau' \mathcal{H}(\pi)}. \end{aligned} \tag{4}$$

The effect of minimizing the other KL direction is well known [Murphy, 2012] and has proved to be effective [Norouzi

[1] To estimate this gradient one would need to use self-normalized importance sampling [Owen, 2013]. We omit the details here since PMD is not our main algorithm; similar techniques can be found in the implementation of ECPO.

Algorithm 1 The ECPO algorithm

Input: temperature parameters τ and τ', number of samples for computing gradient K

Random initialized π_θ
For $t = 1, 2, \dots$ **do**
  Set $\bar{\pi} = \pi_\theta$
  Repeat
    Sample a mini-batch of K trajectories from $\bar{\pi}$
    Compute the gradient according to Eq. (6)
    Update π_θ by gradient descent
  Until t reaches maximum of training steps
end For

et al., 2016; Nachum *et al.*, 2017a]. In particular, minimizing $D_{\mathrm{KL}}(\pi_\theta \| q)$ usually underestimates the support of q, since the objective is infinite if $q = 0$ and $\pi_\theta > 0$. Thus, π_θ is driven to 0 wherever $q = 0$. The problem is that when q changes, π_θ can have zero mass on trajectories that have non-zero probability under the new q, hence π_θ will never capture this part of q, leading to mode collapse. By contrast, minimizing $D_{\mathrm{KL}}(q \| \pi_\theta)$ is zero-avoiding in π_θ, since if $q > 0$ we must ensure $\pi_\theta > 0$. Note that by Eq. (5): (a) the q in our method is nonzero everywhere, (b) we further add entropy in Eq. (4) to avoid q prematurely converging to a deterministic policy, (c) $D_{\mathrm{KL}}(q \| \pi_\theta)$ is zero-avoiding for minimization over π_θ. These ensure that the proposed method does not exhibit the same mode-seeking behavior as MD. As we will see in Section 5, ECPO outperforms PMD significantly in experiments.

3.1 Learning Algorithms

We now provide practical learning algorithms for Eq. (4). The Lift Step has an analytic solution,

$$\bar{\pi}^*_{\tau,\tau'}(\rho) \triangleq \frac{\bar{\pi}(\rho) \exp\left\{\frac{r(\rho) - \tau' \log \bar{\pi}(\rho)}{\tau + \tau'}\right\}}{\sum_{\rho'} \bar{\pi}(\rho') \exp\left\{\frac{r(\rho') - \tau' \log \bar{\pi}(\rho')}{\tau + \tau'}\right\}}. \quad (5)$$

where we take π_{θ_t} as the reference policy $\bar{\pi}$. The Project Step in Eq. (4), $\min_{\pi_\theta \in \Pi} D_{\mathrm{KL}}(\bar{\pi}^*_{\tau,\tau'} \| \pi_\theta)$, can be optimized via stochastic gradient descent, given that one can sample trajectories from $\bar{\pi}^*_{\tau,\tau'}$. The next lemma shows that sampling from $\bar{\pi}^*_{\tau,\tau'}$ can be done using self-normalized importance sampling [Owen, 2013] when it is possible to draw multiple samples from $\bar{\pi}$, following the idea of UREX [Nachum *et al.*, 2017a].

Lemma 1. *Let* $\omega_k = \frac{r(\rho_k) - \tau' \log \bar{\pi}(\rho_k)}{\tau + \tau'}$. *Given* K i.i.d. *samples* $\{\rho_1, \dots, \rho_K\}$ *from the* reference policy $\bar{\pi}$, *we have the following unbiased gradient estimator,*

$$\nabla_\theta D_{\mathrm{KL}}(\bar{\pi}^*_{\tau,\tau'} \| \pi_\theta) \approx -\sum_{k=1}^{K} \frac{\exp\{\omega_k\}}{\sum_{j=1}^{K} \exp\{\omega_j\}} \nabla_\theta \log \pi_\theta(\rho_k), \quad (6)$$

The Pseudocode is presented in Algorithm 1. Derivation for the analytic solution of the Lift step and above Lemma as well as other implementation details can be found in the appendix.

