CSPO: Constraint-Sensitive Policy Optimization for Safe Reinforcement Learning

Ayoub Belouadah, Sylvain Kubler, Yves Le Traon

SnT, University of Luxembourg

ICML International Conference On Machine Learning

Paper & Code

Interdisciplinary Centre for Security, Reliability and Trust

SnT

Motivation

Incorporating constraint sensitivity into policy updates can mitigate oscillations near the safety boundary commonly observed in primal-dual Safe Reinforcement Learning methods.

$$\max_{\pi} J_R(\pi) \quad \text{s.t.} \quad J_C(\pi) \leq d$$

Primal-dual methods enforce safety constraints using a Lagrange multiplier:

$$\lambda_{k+1} = [\lambda_k + \eta_\lambda J_c(\pi_k)]_+$$

Constraint violation → gradual increase of λ → delayed correction

Delayed feedback and geometry-agnostic corrections often result in oscillatory behavior near the safety boundary.

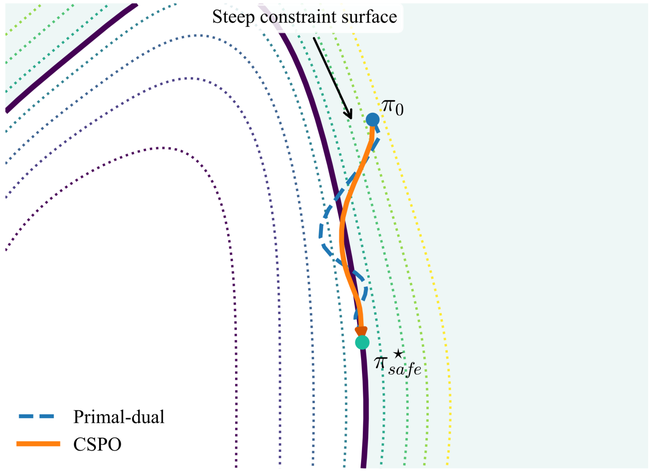


Figure 1. Comparison of optimization trajectories from an infeasible policy. By accounting for constraint sensitivity, CSPO reduces oscillatory behavior near the safety boundary.

Key observation

- **Steep constraint surface** → Cautious updates
- **Flat constraint surface** → Stronger corrective updates

Accounting for local constraint sensitivity enables smoother recovery toward feasibility.

Method

For a parametrized policy π_θ we define the constraint function:

$$g(\theta) = J_c(\pi_\theta) - d$$

Given an infeasible policy at iteration k, we seek the smallest parameter update that reaches the linearized constraint boundary

$$\min_{\Delta\theta} \frac{1}{2}|\Delta\theta|^2 \quad \text{s.t.} \quad g(\theta_k) + \nabla g(\theta_k)^T \Delta\theta = 0 \longrightarrow \Delta\theta^* = -\frac{g(\theta_k)}{\|\nabla g(\theta_k)\|^2}\nabla g(\theta_k)$$

The closed-form solution naturally scales safety recovery according to the local constraint sensitivity captured by $|\nabla g(\theta_k)|$. CSPO augments the original problem with a differentiable correction term that penalizes violations while scaling the penalty according to local constraint sensitivity

$$\pi_{\theta_{k+1}} = \arg\min_{\pi_\theta} \quad -L_R(\theta) + \frac{\alpha}{2}\frac{[g(\theta)]_+^2}{\|\nabla g(\theta_k)\|^2 + \epsilon} \quad \text{s.t.} \quad g(\theta) \leq 0$$

And optimize the corresponding Lagrangian and its multiplier

$$(\theta_{k+1}, \lambda_{k+1}) \in \arg\min_{\theta} \arg\max_{\lambda \in \Lambda} \mathcal{L}_k(\theta, \lambda)$$

With $\mathcal{L}_k(\theta, \lambda) = -L_R(\theta) + \frac{\alpha}{2}\frac{[g(\theta)]_+^2}{|\nabla g(\theta_k)|^2 + \epsilon} + \lambda g(\theta)$

Theoretical properties

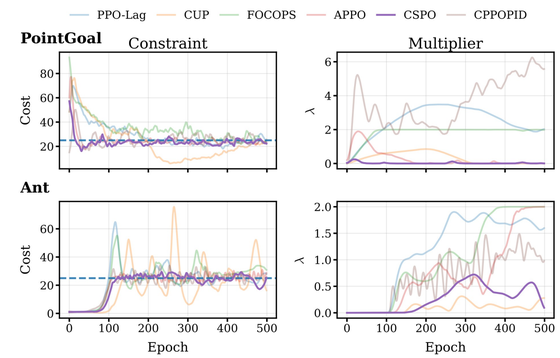
- **KKT equivalence:** CSPO preserves the feasible set and optimal solutions of the original constrained problem
- **Convergence:** CSPO iterates reach an ε-stationary point at rate $O(1/\varepsilon^6)$ in the nonconvex setting
- **Local Constraint Decrease:** Each update guarantees a violation decrease whenever g(θ) exceeds a threshold ∝ 1/a, directly linking a to recovery aggressiveness

Results

CSPO achieves the highest constrained return on most benchmark tasks while satisfying safety constraints

Environment	CSPO (R/C)	Best baseline (R/C)
Ant	**3231.6** / 24.5	3134.2 / 24.3
Humanoid	**6496.8** / 18.0	6213.1 / 18.3
Hopper	**1692.6** / 7.1	1559.0 / 20.5
Point Goal	**23.8** / 23.7	23.1 / 24.3
Point Button	**10.1** / 24.4	9.1 / 24.1
Car Button	**0.7** / 23.4	0.3 / 24.8

By accounting for constraint sensitivity, CSPO reduces oscillatory behavior in both constraint values and multiplier dynamics



CSPO recovers from unsafe states with fewer violations and shorter recovery times while preserving reward performance