Safe, Trust Region Policy Optimization for Constrained Reinforcement Learning

Md Asifur Rahman
Wake Forest University
Winston-Salem, NC, US
rahmm21@wfu.edu

Risal Shahriar Shefin
Wake Forest University
Winston-Salem, NC, US
shefrs24@wfu.edu

Debashis Gupta
Wake Forest University
Winston-Salem, NC, US
guptd23@wfu.edu

Sarra Alqahtani
Wake Forest University
Winston-Salem, NC, US
sarra-alqahtani@wfu.edu

Abstract

Reinforcement learning (RL) holds promise for sequential decision-making, yet real-world adoption in safety-critical settings remains limited by unsafe exploration during training. Constrained Markov Decision Processes (CMDPs) offer a formalism for safe RL, and several methods provide constraint-satisfaction guarantees, but they often fall short on empirical safety, incurring violations during training or deployment. We introduce sTRPO, which augments the traditional trust-region update with explicit exclusion of overlapping unsafe regions, thereby improving safety. sTRPO learns an auxiliary unsafe policy that estimates high-risk regions of the policy space and explicitly reduces distributional overlap with that policy during trust-region updates. Our key algorithmic novelty is a GAE-driven joint advantage: generalized advantage estimation (GAE) supplies reliable N-step local signals for return and safety, while the region-exclusion step provides global planning, stitching those local decisions into trajectories that avoid unsafe occupancy. This dual-stage optimization yields monotonic improvement in both reward and safety objectives; theoretically, we derive per-iteration bounds on worst-case safety degradation and on constraint satisfaction. Practically, the auxiliary unsafe model can be trained with any RL algorithm in simulation, making sTRPO robust and sim-to-real friendly. Empirically, on Safety-Gymnasium, sTRPO outperforms seven state-of-the-art baselines, achieving significantly fewer constraint violations while maintaining competitive task performance. Together, these results position sTRPO as a scalable, theoretically grounded framework for deploying RL in safety-critical environments.

1 Introduction

Reinforcement learning (RL) has emerged as a powerful paradigm for sequential decision-making, demonstrating remarkable success across diverse fields such as robotics Kober et al. [2013], autonomous driving Grigorescu et al. [2020], and gaming. Despite these advancements, its inherent exploratory nature poses significant safety challenges in real-world applications where constraint violations can lead to catastrophic consequences. The fundamental tension between exploration and safety remains unresolved in RL methods, particularly in high-dimensional continuous control tasks like autonomous vehicles or industrial robotics. Standard RL methods typically prioritize reward maximization, allowing unconstrained exploration that can lead to hazardous scenarios or unsafe

2nd Workshop on Aligning Reinforcement Learning Experimentalists and Theorists (ARLET 2025).

system behaviors. Consequently, a significant gap persists in current RL literature: *How can we design theoretically-grounded trust region policy optimization algorithms that effectively balance reward maximization and strict safety constraint adherence throughout the learning process?* This work directly addresses this critical research question by proposing and rigorously analyzing a novel safe policy optimization method.

The constrained Markov Decision Process (CMDP) Kallenberg [1983], Altman [1999] framework provides a principled mathematical foundation for safe RL by incorporating auxiliary cost functions that encode safety constraints alongside reward objectives. CMDPs seek optimal policies that maximize expected cumulative rewards while strictly adhering to predefined safety constraints, expressed as cumulative costs. Although linear programming solutions Ross and Varadarajan [1989], Beutler and Ross [1985] are feasible for low-dimensional CMDPs with known dynamics, these methods become computationally infeasible in complex, high-dimensional environments, as demonstrated by Achiam et al. [2017]. This limitation has motivated the development of policy search algorithms, which predominantly address constraint satisfaction through direct policy optimization strategies to enforce immediate constraint adherence. Constrained Policy Optimization (CPO) Achiam et al. [2017] provides theoretical safety guarantees but relies on second-order approximations, which can compromise empirical safety in practice. Projection-Based Constrained Policy Optimization (PCPO) Yang et al. [2020] employs trust region updates followed by feasibility projections but exhibits sensitivity to approximation errors, leading to safety violations even after convergence. First-Order Constrained Optimization (FOCOPS) Zhang et al. [2020] simplifies computations but demonstrates inconsistent and unsafe empirical performance across domains. However, these methods do not directly address the underlying issue of estimating and explicitly excluding unsafe regions during policy updates. Consequently, these approaches might inadvertently explore risky regions, resulting in occasional safety violations, especially during intermediate policy iterations as shown in our experiments.

In this paper, we propose Safe Trust Region Policy Optimization (sTRPO); a novel dual-stage optimization algorithm that redefines the integration of safety constraints in policy search by combining local GAE-derived learning signals with global region exclusion. Unlike existing methods that enforce cost constraints via direct penalization or projection-based updates, sTRPO introduces a proactive safety estimation mechanism and a conservative update strategy that operates strictly within a redefined safe trust region. Specifically, our approach first learns an auxiliary unsafe policy that explicitly seeks to maximize constraint violations. We train this unsafe policy using a Conditional Value at Risk CVaR-based critic Yang et al. [2019] to capture tail risk thereby effectively modeling the boundary of unsafe regions in the policy space. This policy is fixed and used to define a global exclusion constraint: updates are repelled from regions that resemble unsafe behavior, yielding a safety-informed trust region for subsequent learning. This approach fits naturally into a simulation-to-reality pipeline, enabling robots or UAVs to generalize safety-aware behavior from simulation before deployment in high-stakes, real-world environments. We then formulate the main policy update as a dual-stage problem that (i) uses a *GAE-derived joint advantage* of reward and safety cost scaled with a PID-controlled Lagrange multiplier to adapt cost sensitivity. This GAE-derived joint advantage function is used to drive the trust-region step, while (ii) enforcing *explicit exclusion* of unsafe regions identified by the auxiliary policy. Intuitively, the GAE joint advantage supplies safe local signals, whereas region exclusion provides global trajectory shaping by reducing distributional overlap with the unsafe policy. As a result, each update occurs only within verified safe neighborhoods, yielding monotonic improvement in both reward and safety under our stated conditions.

Building on the foundation of TRPO, sTRPO introduces three key theoretical advances: (1) it ensures monotonic improvement in both reward and safety (cost) objectives, meaning each policy update either improves or maintains performance without regressions; (2) it provides a worst-case safety guarantee by ensuring that any potential increase in safety violations between policy updates remains tightly controlled; and (3) it guarantees per-step constraint satisfaction by explicitly excluding unsafe regions from the policy search space. This represents a paradigm shift from traditional constrained optimization toward explicit unsafe state space exclusion, yielding stronger safety assurances while remaining computationally efficient. Empirically, we evaluate sTRPO on four continuous control tasks from the Safety-Gymnasium benchmark Ji et al. [2023], comparing it against seven state-of-the-art safe RL baselines. Our results demonstrate that sTRPO achieves 2× fewer constraint violations than the best baseline (CPO), maintains competitive reward performance, and ensures monotonic safety improvement throughout training. It consistently performs robustly across diverse safety-critical scenarios, including navigation and velocity control tasks.

Our contributions are:

1. **Region-exclusion trust region.** We introduce sTRPO, which augments TRPO with a *safety-informed trust region* that explicitly excludes policies near an auxiliary unsafe policy π^{unsafe} trained in simulation to maximize constraint violations(with a CVaR-based Yang et al. [2019] critic to capture tail-risk).
2. **Local–global update rule.** We couple *local* GAE-based (Generalized Advantage Estimation) *joint advantages* with *Z-score normalization* (applied independently to reward and cost advantages) and a PID-controlled Lagrange multiplier for adaptive cost sensitivity with a *global* region-exclusion constraint that reduces distributional overlap with π^{unsafe}, yielding monotonic improvement in both reward and safety under stated conditions.
3. **Theory and empirical validation.** We prove *safe updates* within the safe trust region, *bounds on worst-case reward degradation*, and *monotonic safety improvement*; empirically, sTRPO outperforms 7 baselines across 4 continuous Safety-Gymnasium Ji et al. [2023] tasks, achieving significantly fewer violations while maintaining competitive returns.

2 Related Work

In the realm of policy search for safe RL, Chow et al. [2017] introduced a primal-dual approach that converges to policies adhering to cost constraints. Tessler et al. [2019] employed a reward penalization strategy with a multi-timescale training regimen for an actor-critic algorithm, although this method's strict requirements on learning rates pose practical tuning challenges. However, these methods do not ensure cost constraint satisfaction during the training phase. On the contrary, the CPO algorithm by Achiam et al. [2017] directly embeds constraint satisfaction into the policy optimization process, ensuring compliance during policy updates and serving as a cornerstone for our approach. Pham et al. [2018] introduced a method for dynamically shrinking the trust region in response to potential safety breaches. This method relies on an ongoing assessment of risk and dynamically adjusts policies, demonstrating an adaptive approach to safety in RL. Dalal et al. [2020] uses adversarial training techniques to simulate worst-case scenarios that an RL agent might encounter and uses these scenarios as state-action pairs for the agent to avoid during training.

Advances in control theory have also been applied to address the CMDP problem. Chow et al. [2018, 2019] developed a method for constructing Lyapunov functions that ensure constraint adherence throughout training. Yang et al. [2020] proposed the Projection-Based Constrained Policy Optimization (PCPO), which performs a TRPO update followed by a projection back into the feasible cost constraint space using minimum KL-divergence. While PCPO offers theoretical assurances for constraint satisfaction, it relies on complex second-order calculations. Similarly, Zhang et al. [2020] in FOCOPS uses data generated from the current policy to find the optimal update policy by solving a constrained optimization problem in the nonparameterized policy space. Then, it projects the update policy back into the parametric policy space. FOCOPS only utilizes first order approximations which often proves simpler in practice.

