Central Path Proximal Policy Optimization

Anonymous Author(s)
Affiliation
Address
email

Abstract

In constrained Markov decision processes, enforcing constraints during training is often thought of as decreasing the final return. Recently, it was shown that constraints can be incorporated directly in the policy geometry, yielding an optimization trajectory close to the central path of a barrier method, which does not compromise final return. Building on this idea, we introduce Central Path Proximal Policy Optimization (C3PO), a simple modification of the PPO loss that produces policy iterates, which stay close to the central path of the constrained optimization problem. Compared to existing on-policy methods, C3PO delivers improved performance with tighter constraint enforcement, suggesting that central path-guided updates offer a promising direction for constrained policy optimization.

1 Introduction

Reinforcement learning (RL) has demonstrated impressive capabilities across a wide range of domains, yet real-world applications increasingly demand more than just reward maximization. In many real-world high-stakes environments—such as autonomous vehicles, healthcare, or robotic manipulation—agents must also avoid violating domain-specific safety or resource constraints. This motivates the study of constrained Markov decision processes (CMDPs), an extension of the standard RL framework that imposes expected cost constraints alongside the goal of reward maximization Altman [1999]. By treating feasibility and reward objectives separately, CMDPs provide a principled framework for specifying agent behavior in complex environments.

CMDPs are especially relevant in deep reinforcement learning, where designing reward functions that reliably encode which states to avoid is notoriously difficult. Prior work has emphasized the importance of explicit constraint modeling for safe behavior. For instance, Ray et al. [2019] argue that CMDPs offer a natural formalism for benchmarking safe exploration in deep RL and introduce the Safety Gym suite to evaluate algorithms based on both task performance and cumulative safety cost. Similarly, Roy et al. [2022] demonstrate that task specifications expressed via constraints are often more natural and easier to design, especially as tasks increase in complexity, e.g. in finetuning LLMs Dai et al. [2023].

Despite their relatively low sample efficiency, model-free on-policy algorithms continue to play a foundational role in constrained RL. They offer conceptual clarity, support rigorous theoretical analysis, and provide strong baselines for studying the balance between performance and constraint satisfaction. As the field moves toward more scalable and sample-efficient approaches, insights developed in the on-policy setting remain central to both algorithm design and our broader understanding of safe learning, such as the policy improvement guarantees and constraint violation bounds introduced by Achiam et al. [2017].

In this context, there is a growing need for simple, scalable, and effective algorithms for solving CMDPs—ideally with properties similar to widely used algorithms such as proximal policy optimization (PPO; Schulman et al. [2017b]). PPO's robustness, ease of implementation, and scalability have

Submitted to the First Exploration in AI Today Workshop at ICML (EXAIT at ICML 2025). Do not distribute.

made it the method of choice in many deep RL and RLHF pipelines Ouyang et al. [2022]. We aim to extend these strengths to the constrained setting by developing an algorithm that shares PPO's practical benefits while enforcing constraints in a principled CMDP framework. Specifically, we seek to achieve high final reward while approximately satisfying constraints, at least at convergence.

To frame this problem, we distinguish between two commonly conflated settings in constrained RL: (i) *safe exploration*, where constraints must be satisfied throughout training, and (ii) *safe convergence*, where only the final policy is required to satisfy the constraints. Much of the literature has focused on the former, motivated by safety-critical applications in the real world. In contrast, safe convergence—where exploration may be unsafe—better reflects settings like simulation-based training or alignment finetuning Dai et al. [2023].Typically, ensuring safety *during* training is considered to decrease the final performance achieved by an algorithm. We show the contrary and present an algorithm that exhibits strict feasibility during training as well as reliable feasibility and high return at convergence.

In nonlinear CMDPs, the constraint surface is typically curved and nonconvex in policy parameter space. Converging prematurely or oscillating near the constraint boundary during training—as is often the case with Lagrangian or most trust region methods—can lead to unreliable constraint satisfaction at convergence. Furthermore, it can lead the optimizer to local optima that satisfy the constraints but fail to achieve high reward, see Figure 1. Penalty and barrier methods avoid this by maintaining a feasible trajectory toward the constraint surface, yielding feasible solutions more reliably. However, barrier methods introduce bias [Müller and Cayci, 2024], meaning the optimization problem obtained by adding a barrier penalty does not have the same solution set as the original problem, which can lead to degraded reward in policy optimization Milosevic et al. [2025]. Barrier methods either require careful tuning or an interior point approach Liu et al. [2020] to avoid harming reward performance.

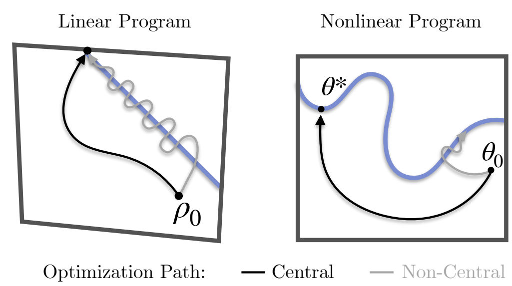

Figure 1: Pictorial visualization of the central path argument from the main text. While a wide range of methods technically converge to an optimal feasible solution in the linear programming formulation of finite CMDPs (left), in the function approximation setting (right), approaching the constraint surface too early may result in higher sensitivity to local optima.

The recently proposed C-TRPO [Milosevic et al., 2025] addresses these challenges by combining the strengths of trust-region and barrier methods by deriving a barrier-inspired trust-region formulation using strictly feasible trust regions. This results in an algorithm that acts like a barrier method with an adaptively receding barrier. This introduces no bias in the optimal feasible solution but still produces strictly feasible policies just like a barrier method. Further, C-TRPO produces policies, which are close to the regularization path obtained by altering the regularization strength, which is commonly known as the *central path* Boyd and Vandenberghe [2004], Müller and Cayci [2024]. This nicely illustrates how the constraints are incorporated in the algorithm's geometry and ensures that C-TRPO and variants thereof produce policies which don't prematurely approach the constraint surface.

However, C-TRPO's scalability remains limited due to computational overhead and sensitivity to large network architectures or batch sizes introduced by the TRPO-inspired update, and the update is defined only in the feasible set. To address the need for a simple and scalable CMDP solver, we propose a proximal version of C-TRPO. It also follows the central path, which we therefore call *Central Path Proximal Policy Optimization (C3PO)*. C3PO is a minibatch-based method that approximates the C-TRPO update using an exact penalty formulation, combining the simplicity and efficiency of PPO-style updates with the feasible geometry of central path methods. At its core, C3PO leverages the central path property of natural policy gradients to gradually guide the policy toward the constraint surface without inducing oscillations or premature convergence. The result is a practical algorithm that retains high reward performance while satisfying constraints at convergence, and which has the potential to scale well to large neural networks and modern deep RL settings.