3.2 Analysis of ECPO

We now present the theoretical analysis of ECPO. Our first result shows that, ECPO enjoys sublinear regret by a particularly designed choice of τ and τ', when the policy class is any convex subset of the probabilistic simplex, recovering the simplex setting as a special case.

Theorem 1. *When the policy class* Π *is a convex subset of the probabilistic simplex, by choosing* $\tau' = 1/\sqrt{T \log n}$, *and* $\tau + \tau' = \sqrt{T}/\sqrt{2 \log n}$, *(or* $\tau' = 1/\sqrt{t \log n}$, *and* $\tau + \tau' = \sqrt{t}/\sqrt{2 \log n}$*),* $\forall \boldsymbol{\pi} \in \Pi$,

$$\sum_{t=1}^{T} \mathop{\mathbb{E}}_{\rho \sim \pi} r(\rho) - \sum_{t=1}^{T} \mathop{\mathbb{E}}_{\rho \sim \pi_t} r(\rho) \leq 4\sqrt{T \log n}.$$

where π_t *is defined by Eq.* (5) *with* π_{t-1} *as the reference policy, and* n *is the total action/trajectory number.*

Our second result shows that ECPO enjoys similar desirable properties (Proposition 2) to PMD in general settings, with respect to the surrogate reward $\mathrm{SR}(\pi_\theta)$.

Theorem 2. *ECPO satisfies the following properties for an arbitrary parametrization* Π.

1. **(Monotonic Improvement)** *If the Project step* $D_{\mathrm{KL}}(\bar{\pi}^*_{\tau,\tau'} \| \pi_\theta)$ *can be globally solved, then*

$$\mathit{SR}(\pi_{\theta_{t+1}}) - \mathit{SR}(\pi_{\theta_t}) \geq 0,$$

where

$$\mathit{SR}(\pi) \triangleq (\tau + \tau') \log \sum_{\rho} \exp\left\{\frac{r(\rho) + \tau \log \pi(\rho)}{\tau + \tau'}\right\}. \quad (7)$$

2. **(Fixed Points)** *If the Project step is optimized by gradient descent, then the fixed points of ECPO are stationary points of* $\mathit{SR}(\pi_\theta)$.

Theorem 2 establishes desirable properties for ECPO of $\mathrm{SR}(\pi_\theta)$, but not necessarily $\mathbb{E}_{\rho \sim \pi_\theta} r(\rho)$. However, $\mathrm{SR}(\pi_\theta)$ is a reasonable surrogate that can provide good guidance for learning. By properly adjusting the two temperature parameters τ and τ', $\mathrm{SR}(\pi_\theta)$ recovers existing performance measures.

Lemma 2. *Let* $\hat{r} = r - \tau' \log \pi$, $\hat{r}_\infty = \|\hat{r}\|_\infty$ *and* $\eta = \tau + \tau'$. *For any policy* π *and* $\tau \geq 0$, $\tau' \geq 0$, *we have*

$$\mathop{\mathbb{E}}_{\rho \sim \pi} r(\rho) + \tau' \mathcal{H}(\pi) \leq \mathit{SR}(\pi) \leq \mathop{\mathbb{E}}_{\rho \sim \pi} \hat{r}(\rho) + \frac{1}{2\eta} \mathop{\mathbb{E}}_{\rho \sim \pi} \left[(\hat{r}(\rho) - \hat{r}_\infty)^2\right].$$

Furthermore,

(i) $\mathit{SR}(\pi) \to \max_\rho r(\rho)$, *as* $\tau \to 0, \tau' \to 0$.

(ii) $\mathit{SR}(\pi) \to \mathop{\mathbb{E}}_{\rho \sim \pi} r(\rho) + \tau' \mathcal{H}(\pi)$, $\tau \to \infty$.

A key question is the feasibility of solving the Project step to global optimality. For a one-layer softmax network policy, the Project step $D_{\mathrm{KL}}(\bar{\pi}^*_{\tau,\tau'} \| \pi_\theta)$ can be solved to global optimality, affording computational advantages over PMD.

