University Mohamed VI Polytechnic
Academic Year 2023-2024

Policy Gradient Methods for Reinforcement Learning

Author: ATMANI Hanan

Supervised by:
Pr. SAADI Omar

September 29, 2024

Contents

List of Figures

Introduction

Reinforcement Learning (RL) is one of the most promising branches of artificial intelligence. Unlike supervised or unsupervised learning, reinforcement learning focuses on how an agent can interact with a dynamic environment to learn how to make optimal decisions. Specifically, the agent learns by receiving reward or punishment signals based on its actions and seeks to maximize cumulative rewards over time. This framework has found applications in various fields, such as robotics, video games, resource management, and autonomous systems.

A crucial aspect of reinforcement learning is how policies—i.e., the decision-making rules the agent follows—are optimized. Policy Gradient Methods have emerged as a powerful approach for this type of optimization. Unlike value-based methods, which estimate a value function for each state or action, policy gradient methods directly model a stochastic policy, i.e., a probability distribution over the actions, and optimize it using gradient calculations. This approach is particularly useful in environments where the state and action spaces are continuous or in scenarios where more sophisticated exploration is required.

The Proximal Policy Optimization (PPO) algorithm, introduced by Schulman et al. in 2017, is one of the most notable advancements in this area. It is a policy gradient approach that introduces a proximity constraint to avoid overly large policy updates, ensuring greater stability and robustness compared to earlier methods such as Trust Region Policy Optimization (TRPO). PPO has demonstrated its effectiveness in various contexts, including complex simulated environments and multi-agent scenarios.

Recent research has focused on improving policy gradient algorithms by introducing PPO variants, integrating regularization methods, or adapting policy update criteria to make algorithms more suited to specific environments. Among these variants are approaches such as PPO with rollback (PPO-RB), Trust Region-PPO (TR-PPO), and neural network-based PPO optimizations that aim to ensure global convergence to optimal policies in complex scenarios.

The objective of this report is to present a detailed analysis of policy gradient methods and their variants based on a review of recent scientific articles. The work conducted during this research internship involved reading, analyzing, and synthesizing key academic publications in this domain to gain a deeper understanding of the theoretical foundations and current advancements. While this report does not contain original experimental results, it aims to provide an overview of policy gradient techniques and discuss their advantages, limitations, and future research perspectives.

I Policy Gradient Methods and Non-Convex Optimization

For a distribution ρ defined over states, we define:

$$V^{\pi}(\rho) := \mathbb{E}_{s_0 \sim \rho}\left[V^{\pi}\left(s_0\right)\right]$$

Consider a class of policies $\left\{\pi_\theta \mid \theta \in \Theta \subset \mathbb{R}^d\right\}$, where $d = |A||S|$.
The optimization problem we now focus on is written as:

$$\max_{\theta \in \Theta} V^{\pi_\theta}(\rho) \qquad \text{(P1)}$$

A deterministic policy π_θ is generally not differentiable, which motivates us to consider π_θ as a class of stochastic policies, allowing for differentiability.

I.1 Policy class

Softmax policy:

$$\pi_\theta(a \mid s) = \frac{\exp\left(\theta_{s,a}\right)}{\sum_{a'} \exp\left(\theta_{s,a'}\right)}, \quad \text{with} \quad \Theta = \mathbb{R}^{|\mathcal{S}||\mathcal{A}|}$$

The closure of the Softmax class contains all stationary and deterministic policies.

Log-linear policies:

$$\pi_\theta(a \mid s) = \frac{\exp\left(\theta \cdot \phi_{s,a}\right)}{\sum_{a' \in \mathcal{A}} \exp\left(\theta \cdot \phi_{s,a'}\right)}$$

$\phi_{s,a} \in \mathbb{R}^d$ is a feature vector for the pair (s, a).

Neural softmax policies:

$$\pi_\theta(a \mid s) = \frac{\exp\left(f_\theta(s, a)\right)}{\sum_{a' \in \mathcal{A}} \exp\left(f_\theta\left(s, a'\right)\right)}$$

where the scalar function $f_\theta(s, a)$ can be parameterized by a neural network.

I.2 Policy gradient

Consider an initial distribution μ, where s_0 is drawn from this distribution. Following a policy π, we define the distribution of the trajectory τ as:

$$\operatorname{Pr}_\mu^\pi(\tau) = \mu\left(s_0\right) \pi\left(a_0 \mid s_0\right) P\left(s_1 \mid s_0, a_0\right) \pi\left(a_1 \mid s_1\right) \cdots .$$

The total discounted reward of a trajectory:

$$R(\tau) := \sum_{t=0}^{\infty} \gamma^t r\left(s_t, a_t\right)$$

$$V^{\pi_\theta}(\mu) = \mathbb{E}_{\tau \sim \operatorname{Pr}_\mu^{\pi_\theta}}[R(\tau)].$$

Theorem I.1 *There are three different expressions for $\nabla_\theta V^{\pi_\theta}(\mu)$:*

- *REINFORCE:*

$$\nabla V^{\pi_\theta}(\mu)=\mathbb{E}_{\tau\sim\operatorname{Pr}_\mu^{\pi_\theta}}\left[R(\tau)\sum_{t=0}^{\infty}\nabla\log\pi_\theta\left(a_t\mid s_t\right)\right]$$

- *Action value expression:*

$$\nabla V^{\pi_\theta}(\mu)=\mathbb{E}_{\tau\sim\operatorname{Pr}_\mu^{\pi_\theta}}\left[\sum_{t=0}^{\infty}\gamma^t Q^{\pi_\theta}\left(s_t,a_t\right)\nabla\log\pi_\theta\left(a_t\mid s_t\right)\right]$$

- *Advantage expression:*

$$\nabla V^{\pi_\theta}(\mu)=\frac{1}{1-\gamma}\mathbb{E}_{s\sim d_\mu^{\pi_\theta}}\mathbb{E}_{a\sim\pi_\theta(\cdot\mid s)}\left[A^{\pi_\theta}(s,a)\nabla\log\pi_\theta(a\mid s)\right]$$

  Where:

$$A^\pi(s,a):=Q^\pi(s,a)-V^\pi(s)$$

  called the advantage of policy π. If $A^\pi(s,a)\leq 0$, then $\pi=\pi^*$.

$$d_\mu^\pi(s)=\mathbb{E}_{s_0\sim\mu}\left[d_{s_0}^\pi(s)\right],\quad d_{s_0}^\pi(s)=(1-\gamma)\sum_{t=0}^{\infty}\gamma^t\operatorname{Pr}^\pi\left(s_t=s\mid s_0\right)$$

  $d_{s_0}^\pi(s)$ is the state visitation measure.

Proof: 1) $\nabla V^{\pi_\theta}(\mu)=\mathbb{E}_{\tau\sim\operatorname{Pr}_\mu^{\pi_\theta}}[R(\tau)]\ =\nabla\sum_\tau R(\tau)\operatorname{Pr}_\mu^{\pi_\theta}(\tau)$

2) $\nabla V^{\pi_\theta}\left(s_0\right)=\nabla\sum_{a_0}\pi_\theta\left(a_0\mid s_0\right)Q^{\pi_\theta}\left(s_0,a_0\right)$

Remark:

$V^{\pi_\theta}(s)$ is not concave when using the Softmax parameterization.

Gradient ascent algorithm with a fixed step size η:

$$\theta_{t+1}=\theta_t+\eta\nabla V^{\pi_{\theta_t}}(\mu).$$

Suppose that $\theta\in\Theta$ and that V^{π_θ} is β-smooth and bounded below by V^*. Suppose we use a fixed step size η. For any T, we have:

$$\min_{t\leq T}\left\|\nabla V^{\pi_{\theta_t}}(\mu)\right\|^2\leq\frac{2\beta\left(V^*(\mu)-V^{\pi_{\theta_0}}(\mu)\right)}{T}.$$

I.3 Optimization

- A challenge arises: even if all the parameters of the MDP are known, calculating the gradient is computationally expensive. We can use unbiased estimates of π based solely on access to our model through simulations. In other words, assuming we can obtain sampled trajectories $\tau\sim\operatorname{Pr}_\mu^{\pi_\theta}$ (simulating possible paths in the model-defined environment).
- We ignore that τ is an infinitely long sequence.

For a trajectory τ (estimator), we define the unbiased gradient estimator:

$$\widehat{\nabla V^{\pi_\theta}}(\mu):=\sum_{t=0}^{\infty}\gamma^t\widehat{Q^{\pi_\theta}}\left(s_t,a_t\right)\nabla\log\pi_\theta\left(a_t\mid s_t\right)$$

Where:

$$\widehat{Q^{\pi_\theta}}\left(s_t,a_t\right):=\sum_{t'=t}^{\infty}\gamma^{t'-t}r\left(s_{t'},a_{t'}\right)$$

Lemma I.1 *(Unbiased gradient estimate) we have:*

$$\mathbb{E}_{\tau \sim \Pr_{\mu}^{\pi_\theta}}\left[\widehat{\nabla V^{\pi_\theta}}(\mu)\right] = \nabla V^{\pi_\theta}(\mu)$$

1. Initialize θ_0.
2. For $t = 0, 1, \ldots$
(2.1) Sample $\tau \sim \Pr_{\mu}^{\pi_\theta}$.
(2.2) Update:

$$\theta_{t+1} = \theta_t + \eta_t \widehat{\nabla V^{\pi_\theta}}(\mu)$$

where η_t is the step size and $\widehat{\nabla V^{\pi_\theta}}(\mu)$ is estimated using τ.

Lemma I.2 *(Stochastic Convergence to Stationary Points) Suppose that for all $\theta \in \Theta$, V^{π_θ} is β-smooth and bounded below by V^*. Suppose that the variance is bounded as follows:*

$$\mathbb{E}\left[\left\|\widehat{\nabla V^{\pi_\theta}}(\mu) - \nabla V^{\pi_\theta}(\mu)\right\|^2\right] \leq \sigma^2$$

For $t \leq \beta\left(V^*(\mu) - V^{(0)}(\mu)\right)/\sigma^2$, suppose we use a constant step size of $\eta_t = 1/\beta$, and subsequently, we use $\eta_t = \sqrt{2/(\beta T)}$. Then for any T, we have:

$$\min_{t \leq T} \|\nabla V^{\pi_{\theta_t}}(\mu)\|^2 \leq \frac{4\beta\left(V^*(\mu) - V^{(0)}(\mu)\right)}{T} + 2\sigma^2.$$

In practice, σ is very high, meaning that the systematic error is very large in the estimator we provided. Even though it is unbiased, it is not precise. To address this problem, we use a form of variance reduction.
Let $f : \mathcal{S} \to \mathbb{R}$.
1) Provide an estimator for $V^{\pi_\theta}(\mu)$.
2) Sample $\tau \sim \Pr_{\mu}^{\pi_\theta}$.
3) Define:

$$\widehat{Q^{\pi_\theta}}\left(s_t, a_t\right) := \sum_{t'=t}^{\infty} \gamma^{t'-t} r\left(s_{t'}, a_{t'}\right)$$

$$\widehat{\nabla V^{\pi_\theta}}(\mu) := \sum_{t=0}^{\infty} \gamma^t \left(\widehat{Q^{\pi_\theta}}\left(s_t, a_t\right) - f\left(s_t\right)\right) \nabla \log \pi_\theta\left(a_t \mid s_t\right)$$

(Unbiased Gradient Estimate with Variance Reduction) For any procedure used to construct the reference function $f : \mathcal{S} \to \mathbb{R}$, if the samples used to construct f are independent of the trajectory τ, where $\widehat{Q^{\pi_\theta}}\left(s_t, a_t\right)$ is constructed using τ, then:

$$\mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t \left(\widehat{Q^{\pi_\theta}}\left(s_t, a_t\right) - f\left(s_t\right)\right) \nabla \log \pi_\theta\left(a_t \mid s_t\right)\right] = \nabla V^{\pi_\theta}(\mu)$$

II Policy gradient methods

In this section, we aim to explore the convergence properties of policy gradient methods. We focus on the softmax policy class, where the parameters θ are defined without constraints, allowing for unconstrained optimization despite dealing with a non-convex (or concave) optimization landscape. Additionally, we investigate how performance is influenced by the choice of the distribution ρ, leading to a modification of the original problem 'P1' to:

$$\max_{\theta \in \Theta} V^{\pi_\theta}(\mu) \quad \text{(P2)}$$

where μ is any distribution defined on the states. Later, we will examine how ρ affects convergence. Furthermore, we address the challenge that the optimal deterministic policy is achieved as θ approaches infinity.
o understand the necessity of optimizing under a distribution μ, we will consider the following MDP example.

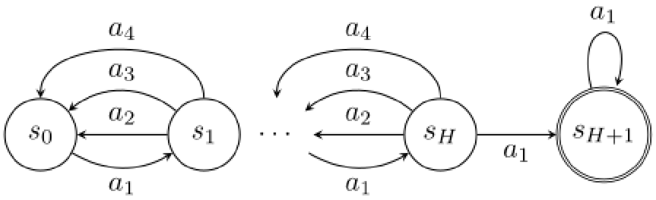


Figure 1: (Vanishing gradient example) A deterministic chain MDP of length $H+2$. We consider a policy where $\pi(a \mid s_i) = \theta_{s_i,a}$ for $i = 1, 2, \ldots, H$. The rewards are zero everywhere else, except at $r(s_{H+1}, a_1) = 1$.

in this example the agent is only rewarded upon viting some small set of states, then the politiy that does not visit any rewarding states will have zero gradient even if it is not optimal

Proposition II.1 *Consider the example of the preceding MDP with* $H+2$ *states,* $\sigma = \frac{H}{1+H}$ *with direct policy parametrezation supppose* $0 < \theta < 1$ *and* $\theta_{s,a_1} < \frac{1}{4}$, $\forall s \in S$. *Then* $\forall k \leq \frac{H}{40\log(2H)} - 1$ *, we have* $\left\|\nabla_\theta^k V^{\pi_\theta}(s_0)\right\| \leq (1/3)^{H/4}$, *where* $\nabla_\theta^k V^{\pi_\theta}(s_0)$ *is a tensor of the* k_{th} *order derivatives of* $V^{\pi_\theta}(s_0)$ *and* $V^\star(s_0) - V^{\pi_\theta}(s_0) \geq (H+1)/8 - (H+1)^2/3^H$.

This property explains that if we choose a strategy in which the agent does not sufficiently explore the environment, it will negatively impact the gradient algorithm. The gradient becomes very small not because we are close to optimal, but rather because the strategy visits advantageous states.

