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 sub-optimal 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) with respect to the underlying policy parameters [Williams and Peng, 1991; Williams, 1992; Sutton *et al.*, 1998]. Unlike standard optimization, stochastic 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,c; Tangkaratt *et al.*, 2017; 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 expected 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 the lines of Mirror Descent [Nemirovskii *et al.*, 1983; Beck and Teboulle, 2003]. Using this reformulation 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 constraint of parametrization, then approximately solves the projection by multiple gradient updates to the policy 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

[*] Equal contribution

non-convex cases, such as one-layer-softmax networks. The 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 SR(π). We further study the properties of SR(π) 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,

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

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:

$$\pi_\theta = \underset{\pi_\theta \in \Pi}{\arg\max} \underset{\rho \sim \pi_\theta}{\mathbb{E}} r(\rho) - \tau D_{\mathrm{KL}}(\pi_\theta \| \bar{\pi}). \tag{2}$$

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 this optimization is to consider an alternating lift-and-project procedure that isolates the different computational challenges.

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

Crucially, the reformulation Eq. (3) remains equivalent to the original problem Eq. (2), in that it preserves the same set of solutions, as established in Proposition 1.

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

$$\underset{\pi_\theta \in \Pi}{\arg\max} \underset{\rho \sim \pi_\theta}{\mathbb{E}} r(\rho) - \tau D_{\mathrm{KL}}(\pi_\theta \| \bar{\pi}) = \underset{\pi_\theta \in \Pi}{\arg\min} D_{\mathrm{KL}}(\pi_\theta \| \bar{\pi}_\tau^*).$$

Note this result holds even for the non-convex setting. The reformulation 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 a convex set, one can show that PMD converges to the optimal policy [Nemirovskii *et al.*, 1983; Beck and Teboulle, 2003]. The next proposition shows that for general Π, PMD still enjoys desirable properties.

Proposition 2. *Let π_{θ_t} denote the policy at step t of the update sequence. Then PMD satisfies the following properties for an arbitrary parametrization of π.*

1. **(Monotonic Improvement)** *If the Project Step $\min_{\pi_\theta \in \Pi} D_{\mathrm{KL}}(\pi_\theta \| \bar{\pi}_\tau^*)$ 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 the stationary points of $\mathbb{E}_{\rho \sim \pi_\theta} r(\rho)$.*

Despite these desirable properties, 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_θ is not convex in θ, limiting the applicability of Proposition 2.

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

3 Exploratory Conservative Policy Optimization

We now propose two modifications to PMD that overcome its 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 to the Lift Step, to improve the exploration behavior. The second modification is to use a reversed, *mean seeking* direction of the KL divergence in the Project Step. In particular, the ECPO algorithm solves the following alternating optimization problems to update the policy $\pi_{\theta_{t+1}}$ at each iteration:

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

[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 $\tau$ and $\tau'$, number of samples
    for computing gradient $K$
Random initialized $\pi_\theta$
For $t = 1, 2, \ldots$ 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 $\pi_\theta$ by gradient descent
    Until $t$ reaches maximum of training steps
end For
```

The effect of minimizing different KL direction is well known [Murphy, 2012] and has proved to be effective [Norouzi *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 nonzero 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* $\{\rho_1, \ldots, \rho_K\}$ *from the* reference policy $\bar{\pi}$, *we have the following unbiased gradient estimator,*

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

Pesudeocode of the learning algorithm 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, with the additional entropy regularizer to the Lift Step $\tau' > 0$, the policy $\bar{\pi}^*_{\tau,\tau'}$ still enjoys sublinear regret by 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 still enjoys similar desirable properties (Proposition 2) to PMD in general settings, with respect to the surrogate reward $\mathrm{SR}(\pi_\theta)$.

Theorem 2. *Let* π_{θ_t} *denote the policy at step* t *of the update sequence. ECPO satisfies the following properties for an arbitrary parametrization of* π.

1. **(Monotonic Improvement)** *If the Project Step* $D_{\mathrm{KL}}(\bar{\pi}^*_{\tau,\tau'} \| \pi_\theta)$ *can be globally solved, then*
$$SR(\pi_{\theta_{t+1}}) - SR(\pi_{\theta_t}) \geq 0,$$
where
$$SR(\pi) \triangleq (\tau + \tau') \log \sum_{\rho} \exp\left\{\frac{r(\rho) + \tau \log \pi(\rho)}{\tau + \tau'}\right\}. \quad (7)$$
2. **(Fixed Points)** *If the Project Step is optimized by gradient descent, then the fixed points of ECPO are the stationary points of* $SR(\pi_\theta)$.

Theorem 2 only establishes desirable properties for ECPO with respect to $\mathrm{SR}(\pi_\theta)$, but not necessarily to $\mathbb{E}_{\rho \sim \pi_\theta} r(\rho)$. However, we can provide theoretical and empirical evidence that $\mathrm{SR}(\pi_\theta)$ is a reasonable surrogate that can provide good guidance for learning. In fact, by properly adjusting the two temperature parameters τ and τ', the resulting surrogate objective $\mathrm{SR}(\pi_\theta)$ 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*

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

Furthermore,

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

(ii) $SR(\pi) \to \mathop{\mathbb{E}}_{\rho \sim \pi} r(\rho) + \tau' \mathcal{H}(\pi)$, $\tau \to \infty$. *Further,* $SR(\pi) \to \mathop{\mathbb{E}}_{\rho \sim \pi} r(\rho)$, *as* $\tau \to \infty$, $\tau' \to 0$.

A key question is the feasibility of solving the Project Step to global optimality. As shown in Proposition 3, for a one-layer-softmax neural network policy, the Project Step $D_{\mathrm{KL}}(\bar{\pi}^*_{\tau,\tau'} \| \pi_\theta)$ can also still be solved to global optimality, affording computational advantages over PMD.

Proposition 3. *Assume* $\pi_\theta(s) = \mathrm{softmax}(\phi_s^\top \theta)$. *Given any* $\bar{\pi}$, *the Projection Step* $\min_{\theta \in \mathbb{R}^d} D_{\mathrm{KL}}(\bar{\pi} \| \pi_\theta)$ *is a convex optimization problem in* θ.

4 An Actor-Critic Extension

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

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

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

where $\rho = (s_1 = s, a_1, s_2, a_2, \ldots)$. To incorporate value function approximation, we need derive the temporal consistency structure for this objective. First, write

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

Let $\bar{\pi}^*_{\tau,\tau'}(\cdot|s) = \arg\max_\pi \mathcal{O}_{\text{ECPO}}(\pi, s)$ denote the optimal policy on state s. Further 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. It can then be verified that

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

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,

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

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,

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

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

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

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]. Implementation details for ECAC are given in Appendix.

5 Experiments

We evaluate the main proposed methods, 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 and the algorithmic task suite from the OpenAI gym [Brockman *et al.*, 2016]. The synthetic multi-armed bandit problem has 10000 distinct actions, where 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 randomly sampled feature vector $\omega_i \in \mathbb{R}^{20}$ from standard normal distribution. Note that these features are fixed during training. We further test our method on five algorithmic tasks from the OpenAI gym library, in rough order of difficulty: Copy, DuplicatedInput, RepeatCopy, Reverse, and ReversedAddition [Brockman *et al.*, 2016]. Second, we test the second ECAC approach on continuous-control benchmarks from OpenAI Gym, utilizing the Mujoco environment [Brockman *et al.*, 2016; Todorov *et al.*, 2012]; including Hopper, Walker2d, HalfCheetah, Ant and Humanoid. The details of the algorithmic and Mujoco tasks are provided in the appendix.

Note that only cumulative rewards are available 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

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

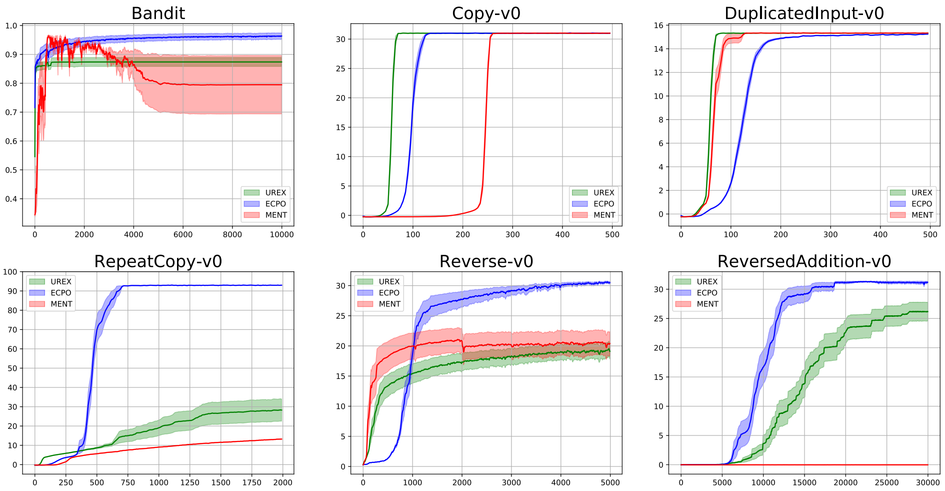

Figure 1: Results of MENT (red), UREX (green), and ECPO (blue) on synthetic bandit problem and algorithmic tasks. Plots show average reward with standard error during training. Synthetic bandit results averaged over 5 runs. Algorithmic task results averaged over 25 random training runs (5 runs × 5 random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

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 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

[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 × 5 random seed for neural network initialization).

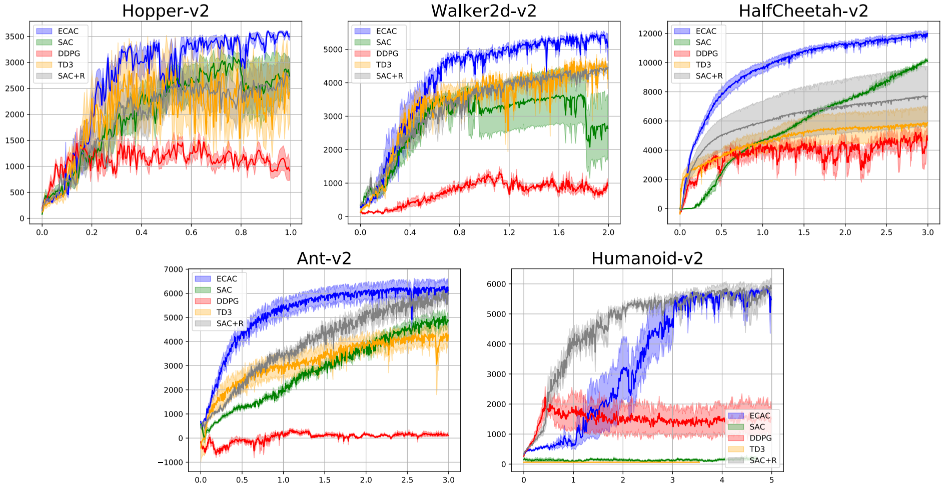

Figure 2: Learning curves of DDPG (red), TD3 (yellow), SAC (green) and ECAC (blue) on Mujoco tasks (with SAC+R (gray) added on Humanoid). Plots show mean reward with standard error during training, averaged over five different instances with different random seeds. X-axis is millions of environment steps.

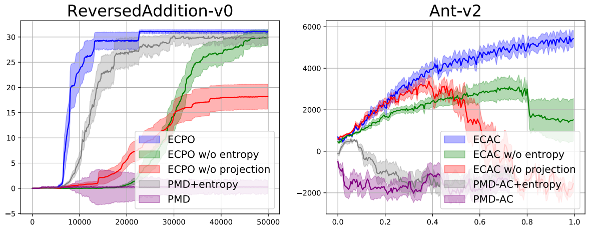

Figure 3: Ablation Study of ECPO and ECAC.

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 regularization 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 also have similar formulations to Eq. (2), using constrained versions with mean seeking KL divergence Schulman *et al.* [2015, 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

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

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

Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. Openai gym. *arXiv:1606.01540*, 2016.

