IPO

: Interior-point Policy Optimization under Constraints

Trust RL 사이드 프로젝트
논문리뷰 스터디

2024.10.06
이정연

Introduction

제약(Constraint)과 함께 RL을 정의: **Constrained Markov Decision Process (CMDP)**

- 1999년도 발표된 논문에서 시작
- 2가지 종류의 제약이 있음
  - **instantaneous** constraint
    - 일시적인
    - 각 timestep마다 실행된 action에 의한 제약
  - **cumulative** constraints
    - 누적적인
    - 특정 시간 동안(e.g. 해당 episode length동안) 누적되어 계산된 제약
    - IPO논문에서는 cumulative constraints의 **discounted cumulative constraints**와 **mean valued constraints**에 초점을 맞춰 진행

일반적으로 CMDP를 어떻게 풀고 있었나?

- **Lagrangian relaxation method**
  - 최적화 식의 목적 함수에 해당하는 **라그랑주 승수(Lagrange multipliers)로 가중치를 둔 제약 함수들의 합**을 추가
  - 라그랑주 승수는 제약 조건을 만족시키기 위해 dual problem에서 업데이트

https://convex-optimization-for-all.github.io/contents/chapter24/2021/03/28/24_04_Relaxations/
https://optboy.github.io/optimization/2020/04/27/duality.html

Introduction

앞서 **Constrained policy optimization (CPO)**가 CMDP를 풀기 위해 제안 되었었다.

- TRPO에서 제약 조건들을 만족하기 위해 확장된 아이디어
- 제약조건들이 한번만 만족되면, 모든 training 과정에서 제약 조건의 만족이 성립
- **하지만 이차미분을 계산해야 하는 계산적 어려움이 있었음**
- **mean valued constraints를 다룰 수 없음**
- **여러개 constraints들을 다룰 수 없음**

그렇다면 IPO는?

- **1차 최적화**
- **다양한 누적 제약 조건들을 포함할 수 있음**
    - 목적식에 제약들을 포함시키기 위해 **logarithmic barrier function** 사용
        - 제약 조건이 만족되면 보상 함수에 추가되는 페널티는 0
        - 제약 조건이 만족되지 않으면 패널티는 -무한
- **PPO를 활용**하여 trust region 성격을 반영

Preliminaries

Constrained Markov Decision Process (CMDP)

사실상 reward랑 매우 비슷한 constraint $C: S \times A \times S \mapsto \mathbb{R}$

several cost functions $\widetilde{C_i} = f\left(C\left(s_0, a_0, s_1\right)\right), \ldots, C\left(s_n, a_n, s_{n+1}\right)\right)$ (n:transition 수, t-th)

constraint **limits** $\epsilon_1, \ldots, \epsilon_m$ (m:constraint 종류 가짓수, i-th)

Expectation over a constraint $J_{C_i}^{\pi_\theta} = \mathbb{E}_{\tau \sim \pi_\theta}\left[\widetilde{C_i}\right]$

$$J_{C_i}^{\pi_\theta} = \mathbb{E}_{\tau \sim \pi_\theta}\left[\sum_{t=0}^{\infty} \gamma^t C\left(s_t, a_t, s_{t+1}\right)\right]$$

discounted cumulative constraint

$$J_{C_i}^{\pi_\theta} = \mathbb{E}_{\tau \sim \pi_\theta}\left[\frac{1}{T} \sum_{t=0}^{T-1} C\left(s_t, a_t, s_{t+1}\right)\right]$$

mean valued constraint

Goal

$$\max_\theta J_R^{\pi_\theta}$$

$$\text{s.t.} \quad J_{C_i}^{\pi_\theta} \leq \epsilon_i$$

Preliminaries

Goal

$$\max_\theta J_R^{\pi_\theta}$$
$$\text{s.t.} \quad J_{C_i}^{\pi_\theta} \le \epsilon_i$$

Policy Gradient Methods

$$A_R^{\pi_\theta}(s,a) = Q_R^{\pi_\theta}(s,a) - V_R^{\pi_\theta}(s)$$

gradient of the objective

$$\nabla J_R^{\pi_\theta} = \mathbb{E}_t\left[\nabla_\theta \log \pi_\theta\left(a_t \mid s_t\right) A_t\right]$$

Trust Region Policy Optimization (TRPO)

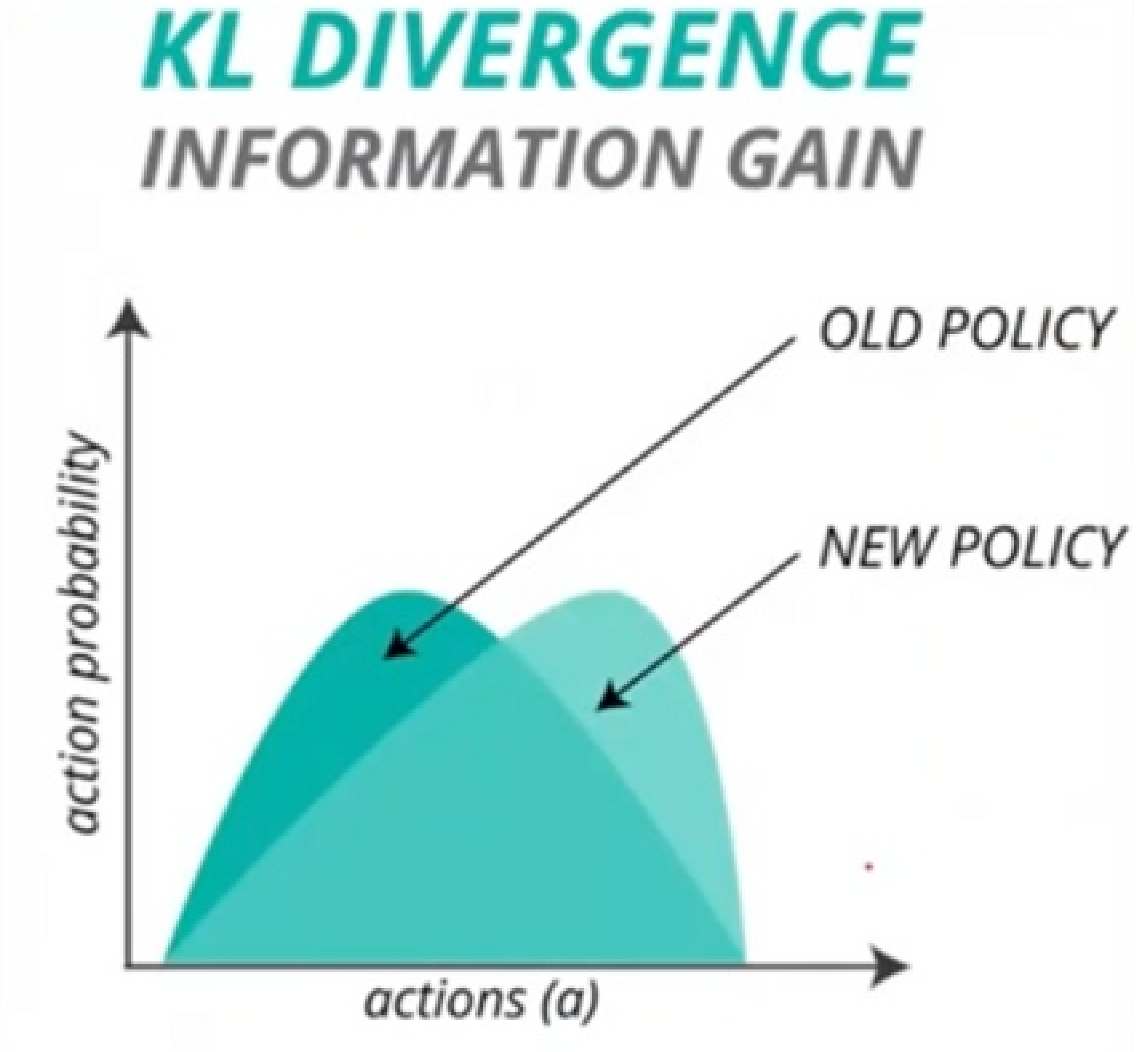
$$\max_\theta L^{TRPO}(\theta) = \mathbb{E}_t\left[\frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{\text{old}}}(a_t|s_t)} A_t\right]$$
$$\text{s.t.} \quad \mathbb{E}_t\left[KL\left[\pi_{\theta_{\text{old}}}\left(a_t \mid s_t\right), \pi_\theta\left(a_t \mid s_t\right)\right]\right] \le \delta$$

Proximal Policy Optimization (PPO)

$$\max_\theta L^{CLIP}(\theta)$$
$$= \mathbb{E}_t\left[\min\left(r_t(\theta) A_t, \text{clip}\left(r_t(\theta), 1, 1-\epsilon, 1+\epsilon\right) A_t\right)\right]$$

