Incentivizing Safer Actions in Policy Optimization for Constrained Reinforcement Learning

Supplementary Material

Somnath Hazra[1] , **Pallab Dasgupta**[2] and **Soumyajit Dey**[1]
[1]Indian Institute of Technology Kharagpur, India
[2]Synopsys, USA
somnathhazra@kgpian.iitkgp.ac.in, pallabd@synopsys.com, soumya@cse.iitkgp.ac.in

A Algorithm

In Algorithm 1 below we outline the pseudocode for policy updation using our penalty function.

Algorithm 1 Policy optimization using IP3O

```
Input: Initial policy $\pi_0$, initial value function $V_{\mathcal{R}}^{\pi_0}$, initial cost value function/s $V_{\mathcal{C}_i}^{\pi_0}$
for $k = 0, ..., K-1$ do
    Sample training batch $\mathcal{D}_k = \{\tau_1, ..., \tau_N\}$ consisting of $N$ trajectories using $\pi_k$
    Compute $V_{\mathcal{R}}^{\pi_k}, V_{\mathcal{C}_i}^{\pi_k}$ for trajectories in $\mathcal{D}_k$
    # Advantage calculation
    Compute $A_{\mathcal{R}}^{\pi_k}(s,a) = Q_{\mathcal{R}}^{\pi_k}(s,a) - V_{\mathcal{R}}^{\pi_k}(s)$
    Compute $A_{\mathcal{C}_i}^{\pi_k}(s,a) = Q_{\mathcal{C}_i}^{\pi_k}(s,a) - V_{\mathcal{C}_i}^{\pi_k}(s)$
    Update: $V_{\mathcal{R}}^{\pi_k}(s), V_{\mathcal{C}_i}^{\pi_k}(s) \rightarrow V_{\mathcal{R}}^{\pi_{k+1}}(s), V_{\mathcal{C}_i}^{\pi_{k+1}}(s)$
    # Policy update
    for $t = 0, ..., T-1$ do
        Compute $\mathcal{L}_{\mathcal{R}}(\pi_k) = \mathbb{E}_{\substack{s \sim d^{\pi_k} \\ a \sim \pi_k}}[-r'(\theta) A_{\mathcal{R}}^{\pi_k}(s,a)]$
        Compute $\mathcal{L}_{\mathcal{C}_i}(\pi_k) = \frac{1}{1-\gamma}\mathbb{E}_{\substack{s \sim d^{\pi_k} \\ a \sim \pi_k}}[r''(\theta) A_{\mathcal{C}_i}^{\pi_k}(s,a)] + (\mathcal{J}_{\mathcal{C}_i}(\pi_k) - d_i)$
        Compute $\mathcal{L}(\pi_k)$ using Equation 6
        $\pi = \pi_k + \omega \cdot \nabla \mathcal{L}(\pi_k)$
        # Trust region criterion (Gradient clipping)
        if $\mathbb{E}_{s \sim d^{\pi_k}}[D_{KL}(\pi || \pi_k)[s]] \notin [\delta^-, \delta^+]$ then
            break
        end if
    end for
    Update $\pi_k \rightarrow \pi_{k+1}$
end for
Return: Trained policy $\pi_K$
```

The output of the algorithm is the final policy. Here ω is the learning rate. For simplicity we have shown trust region updates using the KL divergence criterion, but in practice we use the PPO updates through gradient clipping [Schulman *et al.*, 2017].

B Proofs of Theorems

In this section we discuss the proofs of the theorems stated previously. We start with the dicusssion of the proof for Theorem 1.

B.1 Proof of Theorem 1

Before discussing the proof of the Theorem, we mention the performance difference lemma for expressing the performance bound over policy improvement.

Lemma 1 ([Kakade and Langford, 2002]). *Given a reward function $\mathcal{R}$, for any two policies π and π' and any start state distribution ρ,*

$$\mathcal{J}_{\mathcal{R}}^{\pi'} - \mathcal{J}_{\mathcal{R}}^{\pi} = \frac{1}{1-\gamma}\mathbb{E}_{\substack{s \sim d^{\pi'} \\ a \sim \pi'}}\left[A_{\mathcal{R}}^{\pi}(s,a)\right] \tag{1}$$

In general, a similar performance difference equation can be established in term of the cost function/s $\mathcal{C}_i$ as:

$$\mathcal{J}_{\mathcal{C}_i}^{\pi'} - \mathcal{J}_{\mathcal{C}_i}^{\pi} = \frac{1}{1-\gamma}\mathbb{E}_{\substack{s \sim d^{\pi'} \\ a \sim \pi'}}\left[A_{\mathcal{C}_i}^{\pi}(s,a)\right] \tag{2}$$