Proposition 3. *Suppose* $\pi_\theta(s) = \mathrm{softmax}(\phi_s^\top \theta)$. *Given any* $\bar{\pi}$, $D_{\mathrm{KL}}(\bar{\pi} \| \pi_\theta)$ *is a convex function of* θ.

4 An Actor-Critic Extension

Finally, we develop a natural actor-critic extension of ECPO by incorporating a value function approximator. We refer to this algorithm as Exploratory Conservative Actor-Critic (ECAC).

The data efficiency of policy-based methods can be generally improved by adding a value-based critic. Given $\bar{\pi}$ and an initial state s, the objective in the Lift step of ECPO is

$$\mathcal{O}_{\text{ECPO}}(\pi, s) = \mathop{\mathbb{E}}_{\rho\sim\pi} r(\rho) - \tau D_{\text{KL}}(\pi\|\bar{\pi}) + \tau'\mathcal{H}(\pi),$$

where $\rho = (s_1 = s, a_1, s_2, a_2, \ldots)$. To incorporate value function, we need temporal consistency for this objective:

$$\begin{aligned}\mathcal{O}_{\text{ECPO}}(\pi, s) = \mathbb{E}_{a\sim\pi(\cdot|s)}[r(s,a) + \mathcal{O}_{\text{ECPO}}(\pi, s') \\ + \tau\log\bar{\pi}(a|s) - (\tau+\tau')\log\pi(a|s)].\end{aligned}$$

Denote $\bar{\pi}^*_{\tau,\tau'}(\cdot|s) \triangleq \arg\max_\pi \mathcal{O}_{\text{ECPO}}(\pi, s)$ the optimal policy on state s. Denote the soft optimal state-value function $\mathcal{O}_{\text{ECPO}}(\bar{\pi}^*_{\tau,\tau'}(\cdot|s), s)$ by $\bar{V}^*_{\tau,\tau'}(s)$, and let $\bar{Q}^*_{\tau,\tau'}(s,a) = r(s,a) + \gamma\bar{V}^*_{\tau,\tau'}(s')$ be the soft-Q function. We have,

$$\begin{aligned}\bar{V}^*_{\tau,\tau'}(s) &= (\tau+\tau')\log\sum_a \exp\left\{\frac{\bar{Q}^*_{\tau,\tau'}(s,a) + \tau\log\bar{\pi}(a|s)}{\tau+\tau'}\right\}; \\ \bar{\pi}^*_{\tau,\tau'}(a|s) &= \exp\left\{\frac{\bar{Q}^*_{\tau,\tau'}(s,a) + \tau\log\bar{\pi}(a|s) - \bar{V}^*_{\tau,\tau'}(s)}{\tau+\tau'}\right\}.\end{aligned} \tag{8}$$

We propose to train a soft state-value function V_ϕ parameterized by ϕ, a soft Q-function Q_ψ parameterized by ψ, and a policy π_θ parameterized by θ, based on Eq. (4). The update rules for these parameters can be derived as follows.

The soft state-value function approximates the soft optimal state-value $\bar{V}^*_{\tau,\tau'}$, which can be re-expressed by

$$\bar{V}^*_{\tau,\tau'}(s) = (\tau+\tau')\log\mathbb{E}_{a\sim\bar{\pi}}\left[\exp\left\{\frac{\bar{Q}^*_{\tau,\tau'}(s,a) - \tau'\log\bar{\pi}(a|s)}{\tau+\tau'}\right\}\right].$$

This suggests a Monte-Carlo estimate for $\bar{V}^*_{\tau,\tau'}(s)$: by sampling one single action a according to the reference policy $\bar{\pi}$, we have $\bar{V}^*_{\tau,\tau'}(s) \approx \bar{Q}^*_{\tau,\tau'}(s,a) - \tau'\log\bar{\pi}(a|s)$. Then, given a replay buffer $\mathcal{D}$, the soft state-value function can be trained to minimize the mean squared error,

$$L(\phi) = \mathbb{E}_{s\sim\mathcal{D}}\left[\frac{1}{2}\left(V_\phi(s) - \left[Q_\psi(s,a) - \tau'\log\bar{\pi}(a|s)\right]\right)^2\right]. \tag{9}$$

