P2BPO: Permeable Penalty Barrier-based Policy Optimization for Safe RL (Appendix)

A Permeable Penalty Barrier-based Policy Optimization

A.1 Additional Derivations

The $softplus(x, \beta)$ and $sigmoid(x, \beta)$ can be written as

$$softplus(x, \beta) = \frac{1}{\beta} ln(1 + e^{\beta.x}) \tag{1}$$

$$sigmoid(x, \beta) = \frac{1}{(1 + e^{-\beta.x})} = \frac{e^{\beta.x}}{(1 + e^{\beta.x})} \tag{2}$$

Now, the derivative of the softplus is:

$$\nabla softplus(x, \beta) = \nabla(\frac{1}{\beta} ln(1 + e^{\beta.x})) = \frac{1}{\beta} \frac{1}{(1 + e^{\beta.x})} \nabla(1 + e^{\beta.x}) \tag{3}$$

$$= \frac{1}{\beta} \Big(\frac{1}{(1 + e^{\beta.x})} e^{\beta.x} \Big) \times \beta \nabla x \tag{4}$$

$$= sigmoid(x, \beta) \nabla x \tag{5}$$

A.2 Proof of Lemma 1:

Lemma 1: *Approximation error (k) for approximating* $(x.sigmoid(x, \beta))$ *using* $softplus(x, \beta)$ *is* $\leq \frac{ln(2)}{\beta}$.

Proof:

Note that $softplus(x, \beta) \geq x.sigmoid(x, \beta)$. Hence, to find the maximum distance between two functions, we formulate the distance variable ($Dist$) as follows:

$$Dist = x.sigmoid(x, \beta) - softplus(x, \beta) \tag{6}$$

$$\nabla Dist = \nabla \Big(x \times sigmoid(x, \beta) - softplus(x, \beta) \Big) \tag{7}$$

Now, at the minimum point, $\nabla Dist$ will be zero. Therefore,

$$0 = sigmoid(x, \beta) - \Big(x \times sigmoid(x, \beta) \times (1 - sigmoid(x, \beta)) \Big) - sigmoid(x, \beta)$$

Or,

$$\Big(x \times sigmoid(x, \beta)(1 - sigmoid(x, \beta)) \Big) = 0$$

$$x \times \frac{e^{\beta.x}}{(1 + e^{\beta.x})} \times \frac{e^{-\beta.x}}{(1 + e^{-\beta.x})} = 0 \tag{8}$$

e^x and e^{-x} are never zero in the real domain. Therefore x must be zero, and applying it in (Eq. 6), we get minimum $Dist$ is equal to $-\frac{ln(2)}{\beta}$. or the maximum distance between $softplus(x, \beta)$ and $x.sigmoid(x, \beta)$ is $\frac{ln(2)}{\beta}$. That means,

$$\Big(softplus(x, \beta) - x.sigmoid(x, \beta)\Big) \leq \frac{ln(2)}{\beta} \tag{9}$$

B Experimental Setup

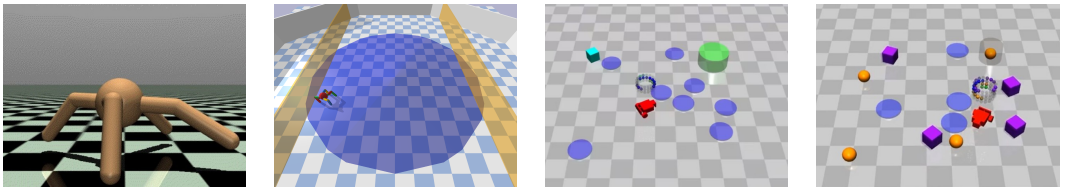
All the experiments run on a system with the following configurations:

- OS: Ubuntu 22.04
- Processor: Intel i7-12th Gen
- GPU: Nvidia RTX 3060
- RAM: 16 GB
- Python: 3.7
- PyTorch: 1.13.1
- Gym: 0.15.7

Baseline implementations are taken from the GitHub Safe-Policy-Optimization repository (PKU-Alignment 2023).

B.1 Environments

To compare our method with the baselines, we considered four different environments from the three most popular safe RL benchmark environments: Safety Gym (Ray, Achiam, and Amodei 2019), Bullet Safety Gym (Gronauer 2022), and Safe MuJoCo (Zhang, Vuong, and Ross 2020). All these four environments are categorized into four specific tasks. For each task, we train two different kinds of agents. Therefore, there is a total of eight different environment configurations. The details of the environments are given below.



