SB-TRPO: TOWARDS SAFE REINFORCEMENT LEARNING WITH HARD CONSTRAINTS

Anonymous authors
Paper under double-blind review

ABSTRACT

In safety-critical domains, reinforcement learning (RL) systems must satisfy strict, zero-cost safety constraints while achieving meaningful task performance. Existing model-free methods can struggle to achieve high safety without substantially compromising task performance. We introduce *Safety-Biased Trust Region Policy Optimisation (SB-TRPO)*, a principled approach to RL with zero-cost constraints, which requires only a fixed fraction of the maximal cost reduction achievable within the trust region, thus retaining flexibility for reward optimisation. We show that the idealised update nevertheless converges to zero cost and maximal reward amongst zero-cost policies in finite MDPs. A practical gradient-based approximation provides local improvements in both safety and reward under suitable gradient alignment. Experiments on *Safety Gymnasium* demonstrate high safety alongside strong task performance.

1 INTRODUCTION

Reinforcement learning (RL) has achieved remarkable success in domains ranging from games to robotics, largely by optimising long-term cumulative rewards through trial-and-error interaction with an environment. However, in many real-world applications unconstrained reward maximisation is insufficient: agents must also satisfy safety or operational constraints. This problem is naturally formulated as a Constrained Markov Decision Process (CMDP, see Altman (2021)), which includes cost signals representing undesirable or unsafe outcomes in addition to rewards. Within this framework, safe RL has been studied under a wide range of objectives and algorithmic paradigms (see (Wachi et al., 2024; Gu et al., 2024b) for surveys).

Most existing work formulates safe RL as maximising discounted expected reward subject to the discounted expected cost remaining below a positive threshold (e.g., (Ray et al., 2019; Achiam et al., 2017; Yang et al., 2022)). This formulation is appropriate when costs represent *soft constraints*, such as resource consumption or wear-and-tear – for example, an autonomous robot that must complete its mission before its battery is depleted. In such cases, incurring costs can be acceptable provided their expected frequency remains low.

However, in many safety-critical domains any non-zero cost corresponds to catastrophic failure. Robots operating near humans cannot afford collisions, and autonomous vehicles must never violate speed or distance limits in safety-critical zones. In these settings, expected-cost formulations with a positive cost threshold are misaligned with the true safety objective: policies must avoid safety violations entirely, rather than merely keeping them small in expectation. This motivates a *hard-constrained formulation*, in which safety violations are unacceptable and the objective is to achieve zero cost with high reliability.

Limitations of Existing Approaches. Despite substantial progress in constrained RL, existing model-free methods struggle in hard-safety regimes (Belardinelli et al., 2025). Lagrangian approaches, such as TRPO- and PPO-Lagrangian (Ray et al., 2019), rely on penalty multipliers to trade off reward and cost. In practice, this often leads to either unsafe policies (when penalties are small) or overly conservative behaviour with poor task completion (when penalties are large), with performance highly sensitive to hyperparameter choices.

Projection-based methods such as Constrained Policy Optimization (CPO) (Achiam et al., 2017) explicitly attempt to maintain feasibility at each update. While appealing in principle, this often

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

results in severely restricted update directions or vanishing step sizes in challenging environments, leading to slow learning and low reward. More recent methods, including CRPO (Xu et al., 2021) and PCRPO Gu et al. (2024a), alternate between cost and reward optimisation or combine gradients via simple heuristics. When safety violations are frequent or hard to eliminate, these approaches tend to prioritise cost reduction almost exclusively, again resulting in poor task performance.

Overall, existing methods are primarily designed for *soft* constraints and degrade qualitatively when applied to hard-safety settings: they either violate safety constraints or achieve safety at the cost of extremely low task performance. This gap motivates algorithms that are explicitly designed for hard-constrained reinforcement learning and can make reliable progress towards safety while still enabling meaningful reward optimisation.

Our Approach. We propose *Safety-Biased Trust Region Policy Optimisation (SB-TRPO)*, a principled approach to reinforcement learning with hard safety constraints. Its idealised update only requires a fixed fraction of the maximal achievable cost reduction within the trust region, controlled by a *safety bias* parameter, while retaining flexibility for reward improvement. Unlike CPO and related methods, SB-TRPO *does not* switch into separate feasibility-recovery phases, avoiding over-conservatism and allowing reward improvement throughout training. In SB-TRPO, we approximate the idealised update using a dynamically computed convex combination of cost and reward TRPO directions.

Contributions. We summarise our main contributions:

- **Trust-Region Formulation for Zero-Cost Constraints.** We introduce an idealised policy update that controls the required cost reduction while retaining flexibility for reward optimisation, and develop a gradient-based approximation for its practical implementation.
- **Asymptotic and Local Guarantees.** For the idealised update we show convergence to zero cost and maximal reward amongst zero-cost policies in finite MDPs. For the gradient-based approximation we instead establish local improvements in both safety and reward under suitable gradient alignment.
- **Empirical Evaluation.** On a suite of *Safety Gymnasium* tasks, SB-TRPO demonstrates strong task performance alongside high levels of safety.

2 Related Work

Primal–Dual Methods. Primal–dual methods for constrained RL optimise a Lagrangian objective jointly over the policy and Lagrange multiplier to enforce cost constraints. RCPO (Tessler et al., 2018) is an early two-timescale scheme using vanilla policy gradient and slow multiplier updates. Under additional assumptions (e.g. that all local optima are feasible), it converges to a locally optimal feasible policy. Likewise, TRPO-Lagrangian and PPO-Lagrangian (Ray et al., 2019) use trust-region or clipped-surrogate updates for the primal policy parameters. PID Lagrangian methods (Stooke et al., 2020) augment the standard Lagrangian update with additional terms, reducing oscillations when overshooting the safety target is possible. Similarly, APPO (Dai et al., 2023), which is based on PPO, augments the Lagrangian with a quadratic deviation term to dampen cost oscillations.

Trust-Region and Projection-Based Methods. CPO (Achiam et al., 2017) enforces a local approximation of the CMDP constraints within a trust region and performs pure cost-gradient steps when constraints are violated, aiming to always keep policy updates feasible. FOCOPS (Zhang et al., 2020) is a first-order method solving almost the same abstract problem for policy updates in the nonparameterised policy space and then projects it back to the parameterised policy. P3O (Zhang et al., 2022) is a PPO-based method motivated by CPO. It uses a Lagrange multiplier, increasing linearly to a fixed upper bound, and clipped surrogate updates to balance reward improvement with constraint satisfaction. CUP (Yang et al., 2022), in contrast to CPO, formulates surrogate reward and cost objectives using generalised performance bounds and GAEs, and projects each policy gradient update into the feasible set defined by these surrogates, allowing updates to jointly respect reward and cost approximations. Milosevic et al. (2024) propose C-TRPO, which reshapes the geometry of the policy space by adding a barrier term to the KL divergence so trust regions contain only safe

policies. Like CPO, C-TRPO switches to a dedicated recovery phase with pure cost-gradient updates when the policy becomes infeasible.

Reward–Cost Switching. CRPO (Xu et al., 2021) is a constrained RL method that alternates between reward maximisation and cost minimisation depending on whether the current cost estimate indicates the policy is infeasible. Motivated by CRPO's tendency to oscillate between purely reward- and cost-focused updates, Gu et al. (2024a) propose PCRPO, which mitigates conflicts between reward and safety gradients using a softer switching mechanism and ideas from gradient manipulation Yu et al. (2020); Liu et al. (2021). When both objectives are optimised, the gradients are averaged, and if their angle exceeds 90°, each is projected onto the normal plane of the other.

Model-Based and Shielding Approaches. Shielding methods (Alshiekh et al., 2018; Belardinelli et al., 2025; Jansen et al., 2020) intervene when a policy proposes unsafe actions, e.g. via lookahead or model predictive control, but typically require accurate dynamic models and can be costly. (Yu et al., 2022) train an auxiliary policy to edit unsafe actions, balancing reward against the extent of editing. Other typically model-based methods leverage control theory (Perkins & Barto, 2002; Berkenkamp et al., 2017; Chow et al., 2018; Wang et al., 2023). These approaches are complementary to our model-free approach.

3 Problem Formulation

We consider a *constrained Markov Decision Process* (CMDP) defined by the tuple $(\mathcal{S}, \mathcal{A}, P, r, c, \gamma)$, where $\mathcal{S}$ and $\mathcal{A}$ are the (potentially continuous) state and action spaces, $P(s' \mid s, a)$ is the transition kernel, $r : \mathcal{S} \to \mathbb{R}$ is the reward function, $c : \mathcal{S} \to \mathbb{R}_{\geq 0}$ is the cost function representing unsafe events, and $\gamma \in (0, 1)$ is the discount factor. A stochastic policy $\pi(a \mid s)$ induces a distribution over trajectories $\tau = (s_0, a_0, s_1, a_1, \dots)$ together with P. Furthermore, we denote the standard discounted occupancy measure by d_π.

In safety-critical domains, the objective is to maximise discounted expected reward while ensuring that unsafe states are never visited, i.e. $c(s_t) = 0$ for all times t almost surely:

$$\max_\pi \quad J_r(\pi) \coloneqq \mathbb{E}_{\tau\sim\pi}\left[\sum_{t=0}^{\infty} \gamma^t \cdot r(s_t)\right] \qquad \text{s.t.} \qquad \mathbb{P}_{\tau\sim\pi}\left[\exists t \in \mathbb{N}.\, c(s_t) > 0\right] = 0 \quad \textbf{(Problem 1)}$$

Since costs are non-negative, any policy feasible under the following standard CMDP with expected discounted cost threshold of 0 is also feasible for the original (**Problem 1**):

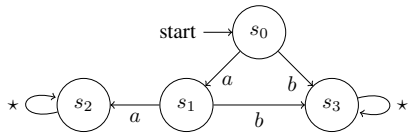
$$\max_\pi \quad J_r(\pi) \coloneqq \mathbb{E}_{\tau\sim\pi}\left[\sum_{t=0}^{\infty} \gamma^t \cdot r(s_t)\right] \quad \text{s.t.} \quad J_c(\pi) \coloneqq \mathbb{E}_{\tau\sim\pi}\left[\sum_{t=0}^{\infty} \gamma^t \cdot c(s_t)\right] = 0 \quad \textbf{(Problem 2)}$$

This re-formulation enables the use of off-the-shelf constrained policy optimisation techniques whilst in principle still enforcing strictly safe behaviour.

Discussion. Most prior work in Safe RL focuses on *positive cost thresholds* (e.g. 25 in *Safety Gymnasium* (Ji et al., 2023)). However, in genuinely safety-critical applications there is no meaningful notion of an "acceptable" amount of catastrophic failure. Positive thresholds conflate the *problem specification* with an *algorithmic hyperparameter*, making training brittle, environment-dependent, and misaligned with true safety objectives (see Section 5.2.2).

Our zero-cost formulation instead directly encodes the intrinsic hard-safety requirement. Conceptually, it provides a clear, principled target for algorithm design: policies should strictly avoid unsafe states, and algorithm design can focus on achieving this reliably rather than tuning a threshold that only approximates strict safety and explicitly permits a certain level of unsafety.

Finally, we focus on the safety of the *learned deployment policy*, rather than safe exploration during training. We assume training takes place in simulation or a controlled environment where exploratory violations are safely contained; safe exploration is beyond our scope.



$$r(s_1) \coloneqq 1, \qquad r(s_i) \coloneqq 0 \quad (i \neq 1)$$
$$c(s_2) \coloneqq 1, \qquad c(s_j) \coloneqq 0 \quad (j \neq 2)$$

Figure 1: Deterministic MDP for which CPO yields reward-suboptimal policies.

4 METHOD

We first derive the idealised SB-TRPO update, before presenting a practical approximation. Finally, we summarise the overall method in a complete algorithmic description.

4.1 SAFETY-BIASED TRUST REGION UPDATE

In the setting of hard (zero-cost) constraints, the idealised CPO (Achiam et al., 2017) update seeks to improve the current policy π_k by maximising reward amongst feasible policies in the KL trust region:

$$\max_{\pi} \quad J_r(\pi) \quad \text{s.t.} \quad J_c(\pi) \leq 0, \qquad \bar{D}_{\mathrm{KL}}(\pi_k \parallel \pi) \leq \delta \tag{CPO}$$

where $\bar{D}_{\mathrm{KL}}(\pi \parallel \pi') \coloneqq \mathbb{E}_{s\sim d_\pi}[D_{\mathrm{KL}}(\pi(\cdot \mid s) \parallel \pi'(\cdot \mid s))]$. When no such policy exists, CPO (and related methods such as C-TRPO) switches to a recovery step:

$$\min_{\pi} \quad J_c(\pi) \quad \text{s.t.} \quad \bar{D}_{\mathrm{KL}}(\pi_k \parallel \pi) \leq \delta \tag{Recovery}$$

This deliberately ignores task reward in favour of maximising cost reduction and can lead to reward-suboptimal zero-cost policies:

Example 1. Consider the MDP in Fig. 1. The policy $\pi^*(a \mid s_0) = 1 = \pi^*(b \mid s_1)$ achieves $J_c(\pi^*) = 0$ and $J_r(\pi^*) = \gamma$. Starting from an initial policy π_0 satisfying $\pi_0(a \mid s_0), \pi_0(a \mid s_1) \in (0, 1)$, subsequent updates can never increase the probability of taking a at s_0: doing so increases the probability of reaching s_1, where action a incurs the cost with positive probability, whilst not decreasing the KL-divergence (see Section B.1.1 for further details). Consequently, $\sup_{k\in\mathbb{N}} J_r(\pi_k) \leq \gamma \cdot \pi_0(a \mid s_0) < J_r(\pi^*)$.

This limitation is also observed empirically: under zero-cost constraints, CPO often yields policies with low cost but poor task reward (cf. Section 5 and (Belardinelli et al., 2025)).

To address this issue, we propose a more general update rule that seeks high reward while requiring only a controlled *reduction* of cost:

$$\max_{\pi} \quad J_r(\pi) \qquad \text{s.t.} \qquad J_c(\pi) \leq J_c(\pi_k) - \epsilon_k, \quad \bar{D}_{\mathrm{KL}}(\pi_k \parallel \pi) \leq \delta \tag{**Update 1**}$$

where for a *safety bias* $\beta \in (0, 1]$, we define

$$\epsilon_k \coloneqq \beta \cdot (J_c(\pi_k) - c_k^*) \qquad \text{where} \quad c_k^* \coloneqq \min_{\pi} \; J_c(\pi) \quad \text{s.t.} \quad \bar{D}_{\mathrm{KL}}(\pi_k \parallel \pi) \leq \delta \tag{1}$$

Intuitively, ϵ_k is a β-fraction of the best achievable cost improvement inside the trust region. This ensures that (**Update 1**) always admits at least one solution, $\epsilon_k \geq 0$, and avoids the need for an explicit recovery step.