II.1 Policy Gradient Ascent

Proposition II.2 *For the softmax policy class:*

$$\frac{\partial V^{\pi_\theta}(\mu)}{\partial \theta_{s,a}} = \frac{1}{1-\gamma} d_\mu^{\pi_\theta}(s) \pi_\theta(a \mid s) A^{\pi_\theta}(s,a)$$

- **Proof:** (Indecation)

  Observe that:

$$\frac{\partial \log \pi_\theta(a \mid s)}{\partial \theta_{s',a'}} = \mathbf{1}[s = s'](\mathbf{1}[a = a'] - \pi_\theta(a' \mid s))$$

$$\frac{\partial V^{\pi_\theta}(\mu)}{\partial \theta_{s',a'}} = \frac{1}{1-\gamma} \mathbb{E}_{s \sim d_\mu^{\pi_\theta}} \mathbb{E}_{a \sim \pi_\theta(\cdot \mid s)} \left[A^{\pi_\theta}(s,a) \frac{\partial \log \pi_\theta(a \mid s)}{\partial \theta_{s',a'}} \right]$$

  Remark: An issue arises: for any policy π that becomes almost deterministic, its gradient tends to 0 even if it's not optimal.

  The update rule for gradient ascent is:

$$\theta^{(t+1)} = \theta^{(t)} + \eta \nabla_\theta V^{\pi_{\theta_t}}(\mu) \tag{1}$$

Theorem II.1 *Assume we follow the gradient ascent update rule as specified above (1) and $\forall s \in S$, $\mu(s) > 0$. Suppose $\eta \leq \frac{(1-\sigma)^3}{8}$ then we have that for all states $s, V^{\pi_{\theta_t}}(s) \rightarrow V^\star(s)$ as $t \rightarrow \infty$*

Remark: If the condition $\forall s \in S$, $\mu(s) > 0$ is not satisfied, then we can find $s \in S$ such that $\mu(s) = 0$, and thus $\frac{\partial V^{\pi_\theta}(\mu)}{\partial \theta_{s',a'}} \rightarrow 0$.

- The exponential nature of the Softmax expression quickly leads to deterministic policies, making the optimization problem very challenging.
- The convergence rate of this algorithm is exponential (very slow) in the number of states. To avoid this issue, we will adopt a regularization approach to ensure polynomial convergence in the number of states.

II.2 Log Barrier Regularization

Definition II.1 *Relative_ entropy for distribution p and q is defined as:*

$$\mathrm{KL}(p, q) := \mathbb{E}_{x \sim p}[-\log q(x)/p(x)]$$

Denote the uniform distribution over a set $\mathcal{X}$ by $\mathrm{Unif}_{\mathcal{X}}$, and define the following log barrier regularized objective as:

$$\begin{aligned} L_\lambda(\theta) &:= V^{\pi_\theta}(\mu) - \lambda \mathbb{E}_{s \sim \ Unif \ \mathcal{S}}\left[\mathrm{KL}\left(\mathrm{Unif}_{\mathcal{A}}, \pi_\theta(\cdot \mid s)\right)\right] \\ &= V^{\pi_\theta}(\mu) + \frac{\lambda}{|\mathcal{S}||\mathcal{A}|} \sum_{s,a} \log \pi_\theta(a \mid s) + \lambda \log |\mathcal{A}| \end{aligned}$$

where λ is a regularization parameter

The policy gradient ascent updates for $L_\lambda(\theta)$ are given by:

$$\theta^{(t+1)} = \theta^{(t)} + \eta \nabla_\theta L_\lambda\left(\theta^{(t)}\right). \tag{2}$$

Theorem II.2 *Suppose θ is such that:*

$$\left\|\nabla_\theta L_\lambda(\theta)\right\|_2 \leq \epsilon_{opt}$$

and $\epsilon_{opt} \ \leq \lambda/(2|\mathcal{S}||\mathcal{A}|)$. Then we have that for all starting state distributions ρ :

$$V^{\pi_\theta}(\rho) \geq V^\star(\rho) - \frac{2\lambda}{1-\gamma}\left\|\frac{d_\rho^{\pi^\star}}{\mu}\right\|_\infty$$

Proof: Let's show that $\max_a A^{\pi_\theta}(s,a) \leq \frac{2\lambda}{\mu(s)|\mathcal{S}|}$ for all states.
Next, we use a lemma from the first chapter:

$$V^\star(\rho) - V^{\pi_\theta}(\rho) = \frac{1}{1-\gamma} \sum_{s,a} d_\rho^{\pi^\star}(s) \pi^\star(a \mid s) A^{\pi_\theta}(s,a)$$

Remark: The importance of choosing the measure μ.

Corollary II.1 *Let* $\beta_\lambda := \frac{8\gamma}{(1-\gamma)^3} + \frac{2\lambda}{|\mathcal{S}|}$. *Starting from any initial* $\theta^{(0)}$, *consider the updates 0.6) with* $\lambda = \frac{\epsilon(1-\gamma)}{2\left\|\frac{\|\pi^*_\beta}{\mu}\right\|_\infty}$ *and* $\eta = 1/\beta_\lambda$. *Then for all starting state distributions* ρ, *we have*

$$\min_{t<T}\{V^\star(\rho) - V^{\pi_{\theta_t}}(\rho)\} \le \epsilon \quad \textit{whenever} \quad T \ge \frac{320|\mathcal{S}|^2|\mathcal{A}|^2}{(1-\gamma)^6\epsilon^2}\left\|\frac{d_\rho^{\pi^*}}{\mu}\right\|_\infty^2 .$$

This corollary emphasizes the importance of choosing λ carefully for the precision of ϵ, and μ to ensure global optimality.
We can consider other regularization examples such as entropy vs log barrier regularization.

II.3 The Natural Policy Gradient NPG

Observe that a policy constitutes a family of probability distributions $\{\pi_\theta(./s) \mid s \in S\}$

Definition II.2 *The Fisher information matrix of a parameterized density* $p_\theta(x)$ *is defined as* $\mathbb{E}_{x\sim p_\theta}\left[\nabla \log p_\theta(x) \nabla \log p_\theta(x)^\top\right]$. *Now we let us define* $\mathcal{F}_\rho^\theta$ *as an (average) Fisher information matrix on the family of distributions* $\{\pi_\theta(\cdot \mid s) \mid s \in \mathcal{S}\}$ *as follows:*

$$\mathcal{F}_\rho^\theta := \mathbb{E}_{s\sim d_\rho^{\pi_\theta}} \mathbb{E}_{a\sim\pi_\theta(\cdot|s)}\left[\left(\nabla \log \pi_\theta(a \mid s)\right) \nabla \log \pi_\theta(a \mid s)^\top\right].$$

Note that the average is under the state-action visitation frequencies

The NPG algorithm performs gradient updates in the geometry induced by this matrix as follows:

$$\theta^{(t+1)} = \theta^{(t)} + \eta F_\rho\left(\theta^{(t)}\right)^\dagger \nabla_\theta V^{\pi_{\theta_t}}(\rho) \tag{3}$$

where $M^\dagger$ denotes the Moore-Penrose pseudoinverse of the matrix M.
We restrict to using the initial state distribution ρ and we restrict attention to states reachable from ρ

Lemma II.1 *For the softmax parameterization, the NPG updates take the form:*

$$\theta^{(t+1)} = \theta^{(t)} + \frac{\eta}{1-\gamma} A^{\pi_{\theta_t}} + \eta v$$

and

$$\pi^{(t+1)}(a \mid s) = \pi^{(t)}(a \mid s) \frac{\exp\left(\eta A^{\pi_{\theta_t}}(s,a)/(1-\gamma)\right)}{Z_t(s)},$$

where $Z_t(s) = \sum_{a\in\mathcal{A}} \pi^{(t)}(a \mid s) \exp\left(\eta A^{\pi_{\theta_t}}(s,a)/(1-\gamma)\right)$

The update rule NGP does not depend on either ρ or $d_\rho^{\pi_{\theta_t}}$.

The Fisher information nullifies the impact of the initial distribution (stabilizes the algorithm).

Theorem II.3 *Suppose we run the NPG updates using* $\rho \in \Delta(\mathcal{S})$ *and with* $\theta^{(0)} = 0$. *Fix* $\eta > 0$. *For all* $T > 0$, *we have:*

$$V^{(T)}(\rho) \ge V^*(\rho) - \frac{\log|\mathcal{A}|}{\eta T} - \frac{1}{(1-\gamma)^2 T}.$$

Le taux de convergance ne depand pas de ρ
Now setting $\eta \geq (1-\gamma)^2 \log |\mathcal{A}|$, we see that NPG finds an ϵ-optimal policy in a number of iterations that is at most:

$$T \leq \frac{2}{(1-\gamma)^2 \epsilon},$$

The number of iterations required for convergence does not depend on the number of states or actions.

For the iterates $\pi^{(t)}$ generated by the NPG updates (3) , we have for all starting state distributions μ

$$V^{\pi_{\theta_{t+1}}}(\mu) - V^{\pi_{\theta_t}}(\mu) \geq \frac{(1-\gamma)}{\eta} \mathbb{E}_{s \sim \mu} \log Z_t(s) \geq 0.$$

III Function Approximation and the NPG

In this section, we analyze the case of using parametric policy classes:

$$\Pi = \left\{ \pi_\theta \mid \theta \in \mathbb{R}^d \right\}$$

where Π may not contain all stochastic policies and may not even include an optimal policy. The policy class Π is not fully expressive, as $d \ll |\mathcal{S}||\mathcal{A}|$, and in fact, $|\mathcal{S}|$ or $|\mathcal{A}|$ do not need to be finite for the results presented in this section to hold.

The objectives of this section are to establish a connection between the Natural Policy Gradient (NPG) algorithm and compatible function approximation, and to assess the effectiveness of NPG updates in the presence of errors arising from statistical estimation, where exact gradients may not be available and approximation is required.

III.1 Compatible function approximation and the NPG

Definition III.1 ***Compatible function approximation :*** *A compatible function is a function chosen to approximate a specific problem in such a way that it fits well with the characteristics of that problem.*

Lemma III.1 *Let $w^\star$ denote the following minimizer:*

$$w^\star \in \operatorname{argmin}_w \mathbb{E}_{s \sim d_\mu^{\pi_\theta}} \mathbb{E}_{a \sim \pi_\theta(\cdot \mid s)} \left[\left(A^{\pi_\theta}(s,a) - w \cdot \nabla_\theta \log \pi_\theta(a \mid s) \right)^2 \right], \tag{4}$$

The loss function mentioned above is referred to as the error of the compatible function approximation.

This optimization problem is a linear regression problem aiming to approximate the function $A^{\pi_\theta}(s,a)$ using the $\nabla_\theta \log \pi_\theta(\cdot \mid s)$ as features
Denote the best linear predictor of $A^{\pi_\theta}(s,a)$ using $\nabla_\theta \log \pi_\theta(a \mid s)$ by $\widehat{A}^{\pi_\theta}(s,a)$, i.e. Zhu and Rosendo [2021]

$$\widehat{A}^{\pi_\theta}(s,a) := w^\star \cdot \nabla_\theta \log \pi_\theta(a \mid s).$$

Proposition III.1 *We have that:*

$$\nabla_\theta V^{\pi_\theta}(\mu) = \frac{1}{1-\gamma}\mathbb{E}_{s\sim d_\mu^{\pi_\theta}}\mathbb{E}_{a\sim\pi_\theta(\cdot|s)}\left[\nabla_\theta \log \pi_\theta(a \mid s)\hat{A}^{\pi_\theta}(s,a)\right].$$

Proof:
Use the first-order optimality condition in (4), and utilize the advantage expression of $\nabla_\theta V^{\pi_\theta}(\mu)$

Lemma III.2 *We have that:*

$$F_\rho(\theta)^\dagger \nabla_\theta V^\theta(\rho) = \frac{1}{1-\gamma}w^\star,$$

- This lemma shows that the weight vector above precisely corresponds to the ascent direction of NPG
- This lemma implies that we might write the NPG update rule as:

$$\theta \leftarrow \theta + \frac{\eta}{1-\gamma}w^\star. \tag{5}$$

In practice, the most common policy classes are of the form:

$$\Pi = \left\{ \pi_\theta(a \mid s) = \frac{\exp\left(f_\theta(s,a)\right)}{\sum_{a'\in\mathcal{A}}\exp\left(f_\theta\left(s,a'\right)\right)} \middle| \; \theta \in \mathbb{R}^d \right\},$$

where f_θ is a differentiable function
Π as the tabular softmax policy class if $f_\theta(s,a) = \theta_{s,a}$.
Π as the Log-linear policies if $f_\theta(s,a) = \theta \cdot \phi_{s,a}$
Π as the Neural softmax policies if $f_\theta(s,a)$ is a neural network parameterized by θ

For any state-action pair (s,a), suppose we have a feature mapping $\phi_{s,a} \in \mathbb{R}^d$. Each policy in the log-linear policy class is of the form Π where $f_\theta(s,a) = \theta\cdot\phi_{s,a}$
Compatible function approximation for the log-linear policy class as:

$$\nabla_\theta \log \pi_\theta(a \mid s) = \bar{\phi}_{s,a}^\theta, \text{ where } \quad \bar{\phi}_{s,a}^\theta = \phi_{s,a} - \mathbb{E}_{a'\sim\pi_\theta(-|s)}\left[\phi_{s,a'}\right],$$

$\bar{\phi}_{s,a}^\theta$ is the centered version of $\phi_{s,a}$.
The NPG update using Log-linear Policy Classes

$$\text{NPG:} \quad \theta \leftarrow \theta + \eta w_\star$$

$$w_\star \in \operatorname{argmin}_w \mathbb{E}_{s\sim d_\rho^\pi}\mathbb{E}_{a\sim\pi(\cdot|s)}\left[\left(A^{\pi_\theta}(s,a) - w\cdot\bar{\phi}_{s,a}^\theta\right)^2\right].$$

We have rescaled the learning rate η in comparison to (5)
Here, the compatible function approximation error assesses how effectively our parameterization can capture the policy's advantage function using linear functions.
The Q-NPG using Log-linear Policy Classes :

$$Q\text{-NPG:} \quad \theta \leftarrow \theta + \eta w_\star,$$

$$w_\star \in \operatorname{argmin}_w \mathbb{E}_{s\sim d_\rho^\pi}\mathbb{E}_{a\sim\pi(\cdot|s)}\left[\left(Q^{\pi_\theta}(s,a) - w\cdot\phi_{s,a}\right)^2\right].$$

We do not center the features for Q-NPG
Observe that $Q^\pi(s,a)$ is also not 0 in expectation under $\pi(\cdot \mid s)$, unlike the

advantage function.

Using the last lemma from Chapter 2, we observe how both NPG and Q-NPG can be seen as an incremental (soft) version of policy iteration. We can write an equivalent update rule directly in terms of the (log-linear) policy π:

$$\text{NPG: } \pi(a \mid s) \leftarrow \pi(a \mid s) \exp\left(w_\star \cdot \phi_{s,a}\right)/Z_s,$$

$$w_\star \in \operatorname{argmin}_w \mathbb{E}_{s\sim d_\rho^\pi}\mathbb{E}_{a\sim\pi(\cdot|s)}\left[\left(A^\pi(s,a) - w\cdot\bar{\phi}^\pi_{s,a}\right)^2\right],$$

where Z_s is normalization
The normalization makes the update invariant to (constant) translations of the features.