(a) Run (b) Circle (c) Goal (d) Button

Figure 1: Considered environments of four different tasks

Run Task: In this task (Figure 1a), the aim is to train the learning agents to run on a plain surface, and the cost function is as follows:

$$cost = \sqrt{(x^2_{velocity} + y^2_{velocity})}$$

We considered two learner robots, Ant and Humanoid, for this task. These environments are part of the Safe MuJoCo benchmark environments.

Circle Task: In this task (Figure 1b), the learner aims to run as fast as possible along a circle periphery. However, the learner agent must be within that circle. Otherwise, the agent will receive a cost. We consider Ball and Car robots as the learning agents for this task, and these environments are part of the Bullet Safety Gym.

Goal Task: This task and the following are part of the Safety Gym environments. We consider two different robots, Point and Car, for both tasks. In this task (Figure 1c), the learning agent aims to move from random goal positions (green-colored blocks) while avoiding obstacles. If the agent collides with any obstacle, it receives a cost.

Button Task: Similar to the above scenario, in this task (Figure 1d), the agent has to go to a goal block and press a button while avoiding the obstacles.

The above task is organized in increasing order of the difficulty level of safety constraints. Also, it is worth mentioning that the agent and environment configurations are different across all three benchmark environments.

B.2 Hyperparameters

Table 1 gives the environment-specific hyperparameters used in our experiments for P2BPO and baselines. Default hyperparameters are common for all the environments except those mentioned in the table with different values. In Table 2, we mentioned the hyperparameters specific to the corresponding algorithm we consider in our experiments for P2BPO and baselines.

Default Hyperparameters		**Ant Run**	
Hyperparameters Name	**Value**	**Hyperparameters Name**	**Value**
Policy Network Size	[64,64]	Lambda Value(λ_value)	0.97
Value Network Size	[64,64]	Lambda Cost(λ_cost)	0.97
Cost Network Size	[64,64]		
Policy Learning Rate(LR)	0.0003		
Value LR	0.0001		
Cost LR	0.0001		
Activation Function	Tanh		
Training Epochs	500		
Steps/Epochs	30000		
Advantage Estimation Method	GAE		
Gamma(γ)	0.99		
Lambda Value(λ_value)	0.95		
Lambda Cost(λ_cost)	0.95		
Cost Limit	25.0		

Table 1: Default and Environment specific hyperparameters used in all experiments

P2BPO		P3O		IPO	
Name	**Value**	**Name**	**Value**	**Name**	**Value**
Clip ratio	0.2	Clip ratio	0.2	Clip ratio	0.2
Penalty Param Init (χ)	0.001	Kappa (κ)	20	Kappa (κ)	0.01
Penalty Param LR	0.2				
Beta (β)	10				
PPO-Lagrangian		**CPPO-PID**		**APPO**	
Name	**Value**	**Name**	**Value**	**Name**	**Value**
Clip ratio	0.2	Clip ratio	0.2	Clip ratio	0.2
Lagrangian Multiplier Init	0.001	Lagrangian Multiplier Init	0.001	Lagrangian Multiplier Init	0.001
Lagrangian Multiplier LR	0.2	PID Kp Init	0.01	Sigma	1
		PID Ki Init	0.01		
		PID Kd Init	0.01		
CPO		**FOCOPS**		**PCPO**	
Name	**Value**	**Name**	**Value**	**Name**	**Value**
Search Steps	25	Lagrangian Multiplier Init	0.001	Search Steps	25
Search Step Decay	0.8	Lagrangian Multiplier LR	0.2	Search Step Decay	0.8
		Eta (η)	0.02		

Table 2: Algorithm-specific hyperparameters used in all experiments

References

Gronauer, S. 2022. Bullet-Safety-Gym: A Framework for Constrained Reinforcement Learning. Technical report, mediaTUM.

PKU-Alignment. 2023. Safe Policy Optimization (SafePO). https://github.com/PKU-Alignment/Safe-Policy-Optimization.git.

Ray, A.; Achiam, J.; and Amodei, D. 2019. Benchmarking Safe Exploration in Deep Reinforcement Learning.

Zhang, Y.; Vuong, Q. H.; and Ross, K. W. 2020. First Order Constrained Optimization in Policy Space. *arXiv: Learning*.