Importantly, (**Update 1**) does *not* correspond to solving a CMDP with a fixed relaxed cost threshold $J_c(\pi) \leq d$, see also Theorem 1: the required cost decrease is dynamically adjusted according to the current policy via Eq. (1), whilst the parameter β controls how aggressively cost is reduced.

$\beta = 1$ **Recovers CPO.** Setting $\beta = 1$ forces each update to pursue *maximal* cost reduction within the trust region, simplifying the cost constraint in (**Update 1**) to $J_c(\pi) \leq c_k^*$. Note that $c_k^* = 0$ iff (CPO) is feasible. Thus, (**Update 1**) for $\beta = 1$ elegantly *captures both the recovery and standard feasible phases of CPO* (breaking ties during recovery favourably for reward).

Intuition for Using a Safety Bias $\beta < 1$. Choosing $\beta < 1$ intentionally relaxes this requirement: the policy must still reduce cost, but the available policy change can be allocated more flexibly rather than being used entirely to maximise the immediate cost reduction. In Example 1, this leaves part of the KL budget available for reducing the probability of the unsafe action a in s_1, rather than forcing the update to reduce the probability of reaching s_1 in the first place.

Asymptotic Optimality. Despite requiring only a fraction of the maximal immediate cost improvement, (**Update 1**) yields asymptotic convergence to the minimal cost (for $\beta \in (0, 1]$) and to the corresponding optimal reward (for $\beta \in (0, 1)$) in finite MDPs:

Theorem 1 (Asymptotic Optimality for (**Update 1**))**.** *Assume the MDP is finite. Let π_k be obtained via (**Update 1**) for arbitrary π_0. Define $J_c^* \coloneqq \min_\pi J_c(\pi)$ and $J_r^* \coloneqq \max_{\pi: J_c(\pi) = J_c^*} J_r(\pi)$.*

1. *If $\beta \in (0, 1]$ then $\lim_{k\to\infty} J_c(\pi_k) = J_c^*$ (and the limit exists).*
2. *If $\beta \in (0, 1)$ then also $\lim_{k\to\infty} J_r(\pi_k) = J_r^*$ (and the limit exists).*

In particular, when zero-cost policies exist, the theorem implies that $J_c(\pi_k) \to 0$.

Proof sketch. Full details are provided in Section B.1.2. The cost sequence is monotonically non-increasing by construction. If an accumulation point were cost-suboptimal, it would admit a cost-improving policy strictly within the δ-KL trust region, which eventually also lies within the trust region around π_k along the corresponding subsequence. Thus, eventually even the required fractional improvement would yield a strict improvement over the accumulation point.

For the reward result, consider a subsequence approaching an accumulation point with suboptimal reward. Mixing this accumulation point with a reward-optimal cost-optimal policy yields a policy π' that remains cost-optimal while lying strictly within the KL trust region. We can then consider policies mixing each iterate with an β-fraction of π', which eventually become feasible (since $\beta < 1$) for the update and yield a strict reward improvement. □

4.2 Approximate Solution

Since direct evaluation of $J_r(\pi)$ and $J_c(\pi)$ for arbitrary π is infeasible, we employ the standard *surrogate objectives* (Schulman et al., 2015) with respect to a reference policy π_k. For reward,

$$\mathcal{L}_{r,\pi_k}(\pi) \coloneqq J_r(\pi_k) + \mathbb{E}_{s,a\sim d_{\pi_k}}\left[\frac{\pi(a \mid s)}{\pi_k(a \mid s)} \cdot A_r^{\pi_k}(s, a)\right]$$

where $A_r^{\pi_k}$ denotes the advantage function. The Policy Improvement Bound (Schulman et al., 2015; Li et al., 2022) bounds $|\mathcal{L}_{r,\pi_k}(\pi) - J_r(\pi)|$ in terms of the KL-divergence between π_k and π (analogously for cost). In particular, $J_r(\pi_k) = \mathcal{L}_{r,\pi_k}(\pi_k)$. Hence, we may approximate (**Update 1**):

$$\max_\pi \quad \mathcal{L}_{r,\pi_k}(\pi) \qquad \text{s.t.} \qquad \mathcal{L}_{c,\pi_k}(\pi) \le \mathcal{L}_{c,\pi_k}(\pi_k) - \epsilon_k, \qquad \bar{D}_{\mathrm{KL}}(\pi_k \parallel \pi) \le \delta \quad \textbf{(Update 2)}$$

As in TRPO, the KL constraint guarantees that the surrogates remain close to the true performances.

Second-Order Approximation. Henceforth, we assume differentiably parameterised policies π_θ and overload notation, e.g. $J_c(\theta)$ for $J_c(\pi_\theta)$. Linearising the reward and cost objectives around the current parameters θ_k and approximating the KL divergence by a quadratic form with the Fisher information matrix F yields the quadratic program

$$\max_{\Delta\in\mathbb{R}^d} \langle g_r, \Delta\rangle \quad \text{s.t.} \quad \langle g_c, \Delta\rangle \le -\epsilon_k, \quad \tfrac{1}{2}\cdot\Delta^\top\cdot F\cdot\Delta \le \delta, \qquad \textbf{(Update 3)}$$

where $g_r \coloneqq \nabla\mathcal{L}_{r,\theta_k}(\theta_k)$ and $g_c \coloneqq \nabla\mathcal{L}_{c,\theta_k}(\theta_k)$. To choose ϵ_k and approximate Eq. (1), we analogously approximate the trust-region step that maximally decreases the cost surrogate

$$\Delta_c \coloneqq \arg\min_\Delta \langle g_c, \Delta\rangle \quad \text{s.t.} \quad \tfrac{1}{2}\cdot\Delta^\top\cdot F\cdot\Delta \le \delta. \tag{2}$$

and set $\epsilon_k \coloneqq -\beta\cdot\langle g_c, \Delta_c\rangle$ with safety-bias $\beta \in (0, 1]$. This guarantees feasibility of (**Update 3**). The new parameters are then updated via $\theta_{k+1} \coloneqq \theta_k + \Delta^*$, with Δ^* the solution of (**Update 3**), analogous to the TRPO step but enforcing an explicit local cost reduction.

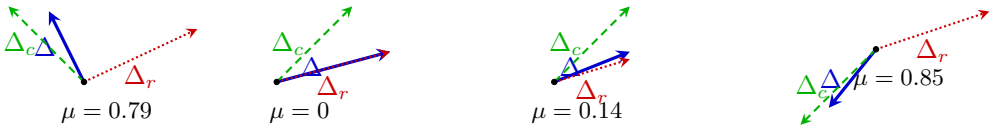


Figure 2: Visualisation of the adaptive convex combination Δ of Δ_r and Δ_c given by Eq. (4) for $\epsilon_k \coloneqq 1.4 = -\beta \cdot \langle g_c, \Delta_c \rangle$, where $\beta \coloneqq 0.7$, and the special case that $\Delta_r = g_r$ and $\Delta_c = -g_c$.

Approximate Solution via Cost-Biased Convex Combination. Solving (**Update 3**) analytically requires computing the natural gradients $F^{-1}g_r$ and $F^{-1}g_c$ and their coefficients (cf. Lemma 8 in Section B.2), which is numerically fragile for large or ill-conditioned F. Instead, we follow TRPO and compute the KL-constrained reward and cost steps separately using the conjugate gradient method (Hestenes et al., 1952):

$$\Delta_r \coloneqq \arg\max_{\Delta} \langle g_r, \Delta \rangle \quad \text{s.t.} \quad \tfrac{1}{2} \cdot \Delta^\top \cdot F \cdot \Delta \leq \delta \tag{3}$$

with Δ_c given by Eq. (2). Both steps satisfy the same KL constraint, and thus any convex combination $\Delta \coloneqq (1-\mu) \cdot \Delta_r + \mu \cdot \Delta_c$, where $\mu \in [0, 1]$, also satisfies it (see Lemma 6). We therefore choose the best reward-improving combination which also satisfies $\langle g_c, \Delta \rangle \leq -\epsilon_k$ (see Fig. 2 for a visualisation and Lemma 7):

$$\mu \coloneqq \begin{cases} \frac{\langle g_c, \Delta_r \rangle + \epsilon_k}{\langle g_c, \Delta_r \rangle - \langle g_c, \Delta_c \rangle} & \text{if } \langle g_c, \Delta_c \rangle \neq \langle g_c, \Delta_r \rangle \text{ and } \langle g_c, \Delta_r \rangle \geq -\epsilon_k \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

Performance Improvement Guarantee. This approximation of (**Update 3**) ensures an approximate cost decreases at each update:

$$J_c(\theta + \Delta) - J_c(\theta) \approx \langle g_c, \Delta \rangle \leq \beta \cdot \langle g_c, \Delta_c \rangle \leq 0$$

Likewise, if the reward gradient is non-zero and its angle with Δ_c does not exceed $90°$, an increase in reward is guaranteed. Formally, sufficiently small steps along Δ are required for the local gradient approximations to hold:

Theorem 2 (Performance Improvement)**.** *Let $\Delta \coloneqq (1-\mu) \cdot \Delta_r + \mu \cdot \Delta_c$ for Δ_r, Δ_c and μ defined in Eqs.* (2) *to* (4)*, respectively. There exists $\eta \in (0, 1]$ such that*

1. *if $g_c \neq 0$ then $J_c(\theta + \eta \cdot \Delta) < J_c(\theta)$*
2. *if $\beta < 1$, $g_r \neq 0$ and $\langle g_r, \Delta_c \rangle \geq 0$ (in particular, $g_c = 0$) then $J_r(\theta + \eta \cdot \Delta) \geq J_r(\theta)$.*

This result complements Theorem 1 with guarantees for the gradient-based approximate solution of (**Update 1**). Unlike CPO and related methods such as C-TRPO (Milosevic et al., 2024), which have a separate feasibility recovery phase, our update can guarantee reward improvement for all steps in which gradients are well-aligned. Once the cost gradient is zero, updates focus on reward improvement (see also Fig. 7). Overall, this scheme dynamically balances cost and reward objectives, naturally stabilising learning.

4.3 Practical Algorithm

Our practical algorithm, implementing the ideas of the preceding subsections, is given in Algorithm 1. We now highlight the key additions.

Line Search. In practice, we follow standard TRPO procedures by performing a line search along the update direction Δ: the KL constraint is enforced empirically to ensure that the surrogate approximations of J_r and J_c remain accurate; and the empirical reduction in the surrogate cost loss is checked to operationalise the cost decrease guarantee of Theorem 2, scaling the step if necessary.

Gradient Estimation. The gradients of the surrogate objective $g_r = \nabla \mathcal{L}_{r,\theta_k}(\theta_k)$ (analogously for cost) coincide with $\nabla J_r(\theta_k)$, which is Monte Carlo estimated using the policy gradient theorem. The Fisher information matrix F is similarly estimated from trajectories.

Algorithm 1 Safety-Biased Trust Region Policy Optimisation (SB-TRPO)

Require: KL divergence limit $\delta > 0$, safety bias $\beta \in [0, 1]$, training epochs $N \in \mathbb{N}$

initialise θ
$\kappa \leftarrow 10^{-8}$ {small constant to avoid division by 0}
for N epochs **do**
  $\mathcal{D} \leftarrow$ collect rollouts under policy π_θ
  $g_r, g_c \leftarrow$ policy gradient estimate of $\nabla\mathcal{L}_{r,\pi_\theta}(\pi_\theta)$ and $\nabla\mathcal{L}_{c,\pi_\theta}(\pi_\theta)$ using $\mathcal{D}$
  $\Delta_r, \Delta_c \leftarrow$ calculated via the conjugate gradient algorithm to optimise Eqs. (2) and (3)
  $\mu \leftarrow \max\left\{0, \frac{\langle g_c, \Delta_r\rangle - \beta\cdot\langle g_c, \Delta_c\rangle}{\langle g_c, \Delta_r\rangle - \langle g_c, \Delta_c\rangle + \kappa}\right\}$ {substituting definition of ϵ_k in Eq. (4)}
  $\Delta \leftarrow (1-\mu)\cdot\Delta_r + \mu\cdot\Delta_c$
  $\eta \leftarrow 1$ {line search for constraint satisfaction}
  repeat
    decrease η
  until $\widehat{D_{KL}}[\pi_\theta \parallel \pi_{\theta+\eta\cdot\Delta}] \leq \delta$ and $\widehat{\mathcal{L}_{c,\pi_\theta}}(\pi_{\theta+\eta\cdot\Delta}) \leq \widehat{\mathcal{L}_{c,\pi_\theta}}(\pi_\theta)$
  $\theta \leftarrow \theta + \eta\cdot\Delta$
end for

5 Empirical Evaluation

5.1 Experimental Setup

Benchmarks. *Safety Gymnasium* provides a broad suite of environments for benchmarking safe RL algorithms. Originally introduced by OpenAI (Ray et al., 2019), it has since been maintained and extended by Ji et al. (2023), who also developed *SafePO*, a library of safe RL algorithms. We focus on two classes of environments: *Safe Navigation* and *Safe Velocity*. In Safe Navigation, a robot equipped with LIDAR sensors must complete tasks while avoiding unsafe hazards. We use the *Point* and *Car* robots on tasks *Circle*, *Push* and *Goal* at *level 2*, which is most challenging due to the greatest number of hazards. In contrast, *Safe Velocity* environments test adherence to velocity limits, and we use the *Hopper* and *Swimmer* robots adapted from the MuJoCo locomotion suite (Todorov et al., 2012). The state spaces are up to 88-dimensional; see Section C.1.1 for more details.

Baselines. We compare our approach against a wide range of state-of-the-art implementations from SafePO (Ji et al., 2023) of model-free constrained RL algorithms: TRPO- and PPO-Lagrangian (Ray et al., 2019), CPO (Achiam et al., 2017), CUP (Yang et al., 2022), CPPO-PID (Stooke et al., 2020), FOCOPS (Zhang et al., 2020), RCPO (Tessler et al., 2018), PCPO (Yang et al., 2020), P3O (Zhang et al., 2022) and C-TRPO (Milosevic et al., 2024).[1] All baselines require a cost tolerance threshold, which we set to 0 to target almost-sure safety (cf. (**Problem 2**)).

Implementation Details. Following the default setup in SafePO, we use a homoscedastic Gaussian policy with state-independent variance, the mean of which is parameterised by a feedforward network with two hidden layers of 64 units. Besides, we use SafePO's defaults for existing hyperparameter (cf. Section C.1.2), and *the same safety-bias* $\beta = 0.75$ *across all tasks* (cf. Section 5.3).

We consider two variants of SB-TRPO: one using Monte Carlo advantage estimates and one using Generalized Advantage Estimation (GAE) with a learned critic. We compare them in Section 5.3 and use the Monte Carlo variant for the main results in Table 1.

Training Details. We train the policies for 2×10^7 time steps (10^3 epochs), running 2×10^4 steps per epoch with 20 vectorised environments per seed. All results are reported as averages over 5 seeded parameter initialisations, with standard deviations indicating variability across seeds.