3 Problem Statement

We model safe reinforcement learning as a Constrained Markov Decision Process (CMDP) Altman [2021] $\mathcal{M} = (\mathcal{S}, \mathcal{A}, \mu, \mathcal{P}, r, \gamma, \mathbf{c})$, with state space $\mathcal{S}$, action space $\mathcal{A}$, initial distribution μ, transition kernel $\mathcal{P}$, reward function $r : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$, discount $\gamma \in (0,1)$, and a vector of nonnegative cost functions $\mathbf{c} = (c_1, \dots, c_m)$ encoding safety. For a (stochastic) policy π, the discounted return and costs are defined as

$$J(\pi) = \mathbb{E}_\pi \left[\sum_{t \geq 0} \gamma^t r(s_t, a_t) \right], \qquad J_{c,i}(\pi) = \mathbb{E}_\pi \left[\sum_{t \geq 0} \gamma^t c_i(s_t, a_t) \right],$$

Now with $\mathbf{J}_c(\pi) = (J_{c,1}(\pi), \dots, J_{c,m}(\pi))$. Given thresholds $\mathbf{d} \in \mathbb{R}^m_{\geq 0}$, the CMDP objective is

$$\max_\pi \ J(\pi) \quad \text{s.t.} \quad \mathbf{J}_c(\pi) \ \leq \ \mathbf{d}. \tag{1}$$

To solve (1), we consider a parameterized policy class $\Pi_\theta = \{\pi_\theta : \theta \in \Theta\}$ and perform *local* updates around the current policy π_{θ_k}. Locality is enforced via an average KL trust region Schulman et al. [2015]:

$$\bar{D}_{\mathrm{KL}}(\pi_{\theta_k} \,\|\, \pi_\theta) = \mathbb{E}_{s\sim d_{\pi_{\theta_k}}}\left[D_{\mathrm{KL}}\big(\pi_{\theta_k}(\cdot|s) \,\|\, \pi_\theta(\cdot|s)\big)\right] \;\le\; \delta,$$

where d_π is the discounted state distribution. Using the performance-difference lemma Kakade and Langford [2002], standard surrogate modeling Schulman et al. [2015], and the cost surrogate from Achiam et al. [2017], a *safe local policy search* step can be written as

$$\begin{aligned} \theta_{k+1} \;=\; \arg\max_\theta \;\; & L_{\pi_{\theta_k}}(\theta) \;=\; \mathbb{E}_{\substack{s\sim d_{\pi_{\theta_k}} \\ a\sim\pi_{\theta_k}}}\left[\rho_\theta(a|s)\, A^r_{\pi_{\theta_k}}(s,a)\right] \\ \text{s.t.} \;\; & \underbrace{J_c(\pi_{\theta_k}) \;+\; \tfrac{1}{1-\gamma}\,\mathbb{E}_{\substack{s\sim d_{\pi_{\theta_k}} \\ a\sim\pi_\theta}}\left[A^c_{\pi_{\theta_k}}(s,a)\right] \;\le\; \mathbf{d}}_{\text{surrogate cost constraint Achiam et al. [2017]}}\,, \\ & \bar{D}_{\mathrm{KL}}(\pi_{\theta_k} \,\|\, \pi_\theta) \;\le\; \delta, \end{aligned} \tag{2}$$

where $\rho_\theta(a|s) = \pi_\theta(a|s)/\pi_{\theta_k}(a|s)$, and $A^r_{\pi_{\theta_k}}$, $A^c_{\pi_{\theta_k}}$ are advantage functions for reward and (vector) cost under the baseline π_{θ_k}. Eq. (2) seeks a policy that increases the (linearized) reward objective while remaining both (i) close to π_{θ_k} in KL and (ii) feasible w.r.t. the cost surrogate evaluated from data collected under π_{θ_k}.

However, the surrogate cost constraint can be brittle in practice, because it relies on high-variance estimates of the cost advantage $A^c_{\pi_{\theta_k}}$ and on an off-policy expectation over actions from π_θ, which can be biased when data are limited or noisy. If the critic underestimates risk or the model is misspecified, the constraint can be falsely certified and the update may drift into unsafe regions; if it overestimates, the step becomes overly conservative which may return an unhelpful step.

4 Safe Trust Region Policy Optimization

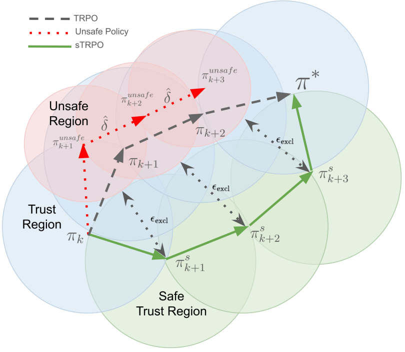


Figure 1: Updates in sTRPO in contrast to conventional trust region update.

We propose sTRPO policy search algorithm for CMDPs that, instead of using the cost constraint in (2), employs a safe trust-region constraint that explicitly excludes unsafe subregions from the policy space (Fig. 1). An auxiliary unsafe policy π^{unsafe} is trained in simulation to maximize constraint violations and identify risky behaviors. This policy is then fixed and used to guide the task policy updates by excluding areas in the trust region that resemble unsafe behavior. The task policy is updated conservatively via a dual optimization process, constrained to remain within both a standard trust region and a safety-informed trust region.

Replacing the traditional cost constraint with a trust-region exclusion constraint yields a clear advantage under critic or model misspecification. When the cost advantage is underestimated or the cost model is biased, updates can drift into unsafe regions; sTRPO bypasses this by enforcing $\bar{D}_{\mathrm{KL}}(\pi^{unsafe} \,\|\, \pi_\theta) > \epsilon_{\text{excl}}$; the policy stays away from behaviors already identified as unsafe and each step corrects prior unsafe tendencies without introducing new risks. Geometrically, sTRPO optimizes in an *annulus*, a safe ring in parameter space that stays close to the incumbent policy via

$\bar{D}_{\mathrm{KL}}(\pi_{\theta_k} \,\|\, \pi_\theta) \leq \delta$ while remaining outside the "unsafe ball" denoting π^{unsafe} (Fig. 1). Operating in this ring stabilizes exploration on the safe manifold rather than chasing a noisy cost boundary. sTRPO thus enables safe, real-world training by leveraging simulation to identify and avoid unsafe policies while preserving TRPO's theoretical properties such as monotonic improvement.

4.1 Unsafe Policy Training for Region Exclusion

To identify unsafe regions in the CMDP, we pre-train an auxiliary policy π_ϕ^{unsafe} with parameters ϕ, optimized to induce maximal cost. The resulting policy approximates the boundary of constraint-violating behaviors which is then excluded during sTRPO updates.

The optimization objective for π_ϕ^{unsafe} follows the standard TRPO framework:

$$\phi_{k+1} = \arg\max_{\phi} \hat{\mathbb{E}}_t \left[\frac{\pi_\phi^{\text{unsafe}}(a_t|s_t)}{\pi_{\phi_{\text{old}}}^{\text{unsafe}}(a_t|s_t)} A_t^c \right] \quad \text{s.t.} \quad \bar{D}_{\mathrm{KL}}(\phi_{\text{old}}, \phi) \leq \delta \tag{3}$$

Here, A_t^c denotes the cost advantage estimator, δ is the trust region constraint radius and $\bar{D}_{\mathrm{KL}}(\phi_{\text{old}}, \phi) = \hat{\mathbb{E}}_t \left[D_{\mathrm{KL}}(\pi_{\phi_{\text{old}}}^{\text{unsafe}}(\cdot|s_t) \,\|\, \pi_\phi^{\text{unsafe}}(\cdot|s_t)) \right]$ represents the average KL divergence between policy updates.

To better capture tail-risk behavior, our approach explicitly models rare but severe safety violations by replacing the standard value estimate in advantage computation with a CVaR-based estimator [citation]. Let $V_{\mathrm{CVaR}_\alpha}^\pi(s)$ denote the CVaR_α of the return distribution; the corresponding TD residual is:

$$\xi_t = r_t + \gamma V_{\mathrm{CVaR}_\alpha}^\pi(s_{t+1}) - V_{\mathrm{CVaR}_\alpha}^\pi(s_t) \tag{4}$$

This modification ensures that the unsafe policy learns extreme or uncommon unsafe scenarios rather than only considering average-case outcomes. The learned π_ϕ^{unsafe} remains fixed throughout sTRPO training (next section) and defines a parametric boundary for the safe trust region constraint.

4.2 Optimization in the Safe Trust Region

In local policy search, policies are updated from sampled rollouts. We therefore restrict attention to a parameterized family $\Pi_\theta = \{\pi_\theta : \theta \in \Theta\} \subset \Pi$ and exclude policies identified as unsafe by the auxiliary cost–maximizing policy π_ϕ^{unsafe} (Sec. 3). Let

$$\mathcal{F}_k \;=\; \Big\{ \theta \;:\; \bar{D}_{\mathrm{KL}}(\pi_{\theta_k} \,\|\, \pi_\theta) \leq \delta \;\text{ and }\; \bar{D}_{\mathrm{KL}}(\pi_\phi^{\text{unsafe}} \,\|\, \pi_\theta) \;>\; \epsilon_{\text{excl}} \Big\}$$

denote the *safe trust region* at iterate k, which enforces both standard TRPO conservatism around π_{θ_k} and *global region exclusion* away from neighborhoods that resemble π_ϕ^{unsafe}. We then update the task policy by maximizing a TRPO surrogate within $\mathcal{F}_k$:

$$\theta_{k+1} \;=\; \arg\max_{\theta \in \mathcal{F}_k} \;\hat{\mathbb{E}}_t \left[\frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_k}(a_t|s_t)} \,\tilde{A}_t^{\text{joint}} \right]. \tag{5}$$

Local joint advantage (GAE, normalization, PID weighting). To provide stable, per-timestep (*local*) learning signals with short-horizon lookahead, we compute *separate* GAEs for reward and cost using distinct critics (and possibly distinct λ's): $\hat{A}_t^r$ and $\hat{A}_t^c$. We Z-score normalize each stream over the batch,

$$\tilde{A}_t^r = \frac{\hat{A}_t^r - \mu_r}{\sigma_r + \varepsilon}, \qquad \tilde{A}_t^c = \frac{\hat{A}_t^c - \mu_c}{\sigma_c + \varepsilon},$$

and combine them into a *GAE-derived joint advantage*

$$\tilde{A}_t^{\text{joint}} \;=\; \tilde{A}_t^r \;-\; \lambda_t^{\text{cost}} \,\tilde{A}_t^c \tag{6}$$

where λ_t^{cost} is a PID-controlled Lagrange multiplier that adapts cost sensitivity online based on deviation from a target cost budget. Intuitively, the reward/cost GAEs supply N-step local signals; Z-score stabilization prevents scale mismatch; and the PID control increases the penalty when constraint violations persist and relaxes it when safety improves.

Performance relation. Following the performance difference lemma Kakade and Langford [2002], the change in the Lagrangian objective between π_θ and $\pi_{\theta'}$ can be expressed (up to standard approximation terms) in expectation over the discounted state distribution $d_{\pi_{\theta'}}$ under the updated policy:

$$J(\pi_{\theta'}) - J(\pi_\theta) \;\approx\; \frac{1}{1-\gamma}\, \mathbb{E}_{\substack{s\sim d_{\pi_{\theta'}} \\ a\sim\pi_{\theta'}}} \big[\, \tilde{A}^{\text{joint}}_{\pi_\theta}(s,a) \,\big],$$

which makes explicit that the joint advantage provides the step-wise signal guiding the trust-region update in (5), while the exclusion constraint shapes the feasible set globally. This local–global coupling yields conservative improvements that correct unsafe tendencies without introducing new risks.