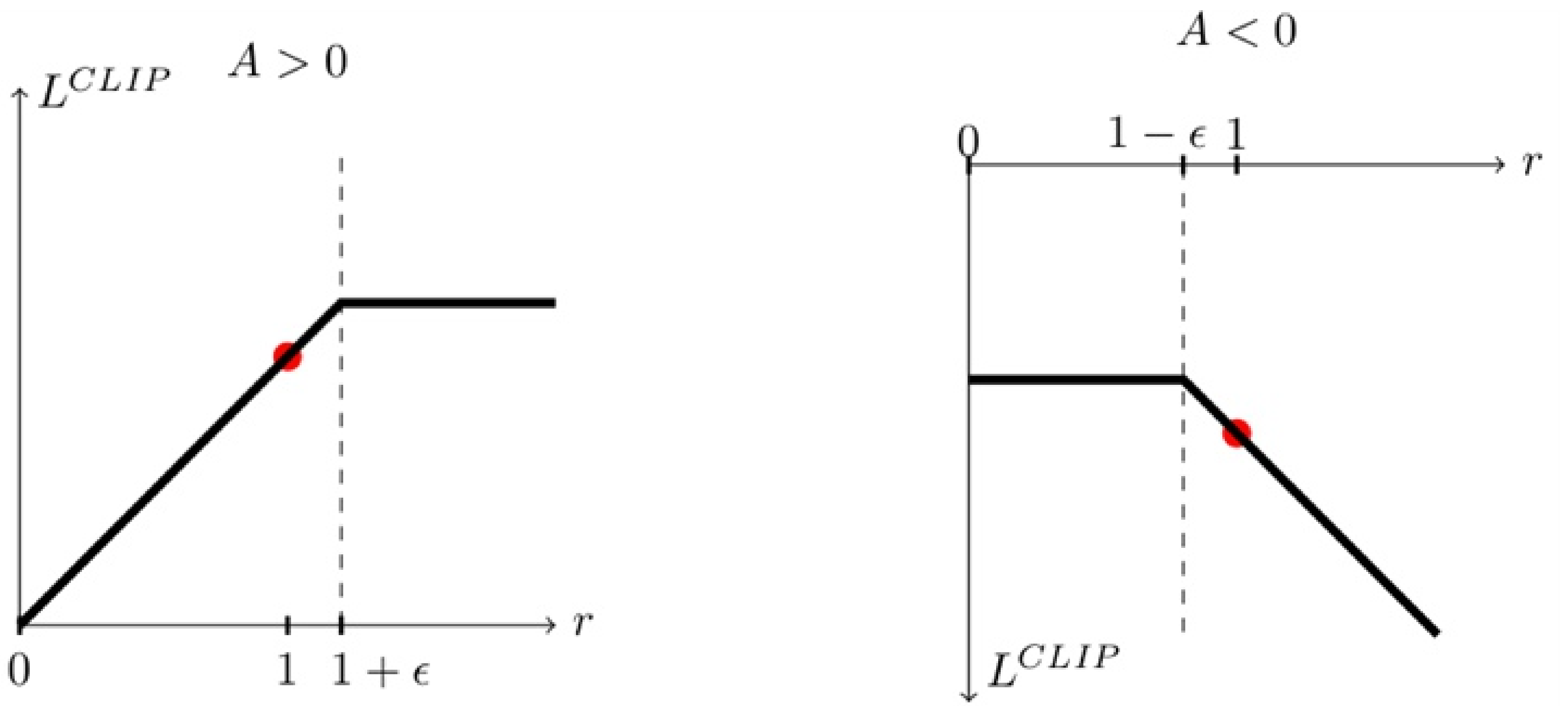
$$r_t(\theta) = \frac{\pi_\theta\left(a_t \mid s_t\right)}{\pi_{\theta_{\text{old}}}\left(a_t \mid s_t\right)}$$

Trust Region Policy Optimization (TRPO)



$$D_{KL}(\pi||\pi') = \pi(a)log\frac{\pi(a)}{\pi'(a)}$$

Proximal Policy Optimization (PPO)



Interior-point Policy Optimization

Interior-point Policy Optimization

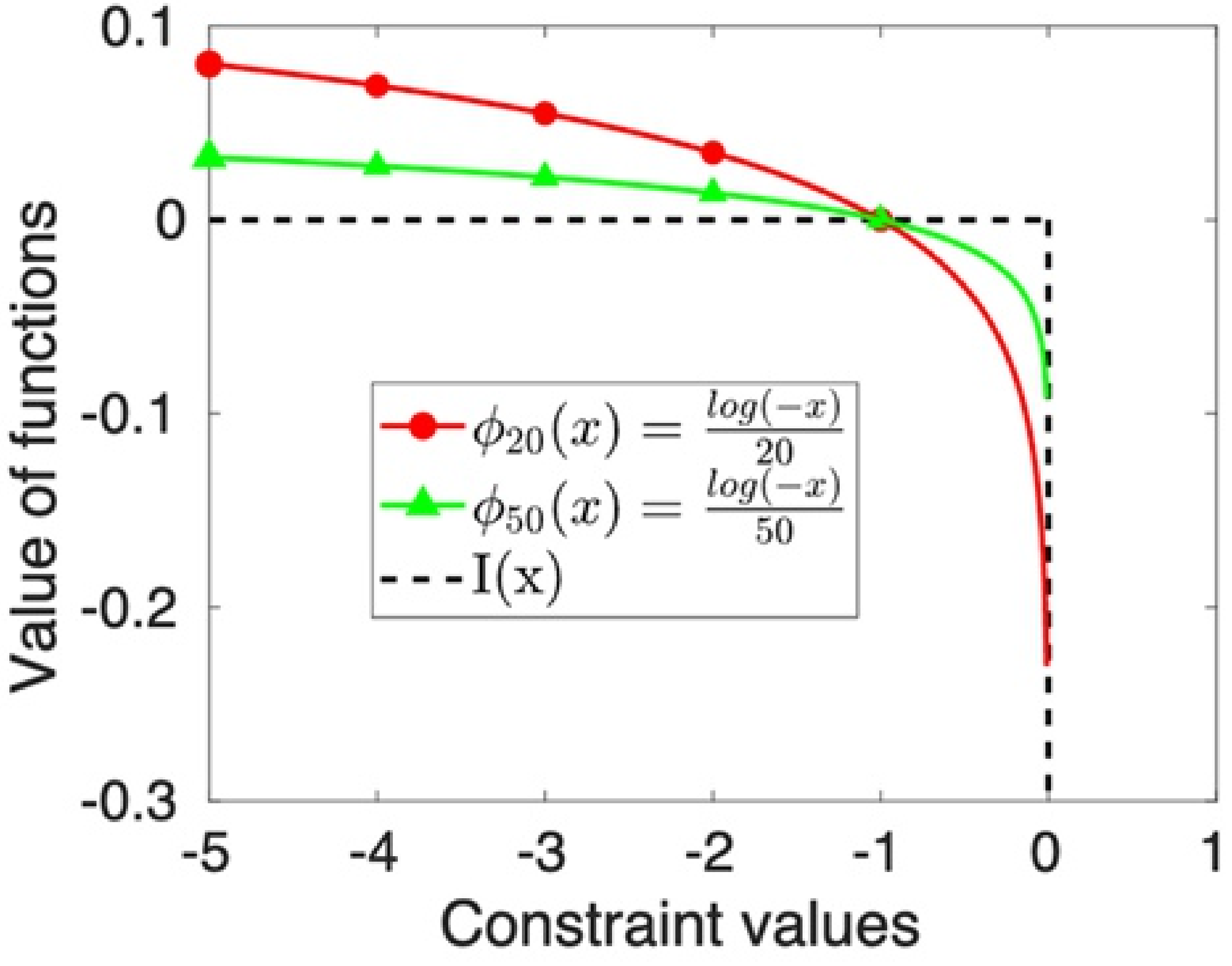
$$\phi(x) = \frac{-\log(-x)}{t}$$

Problem Deifinition

$$\max_{\theta} L^{CLIP}(\theta)$$ PPO의 목적식

$$\text{s.t.} \quad J_{C_i}^{\pi_\theta} \leq \epsilon_i$$

$$\widehat{J}_{C_i}^{\pi_\theta} = J_{C_i}^{\pi_\theta} - \epsilon_i$$



Indicator Function

$$I\left(\widehat{J}_{C_i}^{\pi_\theta}\right) = \begin{cases} 0 & \widehat{J}_{C_i}^{\pi_\theta} \leq 0 \\ -\infty & \widehat{J}_{C_i}^{\pi_\theta} > 0 \end{cases}$$

The larger t is, the better the approximation is to the indicator function

logarithmic barrier function

: differentiable approximation of the indicator function

$$\phi\left(\widehat{J}_{C_i}^{\pi_\theta}\right) = \frac{\log\left(-\widehat{J}_{C_i}^{\pi_\theta}\right)}{t}$$ *hyperparameter

Interior-point Policy Optimization

그래서 정리된 IPO의 objective Function

$$\max_{\theta} L^{IPO}(\theta)$$

$$L^{IPO}(\theta) = L^{CLIP}(\theta) + \sum_{i=1}^{m} \phi\left(\widehat{J}_{C_i}^{\pi_\theta}\right)$$

Algorithm 1 The procedure of IPO

Input: Initialize policy π with parameter $\theta = \theta_0$. Set the hyperparameter r for PPO clip rate and t for logarithmic barrier function

Output: The policy parameters θ

1: Initialize the computational graph structure.
2: **for** iteration k=0,1,2,... **do**
3: Sample N trajectories $\tau_1, ..., \tau_N$ including observations, actions, rewards and costs under the current policy θ_k
4: Process the trajectories to advantages, constraint values, etc
5: Update the policy parameter with first order optimizer $\theta_{k+1} = \theta_k + \alpha \bigtriangledown_\theta L^{IPO}(\theta)$ where α is learning rate based on the processed trajectories.
6: **end for**
7: **return** policy parameters $\theta = \theta_{k+1}$

Performance Guarantee Bound

그렇다면 IPO의 성능 보장을 이론적으로 검증해보자.

Maximization을 Minimization으로 바꾸기 위해 (-) 붙이기

$$\min_\theta -L^{CLIP}(\theta)$$
$$\text{s.t. } \widehat{J}_{C_i}^{\pi_\theta} \leq 0$$

1 Minimization 버전의 IPO 목적식

$$\min_\theta -L^{CLIP}(\theta) - \sum_{i=1}^{m} \frac{\log\left(-\widehat{J}_{C_i}^{\pi_i}\right)}{t}$$

2 라그랑지안 방법으로 했을 때 목적식

$$L(\theta, \lambda_i) = -L^{CLIP}(\theta) + \sum_{i=1}^{m} \lambda_i \widehat{J}_{C_i}^{\pi_\theta}$$

dual function $g(\lambda_i) = \min_\theta -L^{CLIP}(\theta) + \sum_{i=1}^{m} \lambda_i \widehat{J}_{C_i}^{\pi_\theta}$

Performance Guarantee Bound

1의 미분후 optimal θ⋆를 대입 $-\nabla L^{CLIP}(\theta^*) + \sum_{i=1}^{m} \boxed{\frac{1}{-t \times \widehat{J}_{C_i}^{\pi_{\theta^*}}}} \nabla \widehat{J}_{C_i}^{\pi_{\theta^*}} = 0$

$y = \log(-x) \Rightarrow y' = -\frac{1}{x}$

$\lambda_i^* = -\frac{1}{t \times \widehat{J}_{C_i}^{\sigma_i}}$

$-\nabla L^{CLIP}(\theta^*) + \sum_{i=1}^{m} \lambda_i^* \nabla \widehat{J}_{C_i}^{\pi_\theta^*} = 0$ 치환하고 나니 2의 미분 형태와 동일

$\lambda_i = \lambda_i^*$

그렇다면 dual function은 $\lambda_i = \lambda_i^*$ 일 때

$$g(\lambda_i^*) = -L^{CLIP}(\theta^*) + \sum_{i=1}^{m} \lambda_i^* \widehat{J}_{C_i}^{\pi_\theta}$$

$$= -L^{CLIP}(\theta^*) - \frac{m}{t}$$

대입 $\lambda_i^* = -\frac{1}{t \times \widehat{J}_{C_i}^{\sigma_i}}$

$p^* \geq g(\lambda^*)$ duality gap 특성에 의해 $-L^{CLIP}(\theta^*) - p^* \leq \frac{m}{t}$

Performance Guarantee Bound

그래서 결론의 의미가 뭔가요?

$$-L^{CLIP}(\theta^*) - p^* \leq \frac{m}{t}$$

the gap between the optimal value of the original constrained problem with clipped surrogate function (Eq. (7)) and IPO (Eq. (8)) is bounded by m/t

즉, PPO의 최적화 문제-Eq. (7)와 IPO의 최적화 문제-Eq. (8)을 사용한 결과값 간의 차이가 일정한 한계 내에서 유지된다는 것을 의미. 이는 두 방법을 사용할 때 최적 해의 성능 차이가 매우 크지 않음(bounded)을 보장함

검증된 Theorem1으로 알 수 있는 것은?