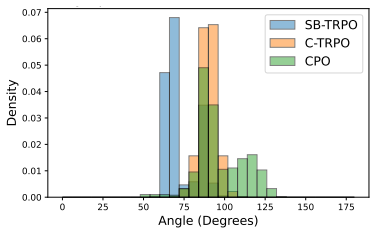
Metrics. We compare our performance with the baselines using the standard metrics of *rewards* and *costs*, averaged over the past 50 episodes during training.

[1] Code of reward–cost switching methods such as (P)CRPO (Xu et al., 2021; Gu et al., 2024a) and APPO (Dai et al., 2023) is not publicly available (Gu et al., 2024b).

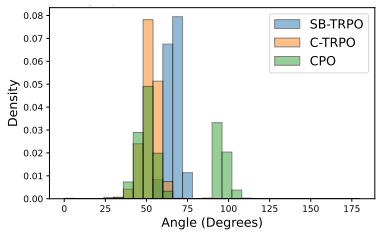
Table 1: Final metrics across level-2 *Safety Gymnasium* tasks. Red: negative (safe) reward, italics: safety probability $< 50\%$, grey: Pareto-dominated pairs, bold: best value per metric (only among methods with positive (safe) reward and safety probability $\geq 50\%$ for the other metric).

Method	Car Circle		Point Push		Car Push		Point Goal		Car Goal		Hopper Velocity		Swimmer Velocity	
	Safe Rew.↑	Safe Prob.↑	Safe Rew.↑	Safe Prob.↑	Safe Rew.↑	Safe Prob.↑	Safe Rew.↑	Safe Prob.↑	Safe Rew.↑	Safe Prob.↑	Safe Rew.↑	Safe Prob.↑	Safe Rew.↑	Safe Prob.↑
Ours	7.5 ± 1.5	$\mathbf{0.99 \pm 0.01}$	$\mathbf{0.33 \pm 0.18}$	$\mathbf{0.79 \pm 0.08}$	$\mathbf{0.38 \pm 0.12}$	0.71 ± 0.07	$\mathbf{1.6 \pm 0.5}$	0.68 ± 0.07	$\mathbf{1.4 \pm 0.4}$	0.66 ± 0.08	860 ± 378	1.0 ± 0.0	48 ± 26	$\mathbf{0.98 \pm 0.02}$
PPO-Lag	7.5 ± 2.4	0.86 ± 0.17	-0.29 ± 0.8	0.59 ± 0.22	-0.13 ± 0.34	0.65 ± 0.1	*1.3 ± 1.1*	*0.47 ± 0.13*	*1.2 ± 1.2*	*0.45 ± 0.16*	510 ± 500	0.58 ± 0.44	18 ± 38	0.57 ± 0.42
TRPO-Lag	11 ± 4	0.81 ± 0.26	-0.49 ± 0.45	0.78 ± 0.16	-0.19 ± 0.25	0.89 ± 0.04	0.09 ± 0.29	$\mathbf{0.79 \pm 0.06}$	0.23 ± 0.38	$\mathbf{0.75 \pm 0.07}$	72 ± 90	0.6 ± 0.5	*1.3 ± 5.1*	*0.33 ± 0.42*
CPO	0.96 ± 0.76	0.99 ± 0.03	-1.2 ± 0.9	0.77 ± 0.19	-1.3 ± 0.3	0.93 ± 0.05	-1.3 ± 0.4	0.86 ± 0.05	-0.73 ± 0.23	0.82 ± 0.09	880 ± 182	$\mathbf{1.0 \pm 0.0}$	-3.8 ± 9.3	0.99 ± 0.03
CUP	7.1 ± 3.5	0.75 ± 0.28	0.12 ± 0.19	0.67 ± 0.19	*-0.0062 ± 0.1729*	*0.47 ± 0.19*	*0.95 ± 0.74*	*0.15 ± 0.09*	*0.93 ± 0.74*	*0.15 ± 0.1*	$\mathbf{1000 \pm 289}$	$\mathbf{0.72 \pm 0.19}$	34 ± 52	0.9 ± 0.1
CPPO-PID	0.96 ± 1.85	0.91 ± 0.12	-2.7 ± 2.7	0.79 ± 0.11	-0.71 ± 0.58	0.72 ± 0.08	-1.8 ± 0.7	0.81 ± 0.07	-0.59 ± 0.5	0.69 ± 0.08	860 ± 263	0.84 ± 0.16	-0.33 ± 9.93	0.95 ± 0.03
FOCOPS	8.1 ± 2.0	0.96 ± 0.09	-0.2 ± 0.5	0.66 ± 0.3	0.05 ± 0.251	0.72 ± 0.08	*2.2 ± 1.2*	*0.34 ± 0.05*	*1.6 ± 2.3*	*0.27 ± 0.3*	$\mathbf{970 \pm 307}$	$\mathbf{1.0 \pm 0.0}$	15 ± 10	0.88 ± 0.09
RCPO	$\mathbf{10 \pm 3}$	$\mathbf{0.9 \pm 0.18}$	-0.48 ± 0.45	0.84 ± 0.08	-0.48 ± 0.4	0.81 ± 0.07	-0.012 ± 0.366	0.85 ± 0.06	$\mathbf{0.21 \pm 0.4}$	$\mathbf{0.77 \pm 0.06}$	420 ± 364	1.0 ± 0.0	*7.0 ± 12.3*	0.67 ± 0.25
PCPO	$\mathbf{11 \pm 2}$	0.89 ± 0.17	-1.1 ± 0.7	0.79 ± 0.11	-0.48 ± 0.38	0.82 ± 0.06	-1.3 ± 0.6	0.81 ± 0.08	-0.75 ± 0.26	0.77 ± 0.08	460 ± 501	0.72 ± 0.36	*5 ± 10*	*0.13 ± 0.26*
P3O	7.1 ± 2.4	0.94 ± 0.07	0.23 ± 0.12	0.78 ± 0.1	$\mathbf{0.29 \pm 0.13}$	$\mathbf{0.76 \pm 0.06}$	-0.024 ± 0.521	0.64 ± 0.06	0.36 ± 0.29	0.52 ± 0.11	110.0 ± 78	0.98 ± 0.017	$\mathbf{94 \pm 51}$	0.88 ± 0.08
C-TRPO	0.54 ± 0.5	0.97 ± 0.05	-1.4 ± 1.2	0.83 ± 0.09	-1.2 ± 0.5	0.88 ± 0.05	-2.2 ± 1.1	0.88 ± 0.06	-1.9 ± 0.5	0.89 ± 0.07	4.2 ± 1.7	$\mathbf{1 \pm 0}$	-18 ± 8	0.97 ± 0.03

Method	Car Circle		Point Push		Car Push		Point Goal		Car Goal		Hopper Velocity		Swimmer Velocity	
	Rewards↑	Costs↓	Rewards↑	Costs↓	Rewards↑	Costs↓	Rewards↑	Costs↓	Rewards↑	Costs↓	Rewards↑	Costs↓	Rewards↑	Costs↓
Ours	7.5 ± 1.5	$\mathbf{0.28 \pm 0.66}$	$\mathbf{0.43 \pm 0.21}$	$\mathbf{8.0 \pm 6.5}$	$\mathbf{0.53 \pm 0.16}$	$\mathbf{14 \pm 7}$	$\mathbf{2.4 \pm 0.6}$	18 ± 6	$\mathbf{2.1 \pm 0.6}$	18 ± 6	860 ± 377	0.00028 ± 0.00208	49 ± 27	0.87 ± 3.24
PPO-Lag	8.7 ± 2.0	22 ± 37	-0.47 ± 1.31	27 ± 21	-0.2 ± 0.52	34 ± 30	*2.8 ± 2.2*	*48 ± 18*	*2.6 ± 2.5*	*54 ± 29*	890 ± 507	2.9 ± 2.8	40 ± 60	6.5 ± 7.2
TRPO-Lag	$\mathbf{13 \pm 1}$	18 ± 33	-0.65 ± 0.55	9.3 ± 14.5	-0.21 ± 0.27	6.3 ± 4.6	0.11 ± 0.35	$\mathbf{17 \pm 9}$	0.29 ± 0.5	$\mathbf{16 \pm 7}$	160 ± 77	2.1 ± 2.6	*21 ± 16*	*6.3 ± 6.0*
CPO	0.96 ± 0.75	0.98 ± 6.63	-1.6 ± 1.1	5.8 ± 9.6	-1.4 ± 0.3	4.2 ± 4.4	-1.5 ± 0.5	9.3 ± 8.0	-0.89 ± 0.27	13 ± 8	$\mathbf{880 \pm 185}$	$\mathbf{0.0 \pm 0.0}$	-3.8 ± 9.4	0.016 ± 0.037
CUP	9.6 ± 2.5	48 ± 86	0.18 ± 0.26	25 ± 22	*-0.026 ± 0.362*	*85 ± 50*	*7.2 ± 2.8*	*110 ± 28*	*6.1 ± 2.8*	*110 ± 31*	$\mathbf{1400 \pm 114}$	$\mathbf{0.43 \pm 0.35}$	38 ± 58	0.14 ± 0.18
CPPO-PID	1.3 ± 1.8	27 ± 55	-3.4 ± 3.3	13 ± 17	-0.99 ± 0.78	34 ± 28	-2.3 ± 0.9	17 ± 10	-0.79 ± 0.69	23 ± 10	$\mathbf{1000 \pm 228}$	$\mathbf{0.081 \pm 0.107}$	$\mathbf{0.71 \pm 10.89}$	$\mathbf{0.044 \pm 0.081}$
FOCOPS	8.4 ± 1.9	3.9 ± 18.2	-0.29 ± 0.76	35 ± 61	0.069 ± 0.338	35 ± 24	*6.5 ± 3.3*	*88 ± 37*	*6.0 ± 5.3*	*86 ± 49*	970 ± 309	0.093 ± 0.135	20 ± 12	3.0 ± 5.7
RCPO	11 ± 2	13 ± 35	-0.57 ± 0.51	6.3 ± 8.8	-0.6 ± 0.49	19 ± 20	-0.00026 ± 0.41018	12 ± 7	0.29 ± 0.5	$\mathbf{16 \pm 7}$	470 ± 361	0.97 ± 1.93	19 ± 18	5.1 ± 6.7
PCPO	$\mathbf{12 \pm 1}$	$\mathbf{4.4 \pm 8.5}$	-1.4 ± 0.8	14 ± 23	-0.58 ± 0.46	17 ± 18	-1.6 ± 0.7	18 ± 13	-0.95 ± 0.3	17 ± 9	750 ± 589	3.9 ± 4.9	*-8.2 ± 17*	*73 ± 39*
P3O	7.6 ± 2.3	11 ± 17	0.3 ± 0.1	15 ± 11	0.38 ± 0.15	28 ± 12	0.05 ± 0.77	36 ± 18	0.73 ± 0.5	40 ± 17	110.0 ± 78	0.028 ± 0.025	$\mathbf{110 \pm 58}$	$\mathbf{0.22 \pm 0.16}$
C-TRPO	0.56 ± 0.48	3.9 ± 7.4	-1.7 ± 1.4	7.8 ± 12.2	-1.3 ± 0.5	8.9 ± 11.5	-2.5 ± 1.3	8.7 ± 5.9	-2.1 ± 0.6	8.5 ± 6.7	4.2 ± 1.7	0.0000078 ± 0.000078	-18 ± 8	0.058 ± 0.18



(a) Policy updates vs. reward gradients



(b) Policy updates vs. negated cost gradients

Figure 3: Angles between policy updates and reward/cost gradients for Car Circle

To capture safety and task performance under hard constraints (as formalised in (**Problem 1**)), we introduce two additional metrics: *safety probability* is the fraction of episodes completed without any safety violation, i.e. with zero cost; *safe reward* is the average return over all episodes, with returns from episodes with any safety violation set to 0 (see Eq. (6) in Section C.1.3).

5.2 Results

The results are reported in Table 1. Our method consistently achieves the best performance in safe reward (among methods with at least 50% safety probability) or safety probability (among methods with positive safe reward), demonstrating its ability to attain both high safety and meaningful task performance. When another method attains a higher safe reward, it typically comes at the expense of a substantial reduction in safety; conversely, methods with higher safety probabilities generally achieve significantly lower safe reward. These trends also manifest for raw rewards and costs. Moreover, regardless of task performance, safety remains below near-perfect levels across all methods on the harder tasks.

PPO-Lagrangian often collapses to poor reward, poor safety, or both. Baselines such as TRPO-Lagrangian, CPO, and C-TRPO can achieve higher safety on some harder tasks, but often at the cost of very low (mostly negative) rewards. CUP and FOCOPS generally exhibit lower safety.

In summary, our approach consistently achieves strong task performance and high levels of safety across challenging *Safety Gymnasium* tasks.

5.2.1 Alignment of Policy Updates with Gradients

To further understand why SB-TRPO avoids the over-conservatism of similar methods, we analyse the angles between policy updates and reward/cost gradients (Figs. 3 and 8). CPO and C-TRPO updates are largely at best orthogonal to the reward gradient (angles around or above $90°$), indicating that these methods spend considerable time in the recovery phase. In contrast, SB-TRPO updates lie below $90°$, mostly around $60°$–$70°$ for Car Circle, confirming the performance improvement

guarantee of Theorem 2 empirically. At the same time, angles relative to the cost gradient can be larger for SB-TRPO than for CPO/C-TRPO but well below $75°$, reflecting a measured approach to feasibility that balances safety with meaningful task progress.

5.2.2 Impact of Cost Limit on Baseline Safety

We study the performance of baselines under a positive cost threshold of 15 (see Table 5). This ad-hoc relaxation departs from the target zero-cost safety objective (**Problem 2**) by *explicitly allowing unsafe events*. On easier tasks such as the Velocity and Car Circle environments, the baselines exploit the higher cost allowance, becoming significantly less safe in line with the threshold, without substantially improving rewards. On harder tasks, rewards increase modestly at a clear cost to safety.

These results highlight the sensitivity to the choice of cost limit: treating it as a tunable, task-dependent hyperparameter can lead to significantly reduced safety without commensurate reward gains.

5.3 Ablation Studies

Safety Bias. We evaluate the effect of the safety bias β over $\beta \in [0.6, 0.9]$. Fig. 9 in Section C.3.1 shows that varying β shifts SB-TRPO policies along a nearly linear reward-safety probability Pareto frontier: higher values emphasise safety while lower values yield greater reward. The baselines typically underperform relative to this frontier (or achieve negative reward). This demonstrates that SB-TRPO is not overly sensitive to the choice of β, with $\beta = 0.75$ achieving excellent results throughout (see Table 1).

Advantage Estimation. We ablate the effect of advantage estimation (see Section 5.1) in Table 6 (Section C.3.2). At a fixed safety bias $\beta = 0.75$, the Monte Carlo (MC) variant of SB-TRPO attains both higher safety and lower reward than its GAE counterpart. Increasing the safety bias for GAE ($\beta = 0.8$) yields comparable safety and reward levels, indicating that the two variants achieve similar safety–reward trade-offs after a simple adjustment of the safety bias. From a practical perspective, the MC variant obviates critic training, reducing computational cost by more than an order of magnitude (see Table 4), motivating its use in our main experiments.