One might note that, in principle, there is no need to include a separate state-value approximation, since it can be directly computed from a soft-Q function and reference policy, using Eq. (8). However, including a separate function approximator for the state-value can help stabilize the training [Haarnoja *et al.*, 2018]. The soft Q-function parameters ψ is then trained to minimize the soft Bellman error using the state-value network,

$$L(\psi) = \mathbb{E}_{(s,a,s')\sim\mathcal{D}}\left[\frac{1}{2}\left(Q_\psi(s,a) - [r(s,a) + \gamma V_{\bar{\phi}}(s')]\right)^2\right]. \tag{10}$$

The policy parameters are updated by performing the Project Step in Eq. (4) with stochastic gradient descent,

$$L(\theta) = \mathbb{E}_{s\sim\mathcal{D}}\left[D_{\text{KL}}\left(\exp\left\{\frac{Q_\psi(s,\cdot) + \tau\log\bar{\pi}(\cdot|s) - V_\phi(s)}{\tau+\tau'}\right\}\middle\|\pi_\theta(\cdot|s)\right)\right], \tag{11}$$

where we approximate $\bar{\pi}^*_{\tau,\tau'}$ by the soft-Q and state-value function approximations.

Finally, we also use a target state-value network [Lillicrap *et al.*, 2015] and the trick of maintaining two soft-Q functions [Haarnoja *et al.*, 2018; Fujimoto *et al.*, 2018].

5 Experiments

We evaluate ECPO and ECAC on a number of benchmark tasks against strong baseline methods. Implementation details are provided in the appendix.

5.1 Settings

We first investigate the performance of ECPO on a synthetic bandit problem, which has 10000 distinct actions. The reward of each action i is initialized by $r_i = s_i^8$ such that s_i is randomly sampled from a uniform $[0, 1)$ distribution. Each action i is represented by a random feature vector $\omega_i \in \mathbb{R}^{20}$ from a standard Gaussian, and it is fixed during training. We further test ECPO on five algorithmic tasks from the OpenAI gym [Brockman *et al.*, 2016] library, in rough order of difficulty: Copy, DuplicatedInput, RepeatCopy, Reverse, and ReversedAddition [Brockman *et al.*, 2016]. Second, we test ECAC on continuous-control benchmarks from the OpenAI Gym, utilizing the MuJoCo environment [Brockman *et al.*, 2016; Todorov *et al.*, 2012]; including Hopper, Walker2d, HalfCheetah, Ant and Humanoid.

Only cumulative rewards are used in the synthetic bandit and algorithmic tasks. Therefore, value-based methods cannot be applied here, which compels us to compare ECPO against REINFORCE with entropy regularization (MENT) [Williams, 1992], and under-appreciated reward exploration (UREX) [Nachum *et al.*, 2017a], which are state-of-the-art policy-based algorithms for the algorithmic tasks. For the continuous control tasks, we compare ECAC with deep deterministic policy gradient (DDPG) [Lillicrap *et al.*, 2015], an efficient off-policy deep RL method; twin delayed deep deterministic policy gradient algorithm (TD3) [Fujimoto *et al.*, 2018], a recent extension of DDPG by using double Q-learning; and Soft-Actor-Critic (SAC) [Haarnoja *et al.*, 2018], a recent state-of-the-art off-policy algorithm on a number of benchmarks. All of these algorithms are implemented in *rlkit*.[2] We do not include TRPO and PPO in these experiments, as their performances are dominated by SAC and TD3, as shown in [Haarnoja *et al.*, 2018; Fujimoto *et al.*, 2018].