2 Background

We consider the infinite-horizon discounted constrained Markov decision process (CMDP) and refer the reader to Altman [1999] for a general treatment. The CMDP is given by the tuple $\mathcal{M} \cup \mathcal{C}$, consisting of a finite MDP $\mathcal{M}$ and a set of constraints $\mathcal{C}$. The finite MDP $\mathcal{M} = \{\mathcal{S}, \mathcal{A}, P, r, \mu, \gamma\}$ is defined by a finite state-space $\mathcal{S}$, a finite action-space $\mathcal{A}$, a transition kernel $P \colon \mathcal{S} \times \mathcal{A} \to \Delta_{\mathcal{S}}$, an extrinsic reward function $r \colon \mathcal{S} \times \mathcal{A} \to \mathbb{R}$, an initial state distribution $\mu \in \Delta_{\mathcal{S}}$, and a discount factor $\gamma \in [0, 1)$. The space $\Delta_{\mathcal{S}}$ is the set of categorical distributions over $\mathcal{S}$. Further, $\mathcal{C} = \{(c_i, b_i)\}_{i=1}^m$ is a set of m constraints, where $c_i \colon \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ are the cost functions and $d_i \in \mathbb{R}$ are the cost thresholds. An agent interacts with the CMDP by selecting a policy $\pi \in \Pi$. Given π, the value function $V_r^\pi \colon \mathcal{S} \to \mathbb{R}$, action-value function $Q_r^\pi \colon \mathcal{S} \times \mathcal{A} \to \mathbb{R}$, and advantage function $A_r^\pi \colon \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ associated with the reward function r are defined as

$$V_r^\pi(s) \coloneqq (1-\gamma)\, \mathbb{E}_\pi \left[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t) \Big| s_0 = s \right],$$

$$Q_r^\pi(s,a) \coloneqq (1-\gamma)\, \mathbb{E}_\pi \left[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t) \Big| s_0 = s, a_0 = a \right],$$

and

$$A_r^\pi(s,a) \coloneqq Q_r^\pi(s,a) - V_r^\pi(s).$$

The expectations are taken over trajectories of the Markov process, meaning with respect to the initial distribution $s_0 \sim \mu$, the policy $a_t \sim \pi(\cdot|s_t)$ and the state transition $s_{t+1} \sim P(\cdot|s_t, a_t)$. They are defined analogously for the i-th cost and denoted as $V_{c_i}^\pi(s)$, $Q_{c_i}^\pi(s,a)$ and $A_{c_i}^\pi(s,a)$.

Constrained Reinforcement Learning addresses the optimization problem

$$\text{maximize}_{\pi \in \Pi}\ R(\pi) \quad \text{subject to} \quad C_i(\pi) \le d_i \tag{1}$$

for all $i = 1, \ldots, m$, where $R(\pi)$ is the expected value under the initial state distribution $R(\pi) \coloneqq \mathbb{E}_{s \sim \mu}[V_r^\pi(s)]$ and $C_i(\pi) \coloneqq \mathbb{E}_{s \sim \mu}[V_{c_i}^\pi(s)]$.

Every stationary policy π induces a discounted state-action occupancy measure $\rho_\pi \in \mathcal{K} \subset \Delta_{\mathcal{S} \times \mathcal{A}}$, which indicates the relative frequencies of visiting a state-action pair, discounted by how far the event lies in the future. We will refer to this measure as the *state-action occupancy* for short.

Definition 2.1. The *state-action occupancy* ρ_π is defined as

$$\rho_\pi(s,a) \coloneqq (1-\gamma) \sum_{t=0}^{\infty} \gamma^t \mathbb{P}_\pi(s_t = s) \pi(a|s), \tag{2}$$

where $\mathbb{P}_\pi(s_t = s)$ is the probability of observing the environment in state s at time t given π.

Note that similar measures can be introduced for the average-reward setting Zahavy et al. [2021], and for continuous state-action spaces Laroche and Des Combes [2023]. For any state-action measure ρ we obtain the associated policy via conditioning, meaning

$$\pi(a|s) \coloneqq \frac{\rho(s,a)}{\sum_{a'} \rho(s,a')}. \tag{3}$$

Assumption 2.2 (State Exploration)**.** For any policy $\pi \in \operatorname{int}(\Delta_{\mathcal{A}}^{\mathcal{S}})$ we have $\rho_\pi(s) > 0$ for all $s \in \mathcal{S}$.

Under Assumption 2.2, this provides a one-to-one correspondence between the space of policies Π and the space of state-action occupancies $\mathcal{K}$, which forms a convex polytope inside of $\mathbb{R}^{|\mathcal{S}||\mathcal{A}|}$ Kallenberg [1994], Mei et al. [2020]. These properties justify the linear programming (LP) approach to solving finite CMDPs Altman [1999], where problem 1 is reformulated as

$$\text{maximize}_{\rho \in \mathcal{K}} \sum_{s,a} \rho(s,a) r(s,a) \quad \text{subject to} \quad \sum_{s,a} \rho(s,a) c_i(s,a) \le d_i \tag{4}$$

which can be solved to obtain an optimal occupancy ρ^* using LP solution methods. The optimal policy can be extracted using relation 3.

In the function approximation setting, approach 4 is not applicable, which has prompted a large body of research in on-policy policy optimization methods. In the analysis of on-policy methods–including for standard MDPs–the *policy advantage* plays an important role. The policy (reward) advantage and *policy cost advantage* are defined as

$$\mathbb{A}_r^{\pi_k}(\pi) = \sum_{s,a} \rho_{\pi_k}(s)\pi(a|s)A_r^{\pi_k}(s,a) \quad \text{and} \quad \mathbb{A}_c^{\pi_k}(\pi) = \sum_{s,a} \rho_{\pi_k}(s)\pi(a|s)A_c^{\pi_k}(s,a). \tag{5}$$

It features prominently in policy optimization methods, as it approximates the performance difference between two nearby policies, i.e. $\mathbb{A}_r^{\pi_k}(\pi) \approx R(\pi) - R(\pi_k)$ if $\pi_k \approx \pi$, and for the policy cost advantage analogously.

2.1 Policy Optimization Methods for Constrained Reinforcement Learning

In the following, we review relevant prior constrained policy optimization methods, thereby focusing on a single constraint to reduce notational clutter. However, all mentioned methods are trivial to extend to multiple constraints.

Constrained Policy Optimization (CPO) Constrained policy optimization (CPO) is a modification of trust region policy optimization (TRPO; Schulman et al. [2017a]), where the classic trust region is intersected with the set of safe policies Achiam et al. [2017]. At each iteration k, the policy of the next iteration π_{k+1} is obtained through the solution of

$$\max_{\pi \in \Pi} \mathbb{A}_r^{\pi_k}(\pi) \text{ s.t. } \bar{D}_{\text{KL}}(\pi, \pi_k) \leq \delta \text{ and } C(\pi_k) + \mathbb{A}_c^{\pi_k}(\pi) \leq d. \tag{6}$$

where $\bar{D}_{\text{KL}}(\pi, \pi_k) = \sum_{s,a} \rho_{\pi_k}(s) D_{\text{KL}}[\pi(\cdot|s)|\pi_k(\cdot|s)]$ and $C(\pi_k) + \mathbb{A}_c^{\pi_k}(\pi)$ is an estimate for $C(\pi)$, see Kakade and Langford [2002], Schulman et al. [2017a], Achiam et al. [2017].