4.3 Policy Update via Surrogate Approximations

Directly solving the policy update in (5) poses practical challenges, as evaluating constraint satisfaction in high-dimensional spaces via sampling can be costly and error-prone Duan et al. [2016], Jiang and Li [2016]. Following Schulman et al. [2015], Achiam et al. [2017], we adopt a trust-region-based surrogate approximation by linearizing the objective and constraints around the current iterate θ_k. Let:

$$\begin{aligned}
g &= \mathbb{E}_{(s,a)\sim\pi_{\theta_k}} \Big[\nabla_\theta \log \pi_\theta(a|s)\; \tilde{A}^{\text{joint}}_{\pi_{\theta_k}}(s,a) \Big] && \text{(gradient of the joint-advantage surrogate)} \\
H &= \nabla^2_\theta\, \bar{D}_{\text{KL}}(\pi_{\theta_k} \,\|\, \pi_\theta)\big|_{\theta=\theta_k} && \text{(Fisher Information at } \theta_k\text{)} \\
\varphi &= \mathbb{E}_{(s,a)\sim\pi_{\theta_k}} \left[-\,\frac{\pi^{\text{unsafe}}_\phi(a|s)}{\pi_{\theta_k}(a|s)}\; \nabla_\theta \log \pi_\theta(a|s) \right] && \text{(approx. gradient of the exclusion constraint).}
\end{aligned} \tag{7}$$

Here, $\tilde{A}^{\text{joint}}_{\pi_{\theta_k}}$ is the GAE-derived joint advantage (6) estimated from rollouts under π_{θ_k}. The vector φ importance-weights actions from π^{unsafe}_ϕ using samples from π_{θ_k} to approximate $\nabla_\theta D_{\text{KL}}(\pi^{\text{unsafe}}_\phi \,\|\, \pi_\theta)$.

Using the linear–quadratic approximation, the update solves the quadratically constrained linear program as:

$$\begin{aligned}
\theta_{k+1} \;=\; \arg\max_{\theta} \quad & g^\top(\theta-\theta_k) \\
\text{s.t.} \quad & \tfrac{1}{2}(\theta-\theta_k)^\top H(\theta-\theta_k) \;\le\; \delta, \\
& \varphi^\top(\theta-\theta_k) \;\ge\; \delta,
\end{aligned} \tag{8}$$

where the first constraint enforces the TRPO trust region and the second increases the KL distance from π^{unsafe}_ϕ by at least δ. Note that the threshold δ here is used in both constraints for convenience but in practice different thresholds can be used as we show in our experiments. Now, let $s = \theta - \theta_k$. The Lagrangian dual has stationarity

$$g - \lambda_1 H \cdot s + \lambda_2 \varphi = 0 \;\Rightarrow\; s = \frac{1}{\lambda_1}\, H^{-1}(g + \lambda_2\varphi).$$

In practice, we set the unscaled search direction as

$$\mathcal{G} \;=\; H^{-1}(g + \lambda_2\varphi),$$

and choose the step size to satisfy the KL constraint:

$$\alpha \;=\; \sqrt{\frac{2\delta}{\mathcal{G}^\top H \mathcal{G}}}, \qquad \theta_{k+1} \;=\; \theta_k + \alpha\, \mathcal{G}. \tag{9}$$

(Equivalently, if one keeps λ_1, the Karush–Kuhn–Tucker (KKT) conditions Boyd and Vandenberghe [2004] for the convex QCQP in Eq. (8), the stationarity condition yields $\lambda_1^* = \sqrt{\frac{(g+\lambda_2\varphi)^\top H^{-1}(g+\lambda_2\varphi)}{2\delta}}$.)

We enforce feasibility via backtracking line search on α and (optionally) adjust $\lambda_2 \ge 0$ (e.g., dual ascent or line search) until the exclusion constraint $\varphi^\top(\theta_{k+1} - \theta_k) \ge \delta$ also holds. When Eq. 8 is infeasible due to estimation noise, we apply a conservative fallback (reverse step) followed by line search:

$$\theta_{k+1} \;=\; \theta_k - \alpha\, \mathcal{G}. \tag{10}$$

This procedure preserves the TRPO KL bound while maintaining separation from unsafe regions.

4.4 Theoretical Guarantees of sTRPO

We present three theoretical results that establish the safety and performance guarantees of the proposed sTRPO algorithm. These results collectively demonstrate that: (1) each policy update lies within a safe trust region, (2) the worst-case degradation in expected return is tightly bounded, and (3) the expected cost under the updated policy cannot increase significantly, thereby ensuring monotonic safety improvement.

Theorem 1: Safe Policy Update. Let π_{θ_k} be the current policy and π_ϕ^{unsafe} the fixed unsafe policy. Then, the sTRPO update

$$\theta_{k+1} = \theta_k + \alpha\mathcal{G}, \quad \text{where} \quad \alpha = \sqrt{\frac{2\delta}{\mathcal{G}^T H \mathcal{G}}}, \quad \mathcal{G} = \frac{1}{\lambda_1^*} H^{-1}(g + \lambda_2^* \varphi),$$

satisfies the following constraints: (1) Trust region constraint: $\frac{1}{2}(\theta_{k+1} - \theta_k)^T H(\theta_{k+1} - \theta_k) \leq \delta$, and (2) Safety exclusion constraint $\varphi^T(\theta_{k+1} - \theta_k) > \delta$.

Proof: Please refer to the supplemental material (Appendix A).

Theorem 2: Worst-Case Performance Bound for Constraint-Satisfying Policies. If the current policy π_{θ_k} satisfies the safety constraints, then the performance improvement from a single sTRPO update is lower bounded by:

$$J(\pi_{\theta_{k+1}}) - J(\pi_{\theta_k}) \geq -\frac{\sqrt{2\delta}\,\gamma\,\epsilon^o_{\pi_{k+1}}}{(1-\gamma)^2} \tag{11}$$

where $\epsilon^o_{\pi_{k+1}} = \max_s \left|\mathbb{E}_{a\sim\pi_{k+1}}\left[A_{\pi_k}(s,a)\right]\right|$ denotes the worst-case advantage estimation error under the updated policy, and δ is the step size imposed by the KL constraint.

Proof: Please refer to the supplemental material (Appendix A).

Theorem 3: Monotonic Safety Guarantee. Let the current policy π_{θ_k} satisfy the safety constraint $J_c(\pi_{\theta_k}) \leq d$. Suppose the updated policy $\pi_{\theta_{k+1}}$ satisfies the trust region constraint

$$\bar{D}_{\text{KL}}(\pi_{\theta_k} \,\|\, \pi_{\theta_{k+1}}) \leq \delta \quad \text{and} \quad \bar{D}_{\text{KL}}(\pi_\phi^{\text{unsafe}} \,\|\, \pi_{\theta_{k+1}}) > \delta.$$

Then, the expected cost under $\pi_{\theta_{k+1}}$ is bounded by:

$$J_c(\pi_{\theta_{k+1}}) \leq J_c(\pi_{\theta_k}) + \epsilon_c(\delta),$$

where $\epsilon_c(\delta) = \frac{2\gamma\epsilon^c_{\pi_{k+1}}}{(1-\gamma)^2}\sqrt{2\delta}$, and $\epsilon^c_{\pi_{k+1}} = \max_s \left|\mathbb{E}_{a\sim\pi_{\theta_{k+1}}}\left[\mathbb{A}^c_{\pi_{\theta_{k+1}}}(s,a)\right]\right|$ denotes the worst-case cost advantage under $\pi_{\theta_{k+1}}$.

Proof. Please refer to the supplemental material (Appendix A).

Finally, we justify that replacing the standard advantage with our GAE-derived joint advantage does not alter the TRPO bound, up to an explicit estimation term; see Lemma A.2 and proof in Appendix A.

5 Experiment

We conducted experiments to answer the following: (1) Does sTRPO succeed at enforcing safety constraints in continuous, high dimensional environments?, and (2) How does sTRPO compare with baselines in terms of safety constraint.

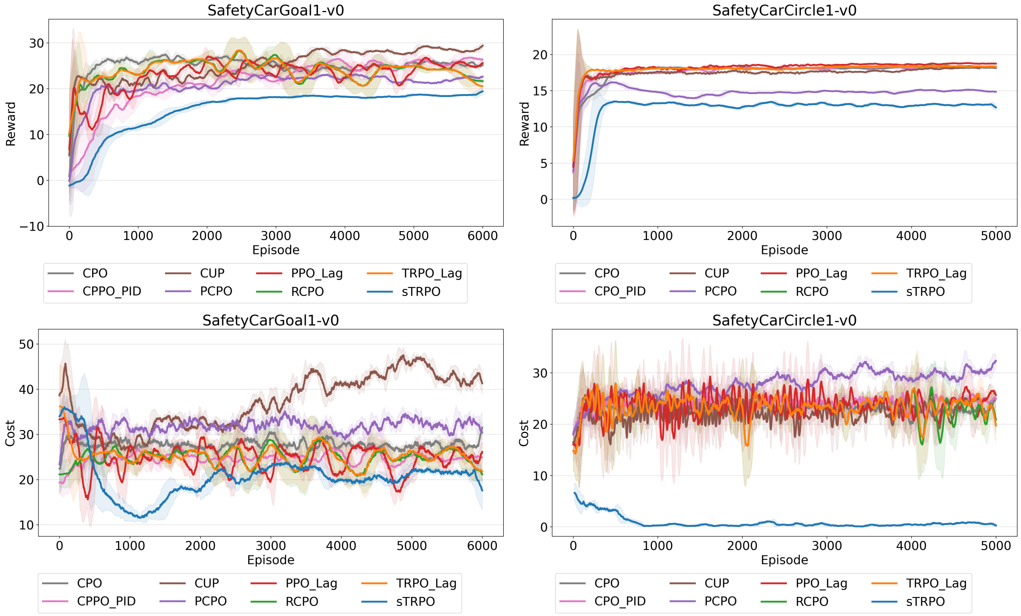
5.1 Experimental Setup: Tasks and Baselines

Tasks (Safety-Gymnasium). We evaluate on four standard tasks from Safety-Gymnasium Ji et al. [2023]: (i) *Goal*—navigate to sequential goal locations while avoiding hazards (goal resets after success); (ii) *Circle*—maximize reward by following a green track while red regions incur safety violations; (iii) *Velocity*—track target forward velocity under safety constraints (robots: Ant, Hopper). Full details and environment visuals are in Appendix B.