Using the above Lemma, the original optimization problem (given in Equation 1 of the main text) is reformulated as follows (Equation 4 of the main text, repeated here for ease of reference).

$$\begin{aligned} \pi_{k+1} &= \arg\max_{\pi} \mathbb{E}_{\substack{s \sim d^{\pi} \\ a \sim \pi}}\left[r(\theta) A_{\mathcal{R}}^{\pi_k}(s,a)\right] \\ \text{s.t.} \quad & \mathcal{J}_{\mathcal{C}_i}(\pi_k) + \frac{1}{1-\gamma}\mathbb{E}_{\substack{s \sim d^{\pi} \\ a \sim \pi}}\left[r(\theta) A_{\mathcal{C}_i}^{\pi_k}(s,a)\right] \leq d_i \end{aligned} \tag{3}$$

Since the problem (4) is difficult to optimize, we reformulated the problem as a penalty problem as follows (Equation 6 of the main text, repeated here for ease of reference).

$$\pi_{k+1} = \arg\min_{\pi} \mathcal{L}_{\mathcal{R}}(\pi_k) + \eta \sum_{i=1}^{m} \text{CELU}(\mathcal{L}_{\mathcal{C}_i}(\pi_k)) \tag{4}$$

where $\mathcal{L}_{\mathcal{R}}(\pi_k) = \mathbb{E}_{\substack{s \sim d^{\pi} \\ a \sim \pi}}[-r(\theta) A_{\mathcal{R}}^{\pi_k}(s,a)]$ and $\mathcal{L}_{\mathcal{C}_i}(\pi_k) = \frac{1}{1-\gamma}\mathbb{E}_{\substack{s \sim d^{\pi} \\ a \sim \pi}}[r(\theta) A_{\mathcal{C}_i}^{\pi_k}(s,a)] + (\mathcal{J}_{\mathcal{C}_i}(\pi_k) - d_i)$. For ease of notation we dilute the divergence requirement here. With respect to the equations 3 and 4, we can state the following.

Lemma 2. *Let λ^* be the Lagrange multiplier for the optimal solution of the dual problem of Equation 3. Given $\hat{\pi}$ is the solution of Equation 3. Then, as long as η is sufficiently large such that $\eta \geq ||\lambda^*||_\infty$, $\hat{\pi}$ also solves Equation 4.*

Proof. When $\mathcal{L}^{\pi_k}_{\mathcal{C}_i} \geq 0$;

$$\begin{aligned}&\mathcal{L}_{\mathcal{R}}(\pi_k) + \eta \sum_{i=1}^{m} \text{CELU}(\mathcal{L}_{\mathcal{C}_i}(\pi_k)) \\ &\qquad \geq \mathcal{L}_{\mathcal{R}}(\pi_k) + \sum_{i=1}^{m} \lambda_i^* \mathcal{L}_{\mathcal{C}_i}(\pi_k) \quad [\text{since } \eta \geq ||\lambda^*||_\infty]\end{aligned}$$

When $\mathcal{L}^{\pi_k}_{\mathcal{C}_i} < 0$;

$$\begin{aligned}&\mathcal{L}_{\mathcal{R}}(\pi_k) + \eta \sum_{i=1}^{m} \text{CELU}(\mathcal{L}_{\mathcal{C}_i}(\pi_k)) \\ &\qquad \geq \mathcal{L}_{\mathcal{R}}(\pi_k) + \sum_{i=1}^{m} \lambda_i^* \mathcal{L}_{\mathcal{C}_i}(\pi_k) \quad [\text{CELU}(x) \geq -\alpha]\end{aligned}$$

Since $\hat{\pi}$ is a solution of Equation 3, it satisfies the KKT conditions. Therefore, for any π_k and $\mathcal{L}^{\pi_k}_{\mathcal{C}_i}$;

$$\begin{aligned}&\mathcal{L}_{\mathcal{R}}(\pi_k) + \sum_{i=1}^{m} \lambda_i^* \mathcal{L}_{\mathcal{C}_i}(\pi_k) \\ &\qquad \geq \mathcal{L}_{\mathcal{R}}(\hat{\pi}) + \sum_{i=1}^{m} \lambda_i^* \mathcal{L}_{\mathcal{C}_i}(\hat{\pi}) \quad [\text{since } \hat{\pi} \text{ solves 3}] \\ &\qquad = \mathcal{L}_{\mathcal{R}}(\hat{\pi}) \quad [\text{complementary slackness at } \hat{\pi}] \\ &\qquad \geq \mathcal{L}_{\mathcal{R}}(\hat{\pi}) + \eta \sum_{i=1}^{m} \text{CELU}(\mathcal{L}_{\mathcal{C}_i}(\hat{\pi}))\end{aligned}$$