6 Discussion

Our empirical results show that framing policy updates as (**Update 3**) — dynamically balancing progress towards safety and task performance — helps avoid collapse into trivially safe but task-ineffective regions under strict safety constraints. Unlike CPO and related methods such as C-TRPO, SB-TRPO has *no separate recovery phase*, instead incorporating reward and cost objectives throughout training. Specifically, SB-TRPO's updates align well with both reward and cost gradients, whereas CPO-style updates are often at best orthogonal to the reward gradient. Consistent with Theorem 2, we observe *steady improvement in both safety and task performance*.

Lagrangian methods face a different limitation in the hard-constraint regime: under a zero cost limit, Lagrange multipliers can only increase, making performance highly sensitive to initialisation. Small values can permit unsafe behaviour, while large values can induce persistent over-conservatism.

Finally, even methods which explicitly focus on maintaining feasibility such as CPO do not achieve near-perfect safety on the harder *Safety Gymnasium* tasks. This suggests that the remaining empirical gap in safety is driven in part by factors shared across methods, particularly sparse cost signals and finite-sample gradient noise. In practice, we therefore recommend pairing SB-TRPO with deployment-time safeguards.

Conclusion. In summary, SB-TRPO is a principled approach to hard-constrained RL that achieves strong task performance alongside high levels of safety on *Safety Gymnasium* tasks.

Limitations. As is common (Schulman et al., 2015; Gu et al., 2024a), Theorem 2 assumes exact gradients, while Theorem 1 assumes finite MDPs. SB-TRPO does not directly support arbitrary positive cost limits, although Theorem 1 guarantees the lowest achievable cost, even when positive.

AI USE STATEMENT

In this work, we used generative AI tools to provide feedback on research methodology and assist with checking lemmas and proofs. We have not used generative AI tools to generate synthetic datasets, propose hypotheses, implement methods, assist with translation, clean or reformat datasets, support qualitative or thematic data analysis, or interpret experimental results, and the remaining required-disclosure tasks are not applicable to this work. Additionally, we used generative AI tools to review related work and improve the clarity and presentation of the manuscript. We have reviewed all AI-assisted work and independently verified mathematical arguments. We take responsibility for the final content of this work, including text, claims, and artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This work does not involve human subjects or personal data. All experiments are conducted in simulation using publicly available *Safety Gymnasium* environments. We are not aware of any additional ethical concerns arising from the methods or experiments presented in this work.

REPRODUCIBILITY STATEMENT

Full details of the experimental setup are provided in Sections C.1 and 5.1, including instructions for reproducing our results in Section C.1.4, whilst code is provided in the supplementary materials. Complete proofs of our theoretical results are provided in Section B.

REFERENCES

Joshua Achiam, David Held, Aviv Tamar, and Pieter Abbeel. Constrained policy optimization. In Doina Precup and Yee Whye Teh (eds.), *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pp. 22–31. PMLR, 06–11 Aug 2017.

Mohammed Alshiekh, Roderick Bloem, Rüdiger Ehlers, Bettina Könighofer, Scott Niekum, and Ufuk Topcu. Safe reinforcement learning via shielding. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

Eitan Altman. *Constrained Markov decision processes*. Routledge, 2021.

Francesco Belardinelli, Alexander W Goodall, et al. Probabilistic shielding for safe reinforcement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 16091–16099, 2025.

Felix Berkenkamp, Matteo Turchetta, Angela Schoellig, and Andreas Krause. Safe model-based reinforcement learning with stability guarantees. *Advances in neural information processing systems*, 30, 2017.

Yinlam Chow, Ofir Nachum, Edgar Duenez-Guzman, and Mohammad Ghavamzadeh. A lyapunov-based approach to safe reinforcement learning. *Advances in neural information processing systems*, 31, 2018.

Juntao Dai, Jiaming Ji, Long Yang, Qian Zheng, and Gang Pan. Augmented proximal policy optimization for safe reinforcement learning. *Proceedings of the AAAI Conference on Artificial Intelligence*, 37(6):7288–7295, Jun. 2023. doi: 10.1609/aaai.v37i6.25888.

Shangding Gu, Bilgehan Sel, Yuhao Ding, Lu Wang, Qingwei Lin, Ming Jin, and Alois Knoll. Balance reward and safety optimization for safe reinforcement learning: A perspective of gradient manipulation. *Proceedings of the AAAI Conference on Artificial Intelligence*, 38(19):21099–21106, Mar. 2024a. doi: 10.1609/aaai.v38i19.30102.

Shangding Gu, Long Yang, Yali Du, Guang Chen, Florian Walter, Jun Wang, and Alois Knoll. A review of safe reinforcement learning: Methods, theories, and applications. *IEEE Trans. Pattern Anal. Mach. Intell.*, 46(12):11216–11235, 2024b. doi: 10.1109/TPAMI.2024.3457538.

Magnus R Hestenes, Eduard Stiefel, et al. Methods of conjugate gradients for solving linear systems. *Journal of research of the National Bureau of Standards*, 49(6):409–436, 1952.

Nils Jansen, Bettina Könighofer, Sebastian Junges, Alex Serban, and Roderick Bloem. Safe reinforcement learning using probabilistic shields. In *31st International Conference on Concurrency Theory (CONCUR 2020)*, pp. 3–1. Schloss Dagstuhl–Leibniz-Zentrum für Informatik, 2020.

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.

Hepeng Li, Nicholas Clavette, and Haibo He. An analytical update rule for general policy optimization. In *International Conference on Machine Learning*, pp. 12696–12716. PMLR, 2022.

Bo Liu, Xingchao Liu, Xiaojie Jin, Peter Stone, and Qiang Liu. Conflict-averse gradient descent for multi-task learning. *Advances in Neural Information Processing Systems*, 34:18878–18890, 2021.

Nikola Milosevic, Johannes Müller, and Nico Scherf. Embedding safety into RL: A new take on trust region methods. *CoRR*, abs/2411.02957, 2024. doi: 10.48550/ARXIV.2411.02957. URL https://doi.org/10.48550/arXiv.2411.02957.

Theodore J Perkins and Andrew G Barto. Lyapunov design for safe reinforcement learning. *Journal of Machine Learning Research*, 3(Dec):803–832, 2002.

Alex Ray, Josh Achiam, and Dario Amodei. Benchmarking safe exploration in deep reinforcement learning. volume 7, pp. 2, 2019.

John Schulman, Sergey Levine, Pieter Abbeel, Michael Jordan, and Philipp Moritz. Trust region policy optimization. In Francis Bach and David Blei (eds.), *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pp. 1889–1897, Lille, France, 07–09 Jul 2015. PMLR.

Adam Stooke, Joshua Achiam, and Pieter Abbeel. Responsive safety in reinforcement learning by pid lagrangian methods, 2020.

Chen Tessler, Daniel J. Mankowitz, and Shie Mannor. Reward constrained policy optimization. *CoRR*, abs/1805.11074, 2018.

Emanuel Todorov, Tom Erez, and Yuval Tassa. Mujoco: A physics engine for model-based control. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 5026–5033. IEEE, 2012. doi: 10.1109/IROS.2012.6386109.

Akifumi Wachi, Xun Shen, and Yanan Sui. A survey of constraint formulations in safe reinforcement learning. In *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence, IJCAI 2024, Jeju, South Korea, August 3-9, 2024*, pp. 8262–8271. ijcai.org, 2024.

Yixuan Wang, Simon Sinong Zhan, Ruochen Jiao, Zhilu Wang, Wanxin Jin, Zhuoran Yang, Zhaoran Wang, Chao Huang, and Qi Zhu. Enforcing hard constraints with soft barriers: Safe reinforcement learning in unknown stochastic environments. In *International Conference on Machine Learning*, pp. 36593–36604. PMLR, 2023.

Tengyu Xu, Yingbin Liang, and Guanghui Lan. Crpo: A new approach for safe reinforcement learning with convergence guarantee. In Marina Meila and Tong Zhang (eds.), *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 11480–11491. PMLR, 18–24 Jul 2021.

Long Yang, Jiaming Ji, Juntao Dai, Linrui Zhang, Binbin Zhou, Pengfei Li, Yaodong Yang, and Gang Pan. Constrained update projection approach to safe policy optimization. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems*, volume 35, pp. 9111–9124. Curran Associates, Inc., 2022.

Tsung-Yen Yang, Justinian Rosca, Karthik Narasimhan, and Peter J Ramadge. Projection-based constrained policy optimization. *arXiv preprint arXiv:2010.03152*, 2020.

Haonan Yu, Wei Xu, and Haichao Zhang. Towards safe reinforcement learning with a safety editor policy. In Sanmi Koyejo, S. Mohamed, A. Agarwal, Danielle Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*, 2022.

Tianhe Yu, Saurabh Kumar, Abhishek Gupta, Sergey Levine, Karol Hausman, and Chelsea Finn. Gradient surgery for multi-task learning. *Advances in neural information processing systems*, 33: 5824–5836, 2020.

Linrui Zhang, Li Shen, Long Yang, Shixiang Chen, Xueqian Wang, Bo Yuan, and Dacheng Tao. Penalized proximal policy optimization for safe reinforcement learning. In Lud De Raedt (ed.), *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI-22*, pp. 3744–3750. International Joint Conferences on Artificial Intelligence Organization, 7 2022. doi: 10.24963/ijcai.2022/520. Main Track.

Yiming Zhang, Quan Vuong, and Keith Ross. First order constrained optimization in policy space. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 15338–15349. Curran Associates, Inc., 2020.

A Motivation of Zero-Cost CMDP Formulation

In many safety-critical domains, the appropriate requirement is that the agent *never* enters unsafe states (e.g. when they encode damage to an expensive robot or even accidents of autonomous cars). In the discounted-cost CMDP formalism with non-negative costs, this requirement is expressed *exactly* by imposing the zero-threshold constraint $J_c(\pi) = 0$. Introducing a positive threshold $b > 0$ effectively permits some violations and, crucially, adds a hyperparameter that encodes an arbitrary notion of "acceptable risk".

The level of violation required for learning to progress varies greatly across tasks (e.g. around 17 on the hardest navigation environments, around 10 for PointPush, and well below 1 for velocity or circle tasks) as well as across algorithms, neural network architectures and optimisation hyperparameters. This makes positive-threshold CMDPs brittle, environment-dependent, and potentially misaligned with true safety objectives. Conceptually, a positive cost threshold in the context of hard safety is undesirable because it conflates the *problem statement/specification* with an *algorithmic hyperparameter*.

For these reasons, we focus on the *intrinsically meaningful* zero-threshold (hard-constraint) CMDP, which avoids the ambiguity of selecting b, removes the tuning burden associated with calibrating allowable violations, and directly matches the requirement of "almost surely no unsafe events".

Finally, we believe that the zero-cost problem is significantly under-explored in model-free Safe RL, despite being the more appropriate formulation for capturing critical safety violations in practice. By explicitly targeting this regime, our work aims to help shift attention towards this practically important but comparatively understudied problem class.

B Supplementary Materials for Section 4

B.1 Supplementary Materials for Section 4.1

B.1.1 Details for Example 1

Recall the MDP in Fig. 1. Suppose π_0 is such that $\pi_0(a \mid s_0), \pi_0(a \mid s_1) \in (0, 1)$ and let π_k for $k \geq 1$ be obtained by iterating the (Recovery) or (CPO) updates (whichever applicable).

First, note that due to the KL constraint for all $k \in \mathbb{N}$, $\pi_k(a \mid s_0), \pi_k(a \mid s_1) \in (0, 1)$ (otherwise the KL is infinite).

Moreover, for all $k \in \mathbb{N}$, $\pi_{k+1}(a \mid s_0) \leq \pi_k(a \mid s_0)$. Otherwise, there exists $k \in \mathbb{N}$ with $\pi_{k+1}(a \mid s_0) > \pi_k(a \mid s_0)$. Define π'_{k+1} by $\pi'_{k+1}(a \mid s_0) \coloneqq \pi_k(a \mid s_0)$ and $\pi'_{k+1}(a \mid s_1) \coloneqq \pi_{k+1}(a \mid s_1)$. Note that $\bar{D}_{\mathrm{KL}}(\pi_k \parallel \pi'_{k+1}) < \bar{D}_{\mathrm{KL}}(\pi_k \parallel \pi_{k+1})$ and $J_c(\pi'_{k+1}) < J_c(\pi_{k+1})$ since $\pi_{k+1}(a \mid s_1) > 0$.

B.1.2 Proof of Theorem 1

In this subsection we assume the MDP is finite. For a policy π, recall the discounted state and state-action occupancy measures

$$d_\pi(s) \coloneqq (1-\gamma) \cdot \sum_{t \in \mathbb{N}} \gamma^t \cdot \mathbb{E}_{\tau \sim \pi} [s_t = s]$$

$$d_\pi(s, a) \coloneqq (1-\gamma) \cdot \sum_{t \in \mathbb{N}} \gamma^t \cdot \mathbb{E}_{\tau \sim \pi} [s_t = s, a_t = a] = \pi(a \mid s) \cdot d_\pi(s)$$

Besides, for policies π_0 and π_1 and $\lambda \in [0, 1]$ define the mixture $(1-\lambda) \cdot \pi_0 + \lambda \cdot \pi_1$ by

$$((1-\lambda) \cdot \pi_0 + \lambda \cdot \pi_1)(a \mid s) \coloneqq \begin{cases} \frac{(1-\lambda) \cdot d_{\pi_0}(s,a) + \lambda \cdot d_{\pi_1}(a,s)}{(1-\lambda) \cdot d_{\pi_0}(s) + \lambda \cdot d_{\pi_1}(s)} & \text{if } (1-\lambda) \cdot d_{\pi_0}(s) + \lambda \cdot d_{\pi_1}(s) \neq 0 \\ \pi_0(a \mid s) & \text{otherwise} \end{cases}$$

Note that $d_{(1-\lambda) \cdot \pi_0 + \lambda \cdot \pi_1} = (1-\lambda) \cdot d_{\pi_0} + \lambda \cdot d_{\pi_1}$ and hence $J_r((1-\lambda) \cdot \pi_0 + \lambda \cdot \pi_1) = (1-\lambda) \cdot J_r(\pi_0) + \lambda \cdot J_r(\pi_1)$ (similarly for cost).