Penalized Proximal Policy Optimization (P3O/P2BPO) Solving the constrained optimization problem equation 8 is difficult to scale up to more challenging tasks and larger model sizes, as it relies on the arguably sample inefficient TRPO update. To circumvent this Zhang et al. [2022] proposed a Constrained RL algorithm derived from the relaxed penalized problem

$$\max_{\pi \in \Pi} \mathbb{A}_r^{\pi_k}(\pi) - \lambda \max\{0, C(\pi_k) + \mathbb{A}_c^{\pi_k}(\pi) - d\}, \text{ s.t. } \bar{D}_{\text{KL}}(\pi, \pi_k) \leq \delta. \tag{7}$$

The appeal of this reformulation is that one can obtain an unconstrained problem that gives the same solution set for λ chosen large enough Zhang et al. [2022] and by employing a PPO-like loss. A similar approach is taken by Dey et al. [2024], where $\max\{0, \cdot\}$ is replaced by a softplus.

Constrained Trust Region Policy Optimization (C-TRPO) Where equation 8 incorporates the safety by intersecting the trust region with the set of safe policies, an alternative approach was taken in Milosevic et al. [2025] where the geometry was modified such that the resulting trust region automatically consists of safe policies. To this end, C-TRPO proceeds as TRPO but with the usual divergence augmented by a barrier term, meaning

$$\max_{\pi \in \Pi} \mathbb{A}_r^{\pi_k}(\pi) \text{ s.t. } \bar{D}_{\text{KL}}(\pi, \pi_k) + \beta D_{\text{B}}(\pi, \pi_k) \leq \delta. \tag{8}$$

where we'll refer to

$$D_{\text{B}}(\pi, \pi_k) = \frac{b - \mathbb{A}_c^{\pi_k}(\pi)}{b} - \log\left(\frac{b - \mathbb{A}_c^{\pi_k}(\pi)}{b}\right) - 1, \text{ for } b > 0 \text{, else } \infty \tag{9}$$

as the *barrier divergence*[1], β is a positive safety parameter, and $b = d - C(\pi_k)$ is its *cost budget*.

This update is justified by the general theory of Bregman divergences and the theory of convex programs. It has desirable theoretical properties and results in state-of-the-art performance compared to other on-policy CMDP algorithms. We refer the reader to Milosevic et al. [2025] and Appendix A for detailed discussions.

[1] Note the similarity to the unbiased KL-Divergence estimator $\hat{D}_{KL} = \frac{\pi(a|s)}{\pi_k(a|s)} - \log \frac{\pi(a|s)}{\pi_k(a|s)} - 1$

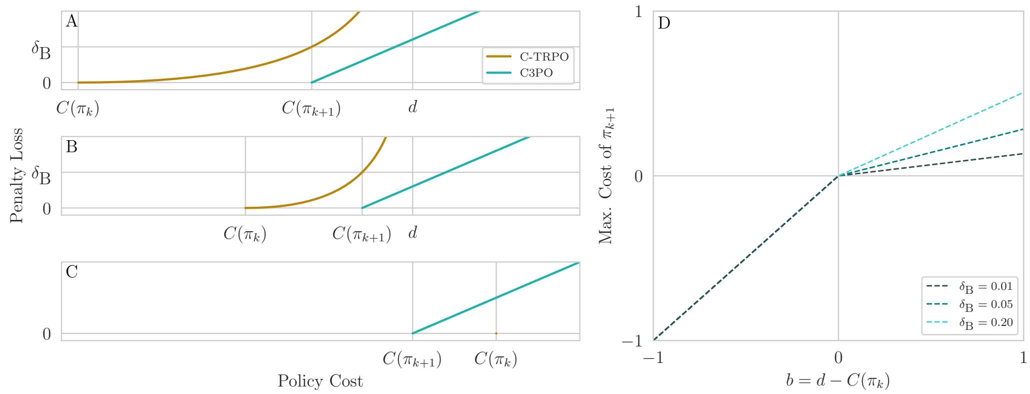

Figure 2: The working principle behind C3PO's exact penalty approach: As the iterate moves closer towards the constraint (A-C), C3PO's ReLU-penalty pulls away at a slower rate $0 < w < 1$, mimicking C-TRPO's barrier divergence. This rate is defined as a function of δ_{B} (D), see main text. While C-TRPO's barrier penalty is undefined if either $d \leq C(\pi_k)$ or $d \leq C(\pi_{k+1})$, C3PO's ReLU-penalty is defined everywhere.

3 Central Path Proximal Policy Optimization

C-TRPO has desirable properties but the practical algorithm 1) scales poorly and is sample-inefficient due to its reliance on the TRPO algorithm and 2) relies on a recovery mechanism (reward-free cost minimization + hysteresis), since the update is not defined if π_k is outside the feasible set.

We propose a first-order approximation of C-TRPO that approximates its central path by solving surrogate optimization problems with the same solution set as C-TRPO's update on every iteration. In addition, by employing an exact penalty approach, it allows unsafe policies during training, also enabling less strict exploration strategies within the safe convergence regime.

C3PO Update Let us consider a slight modification of C-TRPO's update, which is constrained with the KL and Barrier constraints separately, since they can be approximated using different methods which result in different precisions, i.e. we consider

$$\max_{\pi \in \Pi} \mathbb{A}_r^{\pi_k}(\pi) \text{ s.t. } D_{\mathrm{B}}(\pi, \pi_k) \leq \delta_{\mathrm{B}} \text{ and } \bar{D}_{\mathrm{KL}}(\pi, \pi_k) \leq \delta_{\mathrm{KL}}. \tag{10}$$

Note that this is a subtly different problem than that posed by C-TRPO, but δ_{KL} and δ_{B} can always be chosen to include the feasible set entirely in C-TRPO's feasible set for a given δ. Instead of solving this constrained problem directly, we consider the penalized problem given by

$$\max_{\pi \in \Pi} \mathbb{A}_r^{\pi_k}(\pi) - \kappa \max\{0, D_{\mathrm{B}}(\pi, \pi_k) - \delta_{\mathrm{B}}\} \text{ s.t. } \bar{D}_{\mathrm{KL}}(\pi, \pi_k) \leq \delta_{\mathrm{KL}}. \tag{11}$$

Theorem 3.1 (Exactness)**.** *Let λ be the Lagrange multiplier vector for the optimizer of Equation 10. Then for $\kappa \geq |\lambda|$ the solution sets of problem Equation 10 and problem Equation 11 agree.*

Proof. Note that the problem Equation 10 is concave-convex in π. Hence, this is a special case of the general exactness result Theorem B.1. □

C3PO Algorithm The update Equation 11 is still undefined outside the feasible set of the barrier divergence constraint. Since we use the barrier divergence only to define the feasible solution set of the update, we can replace it with another function, as long as it defines the same feasible set. More precisely, this can be achieved with an equivalent linear constraint that is zero where $D_{\mathrm{B}}(\pi, \pi_k) = \delta_{\mathrm{B}}$ for positive cost advantages. The C3PO algorithm approximates update 11 as

$$\max_{\pi \in \Pi} \mathbb{A}_r^{\pi_k}(\pi) - \kappa \max\{0, \mathbb{A}_c^{\pi_k}(\pi) - \min\{b, w \cdot b\}\} \text{ s.t. } \bar{D}_{\mathrm{KL}}(\pi, \pi_k) \leq \delta_{\mathrm{KL}}. \tag{12}$$

where $0 < w < 1$.