Thus the solution for Equation 3, $\hat{\pi}$, is also the solution for Equation 4. This ends the proof. □

Lemma 3. *Let λ^* be the Lagrange multiplier for the optimal solution of the dual problem of Equation 3. Given $\bar{\pi}$ is the solution of Equation 4, and $\hat{\pi}$ solves Equation 3. Then, for $\eta \geq ||\lambda^*||_\infty$, $\bar{\pi}$ also solves Equation 3.*

Proof. Given that $\bar{\pi}$ solves Equation 4 and $\bar{\pi}$ is a feasible solution for Equation 3, i.e., $\mathcal{L}_{\mathcal{C}_i}(\bar{\pi}) \leq 0$;

$$\begin{aligned}&\mathcal{L}_{\mathcal{R}}(\bar{\pi}) + \eta \sum_{i=1}^{m} \text{CELU}(\mathcal{L}_{\mathcal{C}_i}(\bar{\pi})) \\ &\qquad \leq \mathcal{L}_{\mathcal{R}}(\hat{\pi}) + \eta \sum_{i=1}^{m} \text{CELU}(\mathcal{L}_{\mathcal{C}_i}(\hat{\pi})) \\ &\qquad = \mathcal{L}_{\mathcal{R}}(\hat{\pi}) \quad [\text{since } \hat{\pi} \text{ solves 3}]\end{aligned}$$

If $\mathcal{L}_{\mathcal{C}_i}(\bar{\pi}) > 0$;

$$\begin{aligned}&\mathcal{L}_{\mathcal{R}}(\hat{\pi}) + \eta \sum_{i=1}^{m} \text{CELU}(\mathcal{L}_{\mathcal{C}_i}(\hat{\pi})) \\ &\qquad = \mathcal{L}_{\mathcal{R}}(\hat{\pi}) + \sum_{i=1}^{m} \lambda_i^* \mathcal{L}_{\mathcal{C}_i}(\hat{\pi}) [\text{complementary slackness}] \\ &\qquad \leq \mathcal{L}_{\mathcal{R}}(\bar{\pi}) + \sum_{i=1}^{m} \lambda_i^* \mathcal{L}_{\mathcal{C}_i}(\bar{\pi}) \quad [\text{since } \hat{\pi} \text{ solves 3}] \\ &\qquad \leq \mathcal{L}_{\mathcal{R}}(\bar{\pi}) + \eta \sum_{i=1}^{m} \text{CELU}(\mathcal{L}_{\mathcal{C}_i}(\bar{\pi})) [\text{since } \mathcal{L}_{\mathcal{C}_i}(\bar{\pi}) > 0]\end{aligned}$$

which is a contradiction to the assumption that $\bar{\pi}$ is the solution for Equation 4. Thus $\bar{\pi}$ is a feasible solution for Equation 3. This completes the proof. □

From Lemma 2 and Lemma 3, we can deduce that Equation 3 and Equation 4 share the same optimal solution set. This completes the proof of Theorem 1.

B.2 Proof of Theorem 2

Before discussing the proof for Theorem 2 we discuss the following lemma for defining the limits on performance difference given that the trajectories for policy optimization are sampled from the current policy, π_k.

Lemma 4 ([Achiam *et al.*, 2017])**.** *For any reward function, $\mathcal{R}$, and policies π and π', let $\varepsilon^{\pi'}_{\mathcal{R}} = \max_s |\mathbb{E}_{a\sim\pi'}[A^{\pi}_{\mathcal{R}}(s,a)]|$, and $\delta = \mathbb{E}_{s\sim d^\pi}[D_{KL}(\pi'||\pi)[s]]$, and*

$$D^{\pm}_{\pi,\mathcal{R}}(\pi') = \frac{1}{1-\gamma}\mathbb{E}_{\substack{s\sim d^\pi \\ a\sim\pi}}\left[\frac{\pi'(a|s)}{\pi(a|s)}A^{\pi}_{\mathcal{R}}(s,a) \pm \frac{\sqrt{2\delta}\gamma\varepsilon^{\pi'}_{\mathcal{R}}}{1-\gamma}\right]$$

then the following holds:

$$D^{+}_{\pi,\mathcal{R}}(\pi') \geq \mathcal{J}_{\mathcal{R}}(\pi') - \mathcal{J}_{\mathcal{R}}(\pi) \geq D^{-}_{\pi,\mathcal{R}}(\pi') \qquad (5)$$