더 큰 t 값이 원래의 목표 함수에 대한 더 나은 근사를 제공한다는 것을 알 수 있다.

- **Larger t, Higher reward and cost, BUT Lower convergence rate**
- 이러한 단조성(monotonicity)을 이용하여, 수렴 속도와 최적화 성능 사이의 균형을 맞출 수 있는 적절한 t 값을 찾기 위해 **이진 탐색 알고리즘(binary search)을 사용할 수 있게 됨**

Experiment

Experiments

$$J_{C_i}^{\pi_\theta} = \mathbb{E}_{\tau\sim\pi_\theta}\left[\sum_{t=0}^{\infty}\gamma^t C\left(s_t, a_t, s_{t+1}\right)\right] \qquad J_{C_i}^{\pi_\theta} = \mathbb{E}_{\tau\sim\pi_\theta}\left[\frac{1}{T}\sum_{t=0}^{T-1} C\left(s_t, a_t, s_{t+1}\right)\right]$$

실험에서 확인할 수 있는 **IPO의 장점(특성)**

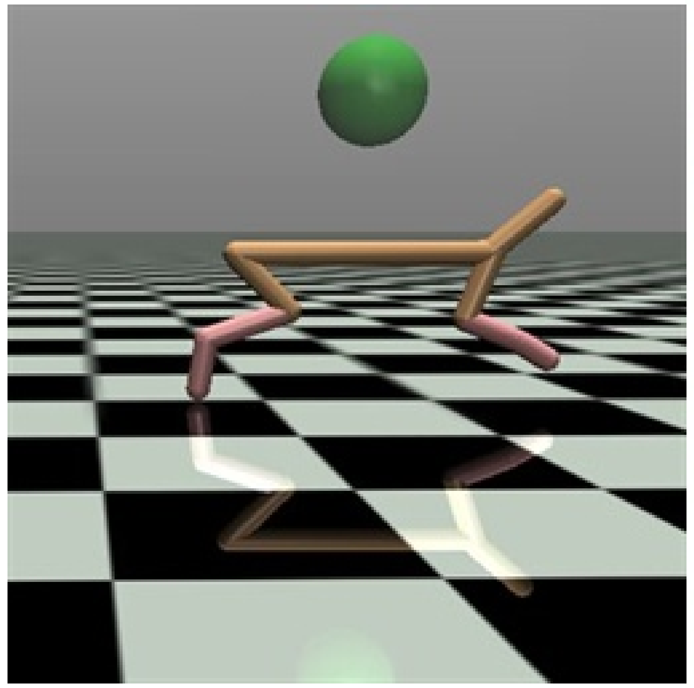
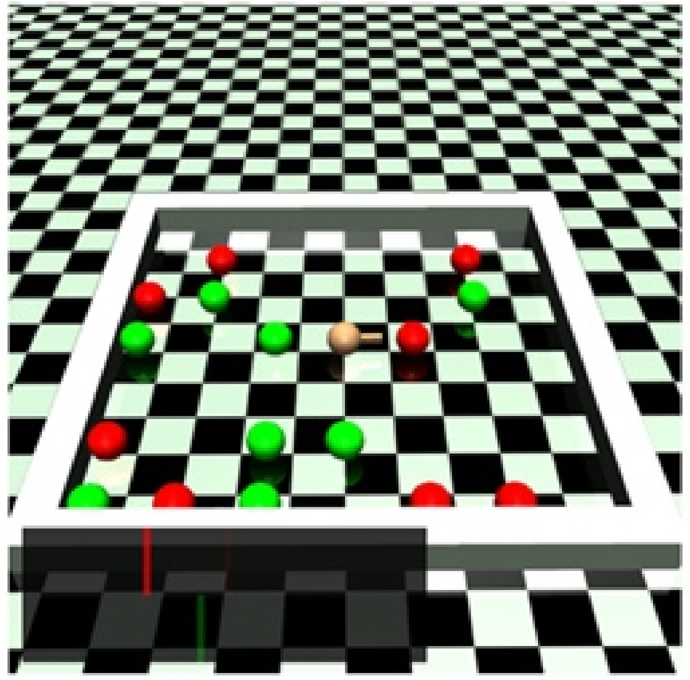
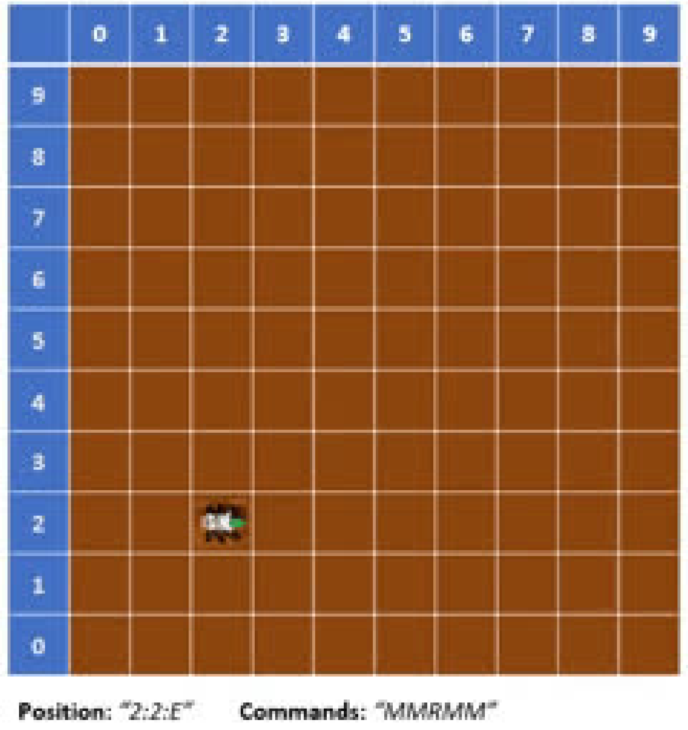
- can handle more general types of cumulative constraints including discounted cumulative constraints and mean valued constraints
- hyperparameter is easy to tune
- can be easily extended to handle optimizations with multiple constraints
- robust in stochastic environments

Baselines

- CPO - Constrained Policy Optimization
- PDO - Primal-dual optimization

Tasks

- Point-Gather, Point-Circle(Mujoco)
- HalfCheetah-Safe
- grid-world task

Experiments

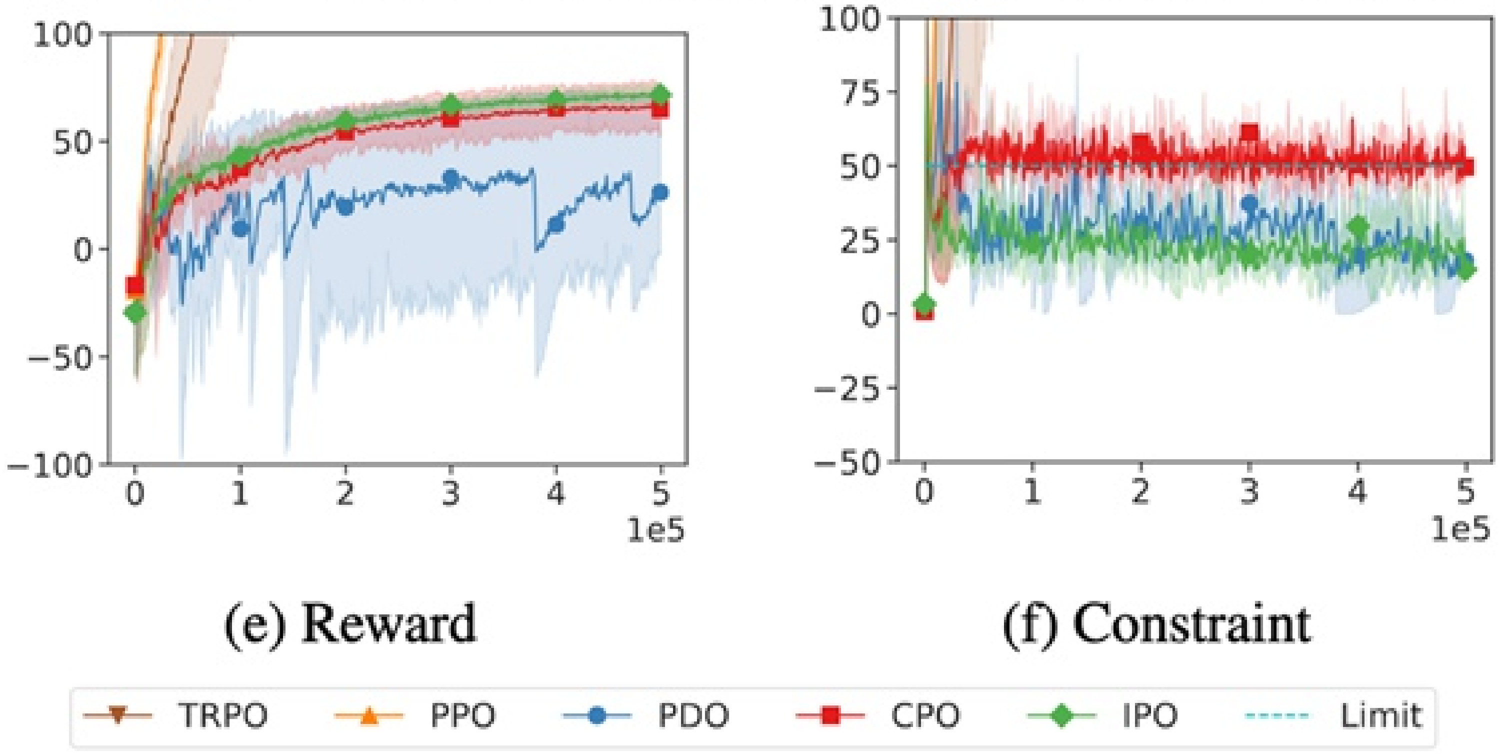
$$J_{C_i}^{\pi_\theta} = \mathbb{E}_{\tau \sim \pi_\theta}\left[\sum_{t=0}^{\infty} \gamma^t C\left(s_t, a_t, s_{t+1}\right)\right]$$