Lemma 1. *Let π and π' be policies and $\lambda \in [0, 1]$. Then $\bar{D}_{\mathrm{KL}}(\pi \parallel (1-\lambda) \cdot \pi + \lambda \cdot \pi') \leq \bar{D}_{\mathrm{KL}}(\pi \parallel \pi')$.*

Proof. Define $\pi'' \coloneqq (1-\lambda)\cdot\pi + \lambda\cdot\pi'$. Let $s \in \mathcal{S}$ be such that $d_\pi(s) > 0$. Note that for $\alpha(s) \coloneqq \frac{\lambda\cdot d_{\pi'}(s)}{(1-\lambda)\cdot d_\pi(s)+\lambda\cdot d_{\pi'}(s)} \in [0,1]$, $\pi''(a \mid s) = (1-\alpha(s))\cdot\pi(a \mid s) + \alpha(s)\cdot\pi'(a \mid s)$. Hence, by convexity of the KL-divergence,

$$\begin{aligned}\bar{D}_{\mathrm{KL}}(\pi(\cdot \mid s) \parallel \pi''(\cdot \mid s)) &\leq (1-\alpha(s))\cdot\bar{D}_{\mathrm{KL}}(\pi(\cdot \mid s) \parallel \pi(\cdot \mid s)) + \alpha(s)\cdot\bar{D}_{\mathrm{KL}}(\pi(\cdot \mid s) \parallel \pi'(\cdot \mid s)) \\ &\leq \bar{D}_{\mathrm{KL}}(\pi(\cdot \mid s) \parallel \pi'(\cdot \mid s))\end{aligned}$$

and the claim follows since $d_\pi(s) > 0$ was arbitrary. □

Lemma 2. *Let π, π' be policies. Then*

$$\lim_{\lambda \searrow 0} \bar{D}_{\mathrm{KL}}(\pi \parallel (1-\lambda)\cdot\pi + \lambda\cdot\pi') = 0$$

Proof. Let $s \in \mathcal{S}$ and $a \in \mathcal{A}$ be such that $d_\pi(s) > 0$ and $\pi(a \mid s) > 0$. Hence, for $\lambda \in [0,1)$,

$$\begin{aligned}(1-\lambda)\cdot d_\pi(s,a) + \lambda\cdot d_{\pi'}(a,s) > 0 \\ (1-\lambda)\cdot d_\pi(s) + \lambda\cdot d_{\pi'}(s) > 0\end{aligned}$$

and by continuity of $\log$ on $(0,\infty)$,

$$\lim_{\lambda\searrow 0} \log \frac{(1-\lambda)\cdot d_\pi(s,a) + \lambda\cdot d_{\pi'}(a,s)}{(1-\lambda)\cdot d_\pi(s) + \lambda\cdot d_{\pi'}(s)} = \log \frac{d_\pi(s,a)}{d_\pi(s)} = \log \pi(a \mid s)$$

Since $\mathcal{S}$ and $\mathcal{A}$ are finite, the result follows. □

Lemma 3. *Let π' be a policy and $(\pi_k)_{k\in\mathbb{N}}$ be a sequence of policies converging to $\bar{\pi}$, i.e. $\lim_{k\to\infty} \pi_k(a \mid s) = \bar{\pi}(a \mid s)$ for all $s \in \mathcal{S}$ and $a \in \mathcal{A}$. Then for $\lambda \in (0,1)$,*

$$\lim_{k\to\infty} \bar{D}_{\mathrm{KL}}(\pi_k \parallel (1-\lambda)\cdot\pi_k + \lambda\cdot\pi') = \bar{D}_{\mathrm{KL}}(\bar{\pi} \parallel (1-\lambda)\cdot\bar{\pi} + \lambda\cdot\pi')$$

Proof. To reduce notational clutter we use d_k for d_{π_k}, $\bar{d}$ for $d_{\bar{\pi}}$, and d' for $d_{\pi'}$. Note that

$$\lim_{k\to\infty} \pi_k(a \mid s)\cdot\log\pi_k(a \mid s) = \bar{\pi}(a \mid s)\cdot\log\bar{\pi}(a \mid s)$$

Hence, it suffices to show

$$\begin{aligned}&\lim_{k\to\infty} d_k(s)\cdot\pi_k(a \mid s)\cdot\log\frac{(1-\lambda)\cdot d_k(s) + \lambda\cdot d'(s)}{(1-\lambda)\cdot d_k(s,a) + \lambda\cdot d'(s,a)} \\ &= \bar{d}(s)\cdot\bar{\pi}(a \mid s)\cdot\log\frac{(1-\lambda)\cdot\bar{d}(s) + \lambda\cdot d'(s)}{(1-\lambda)\cdot\bar{d}(s,a) + \lambda\cdot d'(s,a)}\end{aligned}$$

for all $s \in \mathcal{S}$ and $a \in \mathcal{A}$.

First if $d'(s) = 0$, hence, $d'(s,a) = 0$, then

$$\begin{aligned}&\lim_{k\to\infty} d_k(s)\cdot\pi_k(a \mid s)\cdot\log\frac{(1-\lambda)\cdot d_k(s) + \lambda\cdot d'(s)}{(1-\lambda)\cdot d_k(s,a) + \lambda\cdot d'(s,a)} \\ &= -\lim_{k\to\infty} d_k(s)\cdot\pi_k(a \mid s)\cdot\log\pi_k(s,a) \\ &= -\bar{d}(s)\cdot\bar{\pi}(a \mid s)\cdot\log\bar{\pi}(s,a) \\ &= \bar{d}(s)\cdot\bar{\pi}(a \mid s)\cdot\log\frac{(1-\lambda)\cdot\bar{d}(s) + \lambda\cdot d'(s)}{(1-\lambda)\cdot\bar{d}(s,a) + \lambda\cdot d'(s,a)}\end{aligned}$$

Next, if $d'(s) > 0$ then

$$\lim_{k\to\infty} \log\left((1-\lambda)\cdot d_k(s) + \lambda\cdot d'(s)\right) = \log(1-\lambda)\cdot\bar{d}(s) + \lambda\cdot d'(s)$$

due to $\lambda\cdot d'(s) > 0$ and the continuity of $\log$ on $(0,\infty)$. Hence, it suffices to show

$$\begin{aligned}&\lim_{k\to\infty} d_k(s)\cdot\pi_k(a \mid s)\cdot\log\left((1-\lambda)\cdot d_k(s,a) + \lambda\cdot d'(s,a)\right) \\ &= \bar{d}(s)\cdot\bar{\pi}(a \mid s)\cdot\log\left((1-\lambda)\cdot\bar{d}(s,a) + \lambda\cdot d'(s,a)\right)\end{aligned}$$

If $d'(s,a) > 0$ then this is again due to the continuity of $\log$. Otherwise

$$\begin{aligned}
&\lim_{k\to\infty} d_k(s)\cdot\pi_k(a \mid s)\cdot\log\left((1-\lambda)\cdot d_k(s,a)+\lambda\cdot d'(s,a)\right)\\
&=\lim_{k\to\infty} d_k(s,a)\cdot\log(1-\lambda)+\lim_{k\to\infty} d_k(s,a)\cdot\log d_k(s,a)\\
&=\bar{d}(s,a)\cdot\log(1-\lambda)+\bar{d}(s)\cdot\bar{\pi}(a \mid s)\cdot\log\bar{d}(s,a)\\
&=\bar{d}(s)\cdot\bar{\pi}(a \mid s)\cdot\log\big((1-\lambda)\cdot\bar{d}(s,a)+\lambda\cdot d'(s,a)\big)
\end{aligned}$$

since $\lambda < 1$. □

Lemma 4. *Let $\beta \in (0,1]$, π_0 be a policy and π_k be obtained via (**Update 1**) for $k \geq 1$. Then $\lim_{k\to\infty} J_c(\pi_k) = \min_\pi J_c(\pi)$ (in particular $J_c(\pi_k)$ converges).*

Proof. First, note that since $\bar{D}_{\mathrm{KL}}(\pi \parallel \pi) = 0$ for all policies π, $J_c(\pi_{k+1}) \leq J_c(\pi_k)$ for all $k \in \mathbb{N}$. Hence, since J_c is bounded, $J_c(\pi_k)$ converges.

We assume towards contradiction that there exists a policy π' such that $J_c(\pi') < \lim_{k\to\infty} J_c(\pi_k)$. Clearly, there exists a subsequence $(\pi_{k_i})_{i\in\mathbb{N}}$ of $(\pi_k)_{k\in\mathbb{N}}$ for which π_{k_i} converges to some policy $\bar{\pi}$, i.e. $\lim_{i\to\infty}\pi_{k_i}(a \mid s) = \bar{\pi}(a \mid s)$ for all $s \in \mathcal{S}$ and $a \in \mathcal{A}$. By Lemmas 1 and 2 there exists a policy π'' satisfying $\bar{D}_{\mathrm{KL}}(\bar{\pi} \parallel \frac{1}{2}\bar{\pi}+\frac{1}{2}\pi'') < \delta$ and $J_c(\pi'') < J_c(\bar{\pi})$. By Lemma 3 there exists $K \in \mathbb{N}$ such that $\bar{D}_{\mathrm{KL}}(\pi_k \parallel \frac{1}{2}\pi_k+\frac{1}{2}\pi'') < \delta$ for all $k \geq K$. Moreover, there exists $k \geq K$ such that $(1-\frac{\beta}{2})\cdot J_c(\pi_k)+\beta\cdot J_c(\pi'') < J_c(\bar{\pi})$. Hence, $J_c(\pi_{k+1}) \leq J_c(\pi_k) - \beta(J_c(\pi_k) - (\frac{1}{2}J_c(\pi_k) + \frac{1}{2}J_c(\pi'))) < J_c(\bar{\pi})$ and due to monotonicity, $\lim_{k\to\infty} J_c(\pi_k) < J_c(\bar{\pi})$, which is a contradiction since J_c is continuous. □

Theorem 1 (Asymptotic Optimality for (**Update 1**))**.** *Assume the MDP is finite. Let π_k be obtained via (**Update 1**) for arbitrary π_0. Define $J_c^* \coloneqq \min_\pi J_c(\pi)$ and $J_r^* \coloneqq \max_{\pi: J_c(\pi)=J_c^*} J_r(\pi)$.*

1. *If $\beta \in (0,1]$ then $\lim_{k\to\infty} J_c(\pi_k) = J_c^*$ (and the limit exists).*
2. *If $\beta \in (0,1)$ then also $\lim_{k\to\infty} J_r(\pi_k) = J_r^*$ (and the limit exists).*

Proof. Let π^* be a policy satisfying $J_c(\pi^*) = J_c^*$ and $J_r(\pi^*) = J_r^*$. The first part is due to Lemma 4.

To show the second, note that because of $\lim_{k\to\infty} J_c(\pi_k) = J_c^*$ and continuity, $\limsup_{k\to\infty} J_r(\pi_k) \leq J_r^*$. Hence, it suffices to show $\liminf_{k\to\infty} J_r(\pi_k) \geq J_r^*$.

Otherwise, there exists a subsequence π_{j_i} for which $J_r(\pi_{j_i})$ converges to $\lim_{i\to\infty} J_r(\pi_{j_i}) = \liminf_{k\to\infty} J_r(\pi_k) < J_r^*$.

Due to compactness of $[0,1]^{\mathcal{S}\times\mathcal{A}}$ there exists a subsequence k_i of j_i for which π_{k_i-1} converges to some $\bar{\pi}$.

By Lemma 2 there exists $\lambda \in (0,1]$ such that for $\pi' \coloneqq (1-\lambda)\cdot\bar{\pi}+\lambda\cdot\pi^*$ it holds $\bar{D}_{\mathrm{KL}}(\bar{\pi} \parallel \pi') < \delta$. Note that $J_c(\pi') = c^* = J_c(\pi^*) = J_c(\bar{\pi})$. Besides, by Lemma 1, $\bar{D}_{\mathrm{KL}}(\bar{\pi} \parallel (1-\beta)\cdot\bar{\pi}+\beta\cdot\pi') < \delta$.

Now for $i \in \mathbb{N}$ define $\pi'_{k_i} \coloneqq (1-\beta)\cdot\pi_{k_i-1}+\beta\cdot\pi'$. Since $\beta < 1$, by Lemma 3 there exists $I \in \mathbb{N}$ such that for all $i \geq I$, $\bar{D}_{\mathrm{KL}}(\pi_{k_i-1} \parallel \pi'_{k_i}) < \delta$. Hence, due to $J_c(\pi'_{k_i}) = (1-\beta)\cdot J_c(\pi_{k_i-1})+\beta\cdot J_c^*$, the policy π'_{k_i} is feasible for the update and

$$\begin{aligned}
\liminf_{k\to\infty} J_r(\pi_k) &= \lim_{i\to\infty} J_r(\pi_{k_i}) \geq \liminf_{i\to\infty} J_r(\pi'_{k_i})\\
&= (1-\beta)\cdot\lim_{i\to\infty} J_r(\pi_{k_i-1})+\beta\cdot J_r(\pi') = (1-\beta)\cdot J_r(\bar{\pi})+\beta\cdot J_r(\pi')
\end{aligned}$$

This is a contradiction since $J_r(\bar{\pi}) \geq \liminf_{k\to\infty} J_r(\pi_k)$ and $J_r(\pi') > \liminf_{k\to\infty} J_r(\pi_k)$. □

B.2 Supplementary Materials for Section 4.2

Lemma 5. $\mathcal{L}_{r,\pi_k}(\pi) - C_{r,\pi_k}\cdot D_{\mathrm{KL}}^{\max}(\pi_k \parallel \pi) \leq J_r(\pi) \leq \mathcal{L}_{r,\pi_k}(\pi) + C_{r,\pi_k}\cdot D_{\mathrm{KL}}^{\max}(\pi_k \parallel \pi)$.

Proof. By (Schulman et al., 2015, Thm. 1),

$$\mathcal{L}_{r,\pi_k}(\pi) - C_{r,\pi_k} \cdot D_{\mathrm{KL}}^{\max}(\pi_k \parallel \pi) \le J_r(\pi), \qquad C_{r,\pi_k} \coloneqq \frac{4 \cdot \gamma \cdot \max_{s,a} |A_r^{\pi_k}(s,a)|}{(1-\gamma)^2}$$

Consider the reward function $-r$. Note that $J_{-r}(\pi) = -J_r(\pi)$ and $A_{-r}^{\pi_k}(s,a) = -A_r^{\pi_k}(s,a)$. Thus, $C_{-r,\pi_k} = C_{r,\pi_k}$ and $\mathcal{L}_{-r,\pi_k}(\pi) = -\mathcal{L}_{r,\pi_k}(\pi)$. Finally, again by (Schulman et al., 2015, Thm. 1),

$$\begin{aligned} -J_r(\pi) = J_{-r}(\pi) &\ge \mathcal{L}_{-r,\pi_k}(\pi) - C_{-r,\pi_k} \cdot D_{\mathrm{KL}}^{\max}(\pi_k \parallel \pi) \\ &= -\mathcal{L}_{r,\pi_k}(\pi) - C_{r,\pi_k} \cdot D_{\mathrm{KL}}^{\max}(\pi_k \parallel \pi) \end{aligned}$$

□

We recall the Fisher information matrix:

$$F \coloneqq \mathbb{E}_{s,a\sim\pi_{\theta_k}}\left[\nabla_\theta \log \pi_\theta(a|s)\big|_{\theta=\theta_k} \cdot \nabla_\theta \log \pi_\theta(a|s)^\top\big|_{\theta=\theta_k}\right]$$

Lemma 6. *For Δ_r and Δ_c defined in Eqs.* (2) *and* (3)*,*

$$\frac{1}{2} \cdot \Delta^T \cdot F \cdot \Delta \le \delta$$

where $\Delta = (1-\mu)\cdot\Delta_r + \mu \cdot \Delta_c$ for arbitrary $\mu \in [0,1]$

Proof. Since F is positive semi-definite, $\Delta \mapsto \Delta^T \cdot F \cdot \Delta$ is convex and the bound follows since it is satisfied by both Δ_r and Δ_c. □

Lemma 7. *Assume $\langle g_c, \Delta_c\rangle \le -\epsilon_k$ and define $\Delta_\mu \coloneqq (1-\mu)\cdot\Delta_r + \mu\cdot\Delta_c$. Then*

$$\max_\mu \langle g_r, \Delta_\mu\rangle \qquad \textit{s.t.} \quad \langle g_c, \Delta_\mu\rangle \le -\epsilon_k, \quad \mu \in [0,1]$$

is feasible and its optimal solution is μ defined in Eq. (4)*.*

NB for our choice $\epsilon_k \coloneqq -\beta \cdot \langle g_c, \Delta_c\rangle \ge 0$, where $\beta \in (0,1]$, the assumption clearly holds.