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

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.

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

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

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

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.

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

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.

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.

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

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

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

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.

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

Arkadii Nemirovskii, David Borisovich Yudin, and Edgar Ronald Dawson. Problem complexity and method efficiency in optimization. 1983.

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

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.

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

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

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.

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

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

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.

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

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

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.

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.

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

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

A Proofs

A.1 Proof of Proposition 1

Proof. Note that

$$\begin{aligned}-\tau D_{\mathrm{KL}}(\pi_\theta \| \bar{\pi}^*_\tau) &= -\tau \sum_\rho \pi_\theta(\rho) \log \pi_\theta(\rho) + \tau \sum_\rho \pi_\theta(\rho)(\log \bar{\pi}(\rho) + r(\rho)/\tau) - Z_{\bar{\pi}} \\ &= \mathop{\mathbb{E}}_{\rho \sim \pi_\theta} r(\rho) - \tau D_{\mathrm{KL}}(\pi_\theta \| \bar{\pi}) - Z_{\bar{\pi}},\end{aligned}$$

where $Z_{\bar{\pi}} \triangleq \tau \log \sum_\rho \bar{\pi}(\rho) \exp\{r(\rho)/\tau\}$ is indenpendent of π_θ given the reference policy $\bar{\pi}$. □

A.2 Proof of Proposition 2

Proof. **(Monotonic Improvement)** By the definition of $\pi_{\theta_{t+1}}$, $D_{\mathrm{KL}}(\pi_{\theta_{t+1}} \| \bar{\pi}^*_\tau) = \min_{\pi_\theta \in \Pi} D_{\mathrm{KL}}(\pi_\theta \| \bar{\pi}^*_\tau) \leq D_{\mathrm{KL}}(\pi_{\theta_t} \| \bar{\pi}^*_\tau)$.

$$\begin{aligned}&\mathop{\mathbb{E}}_{\rho \sim \pi_{\theta_{t+1}}} r(\rho) - \mathop{\mathbb{E}}_{\rho \sim \pi_{\theta_t}} r(\rho) = \tau \left[\sum_\rho \pi_{\theta_{t+1}}(\rho) \cdot \frac{r(\rho)}{\tau} - \sum_\rho \pi_{\theta_t}(\rho) \cdot \frac{r(\rho)}{\tau} \right] \\ &= \tau \sum_\rho \left[\pi_{\theta_{t+1}}(\rho) \log \left\{ \frac{\pi_{\theta_t}(\rho) \exp\left\{\frac{r(\rho)}{\tau}\right\}}{\sum\limits_{\rho'} \pi_{\theta_t}(\rho') \exp\left\{\frac{r(\rho')}{\tau}\right\}} \right\} - \pi_{\theta_{t+1}}(\rho) \log \pi_{\theta_t}(\rho) - \pi_{\theta_t}(\rho) \log \left\{ \frac{\pi_{\theta_t}(\rho) \exp\left\{\frac{r(\rho)}{\tau}\right\}}{\sum\limits_{\rho'} \pi_{\theta_t}(\rho') \exp\left\{\frac{r(\rho')}{\tau}\right\}} \right\} + \pi_{\theta_t}(\rho) \log \pi_{\theta_t}(\rho) \right] \\ &= \tau \sum_\rho \left[\pi_{\theta_{t+1}}(\rho) \log \bar{\pi}^*_\tau(\rho) - \pi_{\theta_{t+1}}(\rho) \log \pi_{\theta_t}(\rho) - \pi_{\theta_t}(\rho) \log \bar{\pi}^*_\tau(\rho) + \pi_{\theta_t}(\rho) \log \pi_{\theta_t}(\rho) \right] \\ &= \tau \left[D_{\mathrm{KL}}(\pi_{\theta_t} \| \bar{\pi}^*_\tau) - D_{\mathrm{KL}}(\pi_{\theta_{t+1}} \| \bar{\pi}^*_\tau) + D_{\mathrm{KL}}(\pi_{\theta_{t+1}} \| \pi_{\theta_t}) \right] \geq \tau D_{\mathrm{KL}}(\pi_{\theta_{t+1}} \| \pi_{\theta_t}) \geq 0.\end{aligned}$$

(Fixed Points) The stationary point of $\mathbb{E}_{\rho \sim \pi_\theta} r(\rho)$ is the π_θ which satisfies,

$$\begin{aligned}&\sum_\rho r(\rho) \cdot \frac{d\pi_\theta(\rho)}{d\theta} = 0 \\ \Longleftrightarrow\ &\sum_\rho \left[\log \pi_\theta(\rho) - \log \pi_\theta(\rho) - \frac{r(\rho)}{\tau} \right] \cdot \frac{d\pi_\theta(\rho)}{d\theta} = 0 \qquad (\tau > 0) \\ \Longleftrightarrow\ &\sum_\rho \left[\log \pi_\theta(\rho) - \log\{\pi_\theta(\rho) \exp\{r(\rho)/\tau\}\} \right] \cdot \frac{d\pi_\theta(\rho)}{d\theta} = 0.\end{aligned} \tag{12}$$

On the other hand, the fixed point of PMD indicates at some iteration t,

$$\begin{aligned}&\pi_{\theta_t} = \pi_{\theta_{t+1}}, \\ &\text{where } \pi_{\theta_t} \xrightarrow{\text{Lift Step}} \bar{\pi}^*_\tau \xrightarrow{\text{Project Step}} \pi_{\theta_{t+1}} \text{ in Eq. (3)},\end{aligned} \tag{13}$$

which means π_{θ_t} is the solution of the Project Step,

$$\begin{aligned}&\left.\frac{dD_{\mathrm{KL}}(\pi_\theta \| \bar{\pi}^*_\tau)}{d\theta}\right|_{\theta=\theta_t} = 0 \\ \Longleftrightarrow\ &\sum_\rho \left[\log \pi_\theta(\rho) - \log \bar{\pi}^*_\tau(\rho) + 1 \right] \cdot \left.\frac{d\pi_\theta(\rho)}{d\theta}\right|_{\theta=\theta_t} = 0 \\ \Longleftrightarrow\ &\sum_\rho \left[\log \pi_\theta(\rho) - \log \left\{ \frac{\pi_{\theta_t}(\rho) \exp\{r(\rho)/\tau\}}{\sum\limits_{\rho'} \pi_{\theta_t}(\rho') \exp\{r(\rho')/\tau\}} \right\} + 1 \right] \cdot \left.\frac{d\pi_\theta(\rho)}{d\theta}\right|_{\theta=\theta_t} = 0 \\ &\text{(by Lift Step in Eq. (13))} \\ \Longleftrightarrow\ &\sum_\rho \left[\log \pi_\theta(\rho) - \log\{\pi_{\theta_t}(\rho) \exp\{r(\rho)/\tau\}\} + c \right] \cdot \left.\frac{d\pi_\theta(\rho)}{d\theta}\right|_{\theta=\theta_t} = 0,\end{aligned} \tag{14}$$

where we denote $c = 1 + \log \sum\limits_{\rho'} \pi_{\theta_t}(\rho') \exp\left\{r(\rho')/\tau\right\}$. Note that for c independent of ρ, we have,

$$\sum_{\rho} c \cdot \left.\frac{d\pi_\theta(\rho)}{d\theta}\right|_{\theta=\theta_t} = c \cdot \left.\frac{d\sum_\rho \pi_\theta(\rho)}{d\theta}\right|_{\theta=\theta_t} = c \cdot \left.\frac{d1}{d\theta}\right|_{\theta=\theta_t} = 0.$$

Therefore, Eq. (14) is equivalent with,

$$\iff \sum_{\rho} \left[\log \pi_\theta(\rho) - \log\left\{\pi_{\theta_t}(\rho) \exp\left\{r(\rho)/\tau\right\}\right\}\right] \cdot \left.\frac{d\pi_\theta(\rho)}{d\theta}\right|_{\theta=\theta_t} = 0. \tag{15}$$

Comparing Eq. (15) with Eq. (12), we have the fixed point condition of PMD is the same as the definition of the stationary point of $\mathbb{E}_{\rho\sim\pi_\theta} r(\rho)$. □

A.3 Proof of Theorem 1

Suppose Π is a convex subset of the probabilistic simplex, the policy update is as follows,

$$\bar{\pi}_{t+1}(\rho) = \frac{\pi_t(\rho) \cdot \exp\left\{\frac{r(\rho) - \tau' \log \pi_t(\rho)}{\tau + \tau'}\right\}}{\sum\limits_{\rho} \pi_t(\rho) \cdot \exp\left\{\frac{r(\rho) - \tau' \log \pi_t(\rho)}{\tau + \tau'}\right\}}, \quad \forall \rho,$$

$$\boldsymbol{\pi}_{t+1} = \operatorname*{arg\,min}_{\boldsymbol{\pi}\in\Pi} D_{\mathrm{KL}}\left(\bar{\boldsymbol{\pi}}_{t+1} \| \boldsymbol{\pi}\right).$$

We firstly prove several useful lemmas. For conciseness, we use vector notations.