Proposition 3.2 (Positive Exactness)**.** *For $0 \leq \mathbb{A}_c^{\pi_k}(\pi) < d - C(\pi_k)$, there exist w and δ_{B} for which the solution sets of problems 10, 11 and 12 agree.*

The new update expresses the same constraint using a linear ReLU-penalty. The rate w is a new hyper-parameter and we refer to Appendix B for a proof of Proposition 3.2. Since the original problem's penalty function is not defined outside the interior of the feasible set, we must handle the case $C(\pi_k) \geq d$ differently, which is taken care of by the $\min(b, \cdot)$ term: For $b < 0$, problem 12 reduces to the P3O Zhang et al. [2022] objective Eq. 7. Finally, the additional KL-constraint is approximated as in PPO Schulman et al. [2017b]. The resulting loss only consists of the PPO loss and an additional loss term which is a function of the policy cost advantage estimate. Let $r(\theta) = \frac{\pi_\theta(a|s)}{\pi_k(a|s)}$ denote the likelihood ratio of the optimized and last behavior policies and let

$$\alpha_{\text{clipped}}(\theta) = \mathbb{E}_{s,a\sim\rho_k}\left[\max\left(r(\theta)\hat{A}_c(s,a), \text{clip}(r(\theta), 1-\epsilon, 1+\epsilon)\hat{A}_c(s,a)\right)\right]. \tag{13}$$

The C3PO loss is

$$L^{\text{C3PO}}(\theta) = \text{ReLU}\left(\alpha_{\text{clipped}}(\theta) - \min\{b, w \cdot b\}\right). \tag{14}$$

The penalty coefficient remains a hyperparameter, which can be flexibly scheduled to solve either setting (i) or (ii), as we demonstrate in Section 4, where we use a linear schedule to achieve high final performance across multiple tasks. The final method is summarized in Algorithm 1.

Algorithm 1 C3PO (deviation from PPO in green)

Require: Initial policy π_0 and value functions $\hat{V}_r$, $\hat{V}_{c_i}$, thresholds d_i, scheduled penalty κ_k, rate w
```
for k = 0, 1, 2, ... do
    Collect trajectory data D = {s_0, a_0, r_0, c_0, ...} by running π_k
    Estimate reward advantage Â_t^r and cost advantages Â_t^{c_i} using GAE-λ Schulman et al. [2018]
    Update policy π_{k+1} by minimizing L^PPO + κ_k L^C3PO (Equation 14)
    Update value function estimates V̂_r^{π_{k+1}} and V̂_{c_i}^{π_{k+1}} by regression
end for
```

Relation to other PPO-Penalty methods C3PO is a superset of P3O Zhang et al. [2020]. More precisely, if we set $w = 1$ in C3PO, we obtain the P3O loss exactly. Further, C3PO is conceptually similar to P2BPO Dey et al. [2024], in using a more conservative version of the P3O loss, but C3PO does not use a penalty with a fixed location at the constraint, but a moving penalty which recedes as the iterate gets closer to the constraint. This allows C3PO to approach the optimal feasible solution without regularization bias.

4 Computational Experiments

To evaluate our approach, we conduct experiments aimed at testing the benefits of using central path approximation as a design principle for constrained policy optimization algorithms. We benchmark C3PO against a range of representative constrained reinforcement learning baselines. We include methods from three major algorithmic families: penalty-based methods (P3O, P2BPO), Lagrangian methods (PPO-Lag, CPPO-PID), and trust-region methods (CPO, C-TRPO).

Conceptually, *penalty-based methods*, especially algorithms that augment the PPO loss with a penalty, like P3O Zhang et al. [2022] and P2BPO Dey et al. [2024], are closest to our approach. Like C3PO, those penalize constraint violations directly in the policy gradient loss using a ReLU-penalty. *Lagrangian methods* maintain dual variables to enforce constraints adaptively. PPO-Lagrangian Ray et al. [2019] applies this principle to the PPO algorithm, forming a loss which is similar to C3PO's. For completeness, we consider CPPO-PID Stooke et al. [2020] as a more recent Lagrangian baseline. Finally, *trust region methods*, such as CPO Achiam et al. [2017] and C-TRPO Milosevic et al. [2025], use trust regions and constrained updates to maintain stable reward improvement and feasibility throughout training. They do not aim for scalability, but form strong baselines on the considered benchmark.

We benchmark the algorithms on 4 locomotion tasks and 4 navigation tasks from Safety Gymnasium Ji et al. [2023], as in Milosevic et al. [2025]. For the baseline algorithms, we use the hyper-parameters

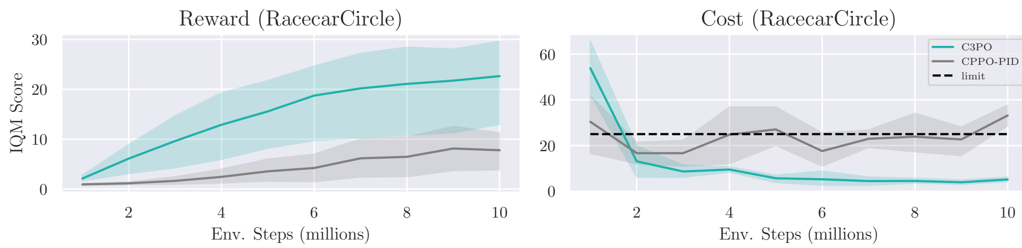

Figure 4: Example of improved performance through approximating the central path: Lagrangian methods tend to converge less reliably towards a safe policy and oscillate around the constraint. This does not yield a higher final reward. Instead, staying feasible from early on in training seems to have a positive effect on final reward.

reported in Ji et al. [2023], and for P3O and C-TRPO the recommended parameters in Zhang et al. [2022] and Milosevic et al. [2025] respectively. For C3PO we use $\kappa = 30.0$ and $w = 0.05$. Each algorithm is trained on each task for 10 million steps with a cost threshold of 25.0. Final iterate performance is measured by aggregating over 5 seeds using `rliable` Agarwal et al. [2021].

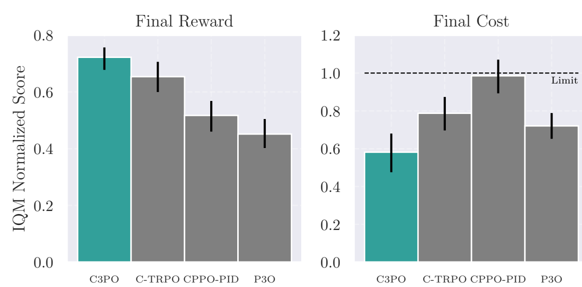

Figure 3: Aggregated performance using the inter quartile mean (IQM) across 8 tasks from Safety Gymansium for a subset of algorithms. The algorithms were chosen as the feasible representatives of their respective group.