5.2 Comparative Evaluation

The results on synthetic bandit and algorithmic tasks are in Fig. 1. ECPO substantially outperforms the baselines. ECPO is able to consistently achieve a higher score substantially faster than UREX. We also find the performance of UREX is unstable. On the difficult tasks, including RepeatCopy, Reverse and ReversedAddition, UREX only finds solutions a few times out of 25 runs, which brings the overall scores down. This observation explains the gap between the results we find here and those in [Nachum *et al.*, 2017a].[3] Note that the performance of ECPO is still significantly better than UREX even compared to the results in [Nachum *et al.*, 2017a].

Fig. 2 presents the continuous control benchmarks, reporting the mean returns on evaluation rollouts obtained by the

[2] https://github.com/vitchyr/rlkit

[3] The results reported in [Nachum *et al.*, 2017a] are averaged over 5 runs of random restarting, while our results are averaged over 25 random training runs (5 runs $\times$ 5 random seed for neural network initialization).

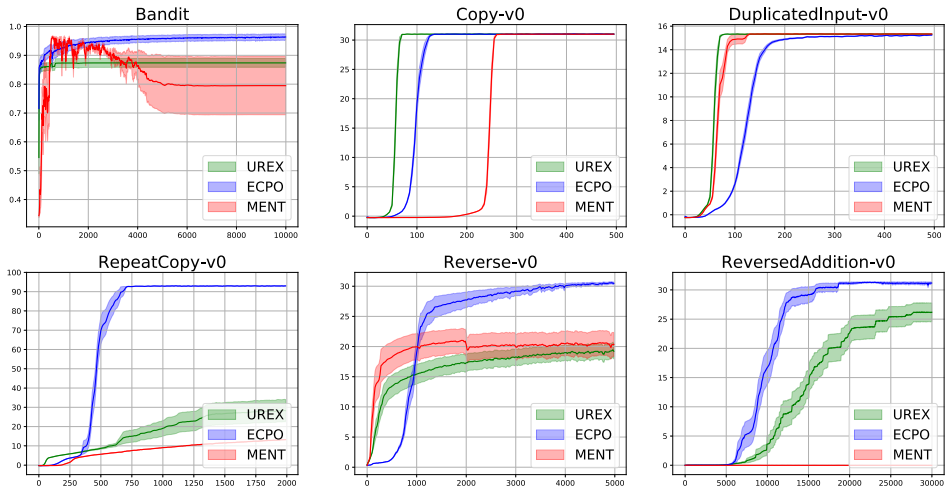

Figure 1: Results of MENT (red), UREX (green), and ECPO (blue) on synthetic bandit problem and algorithmic tasks. Plots show average reward with standard error during training. Synthetic bandit results averaged over 5 runs. Algorithmic task results averaged over 25 random training runs (5 runs $\times$ 5 random seeds for neural network initialization). The x-axis is number of sampled trajectories.

algorithms during learning. The results are averaged over five instances with different random seeds. The solid curves corresponds to the mean and the shaded region to the standard errors over the five trials. We observe that the reparameterization trick dramatically improve the performance of SAC. Therefore, to gain further clarity, we also report the result of SAC with the reparameterization trick, denoted SAC+R. The results show that ECAC matches or, in many cases, surpasses all other baseline algorithms in both final performance and sample efficiency across tasks, except compared to SAC+R in Humanoid. In Humanoid, although SAC+R outperforms ECAC, its final performance is still comparable with SAC+R.

5.3 Ablation Study

The comparative evaluations provided before suggest that our proposed algorithms outperform conventional RL methods on a number of challenging benchmarks. In this section, we further investigate how each novel component of Eq. (4) improves learning performance, by performing an ablation study on ReversedAddition and Ant. The results are presented in Fig. 3, which clearly indicate all of the three major components of Eq. (4) are helpful for achieving better performance.

Importance of entropy regularizer. The main difference between the objective in Eq. (4) and the PMD objective Eq. (3) is the entropy regularizer. We demonstrate the importance of this choice by presenting the results of ECPO and ECAC without the extra entropy regularizer, i.e. $\tau' = 0$.

Importance of KL divergence projection. Another important difference between Eq. (4) with other RL methods is to use a Project Step to update the policy, rather than one SGD. To show the importance of the Project Step, we test ECPO and ECAC without projection, which only performs one step of gradient update at each iteration of training.