Baselines. We compare against seven safe RL baselines: (1) *TRPO_lag*—TRPO with a Lagrangian penalty on costs Ray et al. [2023]; (2) *CPO*—trust-region updates with a cost surrogate and feasibility guarantees Achiam et al. [2017]; (3) *PPO_lag*—PPO with a Lagrangian penalty on constraint costs Ray et al. [2023]; (4) *CPPO_PID*—constrained PPO with a PID-controlled Lagrange multiplier for adaptive cost weighting Stooke et al. [2020]; (5) *RCPO*—reward shaping with a cost penalty term Tessler et al. [2019]; (6) *PCPO*—reward-improving trust-region step followed by a projection onto the constraint set Yang et al. [2020]; (7) *CUP*—update under a KL trust region and subsequent projection to ensure cost-feasibility Yang et al. [2022]. (Algorithm-specific objectives and update equations are deferred to the appendix to conserve space.)

5.2 Evaluation Protocol and Results

Protocol. Each epoch collects on-policy trajectories until a 15,000-step cap. For every episode we log (and average across episodes): *Mean Reward* (task return) and *Mean Safety Cost* (constraint violations). Curves report mean ± one standard deviation across seeds.

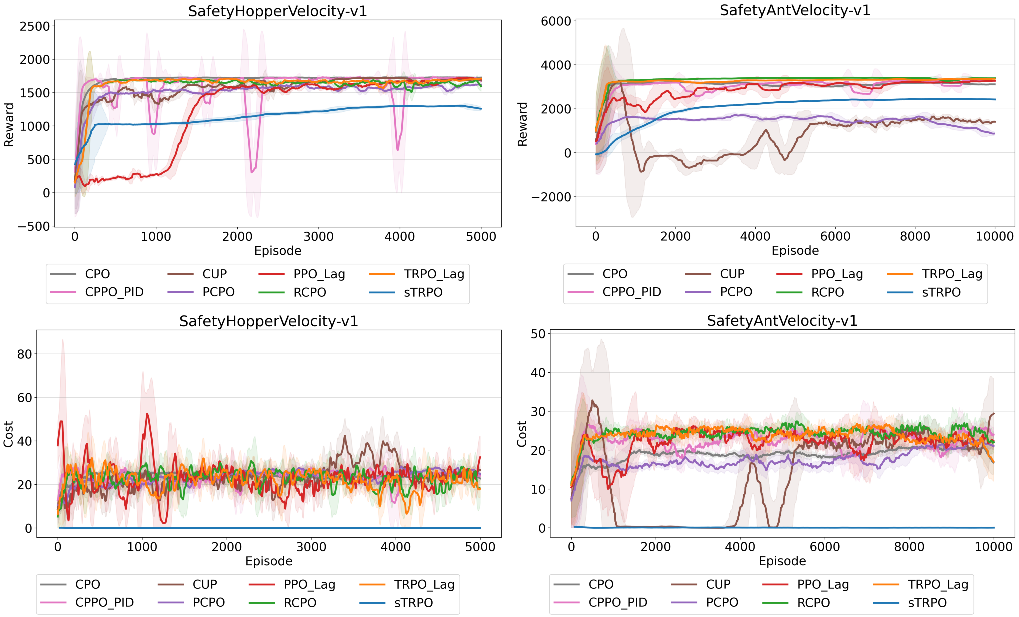


(a) Car–Goal

(b) Car–Circle

Figure 2: **Navigation tasks (Car).** sTRPO consistently attains lower safety cost than all baselines. In *Circle*, it drives cost to ≈ 0 while maintaining moderate reward; in *Goal*, it halves cost relative to the next best method while keeping reasonable returns.

Findings and interpretation. Across all tasks, sTRPO achieves the lowest safety cost while maintaining competitive returns (Figs. 2–3). In *Circle*, *Hopper–Velocity*, and *Ant–Velocity*, the mean episode cost is driven to (and sustained near) zero, whereas baselines stabilize between 15–30. In *Goal*, sTRPO attains rewards comparable to the strongest reward-focused methods yet reduces cost to roughly half of the next best baseline. These outcomes align with the algorithm's design: the global safety-informed region exclusion constrains updates to avoid neighborhoods resembling unsafe behavior, while the local GAE-derived joint advantage provides short-horizon credit assignment that elevates the cost signal whenever violations increase. The TRPO KL line search further stabilizes updates, yielding smoother learning curves and narrower uncertainty bands across seeds. We occasionally observe a brief early rise in cost before convergence; this transient is consistent with the controller ramping the cost weight and the policy relocating away from unsafe regions. Overall, the simultaneous KL and exclusion constraints produce conservative, geometry-aware steps that prioritize monotone safety improvement; the trade-off is slightly slower reward convergence relative to aggressive baselines, offset by markedly lower (often near-zero) violations throughout training.



(a) Hopper–Velocity

(b) Ant–Velocity

Figure 3: **Velocity control.** sTRPO keeps mean cost at (or near) 0 across training for Hopper and Ant, whereas baselines stabilize between 15–30. Rewards remain stable and competitive given the strict safety regime.

Ablation Studies and Robustness Test. We conducted four ablation studies to isolate the impact of key design choices in sTRPO: (i) sweeping the KL trust-region threshold(s) δ (and δ_1/δ_2) to quantify the stability–safety trade-off; (ii) changing the training algorithm for the reference unsafe policy (TRPO vs. distributional TRPO) to assess how its structure shapes the exclusion signal; (iii) varying the exclusion weight λ_2 to map reward–cost Pareto shifts; and (iv) disabling either the GAE-based joint advantage or the unsafe-objective term to measure their individual contributions. Full protocols and results are provided in Appendix C. In addition, we performed a robustness sweep under random-action attacks with per-step rates $\alpha \in \{0.0, 0.1, \ldots, 1.0\}$, measuring reward and cost as attack intensity increases (see Appendix D).

6 Conclusion

We presented sTRPO, a novel safe reinforcement learning algorithm that introduces a safety region exclusion mechanism within trust region optimization. By leveraging an auxiliary unsafe policy to estimate high-risk regions, sTRPO ensures that policy updates occur strictly within safe neighborhoods. This yields monotonic improvement in both reward and safety, with theoretical guarantees and robust empirical performance. Our extensive experiments show that sTRPO outperforms prior safe RL methods in constraint satisfaction without sacrificing task performance. This work bridges theory and practice, advancing RL's reliability in safety-critical domains.

References

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

Eitan Altman. *Constrained Markov Decision Processes*. Stochastic Modeling Series. Chapman & Hall/CRC, Boca Raton, FL, 1999.

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

Frederick J. Beutler and Keith W. Ross. Optimal policies for controlled markov chains with a constraint. *Journal of Mathematical Analysis and Applications*, 112(1):236–252, 1985.

Stephen Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.

Yinlam Chow, Mohammad Ghavamzadeh, Lucas Janson, and Marco Pavone. Risk-constrained reinforcement learning with percentile risk criteria. *J. Mach. Learn. Res.*, 18(1):6070–6120, jan 2017. ISSN 1532-4435.

Yinlam Chow, Ofir Nachum, Edgar Duenez-Guzman, and Mohammad Ghavamzadeh. A lyapunov-based approach to safe reinforcement learning. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, NIPS'18, page 8103–8112, Red Hook, NY, USA, 2018. Curran Associates Inc.

Yinlam Chow, Ofir Nachum, Aleksandra Faust, Edgar Duenez-Guzman, and Mohammad Ghavamzadeh. Lyapunov-based safe policy optimization for continuous control, 2019.

Gal Dalal, Balazs Szorenyi, and Gugan Thoppe. A tale of two-timescale reinforcement learning with the tightest finite-time bound. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(04):3701–3708, Apr. 2020. doi: 10.1609/aaai.v34i04.5779. URL `https://ojs.aaai.org/index.php/AAAI/article/view/5779`.

Yan Duan, Xi Chen, Rein Houthooft, John Schulman, and Pieter Abbeel. Benchmarking deep reinforcement learning for continuous control. In Maria Florina Balcan and Kilian Q. Weinberger, editors, *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pages 1329–1338, New York, New York, USA, 20–22 Jun 2016. PMLR. URL `https://proceedings.mlr.press/v48/duan16.html`.

Sorin Grigorescu, Bogdan Trasnea, Tiberiu Cocias, and Gigel Macesanu. A survey of deep learning techniques for autonomous driving. *Journal of Field Robotics*, 37(3):362–386, 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. *Advances in Neural Information Processing Systems*, 36, 2023.

Nan Jiang and Lihong Li. Doubly robust off-policy value evaluation for reinforcement learning. In Maria Florina Balcan and Kilian Q. Weinberger, editors, *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pages 652–661, New York, New York, USA, 20–22 Jun 2016. PMLR. URL `https://proceedings.mlr.press/v48/jiang16.html`.

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

L. C. M. Kallenberg. *Linear Programming and Finite Markovian Control Problems*. Number 148 in Mathematical Centre Tracts. Mathematisch Centrum, Amsterdam, 1983.

Jens Kober, J Andrew Bagnell, and Jan Peters. Reinforcement learning in robotics: A survey. *The International Journal of Robotics Research*, 32(11):1238–1274, 2013.

Tu-Hoa Pham, Giovanni De Magistris, and Ryuki Tachibana. Optlayer - practical constrained optimization for deep reinforcement learning in the real world. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6236–6243, 2018. doi: 10.1109/ICRA.2018.8460547.

Alex Ray, Joshua Achiam, and Dario Amodei. Benchmarking safe exploration in deep reinforcement learning. *Preprint. Under review*, 2023.

Keith W. Ross and Rajeev Varadarajan. Markov decision processes with sample path constraints. *Operations Research*, 37(5):780–790, 1989.

John Schulman, Sergey Levine, Pieter Abbeel, Michael Jordan, and Philipp Moritz. Trust region policy optimization. In *International conference on machine learning*, pages 1889–1897. PMLR, 2015.

Adam Stooke, Joshua Achiam, and Pieter Abbeel. Responsive safety in reinforcement learning by pid lagrangian methods. In *International Conference on Machine Learning*, pages 9133–9143. PMLR, 2020.

Chen Tessler, Daniel J. Mankowitz, and Shie Mannor. Reward constrained policy optimization. In *International Conference on Learning Representations*, 2019. URL `https://openreview.net/forum?id=SkfrvsA9FX`.