The results provide confirmatory evidence for the usefulness of the central path approach: policies trained with C3PO exhibit a stable progression toward the constrained optimum, maintaining feasibility for most training iterations, see Figure 4. Furthermore, C3PO consistently outperforms prior PPO-style penalty methods in terms of achieved reward, while also adhering more strictly to the specified constraints, see Figure 3. This improved trade-off between reward and feasibility offers additional support for the effectiveness of the central path approach. While C3PO does not outperform trust-region methods across all tasks in the benchmark, it performs well consistently, resulting in high aggregated performance. The full benchmark results table and more examples like Figure 4 are presented in Appendix C.

5 Conclusion

In this work, we use central path approximation as a guiding principle for designing policy optimization methods for constrained RL. We propose C3PO, an algorithm which is obtained through a simple augmentation of the original PPO-loss inspired by the central path approach. Our experimental results support this design principle: Compared to existing PPO-style penalty and Lagrangian methods, C3PO exhibits improved performance with tighter constraint satisfaction, highlighting the benefits of a central path approach in constrained policy optimization. While the current results are limited to small-scale simulations and simplified settings, such as a single constraint per task, they suggest that the central path-approximation is a promising design principle for constrained RL algorithms. We hope this early-stage contribution encourages further discussion and refinement of on-policy safe policy optimization algorithms and other methods that solve constrained MDPs. Future work will explore extensions to high-dimensional tasks, theoretical guarantees, and applications such as safety-critical control and LLM fine-tuning. We hope this early-stage contribution encourages further dialogue on improving constrained RL algorithms using insights from constrained optimization.

References

Joshua Achiam, David Held, Aviv Tamar, and Pieter Abbeel. Constrained policy optimization, 2017.

Rishabh Agarwal, Max Schwarzer, Pablo Samuel Castro, Aaron C Courville, and Marc Bellemare. Deep reinforcement learning at the edge of the statistical precipice. *Advances in Neural Information Processing Systems*, 34, 2021.

Eitan Altman. *Constrained Markov Decision Processes*. CRC Press, Taylor & Francis Group, 1999. URL `https://api.semanticscholar.org/CorpusID:14906227`.

Felipe Alvarez, Jérôme Bolte, and Olivier Brahic. Hessian riemannian gradient flows in convex programming. *SIAM journal on control and optimization*, 43(2):477–501, 2004.

Dimitri P Bertsekas. Nonlinear programming. *Journal of the Operational Research Society*, 48(3): 334–334, 1997.

Stephen P Boyd and Lieven Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

Robert M Corless, Gaston H Gonnet, David EG Hare, David J Jeffrey, and Donald E Knuth. On the lambert w function. *Advances in Computational mathematics*, 5:329–359, 1996.

Josef Dai, Xuehai Pan, Ruiyang Sun, Jiaming Ji, Xinbo Xu, Mickel Liu, Yizhou Wang, and Yaodong Yang. Safe rlhf: Safe reinforcement learning from human feedback, 2023. URL `https://arxiv.org/abs/2310.12773`.

Sumanta Dey, Pallab Dasgupta, and Soumyajit Dey. P2bpo: Permeable penalty barrier-based policy optimization for safe rl. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 21029–21036, 2024.

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. URL `https://openreview.net/forum?id=WZmlxIuIGR`.

Sham Kakade and John Langford. Approximately optimal approximate reinforcement learning. In *Proceedings of the Nineteenth International Conference on Machine Learning*, ICML '02, page 267–274, San Francisco, CA, USA, 2002. Morgan Kaufmann Publishers Inc. ISBN 1558608737.

Lodewijk CM Kallenberg. Survey of linear programming for standard and nonstandard markovian control problems. part i: Theory. *Zeitschrift für Operations Research*, 40:1–42, 1994.

Romain Laroche and Remi Tachet Des Combes. On the occupancy measure of non-markovian policies in continuous mdps. In *International Conference on Machine Learning*, pages 18548–18562. PMLR, 2023.

Yongshuai Liu, Jiaxin Ding, and Xin Liu. Ipo: Interior-point policy optimization under constraints. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34:4940–4947, 04 2020. doi: 10.1609/aaai.v34i04.5932.

Jincheng Mei, Chenjun Xiao, Csaba Szepesvari, and Dale Schuurmans. On the global convergence rates of softmax policy gradient methods. In *International Conference on Machine Learning*, pages 6820–6829. PMLR, 2020.

Nikola Milosevic, Johannes Müller, and Nico Scherf. Embedding safety into rl: A new take on trust region methods, 2025. URL `https://arxiv.org/abs/2411.02957`.

Johannes Müller and Semih Cayci. Essentially sharp estimates on the entropy regularization error in discrete discounted markov decision processes. *arXiv preprint arXiv:2406.04163*, 2024.

Johannes Müller and Guido Montúfar. Geometry and convergence of natural policy gradient methods. *Information Geometry*, pages 1–39, 2023.

Gergely Neu, Anders Jonsson, and Vicenç Gómez. A unified view of entropy-regularized markov decision processes. *arXiv preprint arXiv:1705.07798*, 2017.

Jorge Nocedal and Stephen J Wright. *Numerical optimization*. Springer, 1999.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744, 2022.

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

Julien Roy, Roger Girgis, Joshua Romoff, Pierre-Luc Bacon, and Christopher Pal. Direct behavior specification via constrained reinforcement learning, 2022. URL `https://arxiv.org/abs/2112.12228`.

John Schulman, Sergey Levine, Philipp Moritz, Michael I. Jordan, and Pieter Abbeel. Trust region policy optimization, 2017a.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms, 2017b. URL `https://arxiv.org/abs/1707.06347`.

John Schulman, Philipp Moritz, Sergey Levine, Michael Jordan, and Pieter Abbeel. High-dimensional continuous control using generalized advantage estimation, 2018. URL `https://arxiv.org/abs/1506.02438`.

Adam Stooke, Joshua Achiam, and Pieter Abbeel. Responsive safety in reinforcement learning by pid lagrangian methods, 2020. URL `https://arxiv.org/abs/2007.03964`.

Tom Zahavy, Brendan O'Donoghue, Guillaume Desjardins, and Satinder Singh. Reward is enough for convex mdps. *Advances in Neural Information Processing Systems*, 34:25746–25759, 2021.

Linrui Zhang, Li Shen, Long Yang, Shixiang Chen, Bo Yuan, Xueqian Wang, and Dacheng Tao. Penalized proximal policy optimization for safe reinforcement learning, 2022. URL `https://arxiv.org/abs/2205.11814`.

Yiming Zhang, Quan Vuong, and Keith Ross. First order constrained optimization in policy space. *Advances in Neural Information Processing Systems*, 33:15338–15349, 2020.

A Extended Background

A.1 The Geometry of Policy Optimization

Neu et al. [2017] have shown that the policy divergence used to define the trust-region in TRPO Schulman et al. [2017a] can be derived as the Bregman divergence generated by a mirror function on the state-action polytope. TRPO's mirror function is the negative conditional entropy

$$\Phi_K(\rho) = \sum_{s,a} \rho(s,a) \log \pi_\rho(a|s) \tag{15}$$

which generates

$$D_K(\pi_k||\pi) = \sum_{s,a} \rho_k(s,a)[\log \pi(a|s) - \log \pi_k(a|s)] \tag{16}$$

via the operator

$$D_\Phi(x||y) \coloneqq \Phi(x) - \Phi(y) - \nabla\Phi(y)^\top(x-y). \tag{17}$$

In general, a trust region update is defined as

$$\pi_{k+1} \in \arg\max_{\pi\in\Pi} \mathbb{A}_r^{\pi_k}(\pi) \quad \text{sbj. to } D_\Phi(\rho_{\pi_k}||\rho_\pi) \leq \delta, \tag{18}$$

where $D_\Phi \colon \mathcal{K} \times \mathcal{K} \to \mathbb{R}$ is the *Bregman divergence* induced by a suitably convex function $\Phi\colon \operatorname{int}(\mathcal{K}) \to \mathbb{R}$.