Similarly, an equivalent update for $Q-\text{NPG}$, where we update π directly rather than θ, is:

$$Q\text{-NPG: } \pi(a \mid s) \leftarrow \pi(a \mid s) \exp\left(w_\star \cdot \phi_{s,a}\right)/Z_s$$

$$w_\star \in \operatorname{argmin}_w \mathbb{E}_{s\sim d_\rho^\pi}\mathbb{E}_{a\sim\pi(\cdot|s)}\left[\left(Q^\pi(s,a) - w\cdot\phi_{s,a}\right)^2\right].$$

If the compatible function approximation error is 0 then the NPG and Q-NPG are equivalent algorithms

Now, suppose that Now suppose $f_\theta(s,a)$ in Π is a neural network parameterized by θ
Compatible function approximation in this case is:

$$\nabla_\theta \log \pi_\theta(a \mid s) = g_\theta(s,a)$$

where

$$g_\theta(s,a) = \nabla_\theta f_\theta(s,a) - \mathbb{E}_{a'\sim\pi_\theta(\cdot|s)}\left[\nabla_\theta f_\theta\left(s,a'\right)\right],$$

the NPG update is:

$$\text{NPG: } \theta \leftarrow \theta + \eta w_\star,$$

$$w_\star \in \operatorname{argmin}_w \mathbb{E}_{s\sim d_\rho^{\pi^*_\theta}, a\sim\pi_\theta(\cdot|s)}\left[\left(A^{\pi_\theta}(s,a) - w\cdot g_\theta(s,a)\right)^2\right]$$

The Q-NPG variant of this update rule is:

$$Q\text{-NPG:} \quad \theta \leftarrow \theta + \eta w_\star,$$

$$w_\star \in \operatorname{argmin}_w \mathbb{E}_{s\sim d_\rho^\pi}\mathbb{E}_{a\sim\pi(\cdot|s)}\left[\left(Q^{\pi_\theta}(s,a) - w\cdot\nabla_\theta f_\theta(s,a)\right)^2\right].$$

It is helpful for us to consider NPG more abstractly, as an update rule of the form

$$\theta^{(t+1)} = \theta^{(t)} + \eta\omega^{(t)} \tag{6}$$

where $\omega^{(t)}$ is an arbitrary (bounded) sequence

Lemma III.3 *Fix a copmarison policy $\tilde{\pi}$ and a state distribution . Assume for all $s \in \mathcal{S}$ and $a \in \mathcal{A}$ that $\log \pi_\theta(a/s)$ is a β-smooth function of θ. Consider the update rule (6), where $\pi^{(0)}$ is the uniform distribution (for all states) and where the sequence of weights $\omega^{(0)}, ..., \omega^{(T)}$, satisfies $||\omega^{(t)}||_2 \leq W$ (but is otherwise arbitrary). Define:*

$$err_t = \mathbb{E}_{s\sim\tilde{d}}\mathbb{E}_{a\sim\tilde{\pi}(.|s)}\left[A^{(t)}(s,a) - \omega^{(t)}\nabla_\theta \log \pi^{(t)}(a|s)\right] \tag{7}$$

Using $\eta = \sqrt{2\log|\mathcal{A}|/(\beta W^2 T)}$, we have that:

$$\min_{t<T}\left(V^{\tilde{\pi}}(\rho) - V^{(t)}(\rho)\right) \leq \frac{1}{1-\sigma}\left(W\sqrt{\frac{2\beta\log|\mathcal{A}|}{T}} + \frac{1}{T}\sum_{t=0}^{T-1} err_t\right)$$

For a state-action distribution v, define:

$$L(\omega;\theta;v) := \mathbb{E}_{s,a\sim\sqsubseteq}\left[(Q^{\pi_\theta}(s,a) - \omega.\phi_{s,a})^2\right]$$

The iterates of the $Q-$NPG algorithm can be viewed as minimzing this loss under some (changing) distribution v.

IV Conservative Policy Iteration

We have a data distribution $\mathcal{D}, x_i \sim \mathcal{D}, y_i = f^*(x_i) + \epsilon_i$ where noise $\mathbb{E}(\epsilon_i) = 0$

Unknown target function: $f : X \to Y$

Set of function hypotheses: $\mathcal{F} = \{f \mid f : X \to Y\}$
$\hat{f} = \text{argmin}_{f\in\mathcal{F}} \sum_{i=1}^{N}(f(x_i) - y_i)^2$

Subervised learning theory (e.g VC theory) says that we can indeed genralize i.e we can predict well under the same distribution. Assume $\hat{f} \in \mathcal{F}$ (this is called realizability) we can expect:

$$\mathbb{E}_{x\in\mathcal{D}}(\hat{f}(x) - f^*(x))^2 \leq \delta$$

Where δ is a small number $(\frac{1}{N} or \frac{1}{\sqrt{N}})$
subervised learning can fait if there is train-test distribution mismath.
However, for som $\mathcal{D}' \neq \mathcal{D}$, $\mathbb{E}_{x\in\mathcal{D}'}(\hat{f}(x) - f^*(x))^2$ might be arbiteary large.

IV.1 Fitted Policy-Iteration (FPI) or Approximate Policy Iteration (API)

We consider an infinite horizon discounted MDP $\mathcal{M} = \{\mathcal{S}, \mathcal{A}, \gamma, P, r, \mu\}$, where μ is the initial state distribution. We assume the reward is bounded, i.e., $\sup_{s,a} r(s,a) \in [0,1]$. For notation simplicity, we denote $V_{\max} := \frac{1}{1-\gamma}$.
Visitation measure over just the states:

$$d_{s_0}^{\pi}(s) = (1-\gamma)\sum_{t=0}^{\infty}\gamma^t \overset{\pi}{\Pr}(s_t = s \mid s_0).$$

We write

$$d_{\mu}^{\pi}(s) = \mathbb{E}_{s_0\sim\mu}[d_{s_0}^{\pi}(s)].$$

As we will consider large scale unknown MDP here, we start with a (restricted) function class:

$$\mathcal{F} = \{f \mid f : \mathcal{S}\times\mathcal{A} \to [0, V_{max}]\}$$

IV.2 Approximate Policy iteration

Scherrer [2014] Like policy iteration, we iterate between two steep:
Policy Evaluation $\hat{Q}^t \approx Q^{\pi^t}$ (where $\hat{Q}^t \in \mathcal{F}$)
Policy Improvement $\pi^{(t+1)}(s) = \text{argmax}_a \hat{Q}^t(s,a)$

We use supervised learning (regression) to estimat Q^{π^t}

Haw to get training data?

Roll-in with policy π to time step t, returning (s_t, a_t). We start with $s_0 \sim \mu$ and $a_0 \sim \pi(.|s_0)$, and follow policy π until time step t, returning the pair (s_t, a_t)

Quality of the learned $\hat{Q}^t$?

A roll-out from (s, a) gives an unbaised estimate of $Q^\pi(s, a)$, $\mathbb{E}(y) = Q^\pi(s, a)$
Summary of the dataset generation process.

We roll-in to generate $(s, a) \sim v$
At (s, a), we roll-out with policy π to generate an unbiased estimate of Q^π $(Q^\pi(s, a) = y)$
In other worlds, one roll-in & roll-out gives us a triple (s, a, y)
Estimating the function $Q^\pi(s, a)$ using least square regression:
Given π, repeat N times of the roll-in & roll-out process, we get a training dataset of N samples:

$$\mathcal{D}^\pi = \left\{s^i, a^i, y^i\right\}_{i=1}^N, \quad s^i, a^i \sim v, \quad \mathbb{E}(y^i) = Q^\pi(s^i, a^i)$$

Least square regression:

$$\hat{Q} = argmin_{Q \in \mathcal{F}} \sum_{i=1}^{N} (Q(s^i, a^i) - y^i)^2$$

Assume success ful supervise learning we have,

$$\mathbb{E}_{s,a \sim v}(\hat{Q}^\pi(s, a) - Q^\pi(s, a))^2 \leq \delta$$

Initialize $\hat{Q}^0 \in \mathcal{F}$, $\pi^0 = \text{argmax}_a \hat{Q}^0(s, a)$
Data generalization process (roll-in & roll-out $(\mathcal{D}^{\pi^t} = \{s^i, a^i, y^i\}_{i=1}^N, \quad s^i, a^i \sim v, \quad \mathbb{E}(y^i) = Q^{\pi^t}(s^i, a^i))$
Square Regression oracle

$$\hat{Q}^t = argmin_{Q \in \mathcal{F}} \sum_{i=1}^{N} (Q(s^i, a^i) - y^i)^2$$

Policy Improvement $\pi^{(t+1)}(s) = \text{argmax}_a \hat{Q}^t(s, a)$

Remark

$$\text{argmax}_a A^\pi(s, a) = \text{argmax}_a Q^\pi(s, a)$$

Therefore, we can use the advantage function A in the algorithm (API) instead of using Q.

IV.3 Conservative Policy Iteration

Given the current policy π^t, let's find a new policy that has large adv over π^t, under $d_\mu^{\pi^t}$ (i.e let's aim to solve the following program $\text{argmax}_{\pi \in \Pi} \mathbb{E}_{s \sim d_\mu^{\pi^t}} A^{\pi^t}(s, \pi(s))$
Greedy policy selector)
How to impliment such greedy policy selector?

Implementing Approximate Greedy policy Selector via regression
We can do a reduction to regression via advantage function approximation

$$\mathcal{F} = \{f \mid f : \mathcal{S} \times \mathcal{A} \to \mathbb{R}\}, \ \Pi = \{\pi(s) = \text{argmax}_a f(s,a) \mid f \in \mathcal{F}\}$$

$\{s^i, a^i, y^i\}_{i=1}^N, \ \ s^i \sim d_\mu^{\pi^t}, a^i \sim \mathcal{U}(\mathcal{A}), \ \ \mathbb{E}(y^i) = A^{\pi^t}(s^i, a^i)$
Regression oracle:

$$\hat{A}^t = \text{argmin}_{f \in \mathcal{F}} \sum_{i=1}^{N} (f(s^i, a^i) - y^i)^2$$

At greedy with the estimator $\hat{A}^t$ (as we hope $\hat{A}^t \approx A^{\pi^t}$): $\hat{\pi}(s) = \text{argmax}_a \hat{A}^t(s,a), \forall s$
Assume this regression is successful i.e

$$\mathbb{E}_{s \sim d_\mu^{\pi^t}, a^i \sim \mathcal{U}} (\hat{A}^t(s,a) - A^{\pi^t}(s,a))^2 \leq \epsilon$$

Then, $\hat{\pi}(s) = \text{argmax}_a \hat{A}^t(s,a)$ is the approximate greedy policy:

$$\mathbb{E}_{s \sim d_\mu^{\pi^t}} (A^{\pi t}(s, \hat{\pi}(s)) \geq \text{max}_{\pi \in \Pi} \mathbb{E}_{s \sim d_\mu^{\pi^t}} A^{\pi^t}(s, \pi(s)) - \epsilon,$$

Where $\epsilon = \frac{1}{N}$or$\frac{1}{\sqrt{N}}$, denote $\hat{\pi} = \mathcal{G}_\varepsilon(\pi^t, \Pi, \mu)$
The failure case of the API: An abrupt distribution change means the API cannot guarantee success. To ensure monotonic improvement of the API, we need a strong concentrability ratio assumption $\frac{d^\pi(s,a)}{v(s,a)}$

Key idea of CPI: Incremental update-no abrupt distribution change. Making small incremental update to the policy by forcing that the new policy's state action distribution is not too far away from the current policy's. Achieves that by forming a new polcy is mixture of the current policy and a local greedy policy

Conservative Policy Iteration Algorithm:

1. Initialize π^0,
2. For $t = 0, ...$
   (a) Greedy policy selector:

   $$\pi' = \mathcal{G}_\varepsilon(\pi^t, \Pi, \mu)$$

   (b) If $\mathbb{E}_{s \sim d_\mu^{\pi^t}} A^{\pi t}(s, \pi'(s)) \leq \epsilon$: Return π^t
   (c) Incremental update :

   $$\pi^{(t+1)}(.|s) = (1-\alpha)\pi^t(.|s) + \alpha\pi'(.|s), \forall s$$

3. Fin for

∗ If (2) is true: This means that if I cannot find a policy π' that has a positive advantage over the current policy π^t, then π^t is optimal.
∗ With Probability $1 - \alpha$, $a \sim \pi^t(.|s)$
∗ With Probability α, $\pi'(.|s)$

1. Why this is incremental? In what sense?
2. Can we get monotonic poicy improvement?

Lemma IV.1 *(Similar Policies imply similar state visitations). Consider any t, we have that:*

$$\left\|\pi^{t+1}(\cdot \mid s) - \pi^t(\cdot \mid s)\right\|_1 \leq 2\alpha, \forall s;$$

Further, we have:

$$\left\|d_\mu^{\pi^{t+1}} - d_\mu^{\pi^t}\right\|_1 \leq \frac{2\alpha\gamma}{1-\gamma}.$$

Monotonic improvement before termination: Before termination, we have a non-trivial average local advantage: $\mathbb{A} = \mathbb{E}_{s\sim d_\mu^{\pi^t}} A^{\pi^t}(s, \pi'(s)) \geq \epsilon$.
By the performance difference lemma

$$\begin{aligned} V_\mu^{\pi^{t+1}} - V_\mu^{\pi^t} &= \frac{1}{1-\gamma}\mathbb{E}_{s\sim d_\mu^{\pi^{t+1}}}\mathbb{E}_{a\sim\pi^{t+1}(.|s)}A^{\pi^t}(s,a) \\ &= \frac{\alpha}{1-\gamma}\mathbb{E}_{s\sim d_\mu^{\pi^{t+1}}}A^{\pi^t}(s,\pi'(s)) \end{aligned}$$

Theorem IV.1 *Consider any episode t. Denote $\mathbb{A} = \mathbb{E}_{s\sim d_\mu^{\pi^t}} A^{\pi^t}(s, \pi'(s))$. We have:*

$$V_\mu^{\pi^{t+1}} - V_\mu^{\pi^t} \geq \frac{\alpha}{1-\gamma}\left(\mathbb{A} - \frac{2\alpha\gamma}{(1-\gamma)^2}\right)$$

Set $\alpha = \frac{\mathbb{A}(1-\gamma)^2}{4\gamma}$, we get:

$$V_\mu^{\pi^{t+1}} - V_\mu^{\pi^t} \geq \frac{\mathbb{A}^2(1-\gamma)}{8\gamma}$$

Proof: Indication $|\mathbb{E}_{x\sim p}f(x) - \mathbb{E}_{x\sim q}f(x)| \leq max_x|f(x)|||p-q||_1$