Proof. By definition of Δ_r and Δ_c,

$$\langle g_r, \Delta_r\rangle \ge \langle g_r, \Delta_c\rangle \qquad\qquad \langle g_c, \Delta_c\rangle \le \langle g_c, \Delta_r\rangle \tag{5}$$

Therefore, we can equivalently optimise

$$\min_\mu \mu \qquad \text{s.t.} \quad \langle g_c, \Delta_\mu\rangle \le -\epsilon_k, \quad \mu \in [0,1]$$

If $\langle g_c, \Delta_r\rangle = \langle g_c, \Delta_c\rangle$ or $\langle g_c, \Delta_r\rangle < -\epsilon_k$ then the optimal solution is clearly $\mu \coloneqq 0$.

Otherwise, first note that by the premise of this lemma and Eq. (5), the ratio in Eq. (4) is always in $[0,1]$. Besides, it is straightforward to see that $\langle g_c, \Delta_\mu\rangle \le -\epsilon_k$ iff

$$\mu \ge \frac{\epsilon_k + \langle g_c, \Delta_r\rangle}{\langle g_c, \Delta_r\rangle - \langle g_c, \Delta_c\rangle}$$

□

Lemma 8 (Exact solution of (**Update 3**))**.** *Let $g_r, g_c \in \mathbb{R}^d$, $F \in \mathbb{R}^{d\times d}$ and let $\epsilon_k > 0$, $\delta > 0$. We assume the* non-degenerate case*:*

$$g_r \ne 0, \quad g_c \ne 0, \quad g_c \nparallel g_r, \quad F \succ 0.$$

*Then the optimal solution Δ^* of (**Update 3**) is*

$$\Delta^* = \begin{cases} \sqrt{\frac{2\delta}{a}}\, F^{-1} g_r, & \textit{if } \sqrt{\frac{2\delta}{a}}\, b \le -\epsilon_k, \\ \sqrt{\frac{2\delta}{a - 2\lambda^* b + (\lambda^*)^2 c}}\, F^{-1}(g_r - \lambda^* g_c), & \textit{otherwise,} \end{cases}$$

where

$$a \coloneqq \langle g_r, F^{-1} g_r\rangle, \quad b \coloneqq \langle g_r, F^{-1} g_c\rangle, \quad c \coloneqq \langle g_c, F^{-1} g_c\rangle.$$

and $\lambda^* > 0$ is the unique positive solution of the quadratic

$$A\lambda^2 + B\lambda + C = 0, \quad \textit{with} \quad A = 2\delta c^2 - \epsilon_k^2 c, \quad B = -4\delta bc + 2\epsilon_k^2 b, \quad C = 2\delta b^2 - \epsilon_k^2 a.$$

Proof. We form the Lagrangian with multipliers $\lambda \ge 0$ (linear constraint) and $\nu \ge 0$ (quadratic constraint):

$$\mathcal{L}(\Delta, \lambda, \nu) = \langle g_r, \Delta\rangle + \lambda(-\epsilon_k - \langle g_c, \Delta\rangle) + \nu\left(\delta - \tfrac{1}{2}\Delta^\top F \Delta\right).$$

Stationarity gives

$$g_r - \lambda g_c - \nu F \Delta = 0 \quad \Longrightarrow \quad \Delta = \frac{1}{\nu} F^{-1}(g_r - \lambda g_c), \quad \nu > 0.$$

Justification that $\nu > 0$. The objective is linear in Δ and $g_r \neq 0$, so it is unbounded in the direction of g_r without the quadratic constraint. The linear constraint $\langle g_c, \Delta \rangle \leq -\epsilon_k$ defines a half-space. In the non-degenerate case $g_c \nparallel g_r$, this half-space alone cannot bound the linear objective. Therefore, the quadratic constraint must be active at the solution, which by complementary slackness implies $\nu > 0$.

Form of KKT candidates. Substituting Δ into the quadratic constraint $\frac{1}{2}\Delta^\top F \Delta = \delta$ gives

$$\nu = \sqrt{\frac{(g_r - \lambda g_c)^\top F^{-1}(g_r - \lambda g_c)}{2\delta}}.$$

Hence all KKT candidates satisfy

$$\begin{aligned}\Delta(\lambda) &= \sqrt{\frac{2\delta}{(g_r - \lambda g_c)^\top F^{-1}(g_r - \lambda g_c)}} \cdot F^{-1} \cdot (g_r - \lambda g_c) \\ &= \sqrt{\frac{2\delta}{a - 2\lambda b + \lambda^2 c}} \cdot F^{-1} \cdot (g_r - \lambda g_c)\end{aligned}$$

using the above definitions of a, b and c, so that

$$\langle g_c, \Delta(\lambda) \rangle = \sqrt{\frac{2\delta}{a - 2\lambda b + \lambda^2 c}} \cdot (b - \lambda c)$$

Complementary slackness for the linear constraint. Either $\lambda = 0$ (linear constraint inactive) or $\lambda > 0$ with equality $\langle g_c, \Delta(\lambda) \rangle = -\epsilon_k$.

Case $\lambda = 0$. If $\langle g_c, \Delta(0) \rangle = \sqrt{\frac{2\delta}{a}}\, b \leq -\epsilon_k$, then $\Delta^* = \Delta(0)$.

Case $\lambda > 0$. Solving $\langle g_c, \Delta(\lambda) \rangle = -\epsilon_k$ and squaring both sides gives the quadratic equation $A\lambda^2 + B\lambda + C = 0$ displayed in the proposition. The unique positive root λ^* yields the optimal step $\Delta^* = \Delta(\lambda^*)$.

Summary. In the non-degenerate case, the quadratic constraint is active ($\nu > 0$) to ensure a finite optimum. The linear constraint determines λ^*, and Δ^* follows directly. □

B.2.1 Proof of Theorem 2

For $M > 0$, let $B_M(\theta)$ be the M-ball around θ.

Lemma 9. *Assume J_r and J_c are L-Lipschitz smooth on $B_M(\theta)$ and $\|\Delta_r\|, \|\Delta_c\| \leq M$ for some $M > 0$. Then for all $\eta \in [0, 1]$,*

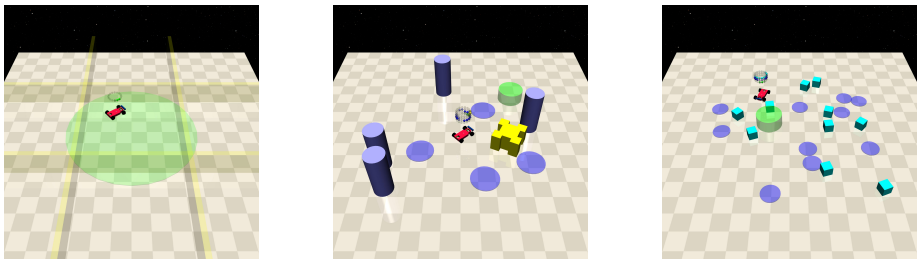
1. $J_c(\theta + \eta \cdot \Delta) \leq J_c(\theta) + \eta \cdot \beta \cdot \langle g_c, \Delta_c \rangle + \frac{L \cdot \eta^2}{2} \cdot \|\Delta\|^2$
2. $J_r(\theta + \eta \cdot \Delta) \geq J_r(\theta) + \eta \cdot (1 - \mu) \cdot \langle g_r, \Delta_r \rangle + \eta \cdot \mu \cdot \langle g_r, \Delta_c \rangle - \frac{L \cdot \eta^2}{2} \cdot \|\Delta\|^2$
3. *if $g_c = 0$ then* $J_r(\theta + \eta \cdot \Delta) \geq J_r(\theta) + \eta \cdot \langle g_r, \Delta_r \rangle - \frac{L \cdot \eta^2}{2} \cdot \|\Delta\|^2$.

Proof. First, note that $g_r = \nabla J_r(\theta)$ and $g_c = \nabla J_c(\theta)$. Therefore, by assumption and Taylor's theorem, for every $\eta \in [0, 1]$,

$$J_c(\theta + \eta \cdot \Delta) \leq J_c(\theta) + \eta \cdot \langle g_c, \Delta \rangle + \frac{L \cdot \eta^2}{2} \cdot \|\Delta\|^2$$

$$J_r(\theta + \eta \cdot \Delta) \geq J_r(\theta) + \eta \cdot \langle g_r, \Delta \rangle - \frac{L \cdot \eta^2}{2} \cdot \|\Delta\|^2$$

since $\|\eta \cdot \Delta\| \leq M$. By the choice of ϵ_k and Lemma 7, $\langle g_c, \Delta \rangle \leq \beta \cdot \langle g_c, \Delta_c \rangle$ and the first claim follows. The second also follows directly by definition of Δ as a convex combination. Finally, if $g_c = 0$ then $\mu = 0$ and the third claim follows from the second. □



(a) Circle (b) Push (c) Goal

Figure 4: Safe navigation tasks of Safety Gymnasium (Ji et al., 2023) (images taken from `https://safety-gymnasium.readthedocs.io/en/latest/environments/safe_navigation.html`)

Theorem 2 (Performance Improvement)**.** *Let* $\Delta := (1-\mu)\cdot\Delta_r + \mu\cdot\Delta_c$ *for* Δ_r, Δ_c *and* μ *defined in Eqs.* (2) *to* (4)*, respectively. There exists* $\eta \in (0, 1]$ *such that*

1. *if* $g_c \neq 0$ *then* $J_c(\theta + \eta\cdot\Delta) < J_c(\theta)$
2. *if* $\beta < 1$, $g_r \neq 0$ *and* $\langle g_r, \Delta_c\rangle \geq 0$ *(in particular,* $g_c = 0$*) then* $J_r(\theta + \eta\cdot\Delta) \geq J_r(\theta)$.

Proof. Since J_r and J_c are smooth, they are L-smooth on the bounded set $B_M(\theta)$ for $M := \max\{\|\Delta_r\|, \|\Delta_c\|\}$ and sufficiently large $L > 0$.

If $g_c \neq 0$ then $\langle g_c, \Delta_c\rangle < 0$ and the claim follows from Lemma 9 for sufficiently small $\eta \in (0, 1]$.

Next, suppose that $g_r \neq 0$ and $\langle g_r, \Delta_c\rangle \geq 0$. Note that $\mu < 1$ if $\beta < 1$. By definition of Δ_r in Eq. (3), $\langle g_r, \Delta_r\rangle > 0$. Besides, by Lemma 9,

$$J_r(\theta + \eta\cdot\Delta) \geq J_r(\theta) + \eta\cdot(1-\mu)\cdot\langle g_r, \Delta_r\rangle - \frac{L\cdot\eta^2}{2}\cdot\|\Delta\|^2$$

and the claim follows for sufficiently small $\eta \in (0, 1]$. □

C Supplementary Materials for Section 5

C.1 Supplementary Materials for Section 5.1

C.1.1 Safety Gymnasium Tasks

We provide additional details on the Safety Gymnasium tasks used in our experiments (see (Ji et al., 2023) and `https://safety-gymnasium.readthedocs.io/` for the full details).

Safety Gymnasium uses reward shaping to provide the agent with auxiliary guidance signals that facilitate more efficient policy optimisation. In the Goal tasks of Safe Navigation environments, for example, the agent receives dense rewards based on its movement relative to the goal, supplementing the sparse task completion reward, so that the cumulative reward can be negative if the agent moves significantly away from the goal. A cost is also computed to quantify constraint violations, with scales that are task dependent. In our selected tasks, these violations include contact with hazards, displacement of hazards, and leaving safe regions in Safe Navigation, and exceeding velocity limits in Safe Velocity.

Safe Navigation Tasks are described in Table 2 and visualised in Fig. 4. Visual depictions and complete descriptions of the Safe Velocity tasks are available at `https://safety-gymnasium.readthedocs.io/en/latest/environments/safe_velocity.html`.

Notes on Agents.

- **Point:** A 2D robot with two actuators for rotation and forward/backward movement.
- **Car:** A 3D robot with two independently driven wheels and a free rear wheel, requiring coordinated steering and movement.

Table 2: Overview of the Safety Gymnasium navigation tasks used in our experiments. The dimensionality refers to the state space for the Point and Car agents, respectively. (Recall that we always use level 2 of the tasks, which is the most difficult.)

Task	Objective	dim.
Circle	Circulate around the center of the circular area (green) while remaining within the boundaries (yellow), with rewards highest along the outermost circumference and increasing with speed.	28/40
Push	Push the box (yellow) to the goal location (green) while avoiding hazards and pillars (both blue).	76/88
Goal	Navigate to the goal (green) while avoiding hazards (blue) and vases (teal).	60/72

Table 3: Hyperparameters for SB-TRPO (Ours) and baselines.