Derek Yang, Li Zhao, Zichuan Lin, Tao Qin, Jiang Bian, and Tieyan Liu. *Fully parameterized quantile function for distributional reinforcement learning*. Curran Associates Inc., Red Hook, NY, USA, 2019.

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. *Advances in Neural Information Processing Systems*, 35:9111–9124, 2022.

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

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

Supplementary Materials

Supplementary Materials

A Theoretical Proofs

A.1 Proof of Theorem 1: Policy Update in Safe Trust Region

Proof. Recall the surrogate-based update direction for sTRPO:

$$\mathcal{G} = \frac{1}{\lambda_1^*} H^{-1}(g + \lambda_2^* \varphi)$$

and the step-size:

$$\alpha = \sqrt{\frac{2\delta}{\mathcal{G}^T H \mathcal{G}}}$$

The updated policy parameter is:

$$\theta_{k+1} = \theta_k + \alpha \mathcal{G}$$

We now verify that this update satisfies both constraints.

(i) Trust Region Constraint. We check:

$$\frac{1}{2}(\theta_{k+1} - \theta_k)^T H (\theta_{k+1} - \theta_k) \leq \delta$$

Substitute $\theta_{k+1} = \theta_k + \alpha \mathcal{G}$:

$$\frac{1}{2}(\alpha \mathcal{G})^T H (\alpha \mathcal{G}) = \frac{1}{2}\alpha^2 \mathcal{G}^T H \mathcal{G}$$

Plug in $\alpha = \sqrt{\frac{2\delta}{\mathcal{G}^T H \mathcal{G}}}$:

$$\frac{1}{2} \cdot \frac{2\delta}{\mathcal{G}^T H \mathcal{G}} \cdot \mathcal{G}^T H \mathcal{G} = \delta$$

Thus, the trust region constraint is satisfied exactly.

(ii) Safety Exclusion Constraint. We verify:

$$\varphi^T(\theta_{k+1} - \theta_k) > \delta$$

Substitute $\theta_{k+1} - \theta_k = \alpha \mathcal{G}$:

$$\varphi^T \alpha \mathcal{G} = \alpha \cdot \varphi^T \mathcal{G}$$

Now compute:

$$\varphi^T \mathcal{G} = \varphi^T \left(\frac{1}{\lambda_1^*} H^{-1}(g + \lambda_2^* \varphi) \right) = \frac{1}{\lambda_1^*} \left[\varphi^T H^{-1} g + \lambda_2^* \varphi^T H^{-1} \varphi \right]$$

Since H is positive definite and $\lambda_2^* > 0$, both inner products are real-valued and bounded. In particular, $\varphi^T H^{-1} \varphi > 0$. Therefore:

$$\varphi^T \mathcal{G} > 0 \quad \Rightarrow \quad \alpha \cdot \varphi^T \mathcal{G} > 0$$

To satisfy $\varphi^T(\theta_{k+1} - \theta_k) > \delta$, we need:

$$\alpha \cdot \varphi^T \mathcal{G} > \delta$$

This condition is enforced by backtracking line search, which ensures the constraint is met in practice. When necessary, feasibility can also be restored using the correction step.

The policy update satisfies both the standard TRPO trust region constraint and the additional safety exclusion condition. This guarantees that each update remains within a safe neighborhood while avoiding unsafe policy behaviors, as estimated by the auxiliary unsafe policy. □

A.2 Proof of Theorem 2: Worst Case Bound on Updating the Constraint-satisfying Policies

Proof. We follow the performance difference bound for trust region methods as derived in Achiam et al. [2017], Zhang et al. [2020], which gives:

$$J(\pi_{\theta_{k+1}}) - J(\pi_{\theta_k}) \geq \frac{1}{1-\gamma}\,\mathbb{E}_{s\sim d_{\pi_{\theta_k}},\, a\sim\pi_{\theta_{k+1}}}\left[A_{\pi_{\theta_{k+1}}}(s,a) - \frac{2\gamma\,\epsilon^o_{\pi_{k+1}}}{1-\gamma}\sqrt{\frac{1}{2}D_{\mathrm{KL}}(\pi_{\theta_{k+1}}\|\pi_{\theta_k})[s]}\right] \tag{A.1}$$

Since $A_{\pi_{\theta_{k+1}}}(s,a)$ is bounded by $\epsilon^o_{\pi_{k+1}}$ in magnitude for all s, we apply the worst case:

$$J(\pi_{\theta_{k+1}}) - J(\pi_{\theta_k}) \geq \frac{1}{1-\gamma}\,\mathbb{E}_s\left[-\frac{2\gamma\,\epsilon^o_{\pi_{k+1}}}{1-\gamma}\sqrt{\frac{1}{2}D_{\mathrm{KL}}(\pi_{\theta_{k+1}}\|\pi_{\theta_k})[s]}\right] \tag{A.2}$$

By Jensen's inequality and the constraint that the average KL divergence is bounded by δ:

$$\mathbb{E}_s\left[\sqrt{D_{\mathrm{KL}}(\pi_{\theta_{k+1}}\|\pi_{\theta_k})[s]}\right] \leq \sqrt{\mathbb{E}_s\left[D_{\mathrm{KL}}(\pi_{\theta_{k+1}}\|\pi_{\theta_k})[s]\right]} \leq \sqrt{2\delta} \tag{A.3}$$

Thus, we have:

$$J(\pi_{\theta_{k+1}}) - J(\pi_{\theta_k}) \geq -\frac{\sqrt{2\delta}\,\gamma\,\epsilon^o_{\pi_{k+1}}}{(1-\gamma)^2} \tag{A.4}$$

This concludes the proof. □

Proof of Theorem 3

We begin from the cost performance bound for trust region methods (as in Achiam et al. [2017]):

$$J_c(\pi_{\theta_{k+1}}) - J_c(\pi_{\theta_k}) \leq \frac{1}{1-\gamma}\mathbb{E}_{s\sim d_{\pi_k}, a\sim\pi_{\theta_{k+1}}}\left[\mathbb{A}^c_{\pi_{\theta_k}}(s,a)\right] + \frac{2\gamma\epsilon^c_{\pi_{k+1}}}{(1-\gamma)^2}\sqrt{2\bar{D}_{\mathrm{KL}}(\pi_{\theta_k}\parallel\pi_{\theta_{k+1}})}$$

The first term is zero or negative if π_{θ_k} already satisfies the constraint, and the policy update is toward lower-cost regions due to the exclusion of the unsafe policy. Therefore, we conservatively bound:

$$J_c(\pi_{\theta_{k+1}}) - J_c(\pi_{\theta_k}) \leq \frac{2\gamma\epsilon^c_{\pi_{k+1}}}{(1-\gamma)^2}\sqrt{2\delta}$$

where we substituted $\bar{D}_{\mathrm{KL}}(\pi_{\theta_k}\parallel\pi_{\theta_{k+1}}) \leq \delta$ from the safe trust region constraint.

Letting $\epsilon_c(\delta) = \frac{2\gamma\epsilon^c_{\pi_{k+1}}}{(1-\gamma)^2}\sqrt{2\delta}$ completes the proof. □

Lemma. Joint-advantage surrogate is compatible with TRPO bounds

Fix a discount $\gamma \in (0,1)$ and freeze the Lagrange weight at $\bar{\lambda}$ within a single policy update (the PID controller may update λ between updates). Let

$$J_{\bar{\lambda}}(\pi) \;=\; \mathbb{E}_\pi\left[\sum_{t\geq 0}\gamma^t\big(r_t - \bar{\lambda}\,c_t\big)\right], \qquad A^{\bar{\lambda}}_{\pi_{\theta_k}}(s,a) \;\text{ be the advantage under the scalarized reward } r-\bar{\lambda}c.$$

Define the *GAE-derived joint advantage*

$$\tilde{A}^{\text{joint}}_{\pi_{\theta_k}}(s,a) \;=\; \tilde{A}^r_{\pi_{\theta_k}}(s,a) \;-\; \bar{\lambda}\,\tilde{A}^c_{\pi_{\theta_k}}(s,a),$$

where $\tilde{A}^r$ and $\tilde{A}^c$ are GAE estimates for reward and cost, each Z-score normalized over the batch with variances bounded away from zero (i.e., $\sigma_r, \sigma_c \geq \sigma_{\min} > 0$). Let the TRPO step satisfy $\bar{D}_{\mathrm{KL}}(\pi_{\theta_k}\parallel\pi_{\theta_{k+1}}) \leq \delta$. Define the (estimation) mismatch

$$\epsilon_{\text{est}} \;=\; \max_s\left|\mathbb{E}_{a\sim\pi_{\theta_{k+1}}}\Big[\tilde{A}^{\text{joint}}_{\pi_{\theta_k}}(s,a) - A^{\bar{\lambda}}_{\pi_{\theta_k}}(s,a)\Big]\right|.$$

Then the TRPO-style performance bound holds with the joint surrogate:

$$J_{\bar{\lambda}}(\pi_{\theta_{k+1}}) - J_{\bar{\lambda}}(\pi_{\theta_k}) \;\geq\; L^{\text{joint}}_{\pi_{\theta_k}}(\pi_{\theta_{k+1}}) \;-\; \frac{4\gamma}{(1-\gamma)^2}\left(\epsilon_{\text{base}} + \epsilon_{\text{est}}\right)\sqrt{2\delta},$$

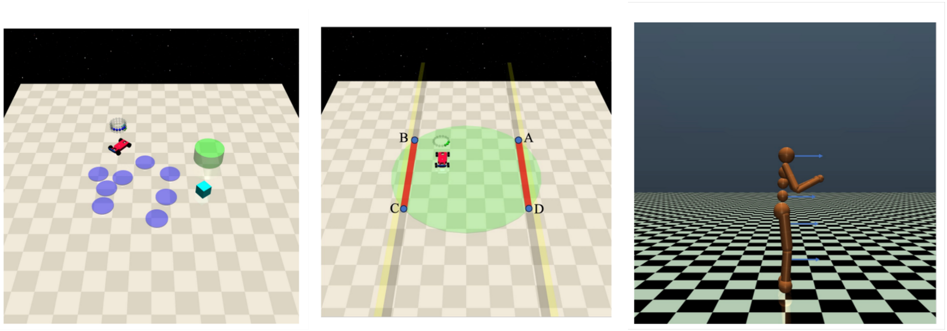
where $L^{\text{joint}}_{\pi_{\theta_k}}(\pi) = \mathbb{E}_{(s,a)\sim\pi_{\theta_k}}\left[\frac{\pi(a|s)}{\pi_{\theta_k}(a|s)}\,\tilde{A}^{\text{joint}}_{\pi_{\theta_k}}(s,a)\right]$ is the linearized surrogate using the joint advantage, and $\epsilon_{\text{base}} = \max_s \left|\mathbb{E}_{a\sim\pi_{\theta_{k+1}}}[A^{\bar{\lambda}}_{\pi_{\theta_k}}(s,a)]\right|$ is the standard TRPO approximation term (state-distribution mismatch).[1]