Theorem IV.2 *Algorithn CPI terminates in at most $8\gamma/c^2$ steps and outputs a policy π^t satisfying $\max_{\pi\in\Pi}\mathbb{E}_{s\sim d_\mu^{\pi^t}}A^{\pi^t}(s,\pi(s)) \leq 2\varepsilon$.*

Theorem IV.3 *Upon termination, we have a policy π such that:*

$$V^\star - V^\pi \leq \frac{2\varepsilon + \epsilon_\Pi}{(1-\gamma)^2}\left\|\frac{d^{\pi^*}}{\mu}\right\|_\infty,$$

where $\epsilon_\Pi := \mathbb{E}_{s\sim d_\mu^\pi}[\max_{a\in\mathcal{A}} A^\pi(s,a)] - \max_{\pi\in\Pi}\mathbb{E}_{s\sim d_\mu^\pi}[A^\pi(s,\pi(s))]$. In other words, if our policy class is rich enough to approximate the policy $\max_{a\in A} A^\pi(s,a)$ under d_μ^π, i.e., ϵ_Π is small, and μ covers d^{π^*} in a sense that $\left\|\frac{d^{d^*}}{\mu}\right\|_\infty \leq \infty$, CPI guarantees to find a near optimal policy.

$$\begin{aligned} V_\mu^{\pi^{t+1}} - V_\mu^{\pi^t} &= \frac{1}{1-\gamma}\mathbb{E}_{s\sim d_\mu^{\pi^{t+1}}}\mathbb{E}_{a\sim\pi^{t+1}(.|s)}A^{\pi^t}(s,a) \\ &= \frac{\alpha}{1-\gamma}\mathbb{E}_{s\sim d_\mu^{\pi^{t+1}}}A^{\pi^t}(s,\pi'(s)) \end{aligned}$$

Why?
PDl :

$$V_\mu^{\pi^{t+1}} - V_\mu^{\pi^t} = \frac{1}{1-\gamma}\mathbb{E}_{s\sim d_\mu^{\pi^{t+1}}}\mathbb{E}_{a\sim\pi^{t+1}(\cdot|s)}A^{\pi^t}(s,a)$$

Definition of $\pi^{(t+1)}$:

$$
\begin{aligned}
\mathbb{E}_{a\sim\pi^{t+1}(\cdot|s)}A^{\pi^t}(s,a) &= \sum_a \pi^{t+1}(a|s)A^{\pi^t}(s,a)\\
&= \sum_a (1-\alpha)\pi^t(a|s)A^{\pi^t}(s,a) + \sum_a \alpha\pi'(a|s)A^{\pi^t}(s,a)\\
&= 0 + \sum_a \alpha\pi'(a|s)A^{\pi^t}(s,a)\\
&= \alpha A^{\pi^t}(s,\pi'(s))
\end{aligned}
$$

$$
\begin{aligned}
\frac{1}{1-\gamma}\mathbb{E}_{s\sim d_\mu^{\pi^{t+1}}}\mathbb{E}_{a\sim\pi^{t+1}(.|s)}A^{\pi^t}(s,a) &= \frac{1}{1-\gamma}\mathbb{E}_{s\sim d_\mu^{\pi^{t+1}}}A^{\pi^t}\left(s,\pi^{t+1}(s)\right)\\
&= \frac{1}{1-\gamma}\mathbb{E}_{s\sim d_\mu^{\pi^{t+1}}}A^{\pi^t}\left(s,(1-\alpha)\pi^t(s)+\alpha\pi'(s)\right)
\end{aligned}
$$

V Trust-Region constrained Policy Optimization

∗ Let's add to the CPI method, which uses a small step size to ensure incremental updates to policies, another popular approach for incremental policy updates that involves explicitly enforcing a small change in the policy distribution via a trust region constraint.

∗ At iteration t with the current policy π_{θ_t} , we are interested in the following local trust-region constrained optimization:Schulman [2015]

$$
\begin{aligned}
&\max_\theta \mathbb{E}_{s\sim d_\mu^{\pi_\theta}}\mathbb{E}_{a\sim\pi_\theta(\cdot|s)}\log\pi_\theta(a\mid s)A^{\pi_\theta}(s,a)\\
&\text{s.t., } D_{KL}\left(\Pr_\mu^{\pi^{\theta_t}}\parallel\Pr_\mu^{\pi_\theta}\right)\leq\delta,
\end{aligned}
$$

∗ where recall $\Pr_\mu^\pi(\tau)$ is the trajectory distribution induced by π starting at $s_0\sim\mu$, and $D_{KL}(P_1\|P_2)$ are KL-divergence between two distribution P_1 and P_2

∗ As we are interested in small local update in parameters, we can perform sequential quadratic programming here, i.e., we can further linearize the objective function at θ_t and quadratize the KL constraint at θ_t to form a local quadratic programming:

$$
\begin{aligned}
&\max_\theta \left\langle \mathbb{E}_{s\sim d_\mu^{\pi_t}}\mathbb{E}_{a\sim\pi_{\theta_t}(\cdot|s)}\nabla_\theta\log\pi_{\theta_t}(a\mid s)A^{\pi^{\theta_t}}(s,a),\theta-\theta\right.\\
-Region constrained Policy label pr\quad &\text{s.t., } \left\langle\nabla_\theta D_{KL}\left(\Pr_\mu^{\pi^{\theta_t}}\parallel\Pr_\mu^{\pi_\theta}\right)\Big|_{\theta=\theta_t},\theta-\theta_t\right\rangle+\\
&\frac{1}{2}(\theta-\theta_t)^\top\left(\nabla_\theta^2 D_{KL}\left(\Pr_\mu^{\pi^{\theta_t}}\parallel\Pr_\mu^{\pi_\theta}\right)\Big|_{\theta=\theta_t}\right)(\theta-\theta_t)\leq
\end{aligned}
\tag{8}
$$

Proposition V.1 *Consider a finite horizon MDP with horizon H. Consider any fixed θ_t. We have:*

$$
\begin{aligned}
&\nabla_\theta D_{KL}\left(\Pr_\mu^{\pi^{\theta_t}}\parallel\Pr_\mu^{\pi_\theta}\right)\Big|_{\theta=\theta_t}=0,\\
&\nabla_\theta^2 D_{KL}\left(\Pr_\mu^{\pi^{\theta_t}}\parallel\Pr_\mu^{\pi_\theta}\right)\Big|_{\theta=\theta_t}=H\mathbb{E}_{s,a\sim d^{\pi_{\theta_t}}}\nabla\log\pi_{\theta_t}(a\mid s)\left(\nabla\log\pi_{\theta_t}(a\mid s)\right)^\top.
\end{aligned}
$$

The above Proposition shows that a second order taylor expansion of the KL constraint over trajectory distribution gives a local distance metric at θ_t :

$$\frac{1}{2}(\theta-\theta_t)F_{\theta_t}(\theta-\theta_t)$$

Where

$$F_{\theta_t}=H\mathbb{E}_{s,a\sim d^{\pi_{\theta_t}}}\nabla\log\pi_{\theta_t}(a\mid s)\left(\nabla\log\pi_{\theta_t}(a\mid s)\right)^{\top}$$

is proportional to the fisher information matrix.

Now using the results from proposition, we can verify that the local policy optimization procedure in Eq. (**??**) exactly recovers the NPG update, where the step size is based on the trust region parameter δ. Denote $\Delta=(\theta-\theta_t)$, we have

$$\begin{aligned}&\max_{\theta}\left\langle\Delta,\nabla_{\theta}V^{\pi_{\theta_t}}\right\rangle,\\&\text{s.t., }\Delta^{\top}F_{\theta_t}\Delta^{\top}\leq 2\delta,\end{aligned}$$

which gives the following update procedure:

$$\theta_{t+1}=\theta_t+\Delta=\theta_t+\sqrt{\frac{2\delta}{\left(\nabla V^{\pi_{\theta_t}}\right)^{\top}F_{\theta_t}^{-1}\nabla V^{\pi_{\theta_t}}}}\cdot F_{\theta_t}^{-1}\nabla V^{\pi_{\theta_t}}$$

where note that we use the self-normalized learning rate computed using the trust region parameter δ

VI Proximal Policy Optimization

- Schulman et al. [2017] Update Gradient asciente:

$$\theta^{(t+1)}=\theta^{(t)}+\eta\nabla_{\theta}J(\theta^{t})$$

- **Unstable update**
  - Step size is very important (If step size is very small, learning process is slow)
  - Next batch is generated from current bad policy → Collect bad samples.
  - Bad sample → worse policy
- **Data Inefficiency**
  - On policy method: for each new policy we need to generate a completely new trajectory
  - the data is throw out after just one gradient update

If we uses the advantage expression of gradient:

$$\nabla J(\theta)=\mathbb{E}_{\tau\sim\pi_{\theta}}\left[\sum_{t=0}^{\infty}\gamma^{t}A^{\pi_{\theta}}(s,a)\nabla\log\pi_{\theta}(a\mid s)\right]$$

Can we estimate an expectation of one distribution without taking samples from it ? Estimate one distribution by sampling from another distribution:

$$
\begin{aligned}
\mathbb{E}_{x\sim p}(f(x)) &= \int f(x)p(x)\,dx,\\
&= \int f(x)\frac{p(x)}{q(x)}q(x)\,dx,\\
&= \mathbb{E}_{x\sim q}\left(f(x)\frac{p(x)}{q(x)}\right) \approx \frac{1}{N}\sum_{i=1,x_i\sim q}^{N}\left(f(x^i)\frac{p(x^i)}{q(x^i)}\right)
\end{aligned}
$$

$$
\begin{aligned}
\nabla J(\theta) &= \mathbb{E}_{\tau\sim\pi_\theta}\left[A^{\pi_\theta}(s,a)\nabla\log\pi_\theta(a\mid s)\right]\\
&= \mathbb{E}_{\tau\sim\pi_{\theta\text{old}}}\left[\frac{\pi_\theta(s_t,a_t)}{\pi_{\theta\text{old}}(s_t,a_t)}A^{\pi_\theta}(s,a)\nabla\log\pi_\theta(a\mid s)\right]
\end{aligned}
$$

Then the surrogate objective function:

$$
J(\theta) = \mathbb{E}_{\tau\sim\pi_{\theta\text{old}}}\left[\frac{\pi_\theta(s_t,a_t)}{\pi_{\theta\text{old}}(s_t,a_t)}A^{\pi_\theta}(s,a)\right]
$$

Two expectations are same, but we are using sampling method to estimate them → variance is also important

We have: $\text{Var}_{x\sim p}(f(x)) = \mathbb{E}_{x\sim p}(f(x)^2) - (\mathbb{E}_{x\sim p}(f(x)))^2$ and $\mathbb{E}_{x\sim p}(f(x)) = \mathbb{E}_{x\sim q}\left(f(x)\frac{p(x)}{q(x)}\right)$
Then

$$
\begin{aligned}
\text{Var}_{x\sim q}\left(f(x)\frac{p(x)}{q(x)}\right) &= \mathbb{E}_{x\sim q}\left(\left(f(x)\frac{p(x)}{q(x)}\right)^2\right) - \left(\mathbb{E}_{x\sim q}\left(f(x)\frac{p(x)}{q(x)}\right)\right)^2,\\
&= \mathbb{E}_{x\sim p}\left(f(x)^2\frac{p(x)}{q(x)}\right) - (\mathbb{E}_{x\sim p}(f(x)))^2
\end{aligned}
$$

We may need to sample more data if $\frac{p(x)}{q(x)}$ is far away from. Make confident update:

- ∗ Adaptive learning rate
- ∗ limit the policy update range

Can we measure the distance between two distributions ? **KL Divergence:** Measure the distance of two distributions:

$$
D_{KL}(p||q) = \sum_x p(x)\log\left(\frac{p(x)}{q(x)}\right)
$$

KL divergence of two policies:

$$
D_{KL}(\pi_1||\pi_2)[s] = \sum_{a\in\mathcal{A}}\pi_1(a|s)\log\left(\frac{\pi_1(a|s)}{\pi_2(a|s)}\right)
$$

The TRPO method involves solving the problem by linearizing the objective function and transforming the constraints into quadratic form.

$$
\max_\theta\hat{\mathbb{E}}_t\left(\frac{\pi_\theta(a_t|s_t)}{\pi_{\text{old}}(a_t|s_t)}\hat{A}_t\right)
$$

Subject to

$$
\hat{\mathbb{E}}_t\left(KL(\pi_{\theta_{\text{old}}},\pi_\theta(.|s_t))\right) \leq \delta
$$

TRPO uses conjugate gradient descent to solve the optimization problem. The Hessian matrix is computationally and memory expensive

VI.1 Proximal Policy Optimization (PPO) with Adaptative KL Penalty

- The Constraint helps in the training process. However, maybe the constraint is not a strict constraint. Does it matter if we only break the constraint just a few times ?
- What if we treat it as a " soft" constraint? add proximal value to objective function?
- PPO with Adaptive KL Penalty:

$$L^{KLpen}(\theta) = \hat{\mathbb{E}}_t \left(\frac{\pi_\theta(a_t|s_t)}{\pi_{\text{old}}(a_t|s_t)} \hat{A}_t - \beta D_{KL}(\pi_{\theta_{\text{old}}}(.|s_t), \pi_\theta(.|s_t)) \right)$$

  - Hard to pick β value → Use adaptive
- Compute $d = \hat{\mathbb{E}}_t \left(D_{KL}(\pi_{\theta_{\text{old}}}(.|s_t), \pi_\theta(.|s_t)) \right)$
  - If $d < d_{targ}/1.5$, $\beta \leftarrow \frac{\beta}{2}$ (more data)
  - If $d > d_{targ} \times 1.5$ (more penalty)
- Still need to setup a KL divergence target value...

Algorithm PPO with adaptive KL penalty

- Input: Initial policy parameters θ_0, initial KL penalty β_0, target KL-divergence δ
- for $k = 0, 1, 2, ...$ do
  - Collect set of partial trajectories on policy $\pi_k = \pi_{\theta_k}$
  - Estimate advantage $\hat{A}_t^{\pi_k}$ using any advantage estimation algorithm
  - Compute policy update:

$$\theta_{k+1} = \text{argmax}_\theta \hat{\mathbb{E}}_t \left(\frac{\pi_\theta(a_t|s_t)}{\pi_{\text{k}}(a_t|s_t)} \hat{A}_t^{\pi_k} - \beta_k D_{KL}(\pi_{\theta_{\text{k}}}(.|s_t), \pi_\theta(.|s_t)) \right)$$

  - if $\hat{\mathbb{E}}_t \left(D_{KL}(\pi_{\theta_k}(.|s_t), \pi_{\theta_{k+1}}(.|s_t)))\right) \geq 1.5\delta$, Then:
  - $\beta_{k+1} = 2\beta_k$
  - Else if $\hat{\mathbb{E}}_t \left(D_{KL}(\pi_{\theta_k(.|s_t), \pi_{\theta k+1}}(.|s_t))\right) \leq \frac{1.5}{\delta}$, then
  - $\beta_{k+1} = \beta_k/2$
- end for

VI.2 Proximal Policy Optimization with Clipped objective

-$max_\theta \hat{\mathbb{E}}_t \left(\frac{\pi_\theta(s_t/a_t)}{\pi_{\theta_{old}}(s_t/a_t)} \hat{A}_t \right)$.
-Denote the probability ratio: $r_t(\theta) = \frac{\pi_\theta(s_t/a_t)}{\pi_{\theta_{old}}(s_t/a_t)}$
Invariance happens when r changes too quickly → limit r within a range?