A.2 The Safe Geometry Approach

Milosevic et al. [2025] consider mirror functions of the form

$$\Phi_{\mathrm{C}}(\rho) \coloneqq \Phi_{\mathrm{K}}(\rho) + \sum_i \beta_i \Phi_{\mathrm{B}}(\rho) \tag{19}$$

$$\coloneqq \sum_{s,a} \rho(s,a) \log \pi_\rho(a|s) + \sum_{i=1}^{m} \beta_i \phi\left(b_i - \sum_{s,a} \rho(s,a) c(s,a)\right), \tag{20}$$

where $\rho \in \mathcal{K}_{\text{safe}}$ is a feasible state-action occupancy, Φ_{K} is the negative conditional entropy, and ϕ is convex. Further, $\phi\colon \mathbb{R}_{>0} \to \mathbb{R}$ with $\phi'(x) \to +\infty$ for $x \searrow 0$. The log-barrier $\phi(x) = -\log(x)$ considered in this work is a possible candidate. In general, the induced divergence takes the form

$$D_{\mathrm{C}}(\rho_1||\rho_2) = D_{\mathrm{K}}(\rho_1||\rho_2) + \sum_{i=1}^{m} \beta_i D_{\mathrm{B}}(\rho_1||\rho_2) \tag{21}$$

$$= D_{\mathrm{K}}(\rho_1||\rho_2) + \sum_{i=1}^{m} \beta_i [\Phi(b_{1;i}) - \Phi(b_{2;i}) + \Phi'(b_{2;i}) C_i(\pi_1) - \Phi'(b_{2;i}) C_i(\pi_2))], \tag{22}$$

where $b_{\pi;i} = d_i - C_i(\pi)$. The corresponding trust-region scheme is

$$\pi_{k+1} \in \arg\max_{\pi \in \Pi} \mathbb{A}_r^{\pi_k}(\pi) \quad \text{sbj. to } D_{\mathrm{C}}(\rho_{\pi_k}||\rho_\pi) \leq \delta. \tag{23}$$

Analogously to the case of unconstrained TRPO, there is a corresponding natural policy gradient scheme:

$$\theta_{k+1} = \theta_k + \epsilon_k G_{\mathrm{C}}(\theta_k)^+ \nabla R(\theta_k), \tag{24}$$

where $G_{\mathrm{C}}(\theta)^+$ denotes an arbitrary pseudo-inverse of the Gramian

$$G_{\mathrm{C}}(\theta)_{ij} = \partial_{\theta_i} \rho_\theta^\top \nabla^2 \Phi_{\mathrm{C}}(\rho_\theta) \partial_{\theta_j} \rho_\theta.$$

The authors discuss that, under suitable parametrizations of $\theta \mapsto \pi$, this gradient preconditioner is a Riemannian metric on Θ_{safe} and natural policy gradient flows based on $G_{\mathrm{C}}(\theta_k)$ leave Θ_{safe} invariant. Further, $G_{\mathrm{C}}(\theta_k)^+$ is equivalent to the Hessian of D_{C}:

$$H_{\mathrm{C}}(\theta) = \mathbb{E}_{s \sim \rho_\theta} F(\theta) + \sum_i \beta_i \Phi''(b_i - C_i(\theta)) \nabla_\theta^2 C_i(\theta)\Big|_{\theta = \theta_k}.$$

where F is the fisher information of the policy. Unlike in TRPO, the divergence itself is not easy to estimate, however, the authors demonstrate that another divergence has the same Hessian, i.e. is equivalent up to second order in the policy parameters. It is derived using a "surrogate advantage trick" for C_i and results in the divergence

$$\bar{D}_{KL}(\pi, \pi_k) + \beta \bar{D}_\Phi(\pi, \pi_k) = \bar{D}_{KL}(\pi, \pi_k) + \beta \cdot [\phi(b_k - \mathbb{A}_c^{\pi_k}(\pi)) - \phi(b_k) - \phi'(b_k) \cdot \mathbb{A}_c^{\pi_k}(\pi)], \tag{25}$$

which is ultimately used as a drop-in replacement for the conventional divergence in TRPO.

A.3 Central Paths

In the small step size limit, the trajectories induced by trust region methods converge to the corresponding natural policy gradient (NPG) flow on the state-action polytope $\mathcal{K}$. The space of state-action occupancies $\rho \in \mathcal{K}$ forms not only a polytope, but a Hessian manifold Müller and Montúfar [2023]. C-TRPO induces such a gradient flow on the LP Equation 4 w.r.t the Hessian geometry induced by the convex function

$$\Phi(\rho) = \sum_{s,a} \rho(s,a) \log \pi(a|s) - \beta \log(\rho - \sum_{s,a} \rho(s,a) c(s,a)). \tag{26}$$

It is well known that Hessian gradient flows (ρ_t) of linear programs follow the central path, meaning that they are characterized as the optimizers of regularized linear programs with regularization strength t^{-1}. In policy space, we obtain for a single constraint

$$\pi_t = \arg\max\{R(\pi) + t^{-1} D_\Phi(\pi, \pi_0) : \pi, C(\pi) \leq d\}. \tag{27}$$

Since Φ curves infinitely towards the boundary of the feasible set of LP Equation 4, solving the problem posed by C-TRPO corresponds to solving LP Equation 4 using an interior point / barrier method with barrier $D_\Phi(\cdot, \pi_0)$. For a more detailed discussion of Hessian geometries and natural policy gradients see Alvarez et al. [2004], Müller and Montúfar [2023], Müller and Cayci [2024].

B Proofs of Section 3

B.1 Exact Penalty Methods

We provide a general result for the exactness of the penalties considered here for general discussions of exact penalty methods, we refer to the standard textbooks in optimization Bertsekas [1997], Nocedal and Wright [1999]. Here, we consider a compact subset $X \subset \mathbb{R}^n$ with non-empty interior a differentiable functions $f, g \in C^1(X)$ and the constrained optimization problem

$$\max f(x) \quad \text{subject to } g(x) \leq b, \tag{28}$$

where we impose Slater's condition $\{x \in X : g(x) < b\} \neq \varnothing$ to be non-empty and f to be concave and g to be convex. We denote the penalized functions by

$$P_\kappa(x) \coloneqq f(x) - \kappa \max\{0, g(x) - b\}. \tag{29}$$

Recall the definition of the Lagrangian

$$\mathcal{L}(x, \lambda) = f(x) - \lambda(g(x) - b). \tag{30}$$

Theorem B.1 (Exactness for convex programs)**.** *Assume that there exists a solution $x^\star \in X$ of equation 28 and denote the corresponding dual variable by $\lambda^\star \geq 0$. For $\kappa > \lambda^\star$ we have*

$$\arg\max\{f(x) : x \in X, g(x) \leq b\} = \arg\max\{P_\kappa(x) : x \in X\}. \tag{31}$$