Proof sketch. Start from the performance difference lemma for the scalarized objective $J_{\bar{\lambda}}$: $J_{\bar{\lambda}}(\pi') - J_{\bar{\lambda}}(\pi) = \frac{1}{1-\gamma}\,\mathbb{E}_{s\sim d_{\pi'}}\mathbb{E}_{a\sim\pi'}[A^{\bar{\lambda}}_{\pi}(s,a)]$. Add and subtract the joint surrogate inside the expectation and apply the triangle inequality to separate (i) the standard TRPO term with $A^{\bar{\lambda}}_{\pi}$ (giving ϵ_{base}) and (ii) the surrogate mismatch with $\tilde{A}^{\text{joint}}_{\pi}$ (giving ϵ_{est}). Convert the state-distribution change to average KL via total-variation bounds (Pinsker), and linearize the objective to obtain $L^{\text{joint}}_{\pi}(\pi')$ in place of the exact expectation, yielding the stated inequality with constant $\frac{4\gamma}{(1-\gamma)^2}$ for average KL Schulman et al. [2015]. The Z-score normalization and GAE introduce only an estimation bias captured by ϵ_{est} (bounded under the assumed finite variances and bounded rewards/costs). □

B Experimental Tasks

B.1 Task Description

- Goal. safety constraints here involve avoiding colliding with 3D obstacles while trying to reach multiple goals (Figure.4(a)).
- Circle. the reward is maximized by moving along the green circle and not allowed to enter the outside of the red region, so its optimal path follows the line segments AD and BC (Figure.4(b)) Ji et al. [2023].
- Velocity. On this task environment we use two different agents - ant and hopper. In this tasks, agents aim for higher reward by moving faster, but they must also adhere to velocity constraints for safety. Specifically, in a two-dimensional plane, the cost is computed as the Euclidean norm of the agent's velocities (v_x and v_y).(Figure.4(c)) Ji et al. [2023].



(a) Goal Task (b) Circle Task (c) Velocity Task

Figure 4: SafetyGym environments we ran our expermints in.

[1] See Schulman et al. [2015] for the derivation of the TRPO bound and use of Pinsker's inequality to relate total variation to average KL.

C Ablation Studies

C.1 KL Divergence Thresholds (δ)

Setup. We assess sTRPO's sensitivity to the KL trust–region bounds by varying a *single* threshold applied to both constraints, $\delta \in \{10^{-2}, 10^{-3}, 10^{-4}\}$, and by testing a *decoupled* setting with a wide update region and a tight exclusion margin: $\delta_1{=}10^{-2}$ for the TRPO step and $\delta_2{=}10^{-4}$ for repulsion from the unsafe policy.

Results.

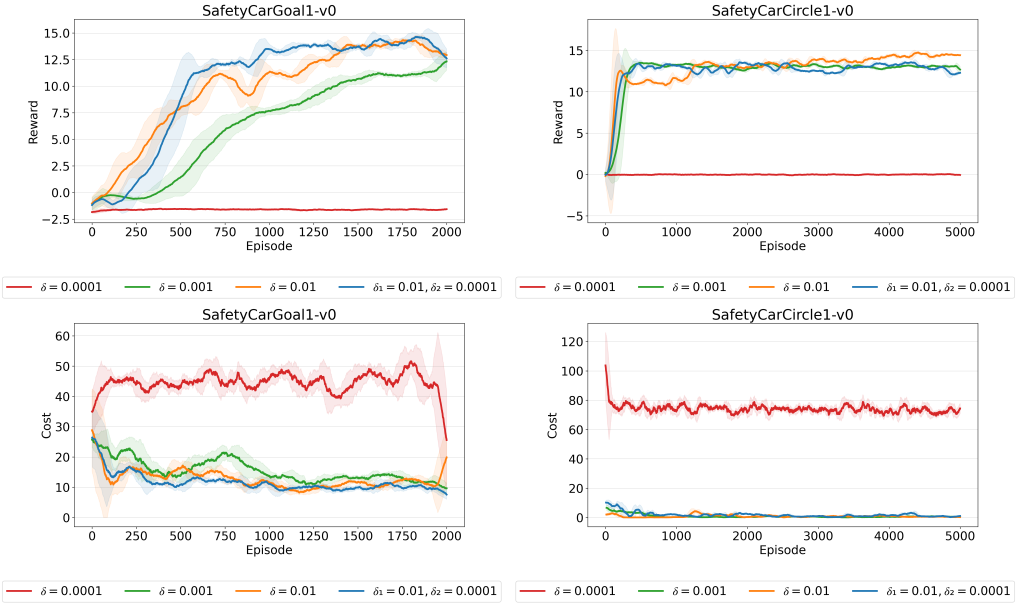
- **Overly small δ is harmful.** With $\delta{=}10^{-4}$, learning becomes over-constrained: in *Car–Circle* the agent remains in high-cost behavior (cost $\approx$ 70–80) and reward stays near zero; in *Car–Goal* it fails to improve; in *Ant–Velocity* learning is very slow; and in *Hopper–Velocity* a pronounced transient cost spike appears despite eventual reward gains.

- **Moderate/large δ stabilize and speed learning.** $\delta{=}10^{-2}$ (and 10^{-3}) drive costs toward zero rapidly in *Circle* and *Ant–Velocity*, with $\delta{=}10^{-2}$ converging fastest; $\delta{=}10^{-3}$ can yield slightly higher asymptotic reward (e.g., *Ant–Velocity*) at the expense of slower convergence.

- **Decoupling the radii helps.** Using $\delta_1{=}10^{-2}$, $\delta_2{=}10^{-4}$ provides the best reward–safety trade-off in *Car–Goal* (faster reward rise while keeping low cost) and matches the stable behavior of $\delta{=}10^{-2}$ in the velocity tasks.

Takeaway. Extremely small KL radii (10^{-4}) over-constrain updates and can *increase* safety cost or stall reward. A wider trust region for the main step ($\delta_1{\approx}10^{-2}$) combined with a tighter margin against the unsafe policy ($\delta_2{\approx}10^{-4}$) offers a robust default across tasks. See Appendix B for the δ-ablation figures.

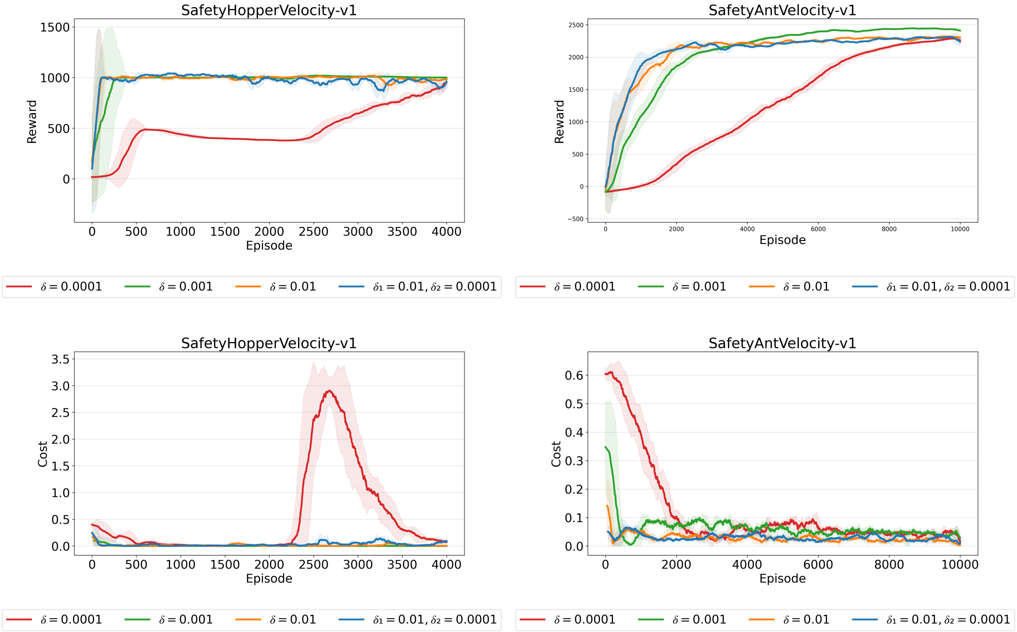
C.1.1 Results



(a) Goal Task

(b) Circle Task

Figure 5: Performance with varying δ



(a) Velocity Hopper

(b) Velocity Ant

Figure 6: Performance with varying δ

C.2 Reference Unsafe Policy: Choice of Training Algorithm

Setup. sTRPO relies on a fixed auxiliary "unsafe" policy π^{unsafe} to shape the exclusion signal. We ablate how π^{unsafe} is trained by comparing (i) **TRPO** and (ii) **DTRPO** (TRPO with a distributional value function). The goal is to see how the training algorithm changes the geometry of the reference policy and, consequently, the quality of the repulsive direction used by sTRPO.

Results. Across tasks, the two choices produce meaningfully different unsafe policies (Figures below):

- **Navigation (CarCircle1).** DTRPO_Unsafe converges much faster to a higher reward plateau and maintains lower cost throughout training than TRPO_Unsafe (right column, top/bottom).
- **Goal-reaching (CarGoal1).** DTRPO_Unsafe attains consistently higher reward while incurring substantially lower cost than TRPO_Unsafe (left column, top/bottom).
- **Locomotion–AntVelocity.** TRPO_Unsafe reaches the highest asymptotic reward, with slightly lower steady-state cost late in training; DTRPO_Unsafe lags by $\sim$ a few hundred reward and stays at a marginally higher cost (top row, center).
- **Locomotion–HopperVelocity.** The two unsafe policies are nearly indistinguishable: both achieve similar reward and drive cost to (near) zero quickly (bottom row, center).

Takeaway. The algorithm used to train π^{unsafe} changes the resulting exclusion signal. DTRPO tends to produce a stronger, more stable reference in the navigation tasks (faster learning, lower cost), whereas in AntVelocity TRPO yields a slightly higher-reward, lower-cost reference. In practice, sTRPO is robust to either choice; environment-specific differences primarily affect learning speed rather than final safety. We therefore report main results with DTRPO_Unsafe and include TRPO_Unsafe results in Appendix B for completeness.