Algorithm PPO with Clipped objective

- Input :initial policy parameters θ_0, clipping parameter ϵ
- For k=0,1,2,... do
  - Collect of partial trajectories $\mathcal{D}_k$ on policy $\pi_k = \pi_{\theta_k}$

- Estmate advantage $\hat{A}_t^{\pi_k}$ using any advantage estimation algorithm.
- Compute policy update:

$$\theta_{k+1} = \text{argmax}_\theta L_{\theta_k}^{CLIP}(\theta)$$

- and for

Where

$$L_{\theta_k}^{CLIP}(\theta) = \hat{\mathbb{E}}_{\tau\sim\pi_k}\left(\sum_{t=0}^{T}\left[min\left(r_t(\theta)\hat{A}_t^{\pi_k}, clip(r_t(\theta), 1-\epsilon, 1+\epsilon)\hat{A}_t^{\pi_k}\right)\right]\right)$$

and

$$clip(x; 1-\epsilon, 1+\epsilon) = \begin{cases} 1-\epsilon & x \leq 1-\epsilon \\ 1+\epsilon & x \geq 1+\epsilon.\pi_\theta, \\ x & \text{else} \end{cases}$$

VII Truly Proximal Policy Optimization

PPO is one of the most effective methods in deep reinforcement learning, but its optimization behavior is still poorly understood. We show that it cannot truly limit the probability ratio and does not impose a well-defined constraint to ensure stability. Therefore, we propose an enhanced PPO method called Trust Region-based PPO with Rollback (TR-PPO-RB).
Consider a Markov Decision Process $(\mathcal{S}, \mathcal{A}, \mathbb{P}, r, \rho_1, \gamma)$. where $\mathbb{P} : \mathcal{S}\cdot\mathcal{A}\times\mathcal{S} \to \mathbb{R}$ is the transition probability distribution, $\rho_1$is the distribution of initial state s_1, The performance of a policy π is defined by $\eta(\pi) = \mathbb{E}_{s\sim\rho^\pi, a\sim\pi}[r(s,a)]$, where $\rho^\pi = (1-\gamma)\sum_{t=1}^{\gamma}{}^{t-1}\rho_t^\pi(s)$, ρ_t^π is the density function of state at time t. Policy gradients methods update the policy by the following surrogate performance objective (Sullon et al.,1999):

$$L_{\pi_{old}}(\pi) = \mathbb{E}_{s,a}\left[r_\pi(s,a)A^{\pi_{old}}(s,a)\right] + \eta(\pi_{old})$$

where $r_\pi(s,a) = \frac{\pi(a|a)}{\pi_{old}(a|s)}$ is the probability ratio between the new policy π and the old policy π_{old}
Schulman et al.(2015) derived the following performance bound:

Theorem VII.1 $C = max_{s,a}|A^{\pi_{old}}(s,a)|\frac{4\gamma}{(1-\gamma)^2}$, $D_{KL}^s(\pi_{old}, \pi) := D_{KL}(\pi_{old}(.|s)||\pi(.|s))$, $M_{\pi_{old}}(\pi) = L_{\pi_{old}} - Cmax_{s\in\mathcal{S}}D_{KL}^s(\pi_{old}, \pi)$. *We have*

$$\eta(\pi) \geq M_{\pi_{old}}(\pi), \eta(\pi_{old}) = M_{\pi_{old}}(\pi_{old})$$

This theorem implies that maximizing $M_{\pi old}(\pi)$ guarantee non-decreasing of the performance of the new policy π.
This theorem explains well the performance and improvement of TRPO.
In practice, if $|\mathbb{A}| = D$ the policy is parametrized by $\pi_\theta(s_t) = f_\theta^p(s_t)$. Where f_θ^p is the DNN outputting a vector which represents a D-dimensional discrete distribution. For continuous action space tasks, it is standard to represent the policy by a Gaussian policy, i.e $\pi_\theta(a|s_t) = \mathcal{N}(A|f_\theta^\mu(s_t), f_\theta^\Sigma(s_t)$. Where f_θ^μ and f_θ^Σ are the DNNs which output the mean and convariance matrix of Gaussian distribution.
PPO employs a clipped surrgate objective to prevent the new policy from straying away from the old one. The clipped objective function of state-action (s_t, a_t) is

$$L_t^{CLIP}(\theta) = min(r_t(\theta)A_t, \mathcal{F}^{CLIP}(r_t(\theta), \epsilon)A_t)$$

- $s_t \sim \rho^{\pi_{\theta_{old}}}$, $a_t \sim \pi_{old}(.|s_t)$ are the sampled states and actions, A_t is the estimated advantage value of $A^{\pi_{\theta_{old}}}(s_t, a_t)$, the clipping function is defien as :

$$\mathcal{F}^{CLIP}(r_t(\theta), \epsilon) = \begin{cases} 1-\epsilon & r_t(\theta) \leq 1-\epsilon \\ 1+\epsilon & r_t(\theta) \geq 1+\epsilon, \\ r_t(\theta) & \text{else} \end{cases}$$

Where $[1-\epsilon, 1+\epsilon]$ is called the clipping range, $0 < \epsilon < 1$
The overall objective function is: $L^{CLIP}(\theta) = \frac{1}{T}\sum_{t=1}^{T} L_t^{CLIP}(\theta)$ Wang et al. [2020]

Can PPO effectively bound the probability ratio as it attempts to do?

$$L_t^{CLIP}(\theta) = \begin{cases} (1-\epsilon)A_t & r_t(\theta) \leq 1-\epsilon \;\text{ and }\; A_t < 0, \;(a) \\ (1+\epsilon)A_T & r_t(\theta) \geq 1+\epsilon \;\text{ and }\; A_t > 0, \;(b) \\ r_t(\theta)A_t & \text{otherwise} \end{cases}$$

The case (a) and (b) are called the clipping condition. When the probability ratio $r_t(\theta)$ is outside the clipping range, the gradient of $L_t^{CLIP}(\theta)$ becomes zero, so there is no longer an incentive to adjust $r_t(\theta)$ to bring it back within the range. The probability ratios on some tasks could even reach a value of 40, which is much larger than the upper clipping range 1.2 (Ilyas and al 2018)

Theorem VII.2 *Given θ_0 that $r_t(\theta_0)$ satisfies the clapping condition ((a)or (b)). Let $\nabla L^{CLIP}(\theta_0)$ denote the gradient of L^{CLIP} at θ_0, and similarly $\nabla r_t(\theta_0)$. Let $\theta_1 = \theta_0 + \beta\nabla L^{CLIP}(\theta_0)$, where β is the step size. If*

$$\langle \nabla L^{CLIP}(\theta_0), \nabla r_t(\theta_0) \rangle A_t > 0$$

then there existe som $\bar{\beta} > 0$ such that for any $\beta \in [0, \bar{\beta}]$, we have

$$|r_t(\theta_1) - 1| > |r_t(\theta_0) - 1| > \epsilon$$

As this theorem implies, even the probability ratio $r_t(\theta_0)$ is already out of the clipping range, it could be driven to go farther beyond the range. Statistics based on over one million samples from benchmark tasks show that the condition occurs in a percentage ranging from 25% to 45% across different tasks

Could PPO enforce a trust region con- straint?

Theorem VII.3 *Assume that for discrete action space tasks where $|\mathcal{A}| = D \leq 3$ and the policy is $\pi_\theta(s) = f_\theta^p(s)$, we have $f_\theta^p(s_t) = \left\{p \mid p \in \mathbb{R}^D, \sum_{i=1}^{D} p^{(i)} = 1\right\}$ for continuous action space tasks where the policy is $\pi_\theta(a|s) = \mathcal{N}(a|f_\theta^\mu(s), f_\theta^\Sigma(s))$, we have $(f_\theta^\mu(s_t), f_\theta^\Sigma(s_t)) = \left\{(\mu, \Sigma)/\mu \in \mathbb{R}^D, \Sigma is\ SSD\ D \times Dmatrix\right\}$. Let*

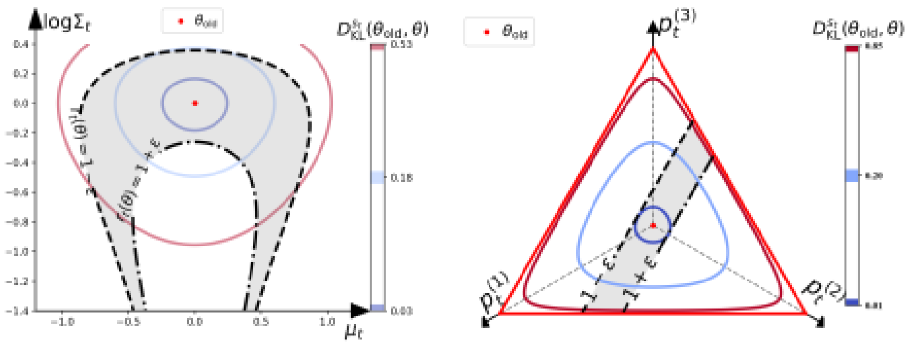
$$E = \{\theta/1-\epsilon \leq r_t(\theta) \leq 1+\epsilon\}$$

we have

$$sup_{\theta \in E} D_{KL}^{s_t}(\theta_{old}, \theta) = +\infty$$

for both discrete and continuous action space tasks

Could PPO enforce a trust region con- straint?



(a) Case of a Continuous Action Space Task (b) Case of a Discrete Action Space Task

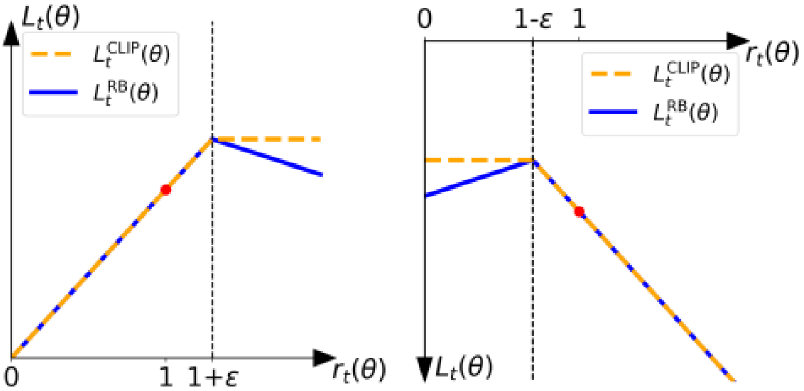
Approaches which manage to bound the probability ratio could not necessarily bound KL divergence theoretically

VII.1 PPO with Rollback (PPO-RB)

What can we do to constrain the probability ratio and the KL divergence? Chen et al. [2018] We address this issue by substituting the clipping function with a rollback function, which is defined as :

$$\mathcal{F}^{RB}(r_t(\theta),\epsilon,\alpha) = \begin{cases} -\alpha r_t(\theta) + (1+\alpha)(1-\epsilon) & r_t(\theta) \leq 1-\epsilon \\ -\alpha r_t(\theta) + (1+\alpha)(1+\epsilon) & r_t(\theta) \geq 1+\epsilon, \\ r_t(\theta) & \text{otherwise} \end{cases}$$

where $\alpha > 0$ is a hyperparameter to decide the force of the rollback. The corresponding objective function at timestep t is donted as $L_T^{RB}(\theta)$ and the overall objective function is $L^{RB}(\theta)$. The rollback function $\mathcal{F}^{RB}(r_t(\theta),\epsilon,\alpha)$ generates a negative incentive when $r_t(\theta)$ is outside of the clipping range.



(a) $A_t > 0$ (b) $A_t < 0$

Where $r_t(\theta)$ is over the clipping range, the slope of L_t^{RB} is reversed, while that of L_t^{CLIP} is zero.

Theorem VII.4 *Let*

$$\theta_1^{CLIP} = \theta_0 + \beta\nabla L^{CLIP}(\theta_0),$$

$$\theta_1^{RB} = \theta_0 + \beta\nabla L^{RB}(\theta_0)$$

The indexes of the samples which satisfy the clipping condition is denoted as

$$\Omega = \{t|1 \leq t \leq T, (A_t > 0 \& r_t(\theta_0) \geq 1+\epsilon) \;\; Or(A_t < 0 \& r_t(\theta_0) \leq 1-\epsilon)\},$$

then there exists some $\beta > 0$ *such that for any* $\beta \in (0, \bar{\beta})$, *we have*

$$|r_t(\theta_1^{RB}) - 1| < |r_t(\theta_1^{CLIP}) - 1|$$

This theorem implies that rollback function can improve its ability in preventing the out-of-the-range ratios from going farther beyond the range.

VII.2 Trust Region-based PPO (TR-PPO)

The original clipping function uses the probability ratio as the element of the trigger condition for clipping. Inspired by the thinking above, we substitute the ratio-based clipping with a trust region-based one

$$\mathcal{F}^{TR}(r_t(\theta), \delta) = \begin{cases} r_t(\theta_{old}) & D_{KL}^{s_t}(\theta_{old}, \theta) \geq \delta \\ r_t(\theta) & \text{otherwise} \end{cases}$$

The incentive for updating policy is removed when the policy π_θ is out of the trust region. TR-PPO combines the strengths of TRPO and PPO: it's theoretically justified, simple to implement, and only needs first-order optimization. Unlike TRPO, TR-PPO doesn't require optimizing θ through KL divergence.