The above lemma can be similarly defined with respect to the cost function/s $\mathcal{C}$. Here we are trying to learn a conservative policy to stay safe considering environmental uncertainties by using CELU function to design our penalty with respect to the cost function. This applies an incentive with respect to the cost function for $\mathcal{L}_{\mathcal{C}_i}(\pi) < 0$. Although this method produces a safer policy, as seen from the empirical results, this penalty function however produces a limit on the optimal reward owing to its restrictiveness, since the policy keeps updating due to the cost function beyond $\mathcal{L}_{\mathcal{C}_i}(\pi) = 0$.

We had stated earlier in the Practical Implementation section that $\text{CELU}(\mathcal{L}_{\mathcal{C}_i}(\pi))$ worked good for our experiments, mathematically it induces a residual gradient. This can be removed by using $\max(\text{CELU}(\mathcal{L}_{\mathcal{C}_i}(\pi)), -\alpha(1-h))$, where $0 < h < \alpha$ and $h \in \mathbb{R}$. This stops updates when $\text{CELU}(\mathcal{L}_{\mathcal{C}_i}(\pi)) \leq -\alpha(1-h)$. From the equation of CELU we get the point at which the loss updates due to cost stop instead of at $\mathcal{L}_{\mathcal{C}_i}(\pi) = 0$.

$$\begin{gathered}\alpha\left(\exp\left(\frac{\mathcal{L}_{\mathcal{C}_i}(\pi)}{\alpha}\right) - 1\right) = -\alpha(1-h) \\ \Rightarrow \mathcal{L}_{\mathcal{C}_i}(\pi) = \alpha \log(h)\end{gathered}$$

This imposes an error bound of $|\alpha \log(h)|$ for each constraint on the optimal problem 3. From Lemma 2, given $\hat{\pi}$ is the optimal solution for Equation 3, combining the above lemma with the error bound we get the total error bound on the optimal solution.

$$\begin{aligned}&|\mathcal{L}(\hat{\pi}) - \mathcal{L}(\pi_K)| \\ &\qquad \leq |\mathcal{L}_{\mathcal{R}}(\hat{\pi}) - \mathcal{L}_{\mathcal{R}}(\pi_K)| \\ &\qquad\qquad + \eta \sum_{i=1}^{m} |\text{CELU}(\mathcal{L}_{\mathcal{C}_i}(\hat{\pi})) - \text{CELU}(\mathcal{L}_{\mathcal{C}_i}(\pi_K))| \\ &\qquad \leq |\mathcal{L}_{\mathcal{R}}(\hat{\pi}) - \mathcal{L}_{\mathcal{R}}(\pi_K)| + \eta \sum_{i=1}^{m} |\mathcal{L}_{\mathcal{C}_i}(\hat{\pi}) - \mathcal{L}_{\mathcal{C}_i}(\pi_K)|\end{aligned}$$

From Equation 5, we derive that,

$$
\begin{aligned}
&|\mathcal{L}(\hat{\pi}) - \mathcal{L}(\pi_K)| \\
&\quad \leq \frac{\sqrt{2\delta}\gamma\varepsilon^{\hat{\pi}}_{\mathcal{R}}}{1-\gamma} + \eta \sum_{i=1}^{m} \left[\frac{\sqrt{2\delta}\gamma\varepsilon^{\hat{\pi}}_{\mathcal{C}_i}}{1-\gamma} + |\alpha \log(h)| \right]
\end{aligned}
$$

This completes the proof of Theorem 2.

C Empirical Details

We showed evaluation results across three widely-used safe RL environments: MuJoCo Safety Velocity [Ji *et al.*, 2023], Safety Gymnasium [Ray *et al.*, 2019], and Bullet Safety Gymnasium [Gronauer, 2022]. We also conducted experiments using the MetaDrive simulator [Li *et al.*, 2022]. All the environments can be defined using a CMDP.

C.1 Single Agent Environments

Given an agent, the objective is to find a policy, $\pi \in \Pi_{\mathcal{C}}$ that maximizes reward. The reward and cost functions vary depending on the environments. We trained different agents in each environment to demonstrate the efficacy of our approach. The details of environments are described below.

Run Tasks: The environments consist of simulation of autonomous robots based on MuJoCo simulator. The objective is to train agents to run on a plain surface, given the cost function defined on the velocity: $\sqrt{(v_x^2) + (v_y^2)}$. We trained four different agents, Ant, Half-Cheetah, Humanoid, and Swimmer from this environment. We also performed some ABlation experiments on these environments.