Lemma 3. $\forall \boldsymbol{\pi} \in \Pi$, $D_{\mathrm{KL}}\left(\boldsymbol{\pi} \| \boldsymbol{\pi}_{t+1}\right) \le D_{\mathrm{KL}}\left(\boldsymbol{\pi} \| \bar{\boldsymbol{\pi}}_{t+1}\right)$, $\forall t \ge 0$.

Proof. Since the KL divergence is convex, and Π is convex, according to the first order optimality condition,

$$\left(-\frac{\bar{\boldsymbol{\pi}}_{t+1}}{\boldsymbol{\pi}_{t+1}}\right)^\top \left(\boldsymbol{\pi}_{t+1} - \boldsymbol{\pi}\right) = \boldsymbol{\pi}^\top \left(\frac{\bar{\boldsymbol{\pi}}_{t+1}}{\boldsymbol{\pi}_{t+1}} - \mathbf{1}\right) \le 0.$$

On the other hand, by $\log x \le x - 1, \forall x \in \mathbb{R}$,

$$D_{\mathrm{KL}}\left(\boldsymbol{\pi} \| \boldsymbol{\pi}_{t+1}\right) - D_{\mathrm{KL}}\left(\boldsymbol{\pi} \| \bar{\boldsymbol{\pi}}_{t+1}\right) = \boldsymbol{\pi}^\top \left(\log \bar{\boldsymbol{\pi}}_{t+1} - \log \boldsymbol{\pi}_{t+1}\right) \le \boldsymbol{\pi}^\top \left(\frac{\bar{\boldsymbol{\pi}}_{t+1}}{\boldsymbol{\pi}_{t+1}} - \mathbf{1}\right).$$

Combining the two inequalities above completes the proof. □

Lemma 4.

$$D_{\mathrm{KL}}(\boldsymbol{\pi}_t \| \bar{\boldsymbol{\pi}}_{t+1}) \le \frac{1}{2\left(\tau + \tau'\right)^2} + \frac{\tau'}{\tau + \tau'} \left(\boldsymbol{\pi}_t - \bar{\boldsymbol{\pi}}_{t+1}\right)^\top \log \boldsymbol{\pi}_t.$$

Proof. By the ℓ_1 norm strongly convex property of the negative entropy, the assumption that $\|\mathbf{r}\|_\infty \le 1$ without loss of generality, and $ax - bx^2 \le \frac{a^2}{4b}, \forall a, b > 0$, we have,

$$\begin{aligned}
D_{\mathrm{KL}}(\boldsymbol{\pi}_t \| \bar{\boldsymbol{\pi}}_{t+1}) &= \boldsymbol{\pi}_t^\top \log \boldsymbol{\pi}_t - \bar{\boldsymbol{\pi}}_{t+1}^\top \log \bar{\boldsymbol{\pi}}_{t+1} - \left(\boldsymbol{\pi}_t - \bar{\boldsymbol{\pi}}_{t+1}\right)^\top \log \bar{\boldsymbol{\pi}}_{t+1} \\
&\le \left(\boldsymbol{\pi}_t - \bar{\boldsymbol{\pi}}_{t+1}\right)^\top \log \boldsymbol{\pi}_t - \frac{1}{2}\left\|\boldsymbol{\pi}_t - \bar{\boldsymbol{\pi}}_{t+1}\right\|_1^2 - \left(\boldsymbol{\pi}_t - \bar{\boldsymbol{\pi}}_{t+1}\right)^\top \log \bar{\boldsymbol{\pi}}_{t+1} \\
&= \left(\boldsymbol{\pi}_t - \bar{\boldsymbol{\pi}}_{t+1}\right)^\top \left(\log \boldsymbol{\pi}_t - \log \bar{\boldsymbol{\pi}}_{t+1}\right) - \frac{1}{2}\left\|\boldsymbol{\pi}_t - \bar{\boldsymbol{\pi}}_{t+1}\right\|_1^2 \\
&= \frac{1}{\tau + \tau'} \left(\bar{\boldsymbol{\pi}}_{t+1} - \boldsymbol{\pi}_t\right)^\top \mathbf{r} + \frac{\tau'}{\tau + \tau'} \left(\boldsymbol{\pi}_t - \bar{\boldsymbol{\pi}}_{t+1}\right)^\top \log \boldsymbol{\pi}_t - \frac{1}{2}\left\|\boldsymbol{\pi}_t - \bar{\boldsymbol{\pi}}_{t+1}\right\|_1^2 \\
&\le \frac{1}{\tau + \tau'} \left\|\bar{\boldsymbol{\pi}}_{t+1} - \boldsymbol{\pi}_t\right\|_1 \cdot \|\mathbf{r}\|_\infty - \frac{1}{2}\left\|\boldsymbol{\pi}_t - \bar{\boldsymbol{\pi}}_{t+1}\right\|_1^2 + \frac{\tau'}{\tau + \tau'} \left(\boldsymbol{\pi}_t - \bar{\boldsymbol{\pi}}_{t+1}\right)^\top \log \boldsymbol{\pi}_t \\
&\le \frac{1}{\tau + \tau'} \left\|\bar{\boldsymbol{\pi}}_{t+1} - \boldsymbol{\pi}_t\right\|_1 - \frac{1}{2}\left\|\boldsymbol{\pi}_t - \bar{\boldsymbol{\pi}}_{t+1}\right\|_1^2 + \frac{\tau'}{\tau + \tau'} \left(\boldsymbol{\pi}_t - \bar{\boldsymbol{\pi}}_{t+1}\right)^\top \log \boldsymbol{\pi}_t \\
&\le \frac{1}{2\left(\tau + \tau'\right)^2} + \frac{\tau'}{\tau + \tau'} \left(\boldsymbol{\pi}_t - \bar{\boldsymbol{\pi}}_{t+1}\right)^\top \log \boldsymbol{\pi}_t.
\end{aligned}$$

□

Lemma 5.

$$D_{\mathrm{KL}}(\bar{\boldsymbol{\pi}}_{t+1}\|\boldsymbol{\pi}_t) \leq \frac{1}{2\tau\left(\tau+\tau'\right)} + \frac{\tau'}{\tau}\log n.$$

Proof.

$$\begin{aligned}
D_{\mathrm{KL}}(\bar{\boldsymbol{\pi}}_{t+1}\|\boldsymbol{\pi}_t) &= \bar{\boldsymbol{\pi}}_{t+1}^\top \log\bar{\boldsymbol{\pi}}_{t+1} - \boldsymbol{\pi}_t^\top \log\boldsymbol{\pi}_t - \left(\bar{\boldsymbol{\pi}}_{t+1}-\boldsymbol{\pi}_t\right)^\top \log\boldsymbol{\pi}_t \\
&\leq \left(\bar{\boldsymbol{\pi}}_{t+1}-\boldsymbol{\pi}_t\right)^\top \log\bar{\boldsymbol{\pi}}_{t+1} - \frac{1}{2}\left\|\boldsymbol{\pi}_t-\bar{\boldsymbol{\pi}}_{t+1}\right\|_1^2 - \left(\bar{\boldsymbol{\pi}}_{t+1}-\boldsymbol{\pi}_t\right)^\top \log\boldsymbol{\pi}_t \\
&= \left(\bar{\boldsymbol{\pi}}_{t+1}-\boldsymbol{\pi}_t\right)^\top \left(\log\bar{\boldsymbol{\pi}}_{t+1}-\log\boldsymbol{\pi}_t\right) - \frac{1}{2}\left\|\boldsymbol{\pi}_t-\bar{\boldsymbol{\pi}}_{t+1}\right\|_1^2 \\
&= \frac{1}{\tau+\tau'}\left(\bar{\boldsymbol{\pi}}_{t+1}-\boldsymbol{\pi}_t\right)^\top \mathbf{r} + \frac{\tau'}{\tau+\tau'}\left(\boldsymbol{\pi}_t-\bar{\boldsymbol{\pi}}_{t+1}\right)^\top \log\boldsymbol{\pi}_t - \frac{1}{2}\left\|\boldsymbol{\pi}_t-\bar{\boldsymbol{\pi}}_{t+1}\right\|_1^2 \\
&\leq \frac{1}{\tau+\tau'}\left\|\bar{\boldsymbol{\pi}}_{t+1}-\boldsymbol{\pi}_t\right\|_1 \cdot \left\|\mathbf{r}\right\|_\infty - \frac{1}{2}\left\|\boldsymbol{\pi}_t-\bar{\boldsymbol{\pi}}_{t+1}\right\|_1^2 + \frac{\tau'}{\tau+\tau'}\left(\boldsymbol{\pi}_t-\bar{\boldsymbol{\pi}}_{t+1}\right)^\top \log\boldsymbol{\pi}_t \\
&\leq \frac{1}{\tau+\tau'}\left\|\bar{\boldsymbol{\pi}}_{t+1}-\boldsymbol{\pi}_t\right\|_1 - \frac{1}{2}\left\|\boldsymbol{\pi}_t-\bar{\boldsymbol{\pi}}_{t+1}\right\|_1^2 + \frac{\tau'}{\tau+\tau'}\left(\boldsymbol{\pi}_t-\bar{\boldsymbol{\pi}}_{t+1}\right)^\top \log\boldsymbol{\pi}_t \\
&\leq \frac{1}{2\left(\tau+\tau'\right)^2} + \frac{\tau'}{\tau+\tau'}\left(\boldsymbol{\pi}_t-\bar{\boldsymbol{\pi}}_{t+1}\right)^\top \log\boldsymbol{\pi}_t.
\end{aligned}$$

Therefore,

$$\begin{aligned}
D_{\mathrm{KL}}(\bar{\boldsymbol{\pi}}_{t+1}\|\boldsymbol{\pi}_t) &\leq \frac{1}{2\left(\tau+\tau'\right)^2} + \frac{\tau'}{\tau+\tau'}\left(\boldsymbol{\pi}_t-\bar{\boldsymbol{\pi}}_{t+1}\right)^\top \log\boldsymbol{\pi}_t \\
&= \frac{1}{2\left(\tau+\tau'\right)^2} + \frac{\tau'}{\tau+\tau'}\left(D_{\mathrm{KL}}(\bar{\boldsymbol{\pi}}_{t+1}\|\boldsymbol{\pi}_t) - \bar{\boldsymbol{\pi}}_{t+1}^\top \log\bar{\boldsymbol{\pi}}_{t+1} + \boldsymbol{\pi}_t^\top \log\boldsymbol{\pi}_t\right).
\end{aligned}$$

Rearranging,

$$D_{\mathrm{KL}}(\bar{\boldsymbol{\pi}}_{t+1}\|\boldsymbol{\pi}_t) \leq \frac{1}{2\tau\left(\tau+\tau'\right)} + \frac{\tau'}{\tau}\left(\boldsymbol{\pi}_t^\top \log\boldsymbol{\pi}_t - \bar{\boldsymbol{\pi}}_{t+1}^\top \log\bar{\boldsymbol{\pi}}_{t+1}\right) \leq \frac{1}{2\tau\left(\tau+\tau'\right)} + \frac{\tau'}{\tau}\log n. \qquad \square$$

Now we prove Theorem 1,

Proof. According to Lemma 3 and Lemma 4, $\forall \boldsymbol{\pi} \in \Pi$,

$$\begin{aligned}
\left(\boldsymbol{\pi}-\boldsymbol{\pi}_t\right)^\top \mathbf{r} &= \left(\boldsymbol{\pi}-\boldsymbol{\pi}_t\right)^\top \left[\left(\tau+\tau'\right)\left(\log\bar{\boldsymbol{\pi}}_{t+1}-\log\boldsymbol{\pi}_t\right) + \tau'\log\boldsymbol{\pi}_t\right] \\
&= \left(\tau+\tau'\right)\left[D_{\mathrm{KL}}(\boldsymbol{\pi}\|\boldsymbol{\pi}_t) - D_{\mathrm{KL}}(\boldsymbol{\pi}\|\bar{\boldsymbol{\pi}}_{t+1}) + D_{\mathrm{KL}}(\boldsymbol{\pi}_t\|\bar{\boldsymbol{\pi}}_{t+1})\right] + \tau'\left(\boldsymbol{\pi}-\boldsymbol{\pi}_t\right)^\top \log\boldsymbol{\pi}_t \\