Importance of the min(·, ·) operation

The objective function is :

$$L_t^{TR}(\theta) = min(r_t(\theta)A_t, \mathcal{F}^{TR}(r_t(\theta), \delta)A_t)$$

* Schulman et al. (2017) explained that the `min(·, ·)` operation ensures $L^{TR}t(\theta)$ remains a lower bound on the unclipped objective $r_t(\theta)A_t$, allowing updates even if the policy violates the trust region and improving learning stability.

$$L_t^{TR}(\theta) = \begin{cases} r_t(\theta_{old})A_t & D_{KL}^{s_t}(\theta_{old}, \theta) \geq \delta \;\; and \;\; r_t(\theta)A_t \geq r_t(\theta_{old})A_t \\ r_t(\theta)A_t & \text{otherwise} \end{cases}$$

As can be seen, the ratio is clipped only if the objective value is improved and if the policy violates the constraint

VII.3 Combbinition of TR-PPO and PPO-RB (TR-PPO-RB)

The trust region-based clipping may fail to handle unbounded probability ratios, so we incorporated a rollback mechanism to address this issue.

$$\mathcal{F}^{TR-RB}(r_t(\theta), \delta, \alpha) = \begin{cases} -\alpha r_t(\theta) & D_{KL}^{s_t}(\theta_{old}, \theta) \geq \delta \\ r_t(\theta) & \text{otherwise} \end{cases}$$

As the equation implies, $\mathcal{F}^{TR-RB}(r_t(\theta), \delta, \alpha)$ generates a negative incentive when π_θ is out of the trust rgion

VII.4 Experiment

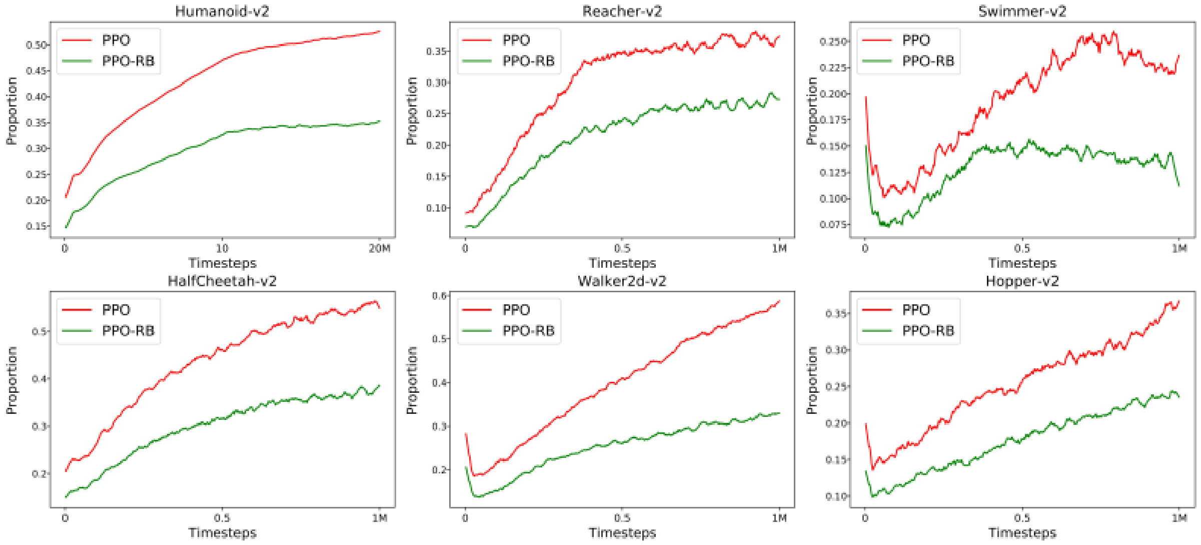


Figure 2: The proportions of the probability ratios which are out of the clipping range.

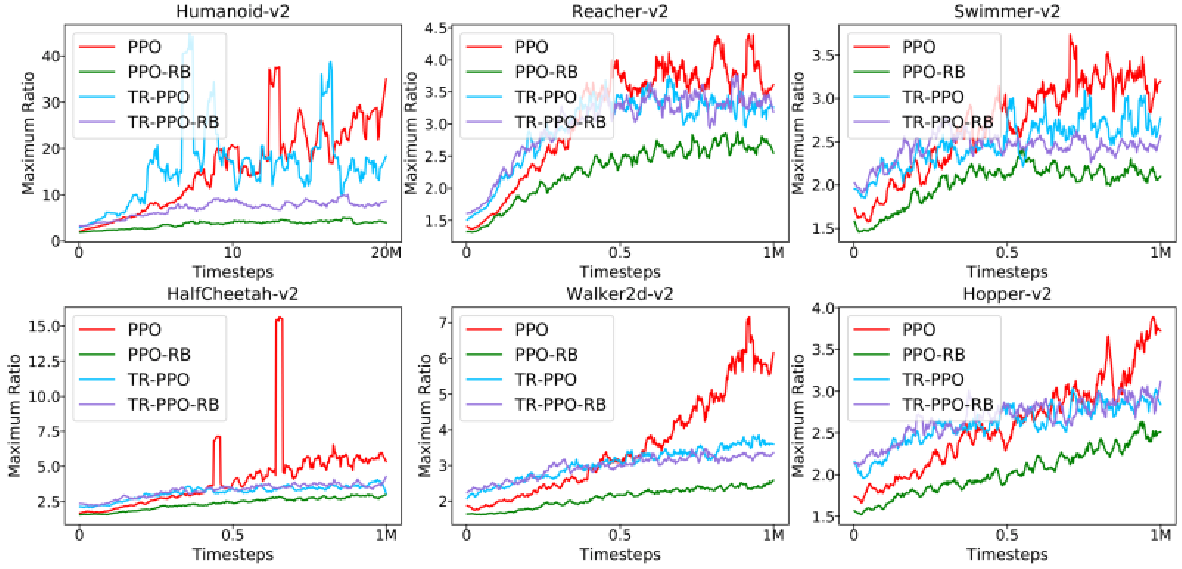


Figure 3: The maximum ratio over all sampled sates of each update during the training process.

VIII Neural Proximal/Trust Region Policy Optimization Attains Globally Optimal Policy

PPO and TRPO have shown significant empirical success, their global convergence remains poorly understood due to the non-convexity of the policy space and neural network parametrization. To bridge this theory-practice gap, three key questions need to be addressed:Liu et al. [2019]

1. How do PPO and TRPO converge to the optimal policy with infinite-dimensional updates?

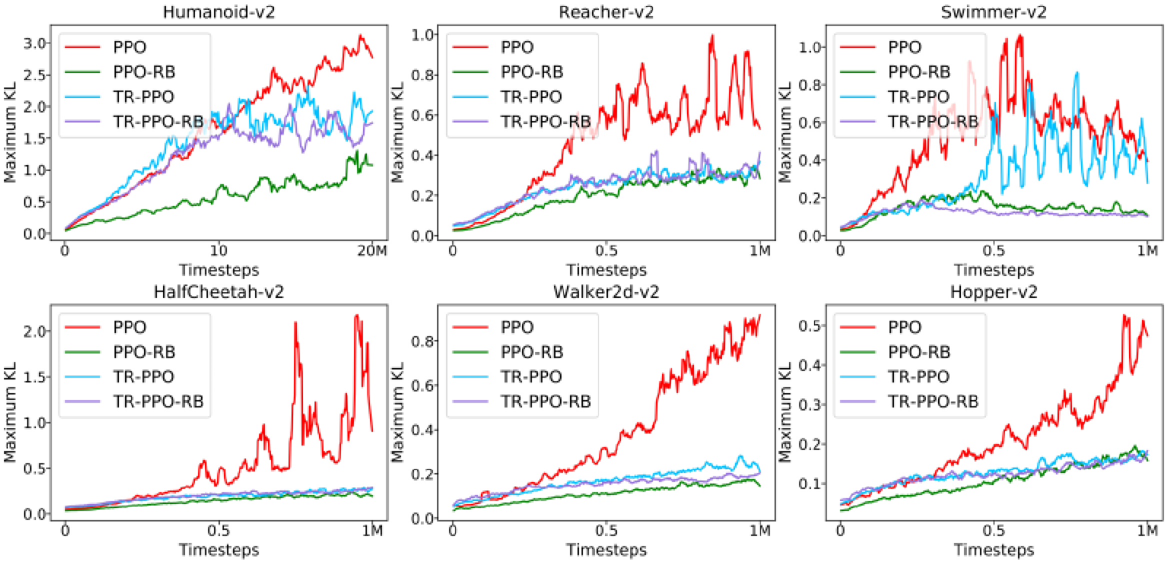


Figure 4: The maximum KL divergence over all sampled states of each update during the training process..

2. How does stochastic gradient descent improve the policy based on this approximate action-value function?

VIII.1 Neural PPO

We consider the Markov decision process $(\mathcal{S}, \mathcal{A}, \mathcal{P}, r, \gamma)$, where $\mathcal{S}$ is a campact state space, $\mathcal{A}$ is a finite action space.
We denote by $v_k := v_{\pi_{\theta_k}}$: The stationary state distribution . $\sigma_k := \sigma_{\pi_{\theta_k}}$: The stationary state-action distribution. $\tilde{\sigma}_k := v_k \pi_0$: The auxiliary distribution.
We assume that $(s, a) \in \mathbb{R}^d$ for all $s \in \mathcal{S}$ and $a \in A$.
We parametrize a functionn $u : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ (policy π action-value function Q^π) by two-layer neural network, which is denoted by $NN(\alpha; m)$,

$$u_\alpha(s, a) = \frac{1}{\sqrt{m}} \sum_{i=1}^{m} b_i * \sigma([\alpha]_i^t (s, a)) \qquad (9)$$

- m: The width of the neural network, $b_i \in [-1, 1] (i \in [m])$: the output weights. $\sigma(.)$: function activation (ReLU) $(\sigma(x) = max\{0, x\})$. $\alpha = ([\alpha]_1^t, ..., [\alpha]_m^t) \in \mathbb{R}^{md}$ with $[\alpha]_i \in \mathbb{R}^d$ $(i \in [m])$ are the input weights.

- We consider the random initialization

$$b_i \sim^{i.i.d} Unif([-1, 1]), [\alpha(0)]_i \sim^{i.i.d} \mathcal{N}(0, I_d/d), \quad \forall i \in [m]$$

- We restrict the input weights α to an L_2-ball centered at the initialization $\alpha(0)$ by the projection:

$$\Pi_{\mathcal{B}^0(R_\alpha)}(\alpha') = argmin_{\alpha \in \mathcal{B}^0(R_\alpha)} \{||\alpha - \alpha'||_2\},$$

where

$$\mathcal{B}^0(R_0) = \{\alpha : ||\alpha - \alpha(0)||_2 \leq R_\alpha\}$$

- Throughout training, we only update α, while keeping $b_i (i \in [m])$fixed at the initialization, we omit the the dependency on b_i $(i \in [m])$ in $NN(\alpha, m)$ and $u_\alpha(s, a)$.

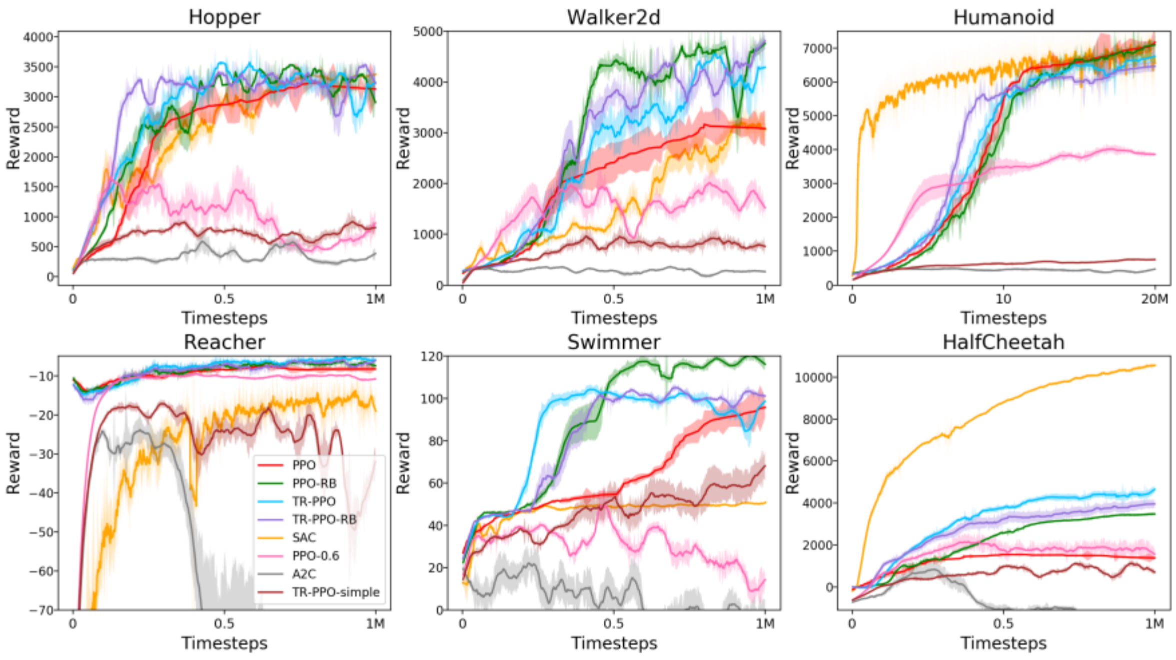


Figure 5: Episode rewards of the policy during the training process.

We consider the population version of the objective function:

$$L(\theta) = \mathbb{E}_{v_k}[\langle Q_{\omega_k}(s,.), \pi_\theta(.|s)\rangle - \beta_k KL(\pi_\theta(.|s)||\pi_{\theta_k}(.|s))]$$

Where Q_{ω_k} is an estimator of $Q^{\pi_{\theta_k}}$.
We consider the energy-based policy $\pi(a|s) \propto \exp\{\tau^{-1}f\}$. Here $f : \mathcal{S}\times\mathcal{A} \to \mathbb{R}$ is the energy function and $\tau > 0$ is the temperature parameter.

Proposition VIII.1 *Let $\pi_{\theta_k} \propto \exp\left\{\tau_k^{-1} f_{\theta_k}\right\}$ be an energy-based policy. Given an estimator Q_{ω_k} of $Q^{\pi_{\theta_k}}$, the update*

$$\widehat{\pi}_{k+1} \leftarrow \operatorname{argmax}_\pi \left\{\mathbb{E}_{\nu_k}\left[\langle Q_{\omega_k}(s,\cdot), \pi(\cdot \mid s)\rangle - \beta_k \cdot \mathrm{KL}\left(\pi(\cdot \mid s) \| \pi_{\theta_k}(\cdot \mid s)\right)\right]\right\}$$

gives

$$\widehat{\pi}_{k+1} \propto \exp\left\{\beta_k^{-1} Q_{\omega_k} + \tau_k^{-1} f_{\theta_k}\right\} \tag{10}$$

To represent the ideal improved policy $\widehat{\pi}_{k+1}$ in Proposition using the energy-based policy $\pi_{\theta_{k+1}} \propto \exp\left\{\tau_{k+1}^{-1} f_{\theta_{k+1}}\right\}$, we solve the subproblem of minimizing the MSE,

$$\theta_{k+1} \leftarrow \underset{\theta\in\mathcal{B}^0\left(R_f\right)}{\operatorname{argmin}} \mathbb{E}_{\tilde{\sigma}_k}\left[\left(f_\theta(s,a) - \tau_{k+1}\cdot\left(\beta_k^{-1} Q_{\omega_k}(s,a) + \tau_k^{-1} f_{\theta_k}(s,a)\right)\right)^2\right] \tag{11}$$

Here we use the neural network parametrization $f_\theta = \mathrm{NN}\left(\theta; m_f\right)$ defined in (9), where θ denotes the input weights and m_f is the width.
To solve (11), we use the SGD update: $\theta(t+1/2) \leftarrow \theta(t) - \eta\cdot\big(f_{\theta(t)}(s,a) - \tau_{k+1}\cdot\big(\beta_k^{-1} Q_{\omega_k}(s,a)$
$\nabla_\theta f_{\theta(t)}(s,a)$
where $(s,a) \sim \widetilde{\sigma}_k$ and $\theta(t+1) \leftarrow \Pi_{\mathcal{B}^\circ\left(R_f\right)}(\theta(t+1/2))$. Here η is the stepsize.
To obtain the estimator Q_{ω_k} of $Q^{\pi_{\theta_k}}$ in (3.3), we solve the subproblem of minimizing the MSBE (Mean Squared Bellman Error),

$$\omega_k \leftarrow \underset{\omega\in\mathcal{B}^0\left(R_Q\right)}{\operatorname{argmin}} \mathbb{E}_{\sigma_k}\left[\left(Q_\omega(s,a) - \left[\mathcal{T}^{\pi_{\theta_k}} Q_\omega\right](s,a)\right)^2\right] \tag{12}$$

The Bellman evaluation operator $\mathcal{T}^\pi$ of a policy π is defined as:
$[\mathcal{T}^\pi Q](s,a) = \mathbb{E}\left[(1-\gamma)\cdot r(s,a) + \gamma \cdot Q(s',a') \mid s' \sim \mathcal{P}(\cdot \mid s,a), a' \sim \pi(\cdot \mid s')\right]$
We use the neural network parametrization $Q_\omega = \mathrm{NN}(\omega; m_Q)$ defined in (9), where ω denotes the input weights and m_Q is the width.