Proof. Consider an infeasible point $\bar{x} \in X$ of P_κ, meaning that $g(\bar{x}) > b$. Note by convexity $x^\star$ maximizes the Lagrangian $\mathcal{L}(\cdot, \lambda^\star)$. Then

$$P_\kappa(\bar{x}) = f(\bar{x}) - \kappa\,(g(\bar{x}) - b) < f(\bar{x}) - \lambda^\star(g(\bar{x}) - b) = \mathcal{L}(\bar{x}, \lambda^\star) \leq \mathcal{L}(x^\star, \lambda^\star) = P_\kappa(x^\star).$$

Hence, every maximizer of P_κ is feasible and thus a solution of the regularized problem, showing the inclusion $\supseteq$. As P_κ agrees with f for feasible points, we also obtain that every maximizer of f over the feasible set is a maximizer of P_κ.

$\square$

B.2 C3PO Exact Penalty

Proposition 3.2 (Positive Exactness)**.** *For $0 \leq \mathbb{A}_c^{\pi_k}(\pi) < d - C(\pi_k)$, there exist w and δ_{B} for which the solution sets of problems 10, 11 and 12 agree.*

Proof. Let

$$P_{\text{Barrier}} \coloneqq \{\pi : D_{\mathrm{B}}(\pi, \pi_k) \leq \delta_B,\ \mathbb{A}_c^{\pi_k}(\pi) \geq 0\}$$

and

$$P_{\text{Lin}} \coloneqq \{\pi : \mathbb{A}_c^{\pi_k}(\pi) - w \cdot b \leq 0,\ \mathbb{A}_c^{\pi_k}(\pi) \geq 0\}.$$

Note that

$$D_{\mathrm{B}}(\pi, \pi_k) = \frac{b - \mathbb{A}_c^{\pi_k}(\pi)}{b} - \log\left(\frac{b - \mathbb{A}_c^{\pi_k}(\pi)}{b}\right) - 1 \tag{32}$$

is a strictly convex increasing function of $\mathbb{A}_c^{\pi_k}$ for $\mathbb{A}_c^{\pi_k} \geq 0$ (see Figure 2). This means that there exists a unique $\mathbb{A}_{\mathrm{B}} > 0$ that solves

$$\frac{b - \mathbb{A}_{\mathrm{B}}}{b} - \log\left(\frac{b - \mathbb{A}_{\mathrm{B}}}{b}\right) - 1 = \delta_{\mathrm{B}} \tag{33}$$

and for $\mathbb{A}_{\mathrm{B}} \geq \mathbb{A}_c^{\pi_k}(\pi) > 0$ it holds that $\delta_{\mathrm{B}} \geq D_{\mathrm{B}}(\pi, \pi_k) > 0$. To solve for $\mathbb{A}_{\mathrm{B}}$, we rewrite 33 as

$$\left(\frac{\mathbb{A}_{\mathrm{B}} - b}{b}\right) \exp\left(\frac{\mathbb{A}_{\mathrm{B}} - b}{b}\right) = -\exp(-\delta_{\mathrm{B}} - 1). \tag{34}$$

and use the definition of Lambert's W-Function Corless et al. [1996] to invert the left hand side as follows

$$\frac{\mathbb{A}_{\mathrm{B}} - b}{b} = W(-\exp(-\delta_{\mathrm{B}} - 1)), \tag{35}$$

where W is the real part of the principle branch of the W-Function. Finally, rearranging yields

$$\mathbb{A}_{\mathrm{B}} = b \cdot (W(-\exp(-\delta_{\mathrm{B}} - 1)) + 1). \tag{36}$$

Note that $b > \mathbb{A}_{\mathrm{B}} > 0$ must still hold. With this result,

$$\begin{aligned} P_{\text{Barrier}} &= \{\pi : \mathbb{A}_c^{\pi_k}(\pi) - \mathbb{A}_{\mathrm{B}} < 0,\ \mathbb{A}_c^{\pi_k}(\pi) \geq 0\}, & (37)\\ &= \{\pi : \mathbb{A}_c^{\pi_k}(\pi) - b\,(W(-\exp(-\delta_{\mathrm{B}} - 1)) + 1) < 0,\ \mathbb{A}_c^{\pi_k}(\pi) \geq 0\}, & (38)\\ &= \{\pi : \mathbb{A}_c^{\pi_k}(\pi) - b\, w < 0,\ \mathbb{A}_c^{\pi_k}(\pi) \geq 0\}, & (39) \end{aligned}$$

showing that $P_{\text{Barrier}} = P_{\text{Lin}}$ for a unique w.

Further, since $\min(b, w \cdot b) = w \cdot b$ for $b > 0$, the solution sets of

$$\max_{\pi \in \Pi} \mathbb{A}_r^{\pi_k}(\pi) \ \text{ s.t. } \ \mathbb{A}_c^{\pi_k}(\pi) - \mathbb{A}_{\mathrm{B}} < 0 \qquad \text{and } \ \bar{D}_{\mathrm{KL}}(\pi, \pi_k) < \delta_{\mathrm{KL}} \tag{40}$$

$$\max_{\pi \in \Pi} \mathbb{A}_r^{\pi_k}(\pi) \ \text{ s.t. } \ D_{\mathrm{B}}(\pi, \pi_k) < \delta_B \qquad \text{and } \ \bar{D}_{\mathrm{KL}}(\pi, \pi_k) < \delta_{\mathrm{KL}} \tag{41}$$

agree for $\mathbb{A}_c^{\pi_k}(\pi) \geq 0$ and $w = W(-\exp(-\delta_{\mathrm{B}} - 1)) + 1$.

Finally, by theorem B.1, they must also agree with the solutions of

$$\max_{\pi \in \Pi} \mathbb{A}_r^{\pi_k}(\pi) - \kappa_k \max\{0, \mathbb{A}_c^{\pi_k}(\pi) - \min(b, w \cdot b)\} \qquad \text{s.t. } \ \bar{D}_{\mathrm{KL}}(\pi, \pi_k) < \delta_{\mathrm{KL}}, \tag{42}$$

$$\max_{\pi \in \Pi} \mathbb{A}_r^{\pi_k}(\pi) - \kappa_k \max\{0, D_{\mathrm{B}}(\pi, \pi_k) - \delta_B\} \qquad \text{s.t. } \ \bar{D}_{\mathrm{KL}}(\pi, \pi_k) < \delta_{\mathrm{KL}}, \tag{43}$$

under the same conditions and for large enough κ. □

Note that the cost budget $b = d - C(\pi_k)$ is multiplied with a fixed function of δ_{B}. Hence, we can use w as the hyper-parameter immediately instead of defining it through δ_{B}.