Importance of direction of KL divergence. We choose PMD Eq. (3) as another baseline to prove the effectiveness of using the *mean seeking* direction of KL divergence in the project step. Similar to ECPO, we add a separate temperature parameter $\tau' > 0$ to the original objective function in Eq. (3) to encourage policy exploration, which gives $\arg\max_{\pi_\theta \in \Pi} \mathbb{E}_{\rho\sim\pi_\theta}\, r(\rho) - \tau \mathrm{KL}(\pi_\theta \| \bar{\pi}) + \tau' \mathcal{H}(\pi_\theta)$. We name it PMD+entropy. The corresponding algorithms in the actor-critic setting, named PMD-AC and PMD-AC+entropy, are also implemented for comparison.

6 Related Work

The lift-and-project approach is distinct from the previous literature on policy search, with the exception of a few recent works: Mirror Descent Guided Policy Search (MDGPS) [Montgomery and Levine, 2016], Guide Actor-Critic (GAC) [Tangkaratt *et al.*, 2017], Maxmimum aposteriori (MPO) [Abdolmaleki *et al.*, 2018], and Soft Actor-Critic (SAC) [Haarnoja *et al.*, 2018]. These approaches also adopt a mirror descent framework, but differ from the proposed approach in key aspects. MDGPS [Montgomery and Levine, 2016] follows a different learning principle, using the Lift Step to learn multiple local policies (rather than a single policy) then aligning these with a global policy in the Project Step. MDGPS does not include the entropy term in the Lift objective, which we have found to be essential for exploration. MPO [Abdolmaleki *et al.*, 2018] also neglects to add the additional entropy term. Alternatively, MPO imposes a KL constraint in its projection to avoid entropy collapse in policy update. Section 5.3 shows that entropy regularization with an appropriate annealing of τ' significantly improves learning efficiency. Both GAC and SAC use the mode seeking KL divergence in the Project Step, in opposition to the mean seeking direction we consider here [Tangkaratt *et al.*, 2017; Haarnoja *et al.*, 2018]. Additionally, SAC only uses entropy in the Lift Step, neglecting the proximal relative entropy. The benefits of regularizing with relative entropy has been discussed in TRPO [Schulman *et al.*, 2015] and MPO [Abdolmaleki *et al.*, 2018], where it is noted that proximal regulariza-

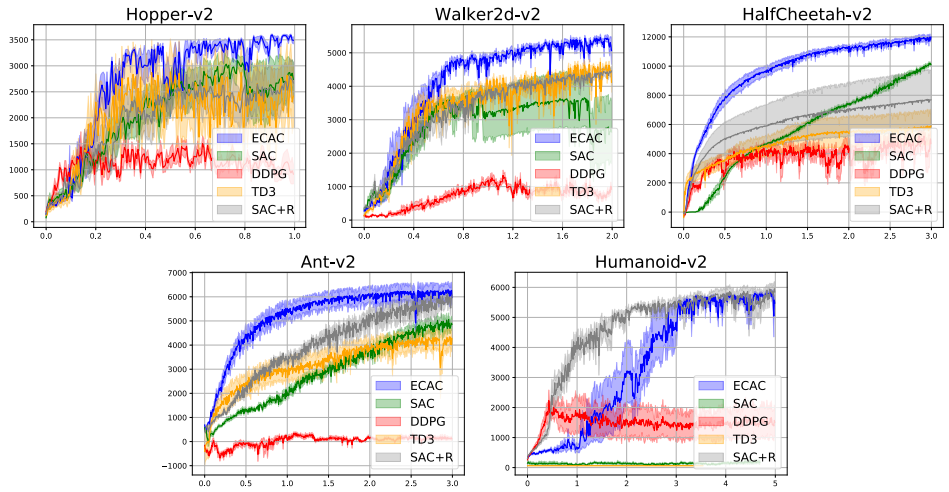

Figure 2: Learning curves of DDPG (red), TD3 (yellow), SAC (green) and ECAC (blue) on MuJoCo tasks (with SAC+R (grey) added on Humanoid). Plots show mean reward with standard error during training, averaged over five different instances with different random seeds. The x-axis is millions of environment steps.