&\leq \left(\tau+\tau'\right)\left[D_{\mathrm{KL}}(\boldsymbol{\pi}\|\boldsymbol{\pi}_t) - D_{\mathrm{KL}}(\boldsymbol{\pi}\|\boldsymbol{\pi}_{t+1}) + D_{\mathrm{KL}}(\boldsymbol{\pi}_t\|\bar{\boldsymbol{\pi}}_{t+1})\right] + \tau'\left(\boldsymbol{\pi}-\boldsymbol{\pi}_t\right)^\top \log\boldsymbol{\pi}_t \\
&\leq \left(\tau+\tau'\right)\left[D_{\mathrm{KL}}(\boldsymbol{\pi}\|\boldsymbol{\pi}_t) - D_{\mathrm{KL}}(\boldsymbol{\pi}\|\boldsymbol{\pi}_{t+1})\right] + \frac{1}{2\left(\tau+\tau'\right)} + \tau'\left(\boldsymbol{\pi}-\bar{\boldsymbol{\pi}}_{t+1}\right)^\top \log\boldsymbol{\pi}_t.
\end{aligned}$$

According to Lemma 5,

$$\begin{aligned}
\left(\boldsymbol{\pi}-\bar{\boldsymbol{\pi}}_{t+1}\right)^\top \log\boldsymbol{\pi}_t &= -D_{\mathrm{KL}}(\boldsymbol{\pi}\|\boldsymbol{\pi}_t) + \boldsymbol{\pi}^\top \log\boldsymbol{\pi} + D_{\mathrm{KL}}(\bar{\boldsymbol{\pi}}_{t+1}\|\boldsymbol{\pi}_t) - \bar{\boldsymbol{\pi}}_{t+1}^\top \log\bar{\boldsymbol{\pi}}_{t+1} \\
&\leq D_{\mathrm{KL}}(\bar{\boldsymbol{\pi}}_{t+1}\|\boldsymbol{\pi}_t) - \bar{\boldsymbol{\pi}}_{t+1}^\top \log\bar{\boldsymbol{\pi}}_{t+1} \\
&\leq \frac{1}{2\tau\left(\tau+\tau'\right)} + \frac{\tau+\tau'}{\tau}\log n.
\end{aligned}$$

Combining the above,

$$\left(\boldsymbol{\pi}-\boldsymbol{\pi}_t\right)^\top \mathbf{r} \leq \left(\tau+\tau'\right)\left[D_{\mathrm{KL}}(\boldsymbol{\pi}\|\boldsymbol{\pi}_t) - D_{\mathrm{KL}}(\boldsymbol{\pi}\|\boldsymbol{\pi}_{t+1})\right] + \frac{1}{2\left(\tau+\tau'\right)} + \frac{\tau'}{2\tau\left(\tau+\tau'\right)} + \frac{\tau'\left(\tau+\tau'\right)}{\tau}\log n.$$

Summing up from $t=1$ to T, and choosing $\tau+\tau' = \frac{\sqrt{T}}{\sqrt{2\log n}}$, $\tau' = \frac{1}{\sqrt{T\log n}}$,

$$\begin{aligned}
\sum_{t=1}^{T}\left(\boldsymbol{\pi}-\boldsymbol{\pi}_t\right)^\top \mathbf{r} &\leq \left(\tau+\tau'\right)D_{\mathrm{KL}}(\boldsymbol{\pi}\|\boldsymbol{\pi}_1) + \frac{T}{2\left(\tau+\tau'\right)} + \frac{T\tau'}{2\tau\left(\tau+\tau'\right)} + \frac{T\tau'\left(\tau+\tau'\right)}{\tau}\log n \\
&\leq \sqrt{2T\log n} + \frac{1}{\frac{\sqrt{T}}{\sqrt{2\log n}} - \frac{1}{\sqrt{T\log n}}}\left(\frac{1}{\sqrt{2}}+1\right) + \sqrt{T\log n},
\end{aligned}$$

by $D_{\mathrm{KL}}(\boldsymbol{\pi}\|\boldsymbol{\pi}_1) \leq \log n$. Note that choosing $\tau+\tau' = \frac{\sqrt{t}}{\sqrt{2\log n}}$, $\tau' = \frac{1}{\sqrt{t\log n}}$ will lead to the same result. □

A.4 Proof of Theorem 2

Proof. **(Monotonic Improvement)** Using $D_{\mathrm{KL}}(\bar{\pi}^*_{\tau,\tau'}\|\pi_{\theta_{t+1}}) = \min_{\pi_\theta\in\Pi} D_{\mathrm{KL}}(\bar{\pi}^*_{\tau,\tau'}\|\pi_\theta) \le D_{\mathrm{KL}}(\bar{\pi}^*_{\tau,\tau'}\|\pi_{\theta_t})$ and Jensen's inequality,

$$
\begin{aligned}
\mathrm{SR}(\pi_{\theta_{t+1}}) - \mathrm{SR}(\pi_{\theta_t}) &= (\tau+\tau')\log\sum_\rho \frac{\exp\left\{\frac{r(\rho)+\tau\log\pi_{\theta_{t+1}}(\rho)}{\tau+\tau'}\right\}}{\sum\limits_\rho \exp\left\{\frac{r(\rho)+\tau\log\pi_{\theta_t}(\rho)}{\tau+\tau'}\right\}} \\
&= (\tau+\tau')\log\sum_\rho \frac{\exp\left\{\frac{r(\rho)+\tau\log\pi_{\theta_t}(\rho)}{\tau+\tau'}\right\}}{\sum\limits_\rho \exp\left\{\frac{r(\rho)+\tau\log\pi_{\theta_t}(\rho)}{\tau+\tau'}\right\}} \cdot \exp\left\{\frac{\tau\log\pi_{\theta_{t+1}}(\rho) - \tau\log\pi_{\theta_t}(\rho)}{\tau+\tau'}\right\} \\
&= (\tau+\tau')\log\sum_\rho \bar{\pi}^*_{\tau,\tau'}(\rho)\cdot\exp\left\{\frac{\tau\log\pi_{\theta_{t+1}}(\rho) - \tau\log\pi_{\theta_t}(\rho)}{\tau+\tau'}\right\} \\
&\ge (\tau+\tau')\sum_\rho \bar{\pi}^*_{\tau,\tau'}(\rho)\log\exp\left\{\frac{\tau\log\pi_{\theta_{t+1}}(\rho) - \tau\log\pi_{\theta_t}(\rho)}{\tau+\tau'}\right\} \\
&= \tau\sum_\rho \bar{\pi}^*_{\tau,\tau'}(\rho)\log\frac{\pi_{\theta_{t+1}}(\rho)}{\pi_{\theta_t}(\rho)} = \tau\left[D_{\mathrm{KL}}(\bar{\pi}^*_{\tau,\tau'}\|\pi_{\theta_t}) - D_{\mathrm{KL}}(\bar{\pi}^*_{\tau,\tau'}\|\pi_{\theta_{t+1}})\right] \ge 0.
\end{aligned}
$$

(Fixed Points) The stationary point of $\mathrm{SR}(\pi_\theta)$ is the π_θ which satisfies,

$$
\begin{aligned}
\frac{d\mathrm{SR}(\pi_\theta)}{d\theta} &= 0 \\
\iff (\tau+\tau')\cdot\frac{\sum\limits_\rho \exp\left\{\frac{r(\rho)+\tau\log\pi_\theta(\rho)}{\tau+\tau'}\right\}\cdot\frac{\tau}{\tau+\tau'}\cdot\frac{1}{\pi_\theta(\rho)}\cdot\frac{d\pi_\theta(\rho)}{d\theta}}{\sum_{\rho'}\exp\left\{\frac{r(\rho')+\tau\log\pi_\theta(\rho')}{\tau+\tau'}\right\}} &= 0 \\
\iff -\sum_\rho \frac{\pi_\theta(\rho)\exp\left\{\frac{r(\rho)-\tau'\log\pi_\theta(\rho)}{\tau+\tau'}\right\}}{\sum_{\rho'}\pi_\theta(\rho')\exp\left\{\frac{r(\rho')-\tau'\log\pi_\theta(\rho')}{\tau+\tau'}\right\}}\cdot\frac{1}{\pi_\theta(\rho)}\cdot\frac{d\pi_\theta(\rho)}{d\theta} &= 0. \qquad (\tau>0)
\end{aligned}
\tag{16}
$$

On the other hand, the fixed point of ECPO indicates at some iteration t,

$$
\begin{gathered}
\pi_{\theta_t} = \pi_{\theta_{t+1}}, \\
\text{where } \pi_{\theta_t} \xrightarrow{\text{Lift Step}} \bar{\pi}^*_{\tau,\tau'} \xrightarrow{\text{Project Step}} \pi_{\theta_{t+1}} \text{ in Eq. (4)},
\end{gathered}
\tag{17}
$$

which means π_{θ_t} is the solution of the Project Step,

$$
\begin{aligned}
\left.\frac{dD_{\mathrm{KL}}(\bar{\pi}^*_{\tau,\tau'}\|\pi_\theta)}{d\theta}\right|_{\theta=\theta_t} &= 0 \\
\iff -\sum_\rho \bar{\pi}^*_{\tau,\tau'}(\rho)\cdot\frac{1}{\pi_\theta(\rho)}\cdot\left.\frac{d\pi_\theta(\rho)}{d\theta}\right|_{\theta=\theta_t} &= 0 \\
\iff -\sum_\rho \frac{\pi_{\theta_t}(\rho)\exp\left\{\frac{r(\rho)-\tau'\log\pi_{\theta_t}(\rho)}{\tau+\tau'}\right\}}{\sum_{\rho'}\pi_{\theta_t}(\rho')\exp\left\{\frac{r(\rho')-\tau'\log\pi_{\theta_t}(\rho')}{\tau+\tau'}\right\}}\cdot\frac{1}{\pi_\theta(\rho)}\cdot\left.\frac{d\pi_\theta(\rho)}{d\theta}\right|_{\theta=\theta_t} &= 0. \\
\text{(by Lift Step in Eq. (17))}
\end{aligned}
\tag{18}
$$

Comparing Eq. (18) with Eq. (16), we have the fixed point condition of ECPO is the same as the definition of the stationary point of $\mathrm{SR}(\pi_\theta)$. □

A.5 Proof of Proposition 3

Proof. Note that $\pi_\theta = \frac{\exp\{\Phi^\top\theta\}}{\mathbf{1}^\top\exp\{\Phi^\top\theta\}}$, where Φ is the feature matrix and θ is the policy parameter. Simply compute the Hessian matrix of the objective,

$$
\frac{d^2 D_{\mathrm{KL}}(\bar{\pi}\|\pi_\theta)}{d\theta^2} = \Phi(\Delta(\pi_\theta) - \pi_\theta\pi_\theta^\top)\Phi^\top \succeq 0.
$$

Thus $D_{\mathrm{KL}}(\bar{\pi}\|\pi_\theta)$ is convex in θ. □

A.6 Proof of Lemma 2

Proof. According to the definition of SR,

$$\begin{aligned}\mathrm{SR}(\pi) &= \max_{\pi' \in \Delta} \mathop{\mathbb{E}}_{\rho \sim \pi'} r(\rho) - \tau D_{\mathrm{KL}}(\pi' \| \pi) + \tau' \mathcal{H}(\pi') \\ &= \eta \log \sum_{\rho} \pi(\rho) \cdot \exp\left\{\frac{\hat{r}(\rho)}{\eta}\right\}.\end{aligned}$$

The lower bound is trivial by directly plugging π into the optimization problem. For the upper bound,

$$\begin{aligned}\exp\left\{\frac{\hat{r}(\rho)}{\eta}\right\} &= \exp\left\{\frac{\hat{r}_\infty}{\eta}\right\} \exp\left\{\frac{\hat{r}(\rho) - \hat{r}_\infty}{\eta}\right\} \\ &\le \exp\left\{\frac{\hat{r}_\infty}{\eta}\right\} \left(1 + \frac{1}{\eta}\left(\hat{r}(\rho) - \hat{r}_\infty\right) + \frac{1}{2\eta^2}\left(\hat{r}(\rho) - \hat{r}_\infty\right)^2\right),\end{aligned}$$