Hyperparameter	**SB-TRPO (Ours)**		**PPO-Lag**	**TRPO-Lag**	**CPO**	**CUP**	**CPPO-PID**	**FOCOPS**	**RCPO**	**PCPO**	**P3O**	**C-TRPO**
	GAE	**MC**										
Discount Factor γ	0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.99
Target KL	0.01	0.01	0.02	0.01	0.01	0.02	0.02	0.02	0.01	0.01	0.02	0.01
GAE λ	0.95	–	0.95	0.95	0.95	0.95	0.95	0.95	0.95	0.95	0.95	0.95
Timesteps per Epoch	20000	20000	20000	20000	20000	20000	20000	20000	20000	20000	20000	20000
Policy Hidden Layers	{64, 64}	{64, 64}	{64, 64}	{64, 64}	{64, 64}	{64, 64}	{64, 64}	{64, 64}	{64, 64}	{64, 64}	{64, 64}	{64, 64}
Policy Batch Size	20000	20000	64	20000	20000	20000	64	64	20000	20000	64	20000
Actor Learning Rate	–	–	0.0003	–	–	0.0003	0.0003	0.0003	–	–	0.0003	–
Actor Optimiser	–	–	ADAM	–	–	ADAM	ADAM	ADAM	–	–	ADAM	–
Actor Update Iterations	–	–	40	–	–	40	40	40	–	–	40	–
Critic Hidden Layers	{64, 64}	{64, 64}	{64, 64}	{64, 64}	{64, 64}	{64, 64}	{64, 64}	{64, 64}	{64, 64}	{64, 64}	{64, 64}	{64, 64}
Critic Batch Size	128	–	64	128	128	64	64	64	128	128	64	256
Critic Learning Rate	0.001	–	0.0003	0.001	0.001	0.0003	0.0003	0.0003	0.001	0.001	0.0003	0.001
Critic Optimiser	ADAM	–	ADAM	ADAM	ADAM	ADAM	ADAM	ADAM	ADAM	ADAM	ADAM	ADAM
Critic Update Iterations	10	–	40	10	10	40	40	40	10	10	40	10
Cost Limit	–	–	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Clip Coefficient	–	–	0.20	–	–	0.20	–	–	–	–	0.20	–
Conjugate Gradient (C.G.) Iterations	50	50	–	15	15	–	–	–	15	15	–	15
C.G. Tikhonov Regularisation Coefficient	0.02	0.02	–	0.1	0.1	–	–	–	0.1	0.1	–	0.1
Update Scaling Steps	100	100	–	15	15	–	–	–	–	200	–	15
Step Fraction	0.80	0.80	–	0.80	0.80	–	–	–	–	0.80	–	0.80
Lagrangian Initial Value	–	–	0.001	0.001	–	0.001	0.001	0.001	–	–	–	–
Lagrangian Learning Rate	–	–	0.035	0.035	–	0.035	–	–	–	–	–	–
Lagrangian Optimiser	–	–	ADAM	ADAM	–	ADAM	–	–	–	–	–	–
CUP, FOCOPS λ	–	–	–	–	–	0.95	–	1.50	–	–	–	–
CUP, FOCOPS ν	–	–	–	–	–	2.00	–	2.00	–	–	–	–
CPPO-PID K_p	–	–	–	–	–	–	0.1	–	–	–	–	–
CPPO-PID K_i	–	–	–	–	–	–	0.01	–	–	–	–	–
CPPO-PID K_d	–	–	–	–	–	–	0.1	–	–	–	–	–
P3O κ	–	–	–	–	–	–	–	–	–	–	10.0	–
C-TRPO β	–	–	–	–	–	–	–	–	–	–	–	1.0
SB-TRPO β	0.75	0.75	–	–	–	–	–	–	–	–	–	–

- The Safe Velocity tasks are based on agents from MuJoCo (Todorov et al., 2012) locomotion tasks, which are extensively documented at `https://gymnasium.farama.org/environments/mujoco/`.

C.1.2 Selection of Hyperparameters

Table 3 lists all hyperparameters. For the baselines, we use the official SafePO defaults (Ji et al., 2023), setting the cost limit to 0 to target almost-sure safety. For SB-TRPO, most hyperparameters are inherited from related methods (TRPO-Lagrangian and CPO), and critic-related settings are kept identical in critic-inclusive runs. In runs without critics we use Monte Carlo returns, i.e. GAE with $\lambda = 1$. The only novel parameter is the safety bias β, for which no established default exists. We therefore performed a light grid search over $[0.6, 0.9]$ as shown in Fig. 9 in Section 5.3 and Fig. 9b in Section C.3.1. Different choices of β trace out an approximate linear reward-cost Pareto frontier, while baselines typically underperform relative to this frontier. For the main results (cf. Table 1), we chose $\beta := 0.75$ *across all tasks*, highlighting the robustness of our approach.

C.1.3 Formal Definition of Safety Metrics

Safety probability is the fraction of episodes completed without any safety violation, i.e. with zero cost; *safe reward* is the average return over all episodes, with returns from episodes with any safety violation set to 0:

$$\text{safety probability} := \frac{1}{N}\sum_{i=1}^{N}\mathbf{1}[C_i = 0] \qquad \text{safe reward} := \frac{1}{N}\sum_{i=1}^{N} R_i \cdot \mathbf{1}[C_i = 0] \tag{6}$$

where R_i and C_i are the total return and cost of episode i.

C.1.4 REPRODUCTION OF RESULTS

All experiments were run with `Python 3.8`. To reproduce our results, first install Safety-Gymnasium as documented at `https://github.com/PKU-Alignment/safety-gymnasium#installation`, and then extract the supplementary code provided with this submission. Inside the extracted folder, install SafePO (Ji et al., 2023) integrated with our approach by running:

```
pip install -e .
```

Experiments can be reproduced in two ways. To train a single baseline (e.g., PPO-Lagrangian) with a specific seed (e.g., 2000) on a given environment (e.g., SafetyPointGoal2-v0), navigate to the `single_agent` directory and run:

```
python ppo_lag.py --task SafetyPointGoal2-v0 --seed 2000 --cost-limit 0
```

Similarly, for our method, SB-TRPO, at a specific safety bias (e.g., $\beta = 0.8$), use:

```
python sb-trpo.py --task SafetyPointGoal2-v0 --seed 2000 --beta 0.8
```

Additional options can be found in `utils/config.py` under `single_agent_args`.

For large-scale benchmarking, multiple algorithms can be launched in parallel with a single command:

```
python benchmark.py
--tasks SafetyHopperVelocity-v1 SafetySwimmerVelocity-v1
--algo ppo_lag trpo_lag cpo cup sb-trpo
--workers 25 --num-envs 20 --steps-per-epoch 20000
--total-steps 20000000 --num-seeds 5
--cost-limit 0 --beta 0.75
```

This command runs all baselines considered in the paper at a cost limit of 0, alongside our method with $\beta = 0.75$, using 25 training processes in parallel with 20 vectorized environments per process and 5 random seeds. Further options are documented in `benchmark.py`.

Finally, our ablations are implemented as `sb-trpo_rcritic` (including a reward critic) and `sb-trpo_critic` (including both reward and cost critics).

C.2 SUPPLEMENTARY MATERIALS FOR SECTION 5.2

The training curves, which are smoothed with averaging over the past 20 epochs, are presented in Fig. 5.

Beyond final performance, our approach also exhibits robustness in practice: temporary increases in cost are typically corrected quickly whilst improving rewards overall (see Fig. 7). Moreover, longer training can continue to improve both reward and feasibility for our method, while the baselines plateau (cf. Figs. 5 and 6 in Section C.2).

Additionally, we train longer on Point Button to confirm further feasibility and reward improvements. This is presented in Fig. 6. As anticipated, the improvements are non-diminishing even for a 125% longer training run.

We compare the compute costs of our approach with the baselines w.r.t. the update times per epoch on a single, non-parallelized training run in Table 4. This demonstrates that training critics is expensive across all critic-integrated approaches.

C.2.1 QUALITATIVE BEHAVIOUR AND VIDEOS

We compare the qualitative behaviours of policies obtained from our approach with the baselines using recorded videos (provided in the supplementary material) across Car Circle, Car Goal, and Swimmer Velocity. The results in Table 1 are strongly corroborated by the behaviours observed in these videos.

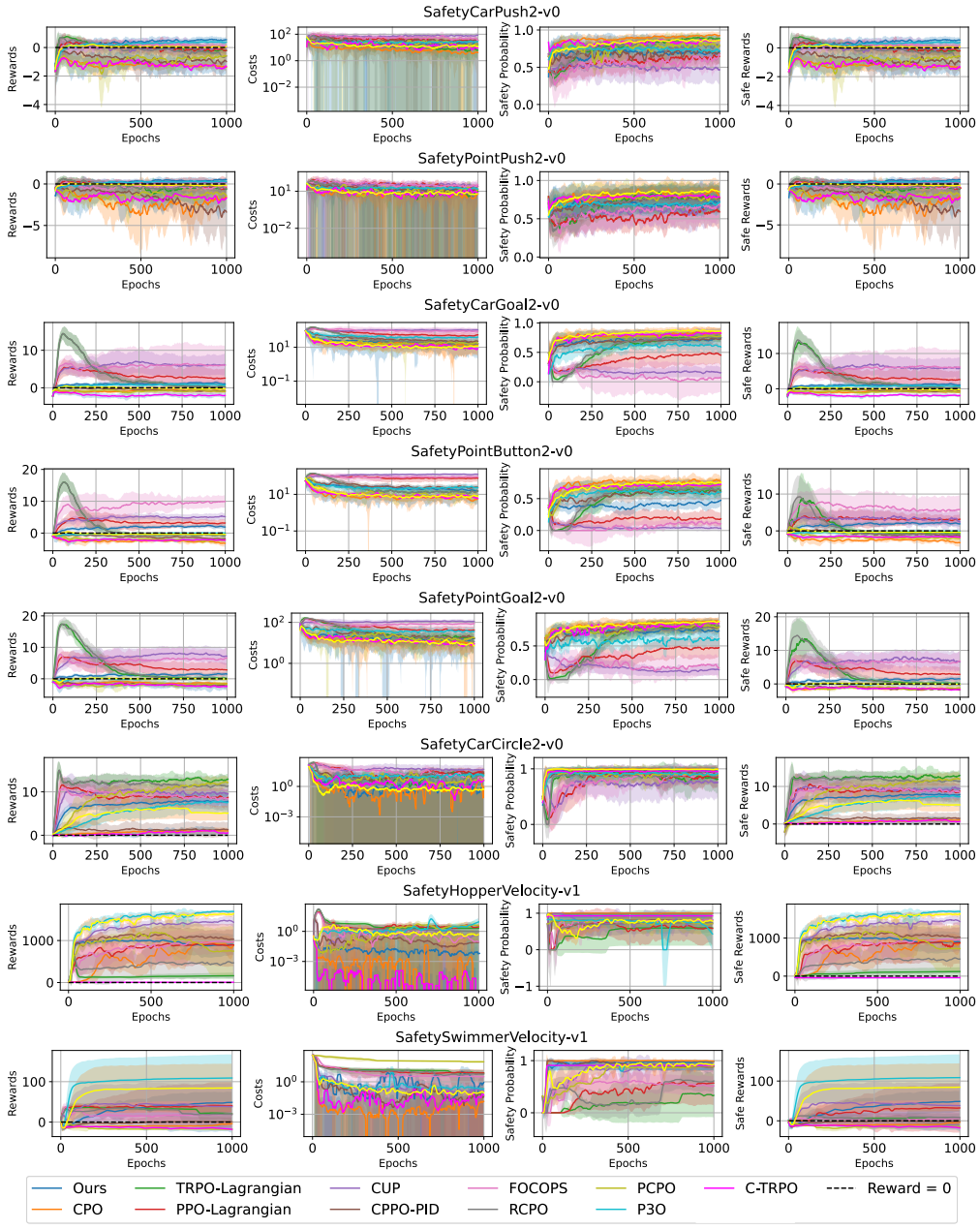


Figure 5: Training curves

Car Circle. CPO and C-TRPO fail to fully associate circling movement with reward acquisition, resulting in back-and-forth movements near the circle's centre and partial rewards. Consequently, these policies rarely violate constraints. P3O, CUP, and PPO-Lag learn to circle effectively but strug-

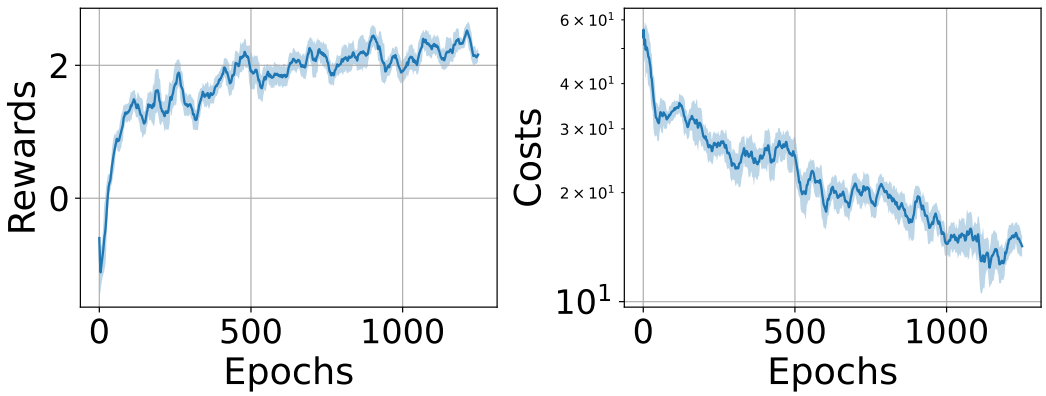


Figure 6: Point Button: training longer, doing better.

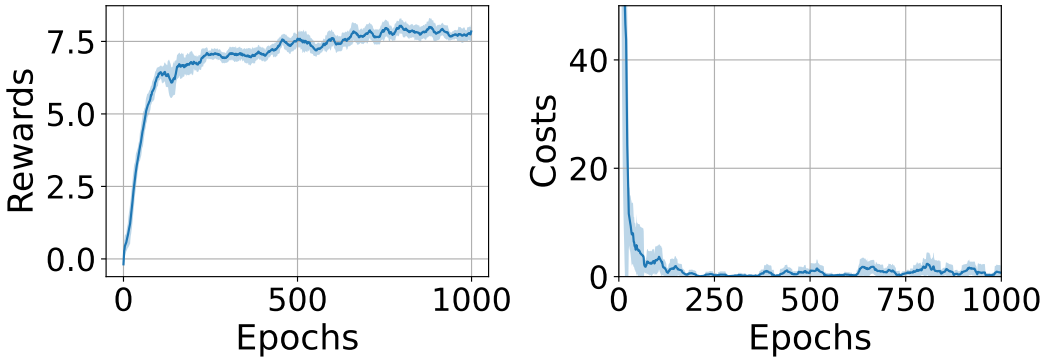


Figure 7: Car Circle: staying safe and improving reward

gle to remain within boundaries, incurring costs. TRPO-Lag and SB-TRPO achieve both effective circling and boundary adherence.

Car Goal. CPO, and C-TRPO display low reward affinity, leading to overconservative behaviours such as avoiding the interactive region. TRPO-Lag is unproductive and does not avoid hazards reliably. CUP and PPO-Lag are highly reward-driven but reckless with hazards. P3O begins to learn safe navigation around the interactive region, though occasional recklessness remains. SB-TRPO achieves productive behaviour while intelligently avoiding hazards, e.g., by slowly steering around obstacles or moving strategically within the interactive region.

Swimmer Velocity. CPO and C-TRPO sometimes move backward or remain unproductively in place. TRPO-Lag and PPO-Lag move forward but engage in unsafe behaviours. CUP and SB-TRPO are productive in moving forward while being less unsafe. P3O is the most effective, moving forward consistently while exhibiting the fewest unsafe behaviours.