1 discounted cumulative constraints

IPO vs. CPO

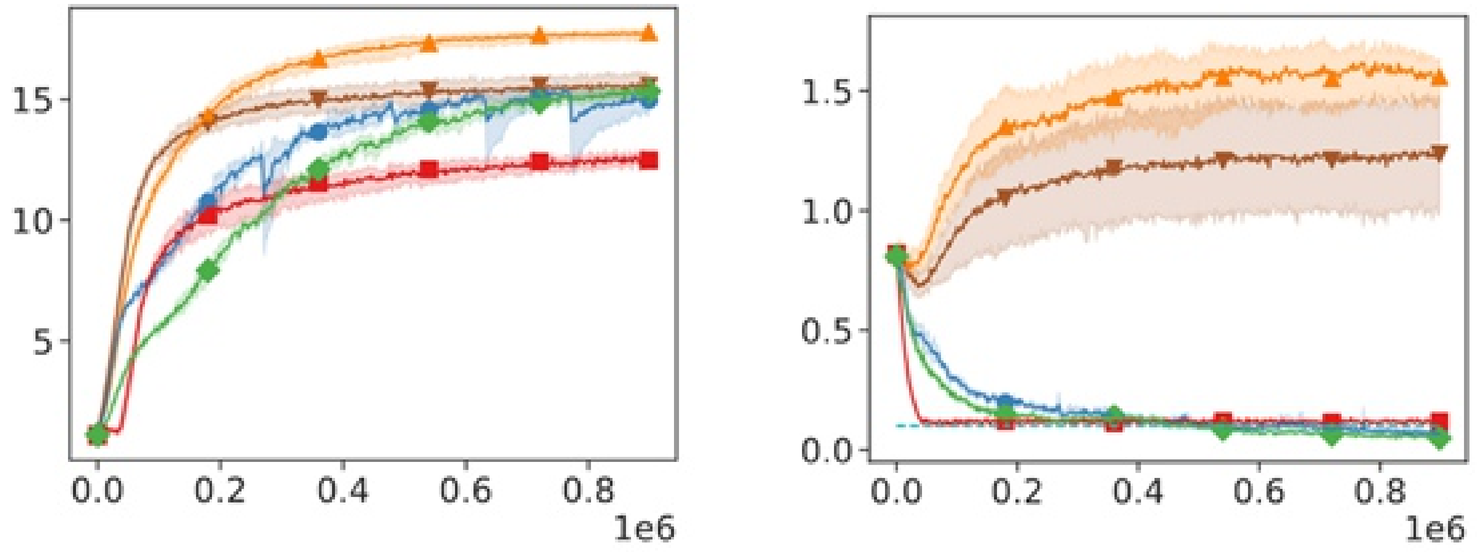
- IPO is best performance
- CPO converges faster than IPO
- CPO always stops improving when the constraint is satisfied
- IPO continues to search for a better policy even if the constraint is satisfied. Hence, it converges to a better reward and lower cost.

HalfCheetah-Safe with discounted cumulative constraint



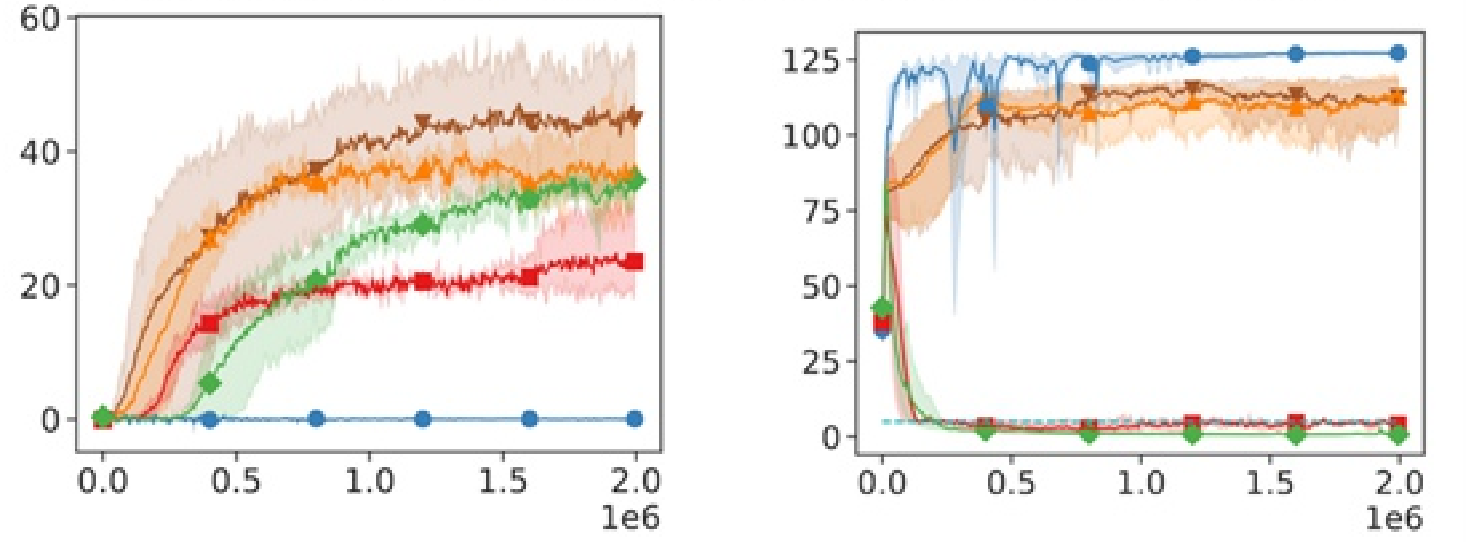
(e) Reward (f) Constraint

Point Gather with discounted cumulative constraint



(a) Reward (b) Constraint

Point Circle with discounted cumulative constraint



(c) Reward (d) Constraint

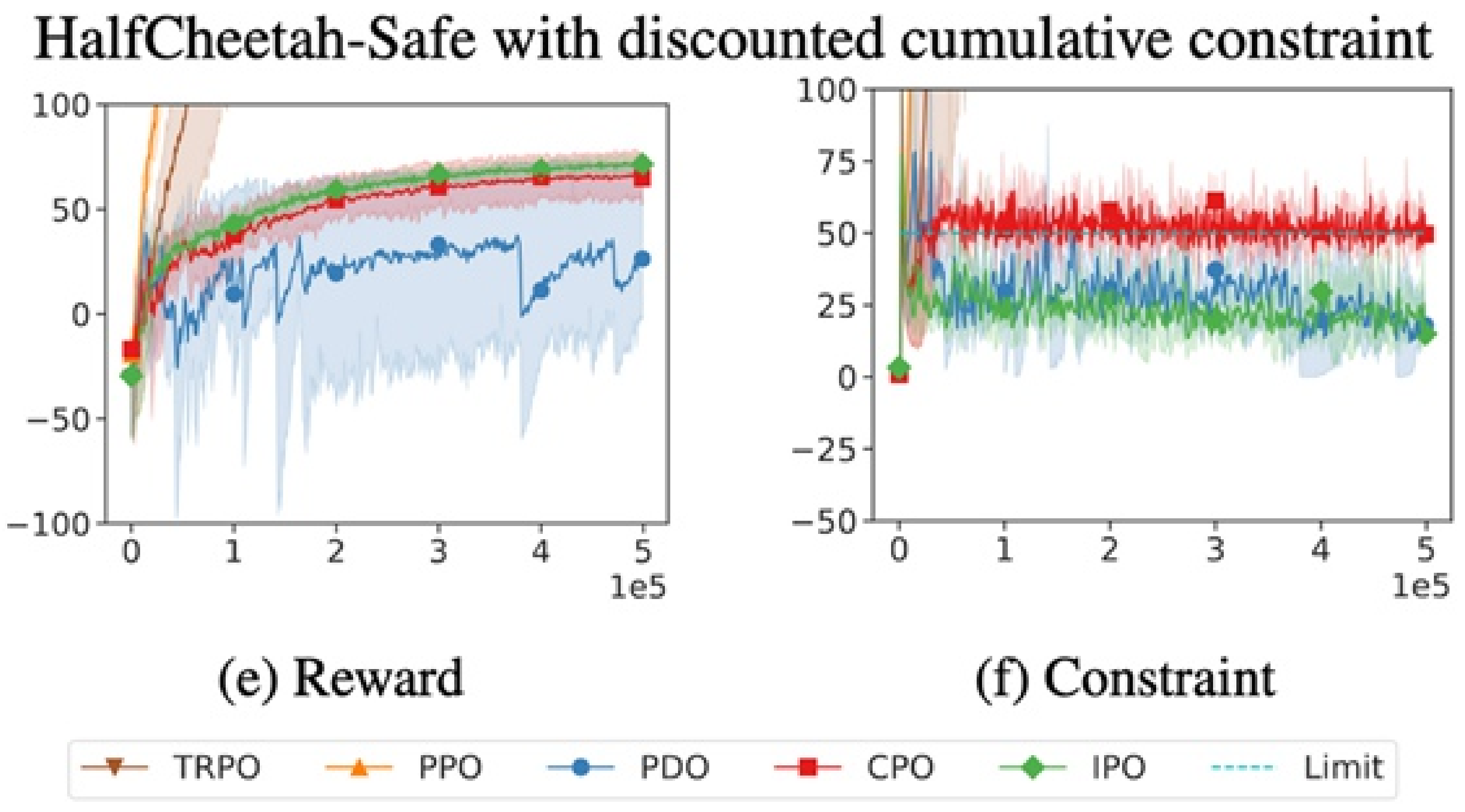
Experiments

$$J_{C_i}^{\pi_\theta} = \mathbb{E}_{\tau\sim\pi_\theta}\left[\sum_{t=0}^{\infty}\gamma^t C\left(s_t, a_t, s_{t+1}\right)\right]$$