where the inequality follows by $e^x \le 1 + x + \frac{x^2}{2}$ for $x \le 0$. Therefore,

$$\begin{aligned}\sum_\rho \pi(\rho) \exp\left\{\frac{\hat{r}(\rho)}{\eta}\right\} &\le \exp\left\{\frac{\hat{r}_\infty}{\eta}\right\} \sum_\rho \pi(\rho) \left(1 + \frac{1}{\eta}\left(\hat{r}(\rho) - \hat{r}_\infty\right) + \frac{1}{2\eta^2}\left(\hat{r}(\rho) - \hat{r}_\infty\right)^2\right) \\ &= \exp\left\{\frac{\hat{r}_\infty}{\eta}\right\} \left(1 + \frac{1}{\eta} \sum_\rho \pi(\rho)(\hat{r}(\rho) - \hat{r}_\infty) + \frac{1}{2\eta^2} \sum_\rho \pi(\rho)\left(\hat{r}(\rho) - \hat{r}_\infty\right)^2\right) \\ &\le \exp\left\{\frac{\hat{r}_\infty}{\eta}\right\} \exp\left\{\frac{1}{\eta} \sum_\rho \pi(\rho)\left(\hat{r}(\rho) - \hat{r}_\infty\right) + \frac{1}{2\eta^2} \sum_\rho \pi(\rho)\left(\hat{r}(\rho) - \hat{r}_\infty\right)^2\right\} \\ &= \exp\left\{\frac{1}{\eta} \sum_\rho \pi(\rho)\hat{r}(\rho) + \frac{1}{2\eta^2} \sum_\rho \pi(\rho)\left(\hat{r}(\rho) - \hat{r}_\infty\right)^2\right\},\end{aligned}$$

where the second inequality follows by $1 + x \le e^x$. Therefore,

$$\mathrm{SR}(\pi) \le \mathop{\mathbb{E}}_{\rho \sim \pi} \hat{r}(\rho) + \frac{1}{2\eta} \mathop{\mathbb{E}}_{\rho \sim \pi}\left[\left(\hat{r}(\rho) - \hat{r}_\infty\right)^2\right].$$

To prove (i), note that as $\tau \to 0$, $\mathrm{SR}(\pi_\theta) \to \tau' \log \sum_\rho \exp\left\{\frac{r(\rho)}{\tau'}\right\}$, the standard softmax value. Taking limit on τ' gives the hardmax value $\max_\rho r(\rho)$ as $\tau' \to 0$.

To prove (ii), we have ,

$$\begin{aligned}\lim_{\tau \to \infty} (\tau + \tau') \log \sum_\rho \exp\left\{\frac{r(\rho) + \tau \log \pi_\theta(\rho)}{\tau + \tau'}\right\} &= \lim_{\tau \to \infty} \frac{\sum_\rho \pi_\theta(\rho) \exp\left\{\frac{r(\rho) - \tau' \log \pi_\theta(\rho)}{\tau + \tau'}\right\} \left(r(\rho) - \tau' \log \pi_\theta(\rho)\right)}{\sum_\rho \pi_\theta(\rho) \exp\left\{\frac{r(\rho) - \tau' \log \pi_\theta(\rho)}{\tau + \tau'}\right\}} \\ &= \sum_\rho \pi_\theta(\rho) \left[r(\rho) - \tau' \log \pi_\theta(\rho)\right] \\ &= \mathop{\mathbb{E}}_{\rho \sim \pi_\theta} r(\rho) + \tau' \mathcal{H}(\pi_\theta).\end{aligned}$$

As $\tau' \to 0$, $\mathrm{SR}(\pi_\theta) \to \mathbb{E}_{\rho \sim \pi_\theta} r(\rho)$. □

B Details of ECPO Learning

This section provides some of the details of learning algorithms for ECPO. We first show the derivation of the analytic solution of the Lift Step.

Lemma 6. *The lift step of Eq.* (4) *has the following closed form expression:*

$$\bar{\pi}^*_{\tau,\tau'}(\rho) \triangleq \frac{\bar{\pi}(\rho) \exp\left\{\frac{r(\rho) - \tau' \log \bar{\pi}(\rho)}{\tau + \tau'}\right\}}{\sum_{\rho'} \bar{\pi}(\rho') \exp\left\{\frac{r(\rho') - \tau' \log \bar{\pi}(\rho')}{\tau + \tau'}\right\}}. \tag{19}$$

Proof. Rewrite the objective function defined in Eq. (4),

$$\mathop{\mathbb{E}}_{\rho\sim\pi} r(\rho) - \tau D_{\mathrm{KL}}(\pi\|\bar{\pi}) + \tau'\mathcal{H}(\pi) = \mathop{\mathbb{E}}_{\rho\sim\pi}[r(\rho) + \tau\log\bar{\pi}(\rho)] + (\tau+\tau')\mathcal{H}(\pi), \tag{20}$$

which is an entropy regularized reshaped reward objective. The optimal policy of this objective can be obtained by directly applying Lemma 4 of Nachum *et al.* [2017b], i.e.,

$$\bar{\pi}^*_{\tau,\tau'}(\rho) \propto \exp\left\{\frac{r(\rho)+\tau\log\bar{\pi}(\rho)}{\tau+\tau'}\right\} = \bar{\pi}(\rho)\exp\left\{\frac{r(\rho)-\tau'\log\bar{\pi}(\rho)}{\tau+\tau'}\right\}. \tag{21}$$

□

The next lemma provides the derivation of the gradient estimation of ECPO (Lemma 1).

Proof. Note that

$$D_{\mathrm{KL}}(\bar{\pi}^*_{\tau,\tau'}\|\pi_\theta) = \mathop{\mathbb{E}}_{\rho\sim\bar{\pi}^*_{\tau,\tau'}}\left[\log\bar{\pi}^*_{\tau,\tau'}(\rho) - \log\pi_\theta(\rho)\right] = \mathop{\mathbb{E}}_{\rho\sim\bar{\pi}}\left[\frac{\bar{\pi}^*_{\tau,\tau'}(\rho)}{\bar{\pi}(\rho)}\left(\log\bar{\pi}^*_{\tau,\tau'}(\rho) - \log\pi_\theta(\rho)\right)\right].$$

Therefore, taking gradient on both sides,

$$\begin{aligned}
\nabla_\theta D_{\mathrm{KL}}(\bar{\pi}^*_{\tau,\tau'}\|\pi_\theta) &\approx -\frac{1}{K}\sum_{k=1}^{K}\frac{\bar{\pi}^*_{\tau,\tau'}(\rho_k)}{\bar{\pi}(\rho_k)}\nabla_\theta\log\pi_\theta(\rho_k) \\
&= -\frac{1}{K}\sum_{k=1}^{K}\frac{\bar{\pi}(\rho_k)\exp\left\{\frac{r(\rho_k)-\tau'\log\bar{\pi}(\rho_k)}{\tau+\tau'}\right\}}{\bar{\pi}(\rho_k)\sum_{\rho'}\bar{\pi}(\rho')\exp\left\{\frac{r(\rho')-\tau'\log\bar{\pi}(\rho')}{\tau+\tau'}\right\}}\nabla_\theta\log\pi_\theta(\rho_k) \qquad \text{by Eq. (19)} \\
&\approx -\frac{1}{K}\sum_{k=1}^{K}\frac{\exp\{\omega_k\}}{\frac{1}{K}\sum_{j=1}^{K}\exp\{\omega_j\}}\nabla_\theta\log\pi_\theta(\rho_k) \\
&= -\sum_{k=1}^{K}\frac{\exp\{\omega_k\}}{\sum_{j=1}^{K}\exp\{\omega_j\}}\nabla_\theta\log\pi_\theta(\rho_k).
\end{aligned}$$

□

Recall that in Algorithm 1 the project step is performed by SGD. In our implementation, the end condition of SGD is controlled by two parameters: $\epsilon > 0$ and F_STEP $\in \{0, 1\}$. First, SGD halts if the change of the KL divergence is below or equal to ϵ. Second, F_STEP decides the maximum number of SGD steps. If F_STEP $= 1$, the maximum number is $\sqrt{t}$ at iteration t; while if F_STEP $= 0$, there is no restriction on the maximum number of gradient steps, and stopping condition only depends on ϵ.

C Details of ECAC Learning

Recall that in Section 4, the losses for training soft state-value function, soft Q-function and policy are as follows,

$$L(\phi) = \mathbb{E}_{s\sim\mathcal{D}}\left[\frac{1}{2}\left(V_\phi(s) - [Q_\psi(s,a) - \tau'\log\bar{\pi}(a|s)]\right)^2\right], \tag{22}$$

$$L(\psi) = \mathbb{E}_{(s,a,s')\sim\mathcal{D}}\left[\frac{1}{2}\left(Q_\psi(s,a) - [r(s,a) + \gamma V_\phi(s')]\right)^2\right], \tag{23}$$

$$L(\theta) = \mathbb{E}_{s\sim\mathcal{D}}\left[D_{\mathrm{KL}}\left(\exp\left\{\frac{Q_\psi(s,\cdot)+\tau\log\bar{\pi}(\cdot|s) - V_\phi(s)}{\tau+\tau'}\right\}\middle\|\,\pi_\theta(\cdot|s)\right)\right]. \tag{24}$$

To increase the stability of the training, we include a target state value network $V_{\bar{\phi}}$, where $\bar{\phi}$ is an exponentially moving average of the value network weights ϕ. Different from Eq. (23), the soft Q-function parameters ψ is then trained to minimize the soft Bellman error using the target state value network,

$$L(\psi) = \mathbb{E}_{(s,a,s')\sim\mathcal{D}}\left[\frac{1}{2}\left(Q_\psi(s,a) - \left[r(s,a) + \gamma V_{\bar{\phi}}(s')\right]\right)^2\right] \tag{25}$$

Our approach also use two soft-Q functions in order to mitigate the overestimation problem caused by value function approximation [Haarnoja *et al.*, 2018; Fujimoto *et al.*, 2018]. Specifically, we apply two soft-Q function approximations, $Q_{\psi_1}(s,a)$ and $Q_{\psi_2}(s,a)$, and train them independently. The minimum of the two Q-functions will be used whenever the soft-Q value is needed.

The next lemma shows that the gradient of Eq. (24) can be computed by importance sampling using the reference policy,

Algorithm 2 The ECAC algorithm