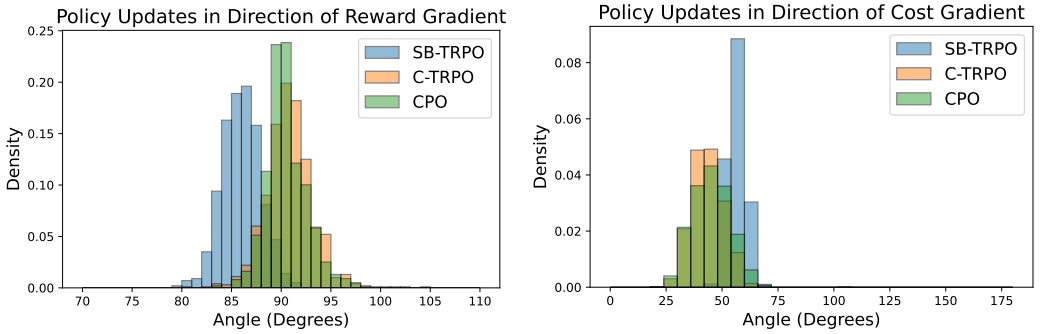
C.2.2 Deviations from Theory in Training Curves

While our theoretical results (Theorem 2) guarantee consistent improvement in both reward and cost for sufficiently small steps, practical training exhibits deviations due to finite sample estimates and the sparsity of the cost signal in some tasks. We summarise task-specific observations:

- In **Swimmer Velocity**, some update steps may not immediately reduce incurred costs due to estimation noise, which explains delays in achieving perfect feasibility.
- In **Car Circle**, policies that previously attained near-zero cost sometimes return temporarily to infeasible behaviour. This arises because the update rule does not explicitly encode "distance" from infeasible regions within the feasible set. Nonetheless, policies eventually return to feasibility thanks to the guarantees of our approach.

Table 4: Update times per epoch on Point Goal on a single, non-parallelized training run, based on the hyperparameters in Table 3. Mean ± STDEV is reported, rounded to 2 decimal places. The least update time consumed per epoch is in **bold**. Note that PPO-Lagrangian, in contrast to TRPO-Lagrangian, has a budget of up to 40 gradient steps per epoch (as per Table 3).

	SB-TRPO (MC)	SB-TRPO (GAE)	TRPO-Lag	CPO	PPO-Lag	CUP	CPPO-PID	FOCOPS	RCPO	PCPO	P3O	C-TRPO
Time↓	**0.26 ± 0.08**	8.12 ± 0.95	5.85 ± 0.79	6.59 ± 1.96	69.83 ± 8.29	84.60 ± 15.54	88.32 ± 18.82	53.54 ± 19.34	4.40 ± 0.35	5.74 ± 1.19	78.05 ± 20.14	3.32 ± 0.16



(a) Point Button (b) Point Button

Figure 8: Angles between policy updates and reward/cost gradients for Point Button

Table 5: Selected baselines with a relaxed cost limit of 15 (using the same highlighting as in Table 1).

Method	Cost limit	Car Circle		Point Push		Point Goal		Swimmer Velocity	
		Safe Rew.	Safe Prob.	Safe Rew.	Safe Prob.	Safe Rew.	Safe Prob.	Safe Rew.	Safe Prob.
Ours	–	7.5 ± 1.5	**0.99 ± 0.01**	0.33 ± 0.18	**0.79 ± 0.08**	**1.6 ± 0.5**	0.68 ± 0.07	**48 ± 26**	**0.98 ± 0.02**
TRPO-Lag	0	**11 ± 4**	0.81 ± 0.26	−0.49 ± 0.45	0.78 ± 0.16	0.09 ± 0.29	**0.79 ± 0.06**	*1.3 ± 5.1*	*0.33 ± 0.42*
TRPO-Lag	15	*6.8 ± 1.5*	*0.44 ± 0.09*	0.29 ± 0.28	0.64 ± 0.12	−0.18 ± 0.53	0.82 ± 0.08	*0.0 ± 0.0*	*0.0 ± 0.0*
CPO	0	0.96 ± 0.76	**0.99 ± 0.03**	−1.2 ± 0.9	0.77 ± 0.19	−1.3 ± 0.4	0.86 ± 0.05	−3.8 ± 9.3	0.99 ± 0.03
CPO	15	8.3 ± 1.6	0.54 ± 0.11	0.24 ± 0.14	0.61 ± 0.07	0.29 ± 0.67	0.69 ± 0.11	*0.0014 ± 0.0098*	*0.001 ± 0.0052*
RCPO	0	10 ± 3	0.9 ± 0.18	−0.48 ± 0.45	0.84 ± 0.08	−0.012 ± 0.366	0.85 ± 0.06	7.0 ± 12.3	0.67 ± 0.25
RCPO	15	9.4 ± 3.6	0.61 ± 0.23	0.43 ± 0.19	0.6 ± 0.1	1.0 ± 0.5	0.71 ± 0.07	*0.0 ± 0.0*	*0.0 ± 0.0*
C-TRPO	0	0.54 ± 0.5	0.97 ± 0.05	−1.4 ± 1.2	0.83 ± 0.09	−2.2 ± 1.1	0.88 ± 0.06	−18 ± 8	0.97 ± 0.03
C-TRPO	15	11 ± 1	0.69 ± 0.08	**0.48 ± 0.3**	0.58 ± 0.08	1.2 ± 0.5	0.66 ± 0.08	*0.05 ± 0.16*	*0.0092 ± 0.0223*

- In **Hopper Velocity**, after quickly achieving high reward, performance shows a slow decline accompanied by gradual improvements in cost. This reflects the prioritisation of feasibility improvement over reward during some updates, without drastically compromising either objective.
- In **navigation tasks**, especially **Goal and Push**, the cost signal is sparse: a cost of 1 occurs only if a hazard is touched. This sparsity makes perfect safety more difficult to achieve, as most updates provide limited feedback about near-misses or almost-unsafe trajectories. Despite this, SB-TRPO steadily reduces unsafe visits and substantially outperforms baselines.

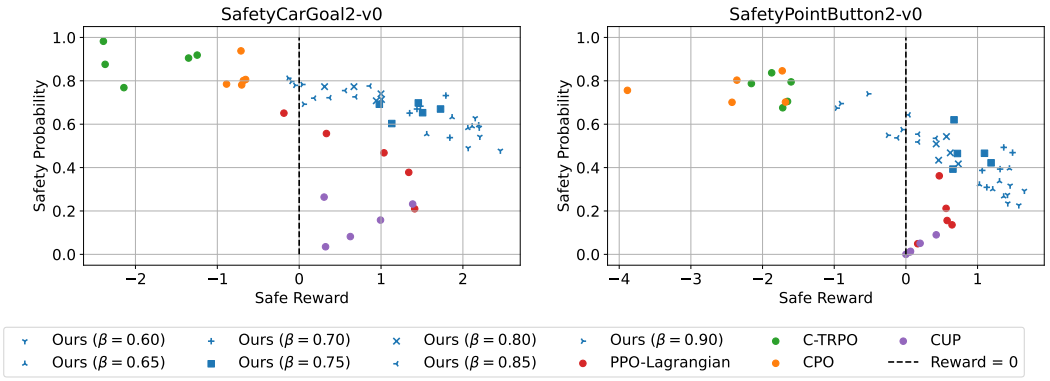
Overall, these deviations are a natural consequence of finite sample estimates, quadratic approximations, and sparse cost signals. In spite of this, SB-TRPO consistently improves reward and safety together, achieving higher returns and lower violations than all considered baselines.

C.2.3 Supplementary Materials for Section 5.2.1

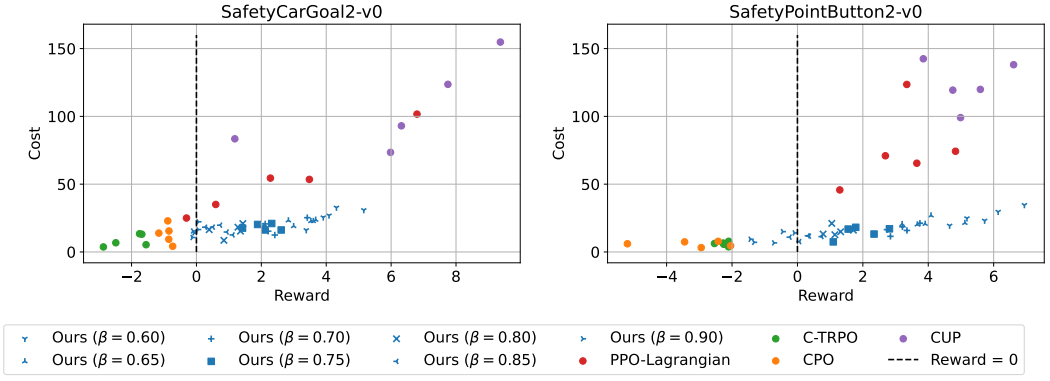
Angles between policy updates and cost gradients are plotted in Fig. 8.

C.2.4 Supplementary Materials for Section 5.2.2

We study the performance of baselines under a positive cost threshold of 15 (see Table 5).



(a) Safe reward and safety probability



(b) Reward and cost

Figure 9: Ablation study of safety bias β

Table 6: Effect of advantage estimation on performance

Advantage	β	Car Goal		Point Goal		Car Circle		Swimmer Velocity	
		Safe Rew.$^{\uparrow}$	Safe Prob.$^{\uparrow}$	Safe Rew.$^{\uparrow}$	Safe Prob.$^{\uparrow}$	Safe Rew.$^{\uparrow}$	Safe Prob.$^{\uparrow}$	Safe Rew.$^{\uparrow}$	Safe Prob.$^{\uparrow}$
MC	0.75	1.4 ± 0.4	0.66 ± 0.08	1.6 ± 0.5	0.68 ± 0.07	7.5 ± 1.5	0.99 ± 0.01	48 ± 26	0.98 ± 0.02
GAE	0.75	5.9 ± 0.9	0.40 ± 0.07	5.0 ± 1.3	0.51 ± 0.09	10.2 ± 2.2	0.96 ± 0.07	0.67 ± 8.1	1.0 ± 0.0
GAE	0.8	2.3 ± 0.35	0.62 ± 0.066	1.7 ± 0.92	0.69 ± 0.098	8 ± 3.1	0.99 ± 0.017	18 ± 35	0.98 ± 0.031

C.3 Supplementary Materials for Section 5.3

C.3.1 Effect of Safety Bias β

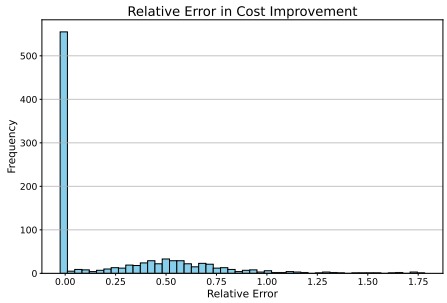
We present scatter plots for our approach compared to selected baselines in Fig. 9. SB-TRPO consistently achieves the best balance of performance and feasibility across all benchmark tasks, demonstrating robustness to the choice of β.

C.3.2 Advantage Estimation

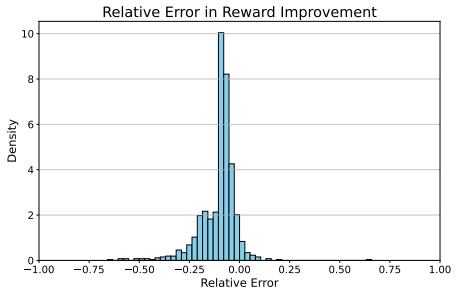
The values of metrics regarding the ablation study about the effect of critics are provided in Tables 4 and 6.

D Approximations

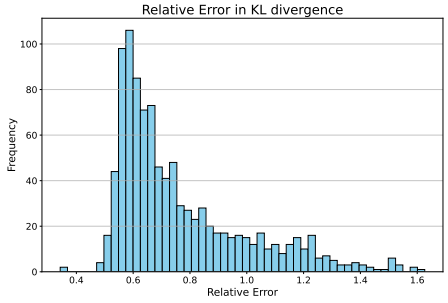
As in most policy optimisation methods, we begin from an idealised (**Update 1**) with theoretical guarantees (Theorem 1) and introduce a sequence of controlled approximations to obtain a tractable algorithm.



(a) Cost



(b) Reward



(c) KL divergence

Figure 10: Histograms of relative approximation errors ($\frac{\widehat{x}-x}{x}$).

The approximations fall into two categories:

- Our innovation: solving (**Update 3**) via a convex combination of reward and cost directions
- Standard trust-region approximations required for tractability

Our Innovation: Convex Combination. Instead of solving (**Update 3**) exactly, we approximate its solution via a convex combination of the reward and cost natural gradient directions (Eqs. (2) and (3)).

We provide the following guarantees:

1. The coefficient μ defined in Eq. (4) ensures feasibility of (**Update 3**), with a linearised decrease in cost of at least ϵ_k (see Lemmas 6 and 7).
2. For sufficiently small step sizes this translates into a decrease in (Theorem 2).
3. Theorem 2 further ensures reward improvement when gradients are not adversarially aligned (angle $\leq 90°$).

Thus, while not necessarily optimal, the update is *provably safe and improving under mild conditions*, which is the key requirement in the constrained setting.

Standard Practical Approximations. The remaining approximations follow established trust-region practices:

- Surrogate Objective: We replace returns with importance-sampling-based surrogates (Update 2). The policy improvement bound, coupled with the KL trust-region constraint, ensures a reliable local approximation.
- Quadratic Approximation: (**Update 2**) is approximated using gradients and the Fisher information matrix for the KL divergence in (**Update 3**).
- Numerical Efficiency: Natural policy gradient directions are computed approximately using Conjugate Gradient.

Line Search and Empirical Safeguards. To bridge the gap between theory and practice, we use backtracking line search (l. 12 in Algorithm 1), enforcing the KL constraint empirically to ensure surrogate accuracy and verifying the empirical reduction in surrogate cost loss.

Gradient Estimation. Finally, we note that gradient estimation is a fundamental challenge in RL. This issue is faced by all policy gradient and trust-region methods, and is not specific to our approach.

Empirical Validation. We empirically evaluate approximation accuracy by comparing predicted (first-/second-order) vs. actual surrogate improvements on the same samples. We include histograms Fig. 10 of relative errors for cost/reward improvement and KL divergence.

We observe the following findings:

- Cost improvement is predicted accurately, supporting the practical validity of Thm. 4.2.
- Reward improvement is under-predicted, indicating a conservative bias.
- KL is over-approximated, resulting in stricter-than-necessary trust regions.

Conclusion. SB-TRPO builds on trusted TRPO/CPO approximations while introducing a convex combination update with performance improvement guarantees (Theorem 2). Empirical validation shows these approximations are conservative, and combined with line search, they ensure robustness even with gradient estimates.