To solve (12) we use the TD update:
$\omega(t+1/2) \leftarrow \omega(t) - \eta \cdot \big(Q_{\omega(t)}(s,a) - (1-\gamma)\cdot r(s,a) - \gamma \cdot Q_{\omega(t)}(s',a')\big) \cdot \nabla_\omega Q_{\omega(t)}(s,a)$
where $(s,a) \sim \sigma_k, s' \sim \mathcal{P}(\cdot \mid s,a), a' \sim \pi_{\theta_k}(\cdot \mid s')$, and $\omega(t+1) = \Pi_{\mathcal{B}^\circ(R_Q)}(\omega(t+1/2))$.

VIII.2 Neural PPO Algorithm

Require:MDP$(\mathcal{S}, \mathcal{A}, \mathcal{P}, r, \gamma)$, penalty parameter β, widths m_f and m_Q, number of SGD and TD iterations T, number of TRPO iterations K, and projection radii $R_f \geq R_Q$

- for $k = 0, \ldots, K-1$ do
    1. Set temperature parameter $\tau_{k+1} \leftarrow \beta\sqrt{K}/(k+1)$ and penalty parameter $\beta_k \leftarrow \beta\sqrt{K}$
    2. Sample $\{(s_t, a_t, a_t^0, s_t', a_t')\}_{t=1}^T$ with $(s_t, a_t) \sim \sigma_k, a_t^0 \sim \pi_0(\cdot \mid s_t), s_t' \sim \mathcal{P}(\cdot \mid s_t, a_t)$ and $a_t' \sim \pi_{\theta_k}(\cdot \mid s_t')$
    3. Solve for $Q_{\omega_k} = \mathrm{NN}(\omega_k; m_Q)$ in (12) (Algorithm 3)
    4. Solve for $f_{\theta_{k+1}} = \mathrm{NN}(\theta_{k+1}; m_f)$ in (11) (Algorithm 2)
    5. Update policy: $\pi_{\theta_{k+1}} \propto \exp\left\{\tau_{k+1}^{-1} f_{\theta_{k+1}}\right\}$
- end for

Definition VIII.1 *For any constant $R > 0$, we define the function class* $\mathcal{F}_{R,m} = \left\{\frac{1}{\sqrt{m}}\sum_{i=1}^m b_i \cdot \nVdash \left\{[\alpha(0)]_i^\top (s,a) > 0\right\} \cdot [\alpha]_i^\top (s,a) : \|\alpha - \alpha(0)\|_2 \leq R\right\}$ *where* $[\alpha(0)]_i$ *and* $b_i (i \in [m])$ *are the random initialization*

- Assumptions
    1. **Bounded Reward:** There exists a constant $R_{\max} > 0$ such that $R_{\max} = \sup_{(s,a)\in\mathcal{S}\times\mathcal{A}} |r(s,a)|$, which implies $|V^\pi(s)| \leq R_{\max}$ and $|Q^\pi(s,a)| \leq R_{\max}$ for any policy π.
    2. **Action-Value Function Class:** It holds that $Q^\pi(s,a) \in \mathcal{F}_{R_Q, m_Q}$ for any π.
    3. **Regularity of Stationary Distribution:** There exists a constant $c > 0$ such that for any vector $z \in \mathbb{R}^d$ and $\zeta > 0$, it holds almost surely that $\mathbb{E}_{\sigma_n}\left[1\left\{\left|z^\top (s,a)\right| \leq \zeta\right\} \mid z\right] \leq c \cdot \zeta / \|z\|_2$ for any π.

VIII.3 Errors of Policy Improvement, Policy Evaluation and Propagation

Theorem VIII.1 *Suppose that Assumptions 1, 2, and 3 hold. We set $T \geq 64$ and the stepsize to be $\eta = T^{-1/2}$. Within the k-th iteration of Algorithm 1, the output $f_{\hat{\theta}}$ of Algorithm 2 satisfies*

$$\mathbb{E}_{init,\, \bar{\sigma}_k}\left[\left(f_{\hat{\theta}}(s,a) - \tau_{k+1}\cdot\left(\beta_k^{-1} Q_{\omega_k}(s,a) + \tau_k^{-1} f_{\theta_k}(s,a)\right)\right)^2\right]$$
$$= O\left(R_f^2 T^{-1/2} + R_f^{5/2} m_f^{-1/4} + R_f^3 m_f^{-1/2}\right)$$

Theorem VIII.2 *Suppose that Assumptions 1, 2, and 3 hold. We set $T \geq 64/(1-\gamma)^2$ and the stepsize to be $\eta = T^{-1/2}$. Within the k-th iteration of Algorithm 1, the output $Q_{\bar{\omega}}$ of Algorithm 3 satisfies*

$$\mathbb{E}_{\textit{init, } \sigma_k}\left[\left(Q_{\bar{\omega}}(s,a) - Q^{\pi_{\theta_k}}(s,a)\right)^2\right] = O\left(R_Q^2 T^{-1/2} + R_Q^{5/2} m_Q^{-1/4} + R_Q^3 m_Q^{-1/2}\right)$$

π^*: Optimal policy.
ν^*: Stationary state distribution under π^*.
σ^*: Stationary state-action distribution under π^*.
π_{k+1}: Improved policy based on $Q^{\pi_{\theta_k}}$, defined as: $\pi_{k+1} = \underset{\pi}{\operatorname{argmax}} \{\mathbb{E}_{\nu_k}[\langle Q^{\pi_{\theta_k}}(s,\cdot), \pi(\cdot,s)\rangle - \beta$
Energy-based policy:

$$\pi_{k+1} \propto \exp\left\{\beta_k^{-1} Q^{\pi_{\theta_k}} + \tau_k^{-1} f_{\theta_k}\right\}$$

$$\phi_k^* = \mathbb{E}_{\widetilde{\sigma}_k}\left[\left|\, \mathrm{d}\sigma^*/\mathrm{d}\widetilde{\sigma}_k - \mathrm{d}\left(\pi_{\theta_k}\nu^*\right)/\mathrm{d}\widetilde{\sigma}_k\right|^2\right]^{1/2}$$
$$\psi_k^* = \mathbb{E}_{\sigma_k}\left[\left|\, \mathrm{d}\sigma^*/\mathrm{d}\sigma_k - \mathrm{d}\nu^*/\mathrm{d}\nu_k\right|^2\right]^{1/2}$$

where $\mathrm{d}\sigma^*/\mathrm{d}\widetilde{\sigma}_k,\ \mathrm{d}\left(\pi_{\theta_k}\nu^*\right)/\mathrm{d}\widetilde{\sigma}_k,\ \mathrm{d}\sigma^*/\mathrm{d}\sigma_k$, and $\mathrm{d}\nu^*/\mathrm{d}\nu_k$ are the Radon-Nikodym derivatives.

Lemma VIII.1 *Suppose that the policy improvement error in Line 4 of Algorithm 1 satisfies*

$$\mathbb{E}_{\tilde{\sigma}_k}\left[\left(f_{\theta_{k+1}}(s,a) - \tau_{k+1}\cdot\left(\beta_k^{-1} Q_{\omega_k}(s,a) - \tau_k^{-1} f_{\theta_k}(s,a)\right)\right)^2\right] \leq \epsilon_{k+1}$$

and the policy evaluation error in Line 3 of Algorithm 1 satisfies

$$\mathbb{E}_{\sigma_k}\left[\left(Q_{\omega_k}(s,a) - Q^{\pi_{\theta_k}}(s,a)\right)^2\right] \leq \epsilon_k'$$

For π_{k+1} and $\pi_{\theta_{k+1}}$ obtained in Line 5 of Algorithm 1, we have

$$\left|\mathbb{E}_{\nu^*}\left[\left\langle \log\left(\pi_{\theta_{k+1}}(\cdot \mid s)/\pi_{k+1}(\cdot \mid s)\right), \pi^*(\cdot \mid s) - \pi_{\theta_k}(\cdot \mid s)\right\rangle\right]\right| \leq \varepsilon_k$$

where $\varepsilon_k = \tau_{k+1}^{-1}\epsilon_{k+1}\cdot\phi_{k+1}^* + \beta_k^{-1}\epsilon_k'\cdot\psi_k^*$.

Lemma VIII.2 *Under the same conditions of last Lemma , we have*

$$\mathbb{E}_{\nu^*}\left[\left\|\tau_{k+1}^{-1} f_{\theta_{k+1}}(s,\cdot) - \tau_k^{-1} f_{\theta_k}(s,\cdot)\right\|_\infty^2\right] \leq 2\varepsilon_k' + 2\beta_k^{-2} M$$

where $\varepsilon_k' = |\mathcal{A}|\cdot\tau_{k+1}^{-2}\epsilon_{k+1}^2$ and $M = 2\mathbb{E}_{\nu^*}\left[\max_{a\in\mathcal{A}}\left(Q_{\omega_0}(s,a)\right)^2\right] + 2R_f^2$.

VIII.4 Global Convergence of Neural PPO

Theorem VIII.3 *Suppose that Assumptions 1, 2 and 3 hold. For the policy sequence $\{\pi_{\theta_k}\}_{k=1}^K$ attained by neural PPO in Algorithm 1, we have*

$$\min_{0\leq k\leq K}\left\{\mathcal{L}\left(\pi^*\right) - \mathcal{L}\left(\pi_{\theta_k}\right)\right\} \leq \frac{\beta^2 \log|\mathcal{A}| + M + \beta^2\sum_{k=0}^{K-1}\left(\varepsilon_k + \varepsilon_k'\right)}{(1-\gamma)\beta\cdot\sqrt{K}}$$

Here $\varepsilon_k = \tau_{k+1}^{-1}\epsilon_{k+1}\cdot\phi_k^* + \beta_k^{-1}\epsilon_k'\cdot\psi_k^*$ and $\varepsilon_k' = |\mathcal{A}|\cdot\tau_{k+1}^{-2}\epsilon_{k+1}^2$, where $\epsilon_{k+1} = O\left(R_f^2 T^{-1/2} + R_f^{5/2} m_f^{-1/4} + R_f^3 m_f^{-1/2}\right), \epsilon_k' = O\left(R_Q^2 T^{-1/2} + R_Q^{5/2} m_Q^{-1/4} + R_Q^3 m_Q^{-1/2}\right)$. Also, we have $M = 2\mathrm{E}_{\nu^*}\left[\max_{a\in\mathcal{A}}\left(Q_{\omega_0}(s,a)\right)^2\right] + 2R_f^2$.

Corollary VIII.1 *Suppose that Assumptions 1, 2 and 3 hold. Let $m_f = \Omega\left(K^6 R_f^{10}\cdot\phi_k^{*4} + K^4 R_f^{10}\cdot|\mathcal{A}|^2\right)$, $m_Q = \Omega\left(K^2 R_Q^{10}\cdot\psi_k^{*4}\right)$, and $T = \Omega\left(K^3 R_f^4\cdot\phi_k^{*2} + K^2 R_f^4\cdot|\mathcal{A}|\right.$ for any $0 \leq k \leq K$. We have*

$$\min_{0\leq k\leq K}\left\{\mathcal{L}\left(\pi^*\right) - \mathcal{L}\left(\pi_{\theta_k}\right)\right\} \leq \frac{\beta^2\log|\mathcal{A}| + M + O(1)}{(1-\gamma)\beta\cdot\sqrt{K}}$$

IX The Surprising Effectiveness of PPO in Cooperative Multi-Agent Games

Proximal Policy Optimization (PPO) has become a leading algorithm in reinforcement learning due to its balance of simplicity and effectiveness.

Multi-agent environments present unique challenges such as coordination and stability, which are critical for effective learning.

While PPO is successful in single-agent settings, its performance and effectiveness in multi-agent scenarios need thorough evaluation.

We aim to assess the performance of PPO and its multi-agent variants IPPO and MAPPO, and identify best practices for applying these algorithms to cooperative multi-agent tasks.

IX.1 Decentralized partially observable Markov decision processes (DEC-POMDP)

Samvelyan et al. [2019] We study decentralized partially observable Markov decision processes (DEC-POMDP) with shared rewards. A DEC-POMDP is defined by $\langle \mathcal{S}, \mathcal{A}, O, R, P, n, \gamma \rangle$.
$\mathcal{S}$ is the state space, and $\mathcal{A}$ is the shared action space for each agent i. $o_i = O(s; i)$ is the local observation for agent i at global state s. $P\left(s' \mid s, A\right)$ denotes the transition probability from s to s' given the joint action $A = (a_1, \ldots, a_n)$ for all n agents. $R(s, A)$ denotes the shared reward function, and γ is the discount factor. Agents use a policy $\pi_\theta\left(a_i \mid o_i\right)$ to produce an action a_i and jointly optimize the discounted accumulated reward $J(\theta) = \mathbb{E}_{A^t, s^t}\left[\sum_t \gamma^t R\left(s^t, A^t\right)\right]$ where $A^t = \left(a_1^t, \ldots, a_n^t\right)$ is the joint action at time step t.
Independent Proximal Policy Optimization : Uses PPO to train local policies π_θ and value functions $V_\phi(s)$ for each agent independently, without access to global information.
Multi-Agent Proximal Policy Optimization : Uses PPO with a centralized policy and value function, where the value function can incorporate global information to optimize the performance of all agents collectively.