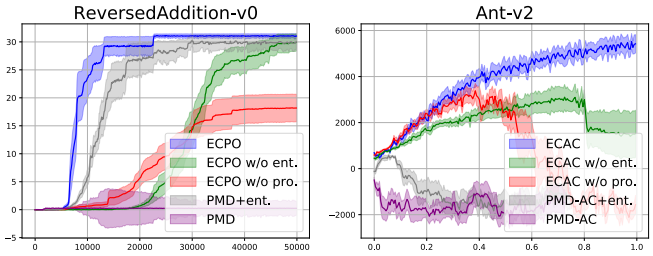

Figure 3: Ablation Study of ECPO and ECAC.

tion significantly improves learning stability. Another point is the reparameterization trick used in SAC and MPO relies on the Gaussian represetation for the continuous action space, which makes them cannot be used in discrete spaces, where our ECPO performs well. GAC seeks to match the mean of Gaussian policies under second order approximation in the Project Step, instead of directly minimizing the KL divergence with gradient descent. Although one might also attempt to interpret "one-step" methods in terms of lift-and-project, these approaches would obliviously still differ from ECPO, given that we use different directions of the KL divergence for the Lift and Project steps respectively.

TRPO and PPO have similar formulations to Eq. (2), using constraints of mean seeking KL divergence [Schulman *et al.*, 2015; Schulman *et al.*, 2017b]. Our proposed method includes additional modifications that, as shown in Section 5, significantly improve performance. UREX also uses mean seeking KL for regularization, which encourages exploration but also complicates the optimization; as shown in Section 5, UREX is significantly less efficient than the method proposed here.

Trust-PCL adopts the same objective Eq. (4), including both entropy and relative entropy regularization [Nachum *et al.*, 2017c]. However, the policy update is substantially different: while ECPO uses KL projection, Trust-PCL minimizes a path inconsistency error between the value and policy along observed trajectories [Nachum *et al.*, 2017b]. Although policy optimization by minimizing path inconsistency error can efficiently utilize off-policy data, this approach loses the desirable monotonic improvement guarantee.

7 Conclusion and Future Work

We have proposed Exploratory Conservative Policy Optimization (ECPO) as an effective new approach for policy based reinforcement learning that also guarantees monotonic improvement in a well motivated objective. We show that the resulting method achieves better exploration than both a directed exploration strategy (UREX) and undirected maximum entropy exploration (MENT). It will be interesting to further extend the follow-on ECAC actor-critic framework with further development of the value function learning approach.

Acknowledgements

Part of the work has been done when the first two authors were interns in Borealis AI Lab. We gratefully acknowledge funding from Canada's Natural Sciences and Engineering Research Council (NSERC).

References

[Abdolmaleki *et al.*, 2018] Abbas Abdolmaleki, Jost Tobias Springenberg, Yuval Tassa, Remi Munos, Nicolas Heess, and Martin Riedmiller. Maximum a posteriori policy optimisation. In *ICLR*, 2018.

[Beck and Teboulle, 2003] Amir Beck and Marc Teboulle. Mirror descent and nonlinear projected subgradient methods for convex optimization. *Operations Research Letters*, 31(3):167–175, 2003.

[Brockman *et al.*, 2016] Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang,

and Wojciech Zaremba. Openai gym. *arXiv preprint arXiv:1606.01540*, 2016.

[Daniel *et al.*, 2012] Christian Daniel, Gerhard Neumann, and Jan Peters. Hierarchical relative entropy policy search. In *Artificial Intelligence and Statistics*, pages 273–281, 2012.

[Deisenroth *et al.*, 2013] Marc Peter Deisenroth, Gerhard Neumann, Jan Peters, et al. A survey on policy search for robotics. *Foundations and Trends® in Robotics*, 2(1–2):1–142, 2013.