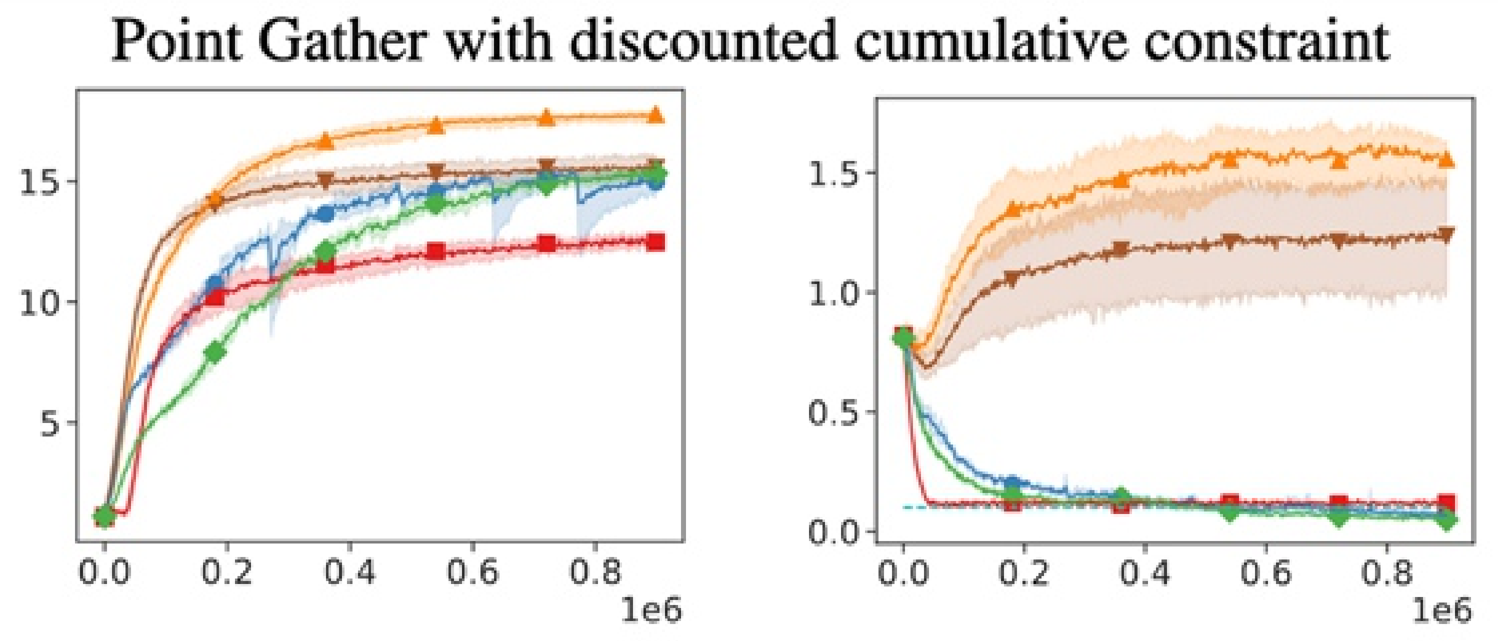
1 discounted cumulative constraints

IPO vs. PDO

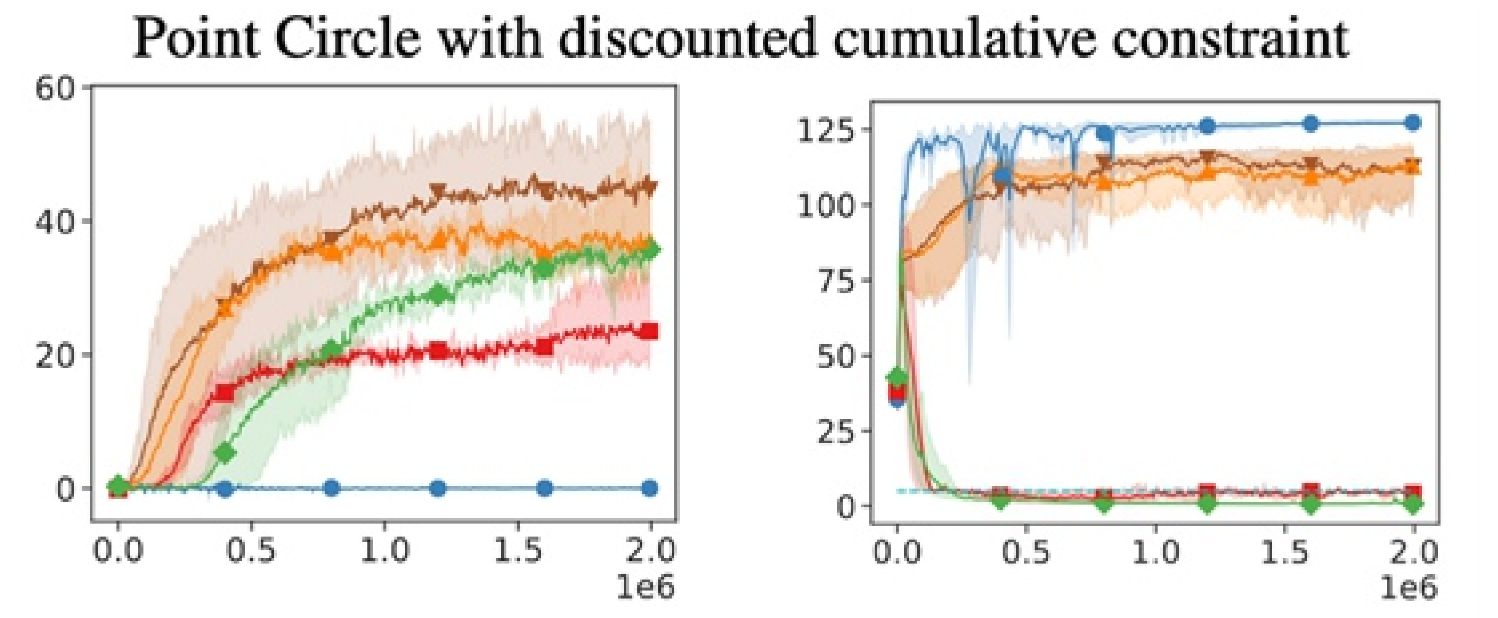
- PDO can converge to a policy as good as IPO, however, the variance of the performance during training is **high**
- PDO achieves a policy whose constraint value is lower than the limit, but the reward is the lowest as well.
- PDO is sensitive to the initialization of the Lagrange multiplier and learning rate



(e) Reward (f) Constraint



(a) Reward (b) Constraint



(c) Reward (d) Constraint

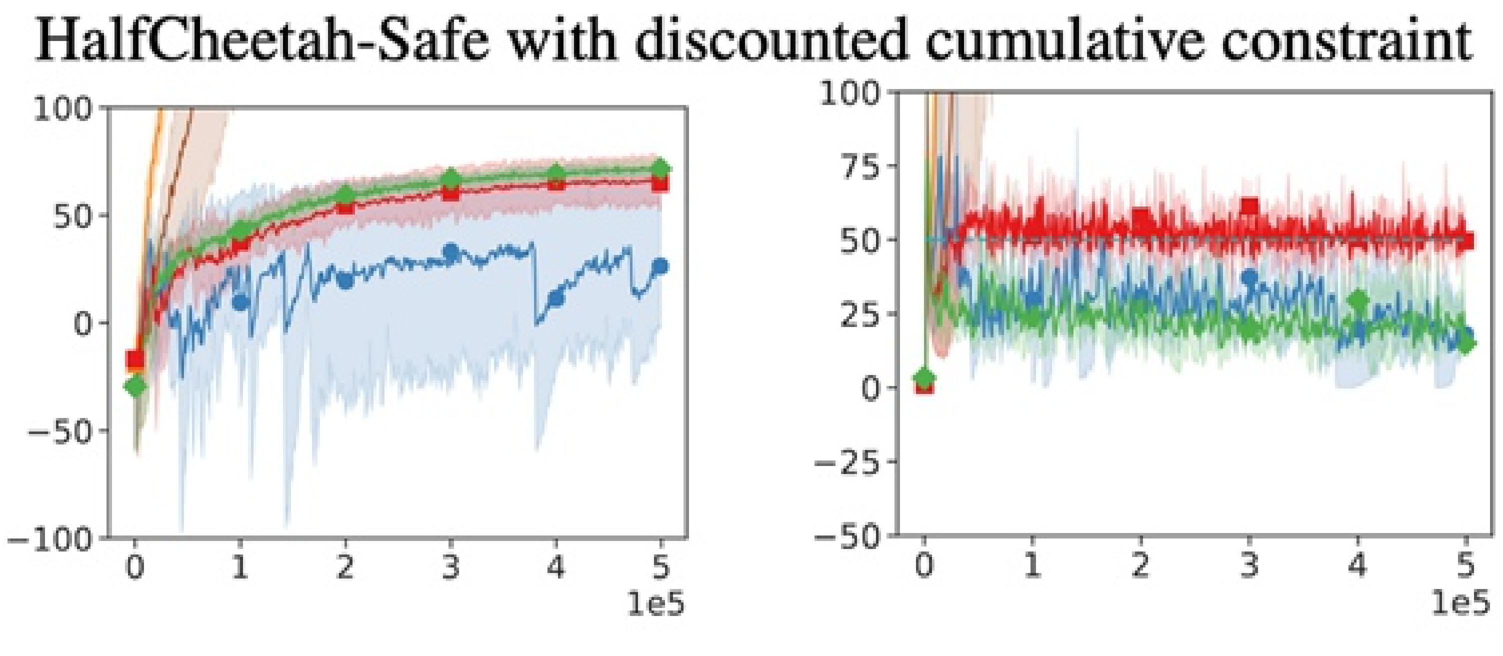
Experiments

$$J_{C_i}^{\pi_\theta} = \mathbb{E}_{\tau \sim \pi_\theta}\left[\sum_{t=0}^{\infty} \gamma^t C\left(s_t, a_t, s_{t+1}\right)\right]$$