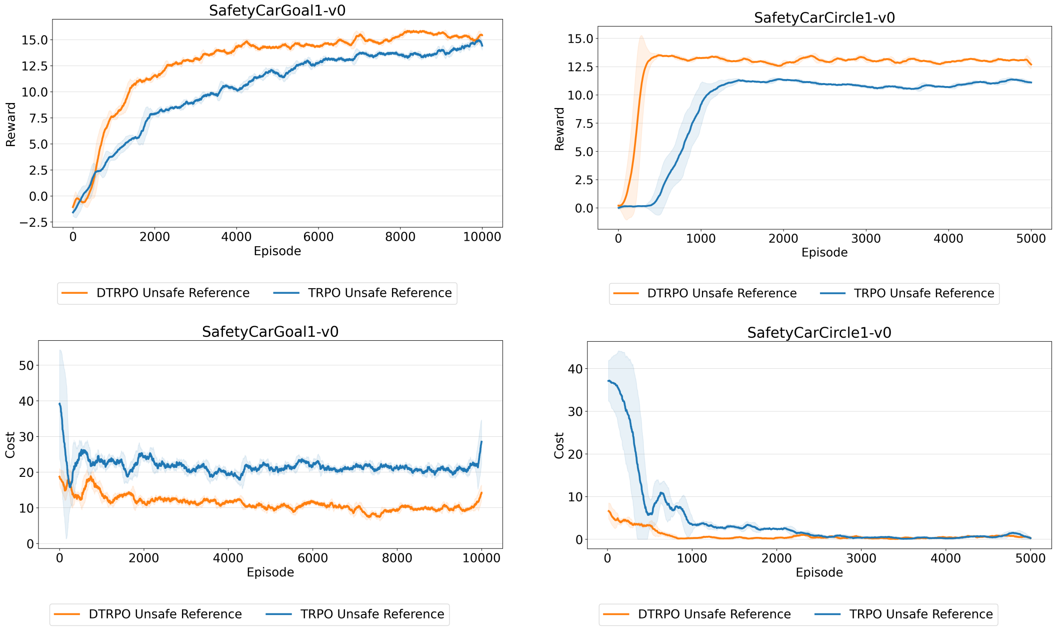


Figure 7: Reference unsafe policy ablation (Car tasks). DTRPO_Unsafe: higher reward and lower cost in both Goal and Circle.

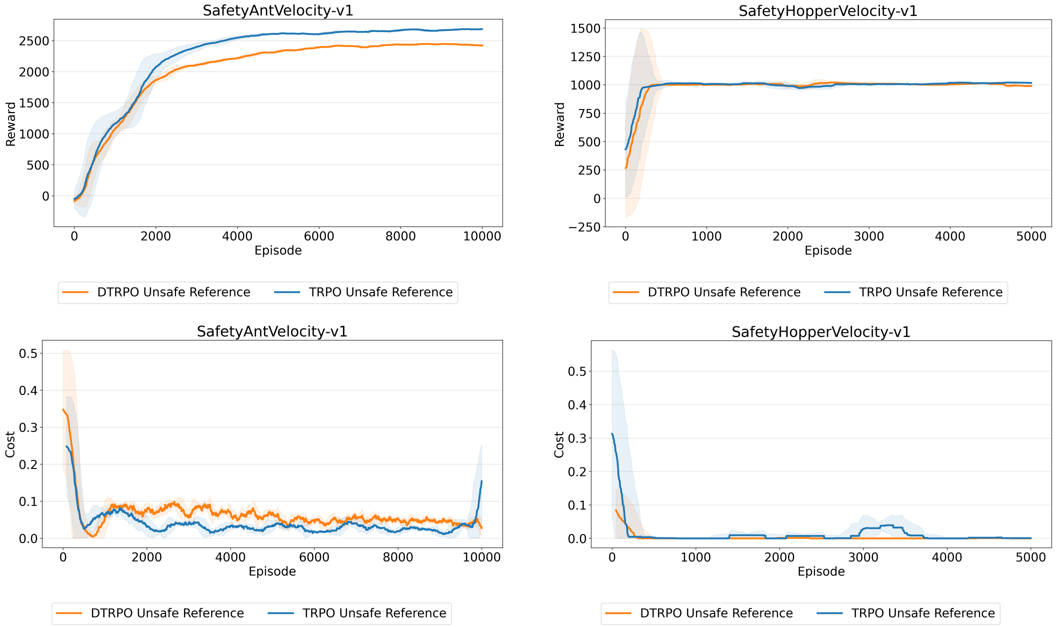


Figure 8: Reference unsafe policy ablation (locomotion tasks). Ant: TRPO_Unsafe slightly higher reward / lower cost; Hopper: negligible differences.

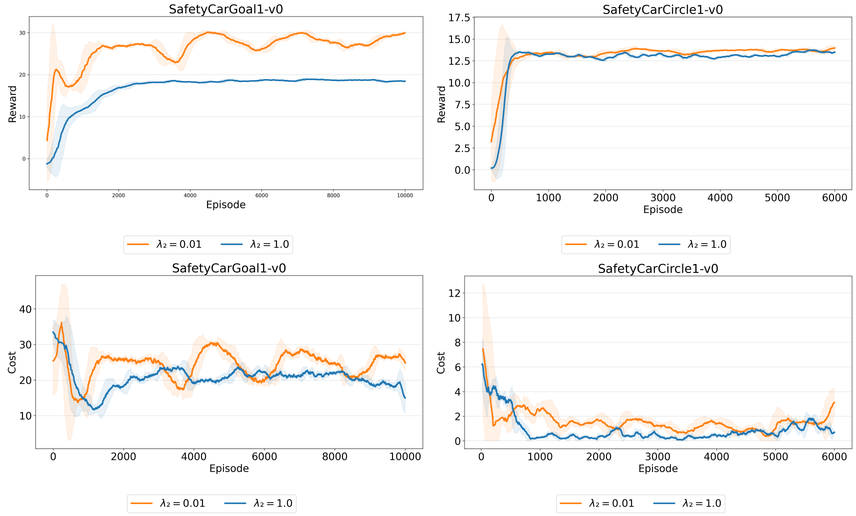
C.3 Effect of the unsafe–objective weight λ_2

Setup. Recall that λ_2 controls the contribution of the unsafe objective φ in the Newton step, $\mathcal{G} = H^{-1}(g + \lambda_2\varphi)$ (Eq. (14)). We sweep $\lambda_2 \in \{0.01,\ 1.0\}$ to study the reward–safety trade-off.

Results. Across the navigation tasks, increasing λ_2 consistently prioritizes safety, while its impact on velocity tasks is negligible:

- **Car–Circle1.** A larger weight ($\lambda_2 = 1.0$) drives the cost rapidly to near zero and keeps it low thereafter, with reward essentially unchanged relative to $\lambda_2 = 0.01$ (Fig. **??**).
- **Car–Goal1.** $\lambda_2 = 1.0$ yields markedly lower cost throughout training (roughly ∼20 vs. ∼25–30 with $\lambda_2 = 0.01$), but at a noticeable reward reduction, illustrating a clear safety–performance trade-off (Fig. **??**).
- **Ant/Hopper Velocity.** Costs are already very small for both settings. Changing λ_2 produces minimal differences in either cost or reward (Figs. **??**, **??**).

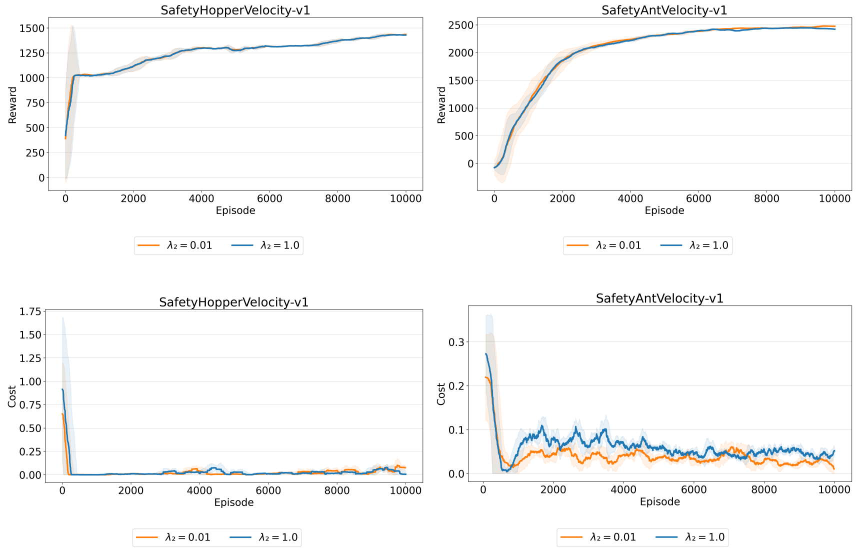
Takeaway. λ_2 is an effective knob to trade reward for safety in navigation-style tasks (where $\lambda_2 = 1.0$ yields the safest behavior), while in low-cost velocity tasks the choice of λ_2 has little effect.