```
Input: temperature parameters $\tau$ and $\tau'$, lag on target value network $\alpha$, number of training steps $M$
Initialize $\pi_\theta, V_\phi, V_{\bar{\phi}}, Q_{\psi_1}, Q_{\psi_2}$ and replay buffer $\mathcal{D}$
For $t = 1, 2, \dots$ do
    For each environment step do
        $a \sim \pi_\theta(\cdot|s)$
        Observe $s'$ and $r$ from environment
        $\mathcal{D} \leftarrow \mathcal{D} \cup \{(s, a, r, s')\}$
    end For
    Set $\bar{\pi} = \pi_\theta$
    For $i = 1, \dots, M$ do
        Sample a mini-batch of data $\{(s_j, a_j, r_j, s_j')\}_{j=1}^{B}$ from $\mathcal{D}$
        Compute gradient $\nabla_\theta L(\theta), \nabla_\phi L(\phi), \nabla_{\psi_1} L(\psi_1), \nabla_{\psi_2} L(\psi_2)$ according to Eqs. (22) to (24)
        Update parameters $\theta, \phi, \psi_1, \psi_2$ by gradient descent
        Update $\bar{\phi}$ by $\alpha\phi + (1-\alpha)\bar{\phi}$
    end For
end For
```

Lemma 7. *The gradient of Eq.* (24) *is,*

$$\nabla_\theta L(\theta) = \nabla_\theta \mathbb{E}_{s\sim\mathcal{D}}\left[\mathbb{E}_{a\sim\bar{\pi}}\left[\exp\left\{\frac{Q_\psi(s,a) - \tau' \log\bar{\pi}(a|s) - V_\phi(s)}{\tau + \tau'}\right\}\log\pi_\theta(a|s)\right]\right]. \tag{26}$$

Proof. Let $\pi(a|s) = \exp\left\{\frac{Q_\psi(s,a) + \tau\log\bar{\pi}(a|s) - V_\phi(s)}{\tau+\tau'}\right\}$, then we have,

$$\begin{aligned}
\nabla_\theta L(\theta) &= \nabla_\theta \mathbb{E}_{s\sim\mathcal{D}}\left[\sum_a \pi(a|s)\log\pi(a|s) - \pi(a|s)\log\pi_\theta(a|s)\right] \\
&= \nabla_\theta \mathbb{E}_{s\sim\mathcal{D}}\left[-\sum_a \exp\left\{\frac{Q_\psi(s,a) + \tau\log\bar{\pi}(a|s) - V_\phi(s)}{\tau+\tau'}\right\}\log\pi_\theta(a|s)\right] \\
&= \nabla_\theta \mathbb{E}_{s\sim\mathcal{D}}\left[-\sum_a \bar{\pi}(a|s)\exp\left\{\frac{Q_\psi(s,a) - \tau'\log\bar{\pi}(a|s) - V_\phi(s)}{\tau+\tau'}\right\}\log\pi_\theta(a|s)\right] \\
&= \nabla_\theta \mathbb{E}_{s\sim\mathcal{D}}\left[\mathbb{E}_{a\sim\bar{\pi}}\left[-\exp\left\{\frac{Q_\psi(s,a) - \tau'\log\bar{\pi}(a|s) - V_\phi(s)}{\tau+\tau'}\right\}\log\pi_\theta(a|s)\right]\right]
\end{aligned}$$

□

Pseudocode for ECAC is presented in Algorithm 2. The major difference between ECAC and ECAC in the lift step is that instead of sampling K actions as described in Algorithm 1, ECAC only samples one action to construct $\bar{\pi}^*_{\tau,\tau'}$, due to the fact both the soft-Q and state value function approximations are adopted. The value function approximations also make ECAC capable of using off-policy data from a reply buffer. Furthermore, in the project step of ECAC, we use a fixed number of iteration for SGD, which is given by an input parameter of the algorithm.

D Stochastic Transition Setting

In Section 1.1, we assume that the state transition function is deterministic for simplicity. For completeness, we consider the general stochastic transition setting here.

D.1 Notations and Settings

Recall in Section 1.1, the policy probability of trajectory $\rho = (s_1, a_1, \dots, a_{T-1}, s_T)$ is denoted as $\pi(\rho) = \prod_{t=1}^{T-1}\pi(a_t|s_t)$. We define transition probability of ρ as $f(\rho) \triangleq \prod_{t=1}^{T-1} f(s_{s+1}|s_t, a_t)$. The total probability of ρ under policy π and transition f is then $p_{\pi,f}(\rho) \triangleq \pi(\rho)f(\rho) = \prod_{t=1}^{T-1}\pi(a_t|s_t)f(s_{s+1}|s_t, a_t)$. We use $\Delta_f \triangleq \{\pi | \sum_\rho p_{\pi,f}(\rho) = \sum_\rho \pi(\rho)f(\rho) = 1, \pi(\rho) \geq 0, f(\rho) > 0, \forall\rho\}$ to refer to the probabilistic simplex over all possible trajectories. It is obvious that $p_{\pi,f}(\rho) = \pi(\rho)$ and $\Delta_f = \Delta$ under deterministic transition setting, i.e., $f(\rho) = 1, \forall\rho$.

D.2 ECPO Optimization Problem

The proposed ECPO algorithm solves Eq. (4) in the deterministic transition setting. In the stochastic setting, the corresponding problem is,

$$\begin{aligned} \textbf{(Project Step)} \quad & \operatorname*{arg\,min}_{\pi_\theta \in \Pi} D_{\mathrm{KL}}(p_{\bar{\pi}^*_{\tau,\tau'},f} \| p_{\pi_\theta,f}), \\ \textbf{(Lift Step)} \quad & \text{where } \; \bar{\pi}^*_{\tau,\tau'} = \operatorname*{arg\,max}_{\pi \in \Delta_f} \mathop{\mathbb{E}}_{\rho \sim p_{\pi,f}} [r(\rho) - \tau' \log \pi(\rho)] - \tau D_{\mathrm{KL}}(p_{\pi,f} \| p_{\pi_{\theta_t},f}). \end{aligned} \tag{27}$$

which also recovers Eq. (4) as a special case when $f(\rho) = 1, \forall \rho$.

Like Eq. (4), $\bar{\pi}^*_{\tau,\tau'}$ in Eq. (27) also has a closed form expression,

Lemma 8. *The unconstrained optimal policy of Eq.* (27) *has the following closed form expression:*

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

Proof. Rewrite the maximization problem in Eq. (27) as (take π_{θ_t} as the reference policy $\bar{\pi}$),

$$\begin{aligned} \operatorname*{maximize}_{\pi} & \sum_\rho \pi(\rho) f(\rho) \left[r(\rho) - (\tau + \tau') \log \pi(\rho) + \tau \log \bar{\pi}(\rho)\right] \\ \text{subject to} & \sum_\rho \pi(\rho) f(\rho) = 1. \end{aligned}$$

The KKT condition of the above problem is,

$$\begin{aligned} f(\rho) \left[r(\rho) - (\tau + \tau') \log \pi(\rho) + \tau \log \bar{\pi}(\rho) + \lambda - (\tau + \tau')\right] = 0, \; \forall \rho \\ \sum_\rho \pi(\rho) f(\rho) = 1. \end{aligned}$$

Using $f(\rho) > 0, \forall \rho$ and solving the KKT condition, we obtain the expression of $\bar{\pi}^*_{\tau,\tau'}$. □

Lemma 8 recovers Lemma 6 as a special case when $f(\rho) = 1, \forall \rho$.

D.3 Theoretical Analysis

In stochastic transition setting, we define the follow softmax approximated expected reward of π_θ

$$\mathrm{SR}_f(\pi_\theta) \triangleq (\tau + \tau') \log \sum_\rho f(\rho) \exp\left\{\frac{r(\rho) + \tau \log \pi_\theta(\rho)}{\tau + \tau'}\right\},$$

which recovers SR(π_θ) when $f(\rho) = 1, \forall \rho$. The monotonic improvement property is for $\mathrm{SR}_f(\pi_\theta)$.

Theorem 3. *Assume that π_{θ_t} is the update sequence of the ECPO algorithm in Eq.* (27)*, then*

$$SR_f(\pi_{\theta_{t+1}}) - SR_f(\pi_{\theta_t}) \geq 0.$$

Proof. Using $D_{\mathrm{KL}}(p_{\bar{\pi}^*_{\tau,\tau'},f} \| p_{\pi_{\theta_{t+1}},f}) = \min_{\pi_\theta \in \Pi} D_{\mathrm{KL}}(p_{\bar{\pi}^*_{\tau,\tau'},f} \| p_{\pi_\theta,f}) \leq D_{\mathrm{KL}}(p_{\bar{\pi}^*_{\tau,\tau'},f} \| p_{\pi_{\theta_t},f})$ and Jensen's inequality,

$$
\begin{aligned}
\mathrm{SR}_f(\pi_{\theta_{t+1}}) - \mathrm{SR}_f(\pi_{\theta_t}) &= (\tau+\tau') \log \sum_\rho \frac{f(\rho)\exp\left\{\frac{r(\rho)+\tau\log\pi_{\theta_{t+1}}(\rho)}{\tau+\tau'}\right\}}{\sum\limits_\rho f(\rho)\exp\left\{\frac{r(\rho)+\tau\log\pi_{\theta_t}(\rho)}{\tau+\tau'}\right\}} \\
&= (\tau+\tau') \log \sum_\rho \frac{f(\rho)\exp\left\{\frac{r(\rho)+\tau\log\pi_{\theta_t}(\rho)}{\tau+\tau'}\right\}}{\sum\limits_\rho f(\rho)\exp\left\{\frac{r(\rho)+\tau\log\pi_{\theta_t}(\rho)}{\tau+\tau'}\right\}} \cdot \exp\left\{\frac{\tau\log\pi_{\theta_{t+1}}(\rho) - \tau\log\pi_{\theta_t}(\rho)}{\tau+\tau'}\right\} \\
&= (\tau+\tau') \log \sum_\rho \bar{\pi}^*_{\tau,\tau'}(\rho) f(\rho) \cdot \exp\left\{\frac{\tau\log\pi_{\theta_{t+1}}(\rho) - \tau\log\pi_{\theta_t}(\rho)}{\tau+\tau'}\right\} \\