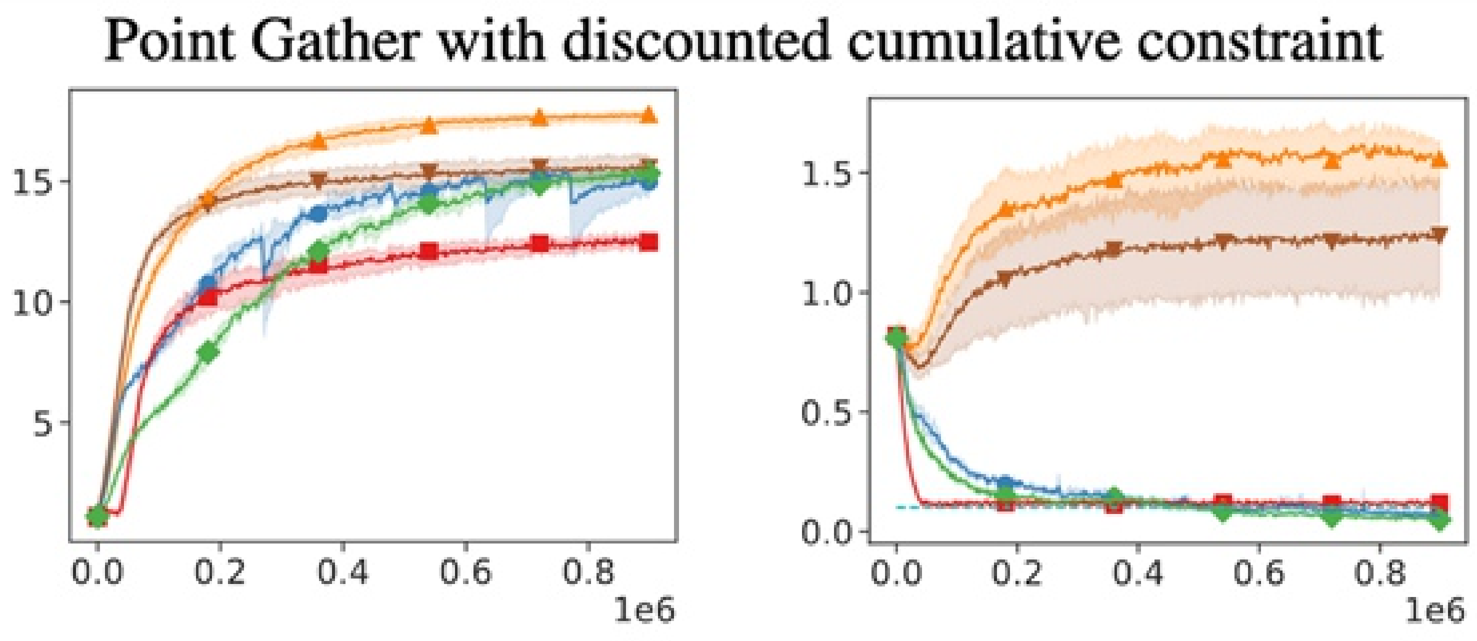
1 discounted cumulative constraints

CPO vs. PPO / TRPO

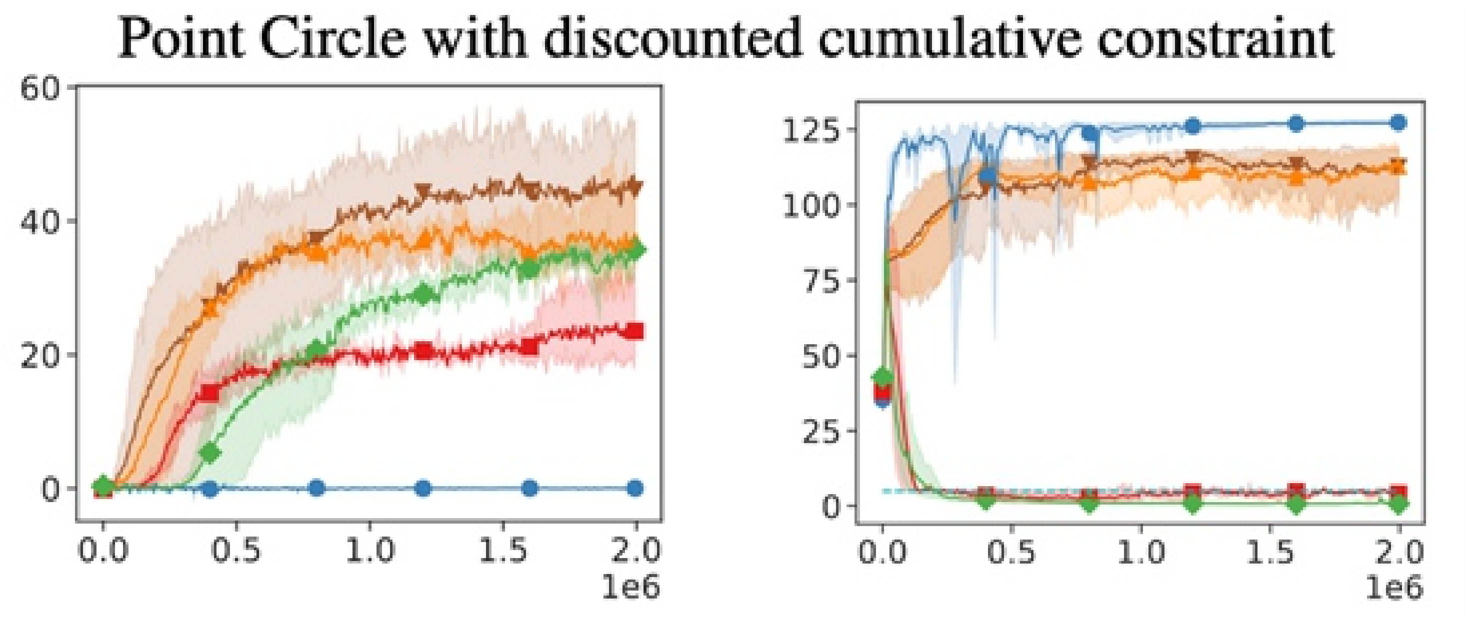
- PPOs consider the optimization without constraints.
- PPOs achieve higher rewards as well as violating the constraints more, compared to IPO, CPO and PDO



(e) Reward (f) Constraint



(a) Reward (b) Constraint



(c) Reward (d) Constraint

Experiments

CPO does not support mean valued constraints

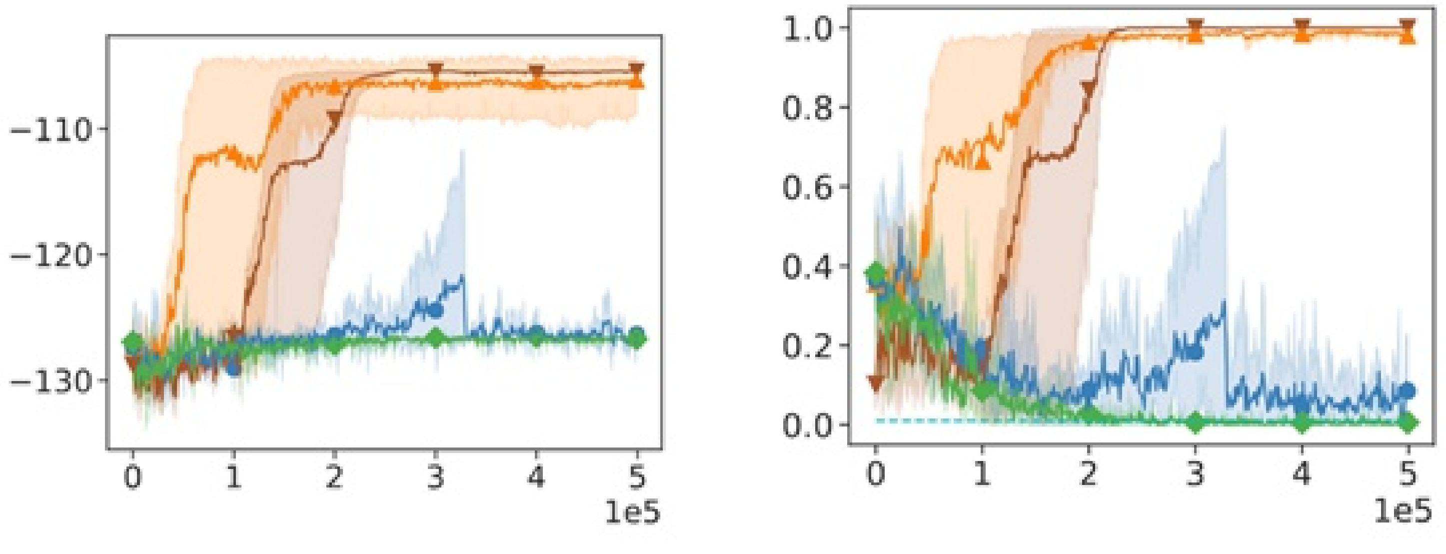
$$J_{C_i}^{\pi_\theta} = \mathbb{E}_{\tau\sim\pi_\theta}\left[\frac{1}{T}\sum_{t=0}^{T-1} C\left(s_t, a_t, s_{t+1}\right)\right]$$

2 mean valued constraints

IPO vs. PDO

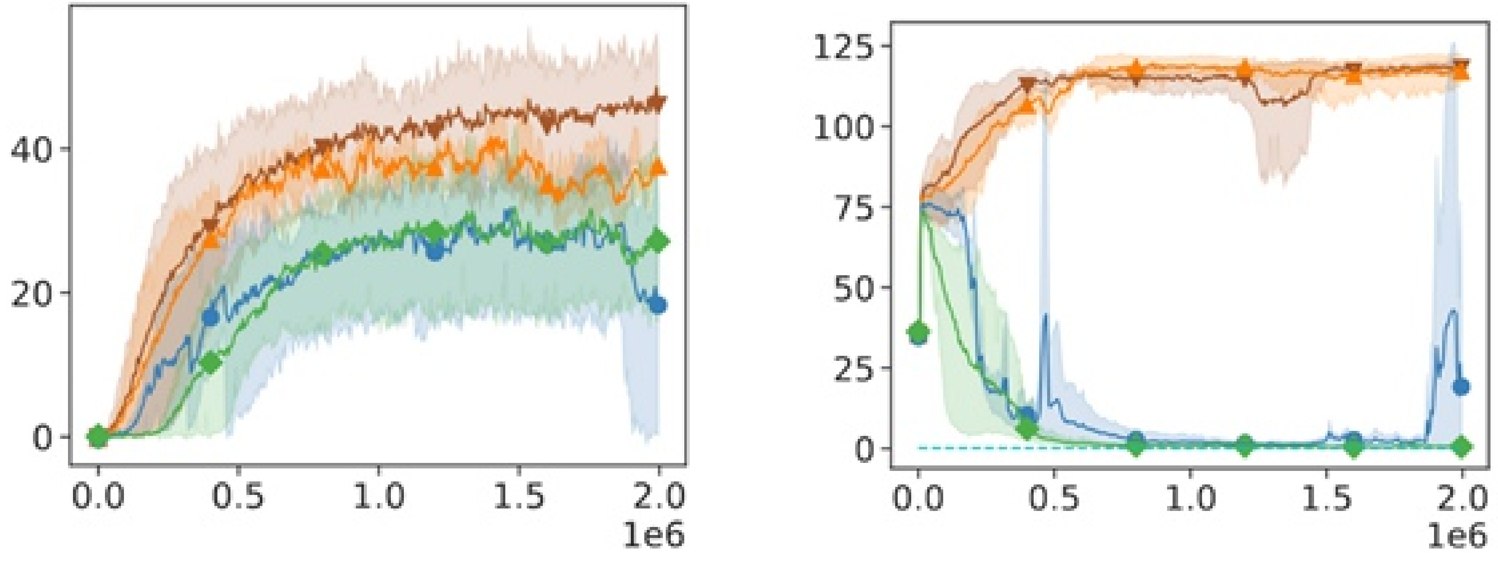
- IPO can consistently converge to a policy with high discounted cumulative reward and satisfy the mean valued constrains on all tasks.
- PDO, however, sometimes converges to a policy **violating the constraints** (Figure 3b) and **has a higher variance** during training (Figure 3d and Figure 3f)