Value Clipping: In addition to clipping the policy updates, our methods (IPPO and MAPPO) also uses value clipping to restrict the update of critic function for each agent a to be smaller than ϵ using:

$$\mathcal{L}(\phi) = \mathbb{E}_{s_t}\Bigg[\min\Bigg\{\left(V_\phi\left(s_t\right) - \hat{V}_t\right)^2, \qquad \Big(V_{\phi_{old}}\left(s_t\right) + \operatorname{clip}\left(V_\phi\left(s_t\right) - V_{\phi_{old}}\left(s_t\right), -\epsilon, +\epsilon\right) - \hat{V}$$

Where ϕ_{old} are old parameters before the update

IX.2 Experimental Evaluation in Various Environments

Experimental Setting: We evaluate three cooperative tasks: Spread, Reference, and Comm. For MAPPO and off-policy methods, a global state is created by combining the agents' local observations. Parameter sharing is not used for Comm due to the diversity of the agents.Witt et al. [2020]

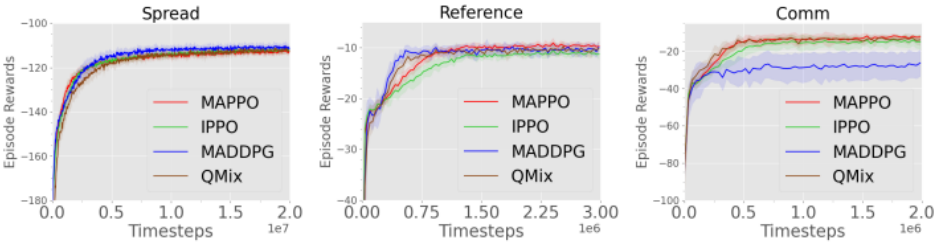
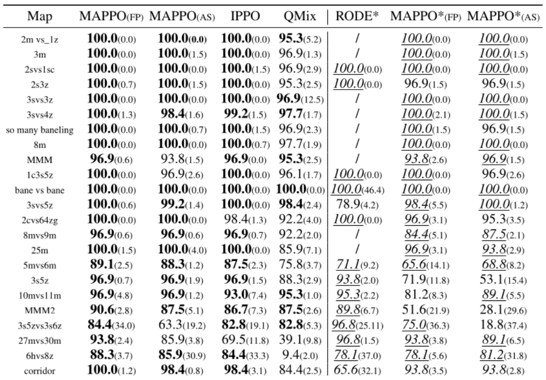


Figure 6: Performance of different algorithms in the MPEs

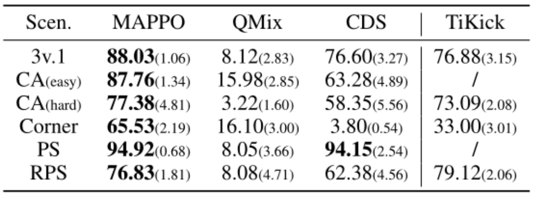
MAPPO and IPPO both show comparable or superior performance to off-policy methods, with MAPPO outperforming IPPO in some tasks.
Experimental Setting: We evaluate three cooperative tasks (Spread, Reference, Comm) using a global state formed by concatenating agents' local observations.



Map	MAPPO(FP)	MAPPO(AS)	IPPO	QMix	RODE*	MAPPO*(FP)	MAPPO*(AS)
2m vs_1z	**100.0**(0.0)	**100.0**(**0.0**)	**100.0**(0.0)	**95.3**(5.2)	/	*100.0*(0.0)	*100.0*(0.0)
3m	**100.0**(0.0)	**100.0**(1.5)	**100.0**(0.0)	96.9(1.3)	/	*100.0*(0.0)	*100.0*(1.5)
2svs1sc	**100.0**(0.0)	**100.0**(0.0)	**100.0**(1.5)	96.9(2.9)	*100.0*(0.0)	*100.0*(0.0)	*100.0*(0.0)
2s3z	**100.0**(0.7)	**100.0**(1.5)	**100.0**(0.0)	95.3(2.5)	*100.0*(0.0)	96.9(1.5)	96.9(1.5)
3svs3z	**100.0**(0.0)	**100.0**(0.0)	**100.0**(0.0)	**96.9**(12.5)	/	*100.0*(0.0)	*100.0*(0.0)
3svs4z	**100.0**(1.3)	**98.4**(1.6)	**99.2**(1.5)	**97.7**(1.7)	/	*100.0*(2.1)	*100.0*(1.5)
so many baneling	**100.0**(0.0)	**100.0**(0.7)	**100.0**(1.5)	96.9(2.3)	/	*100.0*(1.5)	96.9(1.5)
8m	**100.0**(0.0)	**100.0**(0.0)	**100.0**(0.7)	97.7(1.9)	/	*100.0*(0.0)	*100.0*(0.0)
MMM	**96.9**(0.6)	93.8(1.5)	**96.9**(0.0)	**95.3**(2.5)	/	*93.8*(2.6)	*96.9*(1.5)
1c3s5z	**100.0**(0.0)	96.9(2.6)	**100.0**(0.0)	96.1(1.7)	*100.0*(0.0)	*100.0*(0.0)	96.9(2.6)
bane vs bane	**100.0**(0.0)	**100.0**(0.0)	**100.0**(0.0)	**100.0**(0.0)	*100.0*(46.4)	*100.0*(0.0)	*100.0*(0.0)
3svs5z	**100.0**(0.6)	**99.2**(1.4)	**100.0**(0.0)	**98.4**(2.4)	78.9(4.2)	*98.4*(5.5)	*100.0*(1.2)
2cvs64zg	**100.0**(0.0)	**100.0**(0.0)	98.4(1.3)	92.2(4.0)	*100.0*(0.0)	*96.9*(3.1)	95.3(3.5)
8mvs9m	**96.9**(0.6)	**96.9**(0.6)	**96.9**(0.7)	92.2(2.0)	/	*84.4*(5.1)	*87.5*(2.1)
25m	**100.0**(1.5)	**100.0**(4.0)	**100.0**(0.0)	85.9(7.1)	/	*96.9*(3.1)	*93.8*(2.9)
5mvs6m	**89.1**(2.5)	**88.3**(1.2)	**87.5**(2.3)	75.8(3.7)	*71.1*(9.2)	*65.6*(14.1)	*68.8*(8.2)
3s5z	**96.9**(0.7)	**96.9**(1.9)	**96.9**(1.5)	88.3(2.9)	*93.8*(2.0)	71.9(11.8)	53.1(15.4)
10mvs11m	**96.9**(4.8)	**96.9**(1.2)	**93.0**(7.4)	**95.3**(1.0)	*95.3*(2.2)	81.2(8.3)	*89.1*(5.5)
MMM2	**90.6**(2.8)	**87.5**(5.1)	**86.7**(7.3)	**87.5**(2.6)	*89.8*(6.7)	51.6(21.9)	28.1(29.6)
3s5zvs3s6z	**84.4**(34.0)	63.3(19.2)	**82.8**(19.1)	**82.8**(5.3)	*96.8*(25.11)	*75.0*(36.3)	18.8(37.4)
27mvs30m	**93.8**(2.4)	85.9(3.8)	69.5(11.8)	39.1(9.8)	*96.8*(1.5)	*93.8*(3.8)	*89.1*(6.5)
6hvs8z	**88.3**(3.7)	**85.9**(30.9)	**84.4**(33.3)	9.4(2.0)	*78.1*(37.0)	*78.1*(5.6)	*81.2*(31.8)
corridor	**100.0**(1.2)	**98.4**(0.8)	**98.4**(3.1)	84.4(2.5)	*65.6*(32.1)	*93.8*(3.5)	*93.8*(2.8)

Figure 7: Median evaluation win rate and standard deviation on all the SMAC maps for different methods

We observe that IPPO and MAPPO with both the AS and FP inputs achieve strong performance in the vast majority of SMAC.
Experimental Setting: MAPPO was evaluated in several GRF academy scenarios, where a team of agents tries to score against scripted opponents. The results are labeled as "MAPPO" in the tables, even though the agents' local observations make MAPPO and IPPO equivalent. The agents share a single reward, which is the sum of individual rewards. The success rate is measured over 100 game rollouts, and the average success rate from the last 10 evaluations is calculated across 6 seeds.



Scen.	MAPPO	QMix	CDS	TiKick
3v.1	**88.03**(1.06)	8.12(2.83)	76.60(3.27)	76.88(3.15)
CA(easy)	**87.76**(1.34)	15.98(2.85)	63.28(4.89)	/
CA(hard)	**77.38**(4.81)	3.22(1.60)	58.35(5.56)	73.09(2.08)
Corner	**65.53**(2.19)	16.10(3.00)	3.80(0.54)	33.00(3.01)
PS	**94.92**(0.68)	8.05(3.66)	**94.15**(2.54)	/
RPS	**76.83**(1.81)	8.08(4.71)	62.38(4.56)	79.12(2.06)

Figure 8: Average evaluation success rate and standard deviation (over six seeds) on GRF scenarios for different methods.

Experimental Setting: We evaluate MAPPO and IPPO in the full-scale Hanabi game with varying numbers of players (2-5 players). We compare MAPPO and IPPO to strong off-policy methods, namely Value Decomposition Networks (VDN) and Simplified Action Decoder (SAD), a Q-learning variant that has been successful

# Players	Metric	MAPPO	IPPO	SAD	VDN
2	Avg.	23.89(0.02)	**24.00**(0.02)	23.87(0.03)	23.83(0.03)
	Best	**24.23**(0.01)	24.19(0.02)	24.01(0.01)	23.96(0.01)
3	Avg.	**23.77**(0.20)	23.25(0.33)	23.69(0.05)	23.71(0.06)
	Best	**24.01**(0.01)	23.87(0.03)	23.93(0.01)	23.99(0.01)
4	Avg.	**23.57**(0.13)	22.52(0.37)	23.27(0.26)	23.03(0.15)
	Best	23.71(0.01)	23.06(0.03)	**23.81**(0.01)	23.79(0.00)
5	Avg.	**23.04**(0.10)	20.75(0.56)	22.06(0.23)	21.28(0.12)
	Best	**23.16**(0.01)	22.54(0.02)	23.01(0.01)	21.80(0.01)

Figure 9: Best and Average evaluation scores of MAPPO, IPPO, SAD, and VDN on Hanabi-Full. Results are reported over at-least 3 seeds.

in Hanabi. We study the impact of PPO clipping strengths, controlled by the ϵ hyperparameter, in SMAC . Note that ϵ is the same for both policy and value clipping. We generally find that with small ϵ terms such as 0.05, MAPPO's learning speed is slowed in several maps, including hard maps such as MMM2 and 3s5z vs. 3s6z. However, final performance when using $\epsilon = 0.05$ is consistently high and the performance is more stable, as demonstrated by the smaller standard deviation in the training curves. We also observe that large ϵ terms such as 0.2, 0.3, and 0.5, which allow for larger updates to the policy and value function per gradient step, often result in sub-optimal performance.

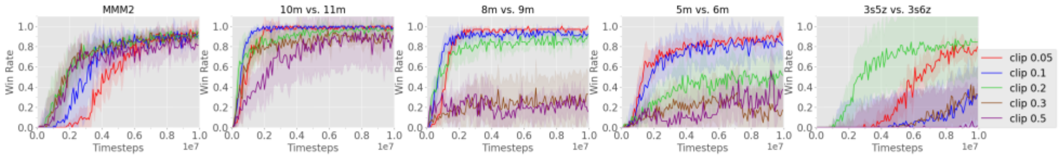


Figure 7: Effect of different clipping strengths on MAPPO's performance in SMAC.

Figure 10: Effect of different clipping strengths on MAPPO's performance in SMAC

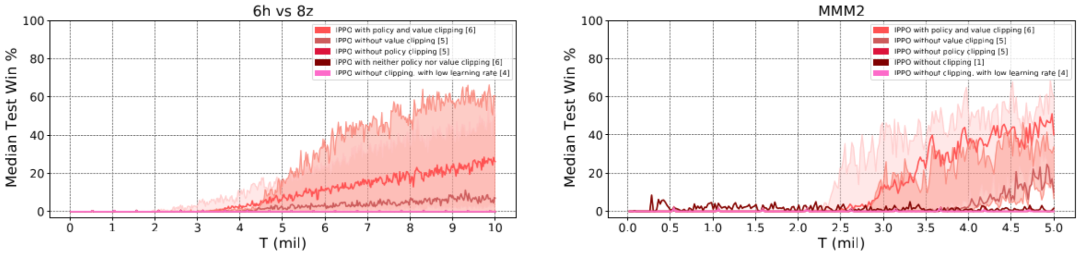


Figure 11: Ablation study for IPPO with different combinations of policy and value clipping

PPO and its variants IPPO and MAPPO achieve strong results on cooperative challenges. Clipping is crucial for stabilizing learning, with smaller values of ϵ enhancing performance

Conclusion

This bibliographic study has highlighted the significant progress made in the field of policy gradient methods for reinforcement learning. In particular, algorithms such as Proximal Policy Optimization (PPO) and its variants have demonstrated high efficiency across various environments, ensuring a good balance between stability and performance.

The reviewed works show that improvements, such as the integration of trust regions and regularization mechanisms, help address some of the limitations of the original algorithms, particularly in terms of robustness and convergence. While these results are promising, they also pave the way for future research to further refine these methods and adapt them to more complex environments.

This review of the literature shows that policy gradient methods, and in particular PPO, represent a key approach for reinforcement learning applications. However, many challenges remain, particularly in optimizing in multi-agent environments and handling high-dimensional state and action spaces.

Bibliography

G. Chen, Y. Peng, and M. Zhang. An adaptive clipping approach for proximal policy optimization. *ArXiv preprint arXiv:1804.06461*, 2018.

B. Liu, Q. Cai, Z. Yang, and Z. Wang. Neural trust region/proximal policy optimization attains globally optimal policy. In *Advances in Neural Information Processing Systems*, 2019.

M. Samvelyan, T. Rashid, and C. S. De Witt. Arxiv preprint arxiv:2019. 2019.

B. Scherrer. Approximate policy iteration schemes: A comparison. In *International Conference on Machine Learning*, 2014.

J. Schulman. Arxiv preprint arxiv:1502.05477. 2015.

J. Schulman, F. Wolski, P. Dhariwal, and A. Radford. Proximal policy optimization algorithms. *ArXiv preprint arXiv:2017*, 2017.

Y. Wang, H. He, and X. Tan. Truly proximal policy optimization. In *Uncertainty in artificial intelligence*, 2020.

C. S. De Witt, T. Gupta, D. Makoviichuk, V. Makoviychuk, P. H. S. Torr, M. Sun, and S. Whiteson. Arxiv preprint arxiv:2011.09533. 2020.

W. Zhu and A. Rosendo. A functional clipping approach for policy optimization algorithms. *IEEE Access*, 2021.