[Fox *et al.*, 2015] Roy Fox, Ari Pakman, and Naftali Tishby. Taming the noise in reinforcement learning via soft updates. *arXiv preprint arXiv:1512.08562*, 2015.

[Fujimoto *et al.*, 2018] Scott Fujimoto, Herke van Hoof, and Dave Meger. Addressing function approximation error in actor-critic methods. *arXiv preprint arXiv:1802.09477*, 2018.

[Haarnoja *et al.*, 2017] Tuomas Haarnoja, Haoran Tang, Pieter Abbeel, and Sergey Levine. Reinforcement learning with deep energy-based policies. *arXiv preprint arXiv:1702.08165*, 2017.

[Haarnoja *et al.*, 2018] Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. *arXiv preprint arXiv:1801.01290*, 2018.

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

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

[Mnih *et al.*, 2015] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529, 2015.

[Montgomery and Levine, 2016] William H Montgomery and Sergey Levine. Guided policy search via approximate mirror descent. In *Advances in Neural Information Processing Systems*, pages 4008–4016, 2016.

[Murphy, 2012] Kevin P Murphy. *Machine learning: a probabilistic perspective*. Cambridge, MA, 2012.

[Nachum *et al.*, 2017a] Ofir Nachum, Mohammad Norouzi, and Dale Schuurmans. Improving policy gradient by exploring under-appreciated rewards. In *ICLR*, 2017.

[Nachum *et al.*, 2017b] Ofir Nachum, Mohammad Norouzi, Kelvin Xu, and Dale Schuurmans. Bridging the gap between value and policy based reinforcement learning. In *Advances in Neural Information Processing Systems*, pages 2772–2782, 2017.

[Nachum *et al.*, 2017c] Ofir Nachum, Mohammad Norouzi, Kelvin Xu, and Dale Schuurmans. Trust-pcl: An off-policy trust region method for continuous control. In *ICLR*, 2017.

[Nemirovskii *et al.*, 1983] Arkadii Nemirovskii, David Borisovich Yudin, and Edgar Ronald Dawson. Problem complexity and method efficiency in optimization. 1983.

[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.

[Norouzi *et al.*, 2016] Mohammad Norouzi, Samy Bengio, Navdeep Jaitly, Mike Schuster, Yonghui Wu, Dale Schuurmans, et al. Reward augmented maximum likelihood for neural structured prediction. In *Advances In Neural Information Processing Systems*, pages 1723–1731, 2016.

[Owen, 2013] Art B. Owen. *Monte Carlo theory, methods and examples*. 2013.

[Peters *et al.*, 2010] Jan Peters, Katharina Mülling, and Yasemin Altun. Relative entropy policy search. In *AAAI*, pages 1607–1612. Atlanta, 2010.

[Schulman *et al.*, 2015] John Schulman, Sergey Levine, Pieter Abbeel, Michael Jordan, and Philipp Moritz. Trust region policy optimization. In *International Conference on Machine Learning*, pages 1889–1897, 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.

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

[Sutton *et al.*, 1998] Richard S Sutton, Andrew G Barto, et al. *Reinforcement learning: An introduction*. MIT press, 1998.

[Tangkaratt *et al.*, 2017] Voot Tangkaratt, Abbas Abdolmaleki, and Masashi Sugiyama. Guide actor-critic for continuous control. *arXiv preprint arXiv:1705.07606*, 2017.

[Todorov *et al.*, 2012] Emanuel Todorov, Tom Erez, and Yuval Tassa. Mujoco: A physics engine for model-based control. In *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, pages 5026–5033. IEEE, 2012.

[Van Hoof *et al.*, 2015] Herke Van Hoof, Jan Peters, and Gerhard Neumann. Learning of non-parametric control policies with high-dimensional state features. In *Artificial Intelligence and Statistics*, pages 995–1003, 2015.

[Williams and Peng, 1991] Ronald J Williams and Jing Peng. Function optimization using connectionist reinforcement learning algorithms. *Connection Science*, 3(3):241–268, 1991.

[Williams, 1992] Ronald J Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning*, 8(3-4):229–256, 1992.