Mars Rover with mean valued constraint



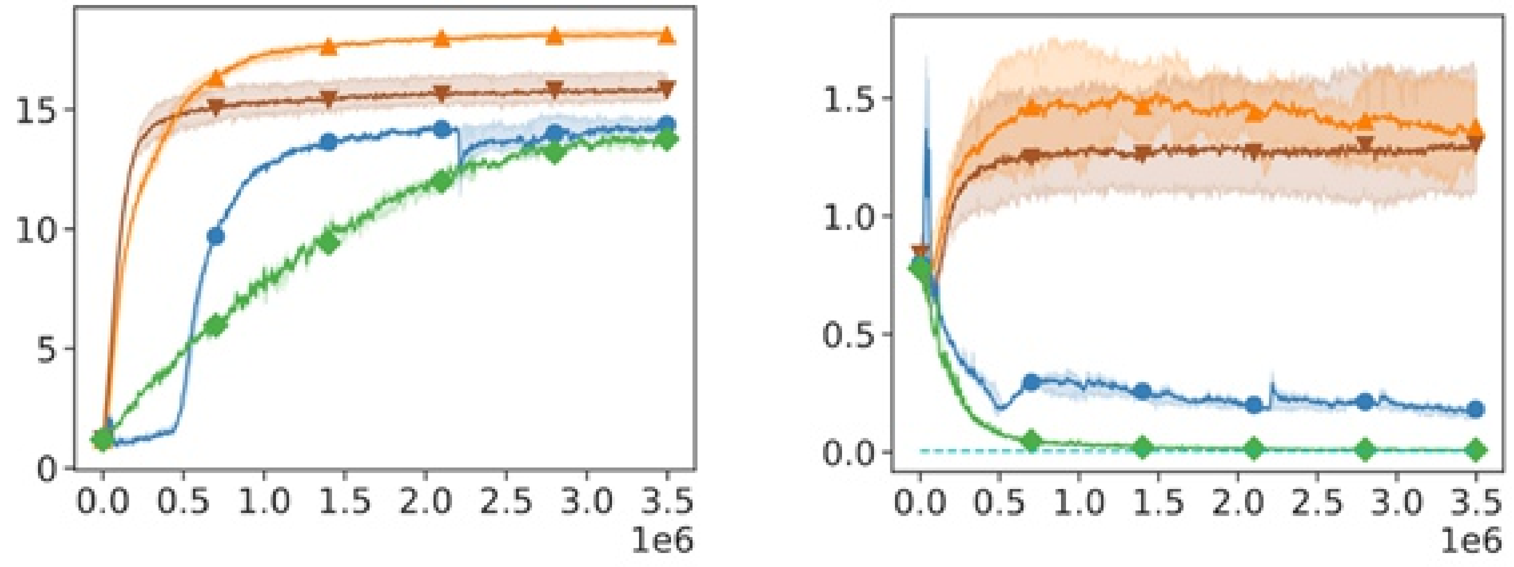
(e) Reward (f) Constraint

Point Circle with mean valued constraint



(c) Reward (d) Constraint

Point Gather with mean valued constraint



(a) Reward (b) Constraint

Experiments

3️⃣ Constraint Effects

- loosen the constraint in Point Gather with a larger threshold, to be 1
  - Point agent can collect at most one bomb on average in each play
  - 평균적으로 1개 이하의 bomb 수집
- so loose that the performance of the constrained optimization is equivalent to the unconstrained one.
- **CPO still increases its cost to reach the constraint 1**, which is even worse than the randomly initialized policy
  - CPO always makes efforts to push its cost to the constraint threshold
- **IPO keeps decreasing its cost after the constraint is satisfied**
- **CPO is around 1 and the number for IPO is around 0.25**

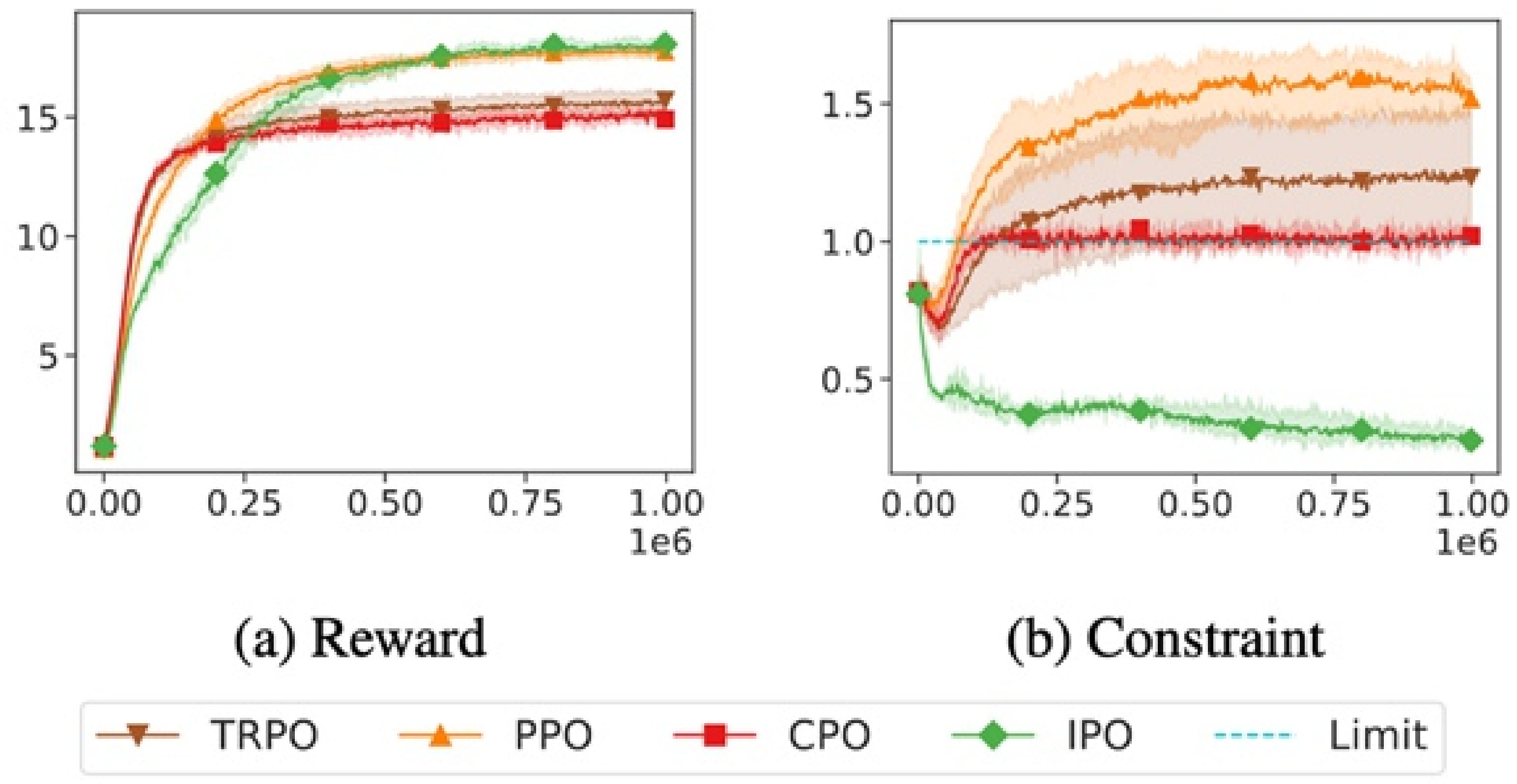


Figure 4: Average performance of TRPO, PPO, CPO and IPO under constraint limit 1.

Experiments

4 Hyperparameter Tuning

- **IPO hyperparameter t is easier to tune**
- **Tuning the initial Lagrange multiplier and learning rate takes a lot of efforts** in PDO
    - PDO is **sensitive** to the initialization of the Lagrange multiplier λ from 0.01 to 0.1
    - PDO is affected by the learning rate which changes from 0.01 to 0.001. The smaller learning rate slows down the policy convergence pace
- **Reward and cost of IPO are positively correlated with the hyperparameter t**
    - Binary search가 가능한 이유
    - higher reward and cost with larger t