Safety Gymnasium: This environment consists of multiple robots to be trained under various constrained environments. We used two agents, Point and Car for our experiments. Depending on the scenarios the cost function is designed. Such as in Goal tasks the objective is to reach a predefined goal, in Button tasks, the objective is to go near button objects. In all tasks, the agent has to avoid certain predefined regions that incur costs. The objects are identified via lidar beams, that are a part of the observation.

Bullet Safety Gymnasium: Here, the task is to control objects such as Ball, and Car to perform tasks such as going in a circle very fast, while staying within predefined boundaries, or reaching a certain goal. We trained the above two agents on four scenarios from this benchmark environment.

C.2 Multi-Agent Environments

The Multi-Agent environments consist of multiple agents in a single scenario. We used MetaDrive scenarios, where the task is to control multiple cars, given a certain road situation. We performed experiments on scenarios that have an innate safety risk (due to crashing or going away from road) among agents, i.e. Parking Lot and unmanned Intersection. We used agent termination factor as our safety objective. The observation of each agent may only inform a partial scenario of the complete environment. The problem is generally solved by using an RNN in the policy network. We used GRU here. Also, for enforcing co-operation, we used summation of the local rewards for each agent, to train our policy following the theory of Value Decomposition Network [Sunehag *et al.*, 2017].

Each vehicle (agent) has continuous action space. Crashes or road exits result in episode termination for the respective agent. During training, episodes are terminated when more than half the agents are eliminated, and during evaluation, termination occurs upon the first agent's crash or completion of the scenario. We used lidar-based observations as inputs to the policy, where each agent acts upon their own local observation, and trained 10 agents in each environment. The evaluation was conducted in two challenging scenarios: Intersection and Parking Lot, both characterized by high probabilities of collisions, making them suitable for evaluating safety objectives.

D Hyper-parameter Settings

Following are the hyper-parameters we used for our experiments.

Table 1: Hyper-parameters used for our experiments

Hyper-parameter	Value
Policy Network Size	[64, 64]
Value Network size	[64, 64]
Policy Learning rate	0.0003
Value Learning rate	0.0003
batch size	64
Activation	Tanh
Epochs	500
Steps / Epoch	5000
Gamma (γ)	0.99
clip factor (ε)	0.2
η	20.0
Advantage Estimation	GAE
λ	0.95
$\lambda_{\mathcal{C}}$	0.95

We conducted our experiments using a system equipped with 12th generation Intel(R) Core(TM) i7 CPU processor with 16GB system memory, and a NVIDIA GeForce RTX 3060 GPU with 6GB graphic memory.

References

[Achiam *et al.*, 2017] Joshua Achiam, David Held, Aviv Tamar, and Pieter Abbeel. Constrained policy optimization. In *International conference on machine learning*, pages 22–31. PMLR, 2017.

[Gronauer, 2022] Sven Gronauer. Bullet-safety-gym: A framework for constrained reinforcement learning. 2022.

[Ji *et al.*, 2023] Jiaming Ji, Borong Zhang, Jiayi Zhou, Xuehai Pan, Weidong Huang, Ruiyang Sun, Yiran Geng, Yifan Zhong, Josef Dai, and Yaodong Yang. Safety gymnasium: A unified safe reinforcement learning benchmark. In *Thirty-seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2023.

[Kakade and Langford, 2002] Sham Kakade and John Langford. Approximately optimal approximate reinforcement learning. In *Proceedings of the Nineteenth International Conference on Machine Learning*, pages 267–274, 2002.

[Li *et al.*, 2022] Quanyi Li, Zhenghao Peng, Lan Feng, Qihang Zhang, Zhenghai Xue, and Bolei Zhou. Metadrive: Composing diverse driving scenarios for generalizable reinforcement learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022.

[Ray *et al.*, 2019] Alex Ray, Joshua Achiam, and Dario Amodei. Benchmarking safe exploration in deep reinforcement learning. *arXiv preprint arXiv:1910.01708*, 7(1):2, 2019.

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

[Sunehag *et al.*, 2017] Peter Sunehag, Guy Lever, Audrunas Gruslys, Wojciech Marian Czarnecki, Vinicius Zambaldi, Max Jaderberg, Marc Lanctot, Nicolas Sonnerat, Joel Z Leibo, Karl Tuyls, et al. Value-decomposition networks for cooperative multi-agent learning. *arXiv preprint arXiv:1706.05296*, 2017.