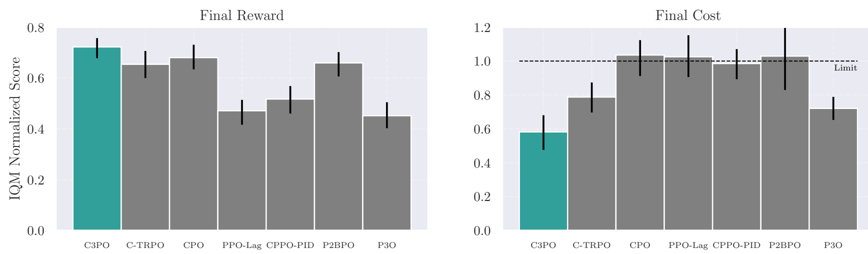

Figure 5: Aggregated performance using the inter quartile mean (IQM) across 8 tasks from Safety Gymansium for all algorithms (except P2BPO) across 8 tasks. P2BPO has been excluded, since the final cost (right) was off the charts. This may be due to the missing penalty coefficient in the algorithm.

C Experiment Details

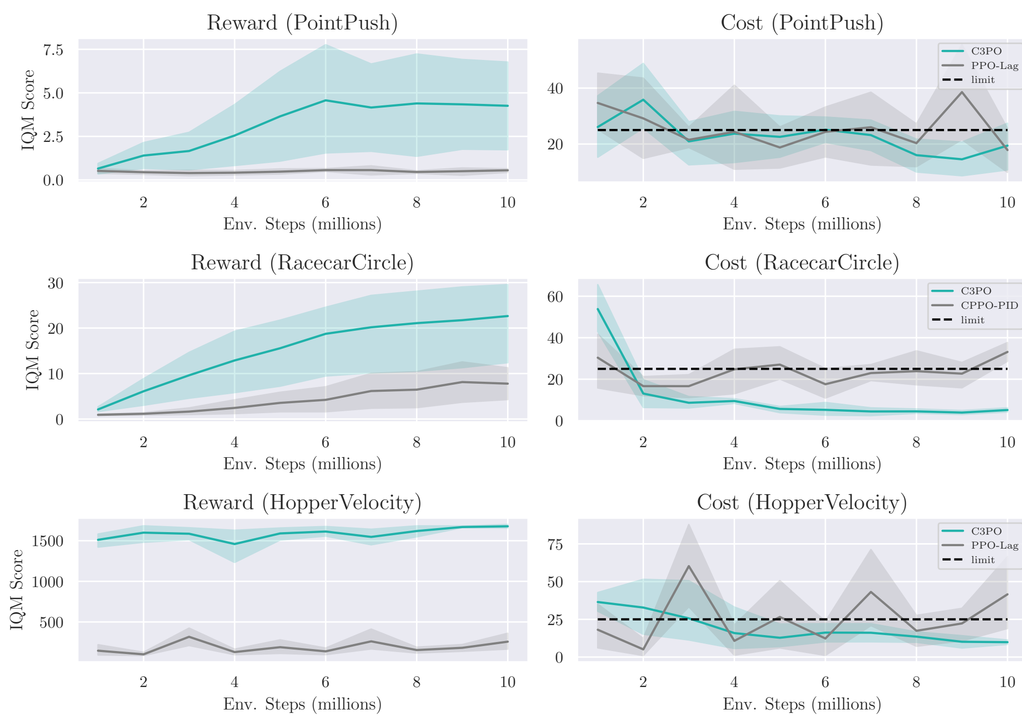

Figure 6: Hand-picked examples where central path approximation improves final reward performance.

Table 1: Performance of 8 representative safe policy optimization algorithms on 8 tasks from Safety Gymnasium for 10 million steps and a cost threshold of 25.0 aggregated over 5 seeds each. Bold marks the algorithm with the highest mean cumulative reward among the admissible ones. An algorithm is admissible, if its average cumulative cost achieved at the end of training is below the threshold.

		Ant	HalfCheetah	Humanoid	Hopper	CarButton	PointGoal	RacecarCircle	PointPush
C3PO	R	3043 ± 44	2458 ± 436	5389 ± 93	**1674 ± 35**	2.3 ± 0.7	23.8 ± 0.9	25.9 ± 5.1	**4.5 ± 2.6**
	C	15.0 ± 4.7	13.3 ± 6.4	1.2 ± 0.9	9.9 ± 1.7	53.4 ± 22.3	37.9 ± 1.7	5.0 ± 1.7	20.2 ± 10.0
C-TRPO	R	3019 ± 149	2841 ± 41	5746 ± 248	1621 ± 82	1.1 ± 0.2	**19.3 ± 0.9**	29.5 ± 3.1	1.0 ± 6.6
	C	13.2 ± 9.2	12.1 ± 7.6	12.2 ± 5.9	17.7 ± 8.0	34.0 ± 10.2	23.3 ± 3.6	20.2 ± 4.0	25.3 ± 7.0
CPO	R	3106 ± 21	2824 ± 104	5569 ± 349	1696 ± 19	1.1 ± 0.2	20.4 ± 2.0	**29.8 ± 1.9**	0.7 ± 2.9
	C	25.1 ± 11.3	23.1 ± 8.0	16.2 ± 8.6	25.7 ± 4.4	33.5 ± 8.7	28.2 ± 4.1	23.1 ± 4.5	28.9 ± 20.0
PPO-LAG	R	3210 ± 85	**3033 ± 1**	5814 ± 122	240 ± 159	0.3 ± 0.8	9.4 ± 1.8	30.9 ± 1.8	0.6 ± 0.0
	C	28.9 ± 8.7	23.2 ± 1.9	12.7 ± 31.0	38.8 ± 36.4	39.2 ± 41.1	22.5 ± 10.1	31.7 ± 2.7	18.2 ± 9.5
CPPO-PID	R	3205 ± 76	3036 ± 10	**5877 ± 84**	1657 ± 61	**-1.2 ± 0.6**	6.1 ± 4.8	8.1 ± 4.3	1.0 ± 1.1
	C	26.2 ± 4.4	26.5 ± 7.2	20.3 ± 6.0	18.6 ± 8.1	23.8 ± 6.0	21.8 ± 6.8	33.3 ± 5.9	22.8 ± 9.9
P2BPO	R	3269 ± 18	2928 ± 46	5293 ± 171	1573 ± 85	6.1 ± 0.9	25.9 ± 0.2	15.7 ± 7.5	1.1 ± 0.5
	C	32.3 ± 8.9	26.0 ± 19.7	1.5 ± 1.1	13.2 ± 11.7	125 ± 14	39.6 ± 5.7	5.5 ± 8.0	43.8 ± 28.9
P3O	R	**3122 ± 24**	3020 ± 12	5492 ± 118	1633 ± 49	0.2 ± 0.3	5.7 ± 0.3	0.9 ± 0.1	0.7 ± 0.6
	C	21.2 ± 2.5	27.0 ± 1.1	4.2 ± 2.2	14.6 ± 1.6	40.9 ± 18.2	17.1 ± 6.2	13.1 ± 4.6	14.1 ± 9.4
PPO	R	5402 ± 274	6583 ± 954	6138 ± 699	1810 ± 390	18.2 ± 1.2	26.6 ± 0.2	40.8 ± 0.5	0.9 ± 0.7
	C	887 ± 27	976 ± 1	783 ± 60	435 ± 85	378 ± 18	50.7 ± 3.3	200 ± 4	42.9 ± 24.0