&\geq (\tau+\tau') \sum_\rho \bar{\pi}^*_{\tau,\tau'}(\rho) f(\rho) \cdot \log\exp\left\{\frac{\tau\log\pi_{\theta_{t+1}}(\rho) - \tau\log\pi_{\theta_t}(\rho)}{\tau+\tau'}\right\} \\
&= \tau \sum_\rho \bar{\pi}^*_{\tau,\tau'}(\rho) f(\rho) \cdot \log \frac{\pi_{\theta_{t+1}}(\rho)}{\pi_{\theta_t}(\rho)} \\
&= \tau \sum_\rho \bar{\pi}^*_{\tau,\tau'}(\rho) f(\rho) \cdot \log \frac{\pi_{\theta_{t+1}}(\rho) f(\rho)}{\pi_{\theta_t}(\rho) f(\rho)} \\
&= \tau \left[D_{\mathrm{KL}}(p_{\bar{\pi}^*_{\tau,\tau'},f} \| p_{\pi_{\theta_t},f}) - D_{\mathrm{KL}}(p_{\bar{\pi}^*_{\tau,\tau'},f} \| p_{\pi_{\theta_{t+1}},f}) \right] \geq 0. \qquad \square
\end{aligned}
$$

$\mathrm{SR}_f(\pi_\theta)$ also recovers corresponding performance measures in the stochastic transition setting.

Proposition 4. *$SR_f(\pi_\theta)$ satisfies the following properties:*

(i) $SR_f(\pi_\theta) \to \max_\rho r(\rho)$, as $\tau \to 0, \tau' \to 0$.

(ii) $SR_f(\pi_\theta) \to \mathop{\mathbb{E}}\limits_{\rho \sim p_{\pi_\theta,f}} r(\rho)$, as $\tau \to \infty, \tau' \to 0$.

Proof. To prove (i), note that as $\tau \to 0$, $\mathrm{SR}_f(\pi_\theta) \to \tau' \log \sum_\rho f(\rho) \exp\left\{\frac{r(\rho)}{\tau'}\right\}$. Taking limit on τ' gives the hardmax value $\max_\rho r(\rho)$ as $\tau' \to 0$.

To prove (ii), we have

$$
\begin{aligned}
\lim_{\tau\to\infty} (\tau+\tau') \log \sum_\rho f(\rho) \exp\left\{\frac{r(\rho)+\tau\log\pi_\theta(\rho)}{\tau+\tau'}\right\} &= \lim_{\tau\to\infty} \frac{\sum_\rho \pi_\theta(\rho) f(\rho) \exp\left\{\frac{r(\rho)-\tau'\log\pi_\theta(\rho)}{\tau+\tau'}\right\}(r(\rho) - \tau'\log\pi_\theta(\rho))}{\sum_\rho \pi_\theta(\rho) f(\rho) \exp\left\{\frac{r(\rho)-\tau'\log\pi_\theta(\rho)}{\tau+\tau'}\right\}} \\
&= \sum_\rho \pi_\theta(\rho) f(\rho) \left[r(\rho) - \tau' \log \pi_\theta(\rho) \right] \\
&= \mathop{\mathbb{E}}_{\rho\sim p_{\pi_\theta,f}} r(\rho) - \tau' \cdot \mathop{\mathbb{E}}_{\rho\sim p_{\pi_\theta,f}} \log \pi_\theta(\rho)
\end{aligned}
$$

As $\tau' \to 0$, $\mathrm{SR}_f(\pi_\theta) \to \mathbb{E}_{\rho\sim p_{\pi_\theta,f}} r(\rho)$. $\square$

D.4 Learning

The ECPO learning process is intact under the stochastic transition setting. Similar with Appendix B, we can estimate the KL divergence in the projection step of Eq. (27) by drawing K *i.i.d.* samples $\{\rho_1, \ldots, \rho_K\}$ from $p_{\bar{\pi},f}$, i.e., the mixture of $\bar{\pi}$ and f, which is exactly the process of sampling from $\bar{\pi}$ and interacting with the environment,

$$
\begin{aligned}
D_{\mathrm{KL}}(p_{\bar{\pi}^*_{\tau,\tau'},f} \| p_{\pi_\theta,f}) &= \mathop{\mathbb{E}}_{\rho\sim p_{\bar{\pi}^*_{\tau,\tau'},f}} \left[\log \bar{\pi}^*_{\tau,\tau'}(\rho) - \log \pi_\theta(\rho) \right] \\
&= \mathop{\mathbb{E}}_{\rho\sim p_{\bar{\pi},f}} \frac{\bar{\pi}^*_{\tau,\tau'}(\rho)}{\bar{\pi}(\rho)} \left[\log \bar{\pi}^*_{\tau,\tau'}(\rho) - \log \pi_\theta(\rho) \right].
\end{aligned} \tag{28}
$$

We can then approximate the gradient of $D_{\mathrm{KL}}(p_{\bar{\pi}^*_{\tau,\tau'},f} \| p_{\pi_\theta,f})$ by averaging these K samples according to Eq. (28).

Theorem 4. *Let* $\omega_k = \frac{r(\rho_k) - \tau' \log \bar{\pi}(\rho_k)}{\tau + \tau'}$*. Given* K i.i.d. *samples* $\{\rho_1, \ldots, \rho_K\}$ *from the* reference policy $\bar{\pi}$*, we have the following unbiased gradient estimator,*

$$\nabla_\theta D_{\mathrm{KL}}(p_{\bar{\pi}^*_{\tau,\tau'},f} \| p_{\pi_\theta,f}) \approx -\sum_{k=1}^{K} \frac{\exp\{\omega_k\}}{\sum_{j=1}^{K} \exp\{\omega_j\}} \nabla_\theta \log \pi_\theta(\rho_k), \quad (29)$$

Proof. See the proof of Lemma 1. □

Similar argument could be applied for ECAC learning objectives.

E Experiments Details

We describe the algorithmic and mujoco tasks we experimented on as well as details of experimental setup in this section.

E.1 Algorithmic and Mujoco Tasks

In each algorithmic task, the agent operates on a tape of characters or digits. At each time step, the agent read one character or digit, and then decide to either move the read pointer one step in any direction of the tape, or write a character or digit to output. The total reward of each sampled trajectory is only observed at the end. The goal of each task is:

- **Copy:** Copy a sequence of characters to output.
- **DuplicatedInput:** Duplicate a sequence of characters.
- **RepeatCopy:** Copy a sequence of characters, reverse it, then forward the sequence again.
- **Reverse:** Reverse a sequence of characters.
- **ReversedAddition:** Observe two numbers in base 3 in little-endian order on a $2 \times n$ grid tape. The agent should add the two numbers together.

The Mujoco library contains various of continuous control tasks [Todorov *et al.*, 2012]. The specific action dimensions of each problem is summarized in Table 1.

Table 1: Action Dimensions of Mujoco Tasks

Task	Action Dimensions
Hopper	3
Walker2d	6
HalfCheetah	6
Ant	8
Humanoid	17

E.2 Implementation Details

For the synthetic bandit problem, we parameterize the policy by a weight vector $\theta \in \mathbb{R}^{20}$. Let $\Omega = (\omega_1, \ldots, \omega_{10,000})$ be the feature matrix. The policy is defined by softmax$(\Omega^\top \theta)$. The ECPO parameters used in Fig. 1 are summarized in Table 2.

Table 2: ECPO Hyperparameters in Synthetic Bandit

Parameter	Values
τ	0.1
τ'	0.0
learning rate	0.01
ϵ	$5 \cdot 10^{-4}$
F_STEP	0

For the algorithmic tasks, policy is parameterized by a recurrent neural network with LSTM cells of hidden dimension 256. In all algorithms, N distinct environments are used to generate samples. On each environment, K random trajectories are sampled using the agent's policy to estimate gradient according to Eq. (6), which gives the batch size $N \times K$ in total. We apply the same batch training setting as in UREX [Nachum *et al.*, 2017a], where $N = 40$ and $K = 10$. F_STEP of REMPD is set to 1 in all tasks (See Appendix B). The ECPO parameters used in Fig. 1 are summarized in Table 3.

We use standard gaussian policy for all experimented algorithms in the mujoco tasks. Two layer fully-connected feed-forward neural networks with hidden dimension 300 and ReLU nonlinearity are applied to parameterize policy, soft state value, and

Table 3: ECPO Hyperparameters in Algorithmic Tasks

	Copy	DuplicatedInput	RepeatCopy	Reverse	ReversedAddition
τ	0.5	0.5	2.0	0.2	0.5
τ'	0.01	0.01	0.01	0.02	0.01
learning rate	0.01	0.01	0.01	0.001	0.001
clip norm	20	20	20	20	20
ϵ	0.01	0.01	0.005	0.005	0.005

soft-Q value. We batch size 256 for all algorithms on all tasks. The lag parameter α of ECAC for target value network update is 0.01, and the number of training steps is set as $M = 100$ in all tasks. The other domain-dependent ECAC parameters are summarized in Table 4.

Table 4: ECAC Hyperparameters in Mujoco Tasks

	Walker2d	Hopper	HalfCheetah	Ant	Humanoid
τ	1.5	0.5	0.5	1.0	2.0
τ'	0.2	0.05	0.2	0.1	0.05
ψ learning rate	$5 \cdot 10^{-4}$	$5 \cdot 10^{-4}$	$3 \cdot 10^{-4}$	$3 \cdot 10^{-4}$	$3 \cdot 10^{-4}$
ϕ learning rate	$5 \cdot 10^{-4}$	$5 \cdot 10^{-4}$	$3 \cdot 10^{-4}$	$3 \cdot 10^{-4}$	$3 \cdot 10^{-4}$
θ learning rate	$5 \cdot 10^{-4}$	$5 \cdot 10^{-4}$	$3 \cdot 10^{-4}$	$3 \cdot 10^{-4}$	$3 \cdot 10^{-4}$