(a) Goal Task

(b) Circle Task

Figure 9: Performance with varying λ_2



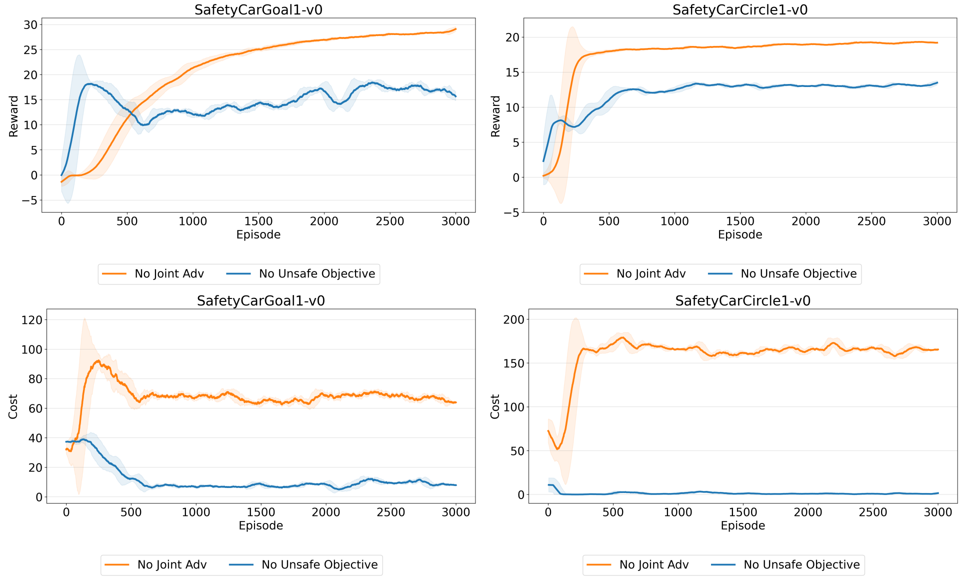
(a) Velocity Hopper

(b) Velocity Ant

Figure 10: Performance with varying λ_2

C.4 Contribution of Joint Advantage & Unsafe Objective

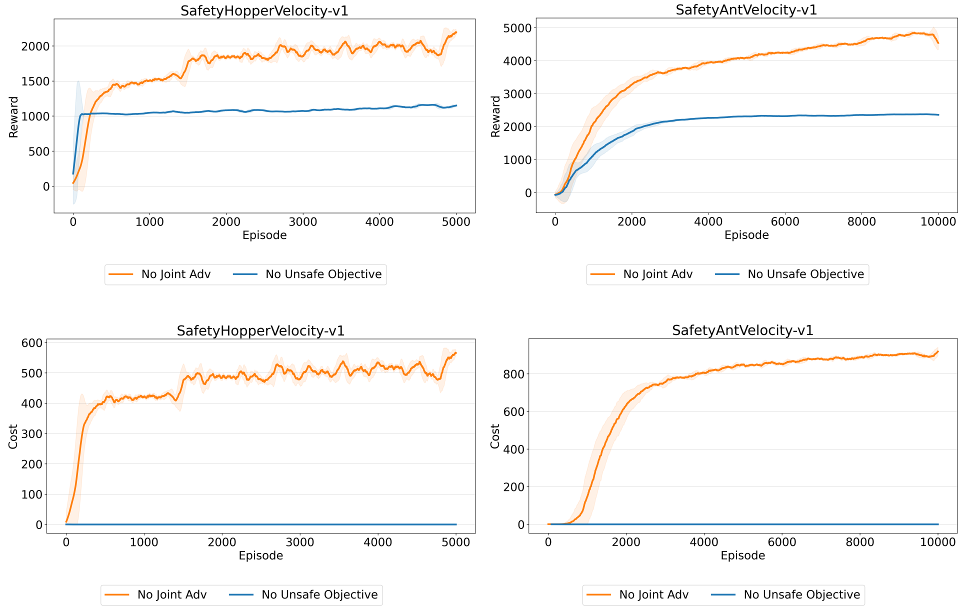
To measure the contribution of the joint advantage function and unsafe objective ϕ on the performance, we disabled the corresponding component and observed the results. Without the joint advantage function, the cost increased drastically. In contrast, the performance was reasonably well without the unsafe objective.



(a) Goal Task

(b) Circle Task

Figure 11: Performance without specific components



(a) Velocity Hopper

(b) Velocity Ant

Figure 12: Performance without specific components

D Robustness Test

We evaluate robustness under a *random action attack*: at each timestep the agent's action is replaced by a uniformly random action with probability (attack rate) $\alpha \in \{0.0, 0.1, \ldots, 1.0\}$. For each α we run 1000 episodes and report average reward and cost. We compare sTRPO to CPO, CPPO_PID, CUP, FOCOPS, PCPO, PPO_LAG, RCPO, and TRPO_LAG (Figs. 13–14).

SafetyCarCircle1-v0. sTRPO yields **substantially lower cost** than all baselines across attack rates (Fig. 13 right). For example, at $\alpha \le 0.7$ sTRPO stays near single-digit cost while baselines range from ~ 40 to > 250. This reflects a conservative, safety-first policy. The reward is slightly below the best baselines at $\alpha = 0$ and declines faster with increasing α (Fig. 13 top-right), indicating an explicit safety–reward trade-off on this navigation task.

SafetyCarGoal1-v0. All methods lose reward roughly linearly as attacks intensify (Fig. 13 top-left). sTRPO's reward curve is among the steepest (more conservative control). Costs are mid-pack overall but exhibit a **pronounced spike** around $\alpha \approx 0.6$ (Fig. 13 left), suggesting a vulnerability at intermediate attack rates (likely due to compounding action flips near goal transitions). This is a target for adding a stronger runtime shield or attack detector.

SafetyAntVelocity-v1. sTRPO attains the **highest reward** over a wide range of α (e.g., ~ 2.6k at $\alpha = 0$ vs. ~ 1.6k for the next best) and degrades more gracefully up to $\alpha \approx 0.8$ (Fig. 14 top-right). Costs remain among the lowest across all attack rates (bottom-right), indicating that the trust-region update plus "diverge-from-unsafe" guidance help preserve stability in continuous control.

SafetyHopperVelocity-v1. sTRPO maintains **state-of-the-art reward** from $\alpha = 0$ to roughly $\alpha \approx 0.7$ (Fig. 14 top-left) while keeping costs near zero for most α (bottom-left). Beyond $\alpha 0.7$ reward drops sharply—consistent with the attack overwriting a large fraction of actions—yet sTRPO remains competitive with or above baselines.

Extreme attacks. At $\alpha \to 1.0$ all methods converge to near-zero (or negative) reward and similar costs, as expected when actions are almost entirely random (sanity check).

Takeaways. (i) On continuous-control tasks (Ant/Hopper), sTRPO shows **graceful degradation** and the best reward–cost profile over a broad attack range. (ii) On navigation tasks (CarCircle/CarGoal), sTRPO **prioritizes safety**: dramatically lower costs on `Circle1` with a corresponding reward reduction; a mid-attack cost spike on `Goal1` reveals a corner case for improvement. (iii) Overall, the results support sTRPO's robustness: enforcing a trust region while repelling the learned policy from an unsafe reference yields safer behavior under action perturbations, with competitive or superior reward whenever safety does not require aggressive conservatism.

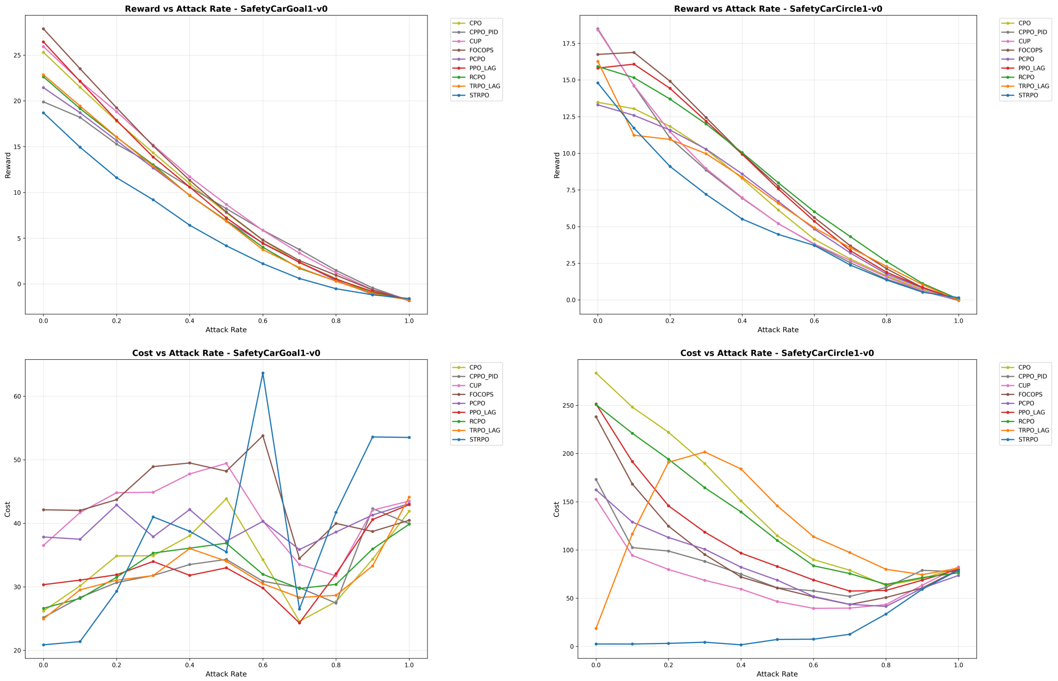


Figure 13: Robustness on car navigation tasks under random action attacks.

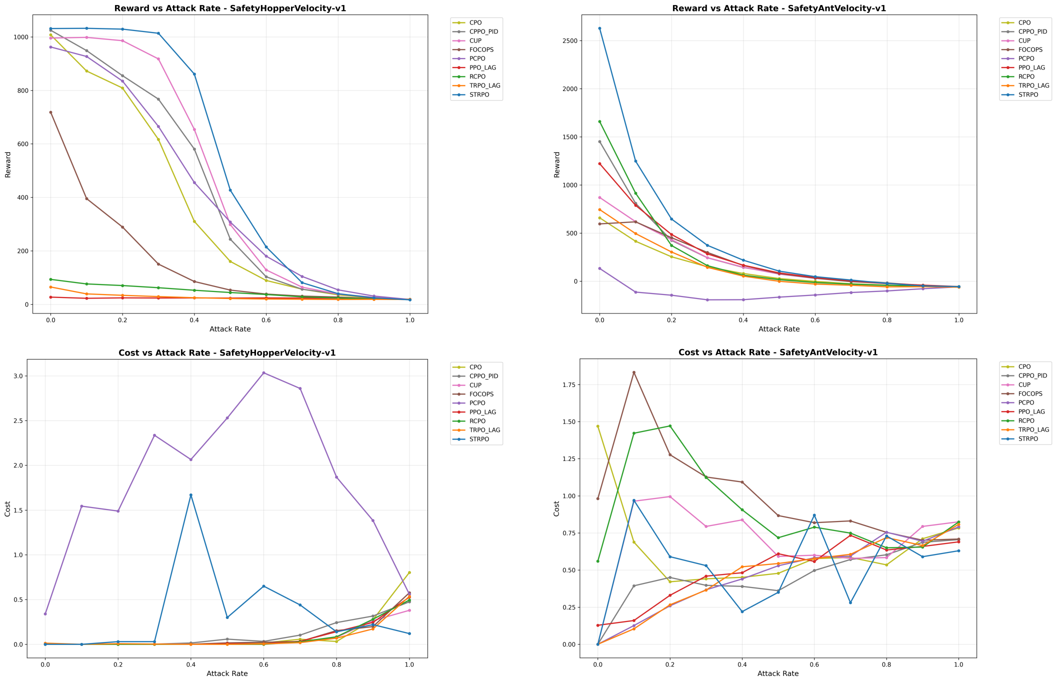


Figure 14: Robustness on MuJoCo velocity tasks under random action attacks.