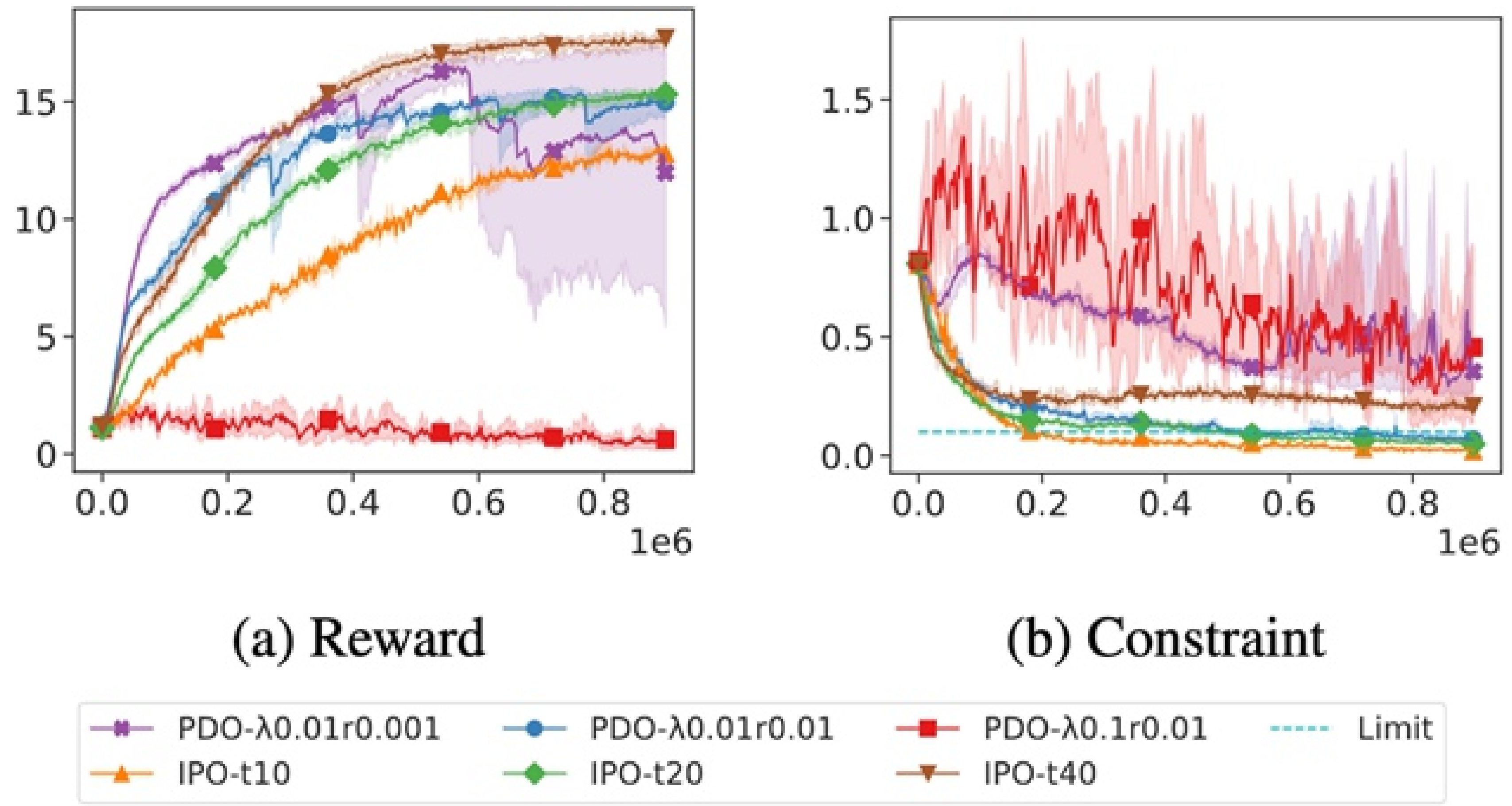


Figure 5: Average performance of PDO and IPO with different hyperparameters.

Experiments

5 Multiple Constraints

- IPO에서 constraint를 추가하고 싶다면 logarithm barrier function을 이용해서 term을 추가하기만 하면 됨
  - CPO보다 쉬움
- constraint에 해당하는 ball의 타입을 추가하여 constraint를 여러개로 만들어서 Point-Gather 실험

1. two apples, three bomb balls (0.04), five mine balls (0.06);
2. two apples, four bomb balls (0.05), four mine balls (0.05);
3. two apples, eight bomb balls (0.1), eight mine balls (0.1);

***()안에 있는 값들은 한 번의 플레이에서 모을 수 있는 해당 공의 최대 기대값에 대한 제약 조건**

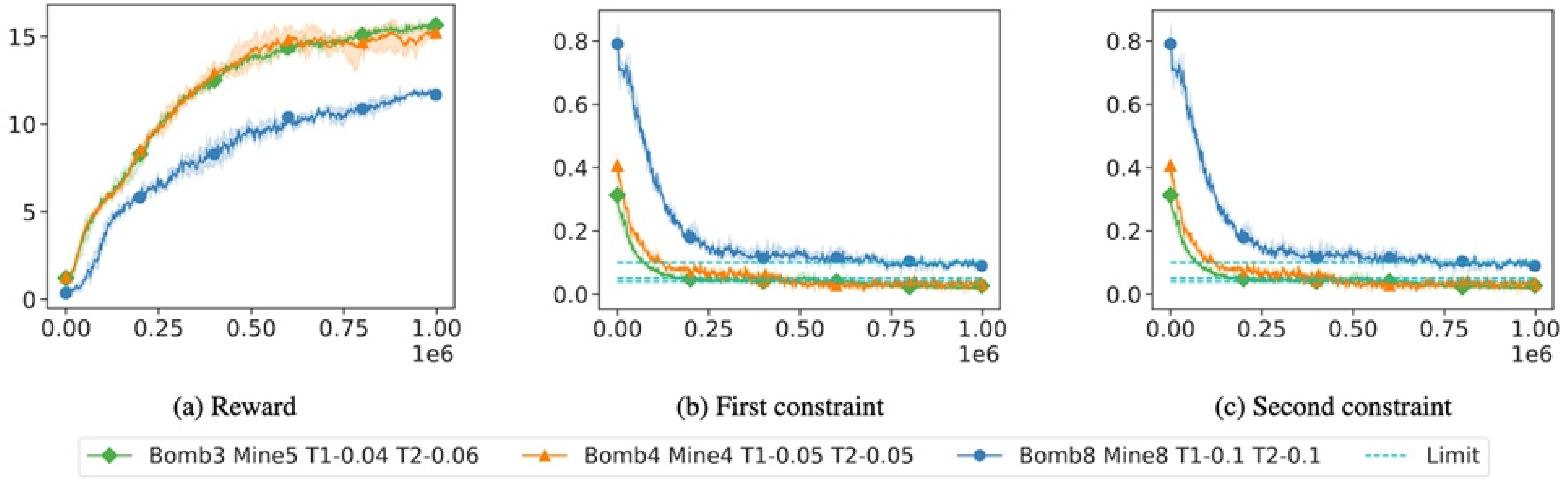


Figure 6: Average performance of IPO under multi-constraints. T1 and T2 correspond to the the limits in (b) and (c) separately. The dash lines are limits for different task settings

Experiments

6 Stochastic Environment Effects

- in real-world scenarios, there is always uncertainty from the environment
  - the outcome of an action is **affected by random noise**
- action: 속도(velocity)와 진행(heading) 방향의 vector로 -1~1 사이의 값으로 정의
- action에 평균 0 분산 각각 0.2, 0.5, 1.0으로 random noise를 추가
- 0.5일때도 학습이 수렴하는 것을 확인할 수 있었음

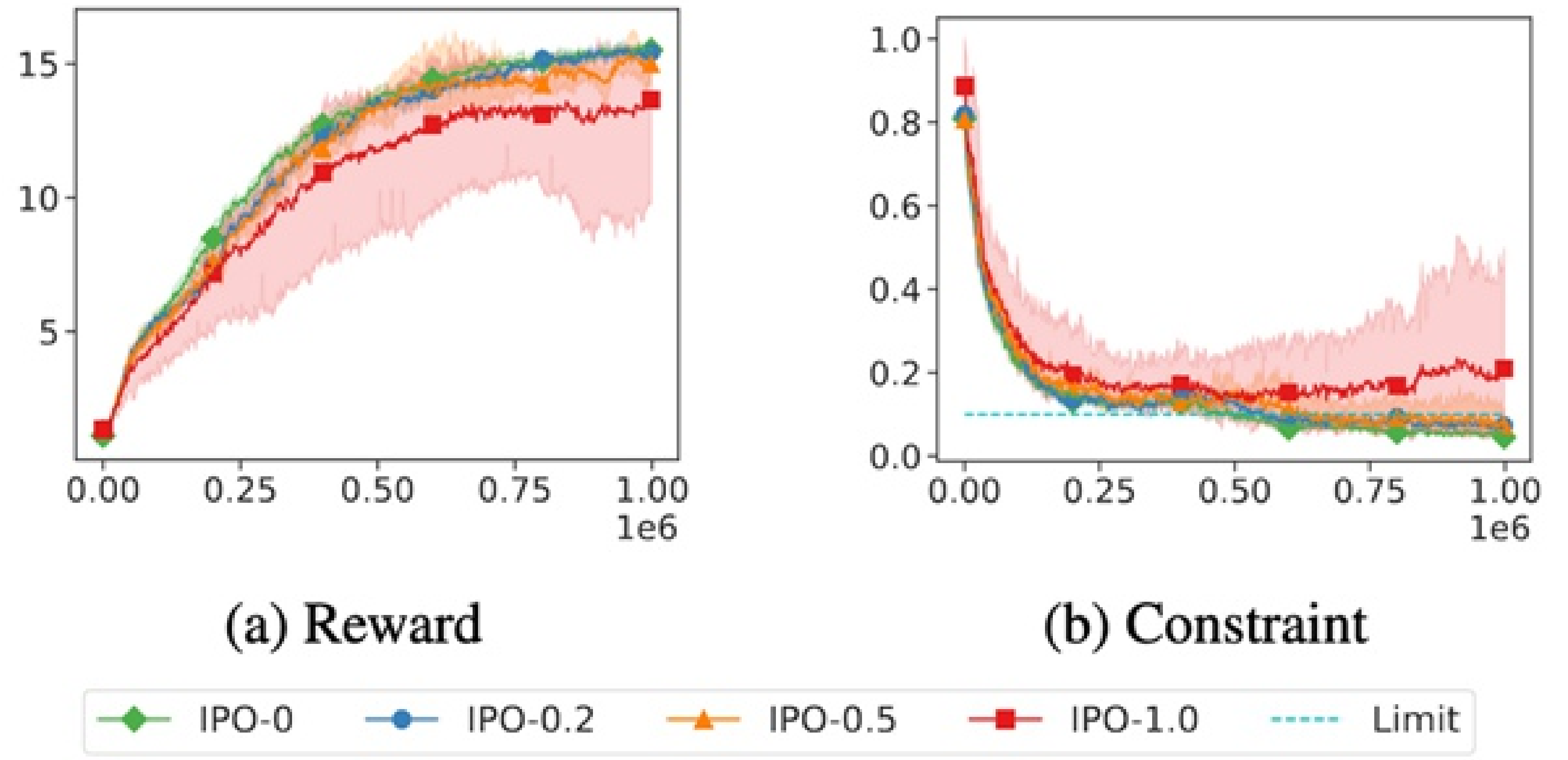


Figure 7: Average performance of IPO under different noise scale. IPO-0 means no noise is added.

END