Provably Optimal Reinforcement Learning under Safety Filtering

Donggeon David Oh[*†‡] **Duy P. Nguyen**[*‡] **Haimin Hu**[‡§] **Jaime F. Fisac**[‡]

Abstract

Recent advances in reinforcement learning (RL) enable its use on increasingly complex tasks, but the lack of formal safety guarantees still limits its application in safety-critical settings. A common practical approach is to augment the RL policy with a *safety filter* that overrides unsafe actions to prevent failures during both training and deployment. However, safety filtering is often perceived as sacrificing performance and hindering the learning process. We show that this perceived safety–performance tradeoff is not inherent and prove, for the first time, that enforcing safety with a sufficiently permissive safety filter does *not* degrade asymptotic performance. We formalize RL safety with a *safety-critical Markov decision process (SC-MDP)*, which requires categorical, rather than high-probability, avoidance of catastrophic failure states. Additionally, we define an associated *filtered MDP* in which all actions result in safe effects, thanks to a safety filter that is considered to be a part of the environment. Our main theorem establishes that (i) learning in the filtered MDP is safe categorically, (ii) standard RL convergence carries over to the filtered MDP, and (iii) any policy that is optimal in the filtered MDP—when executed through the same filter—achieves the *same* asymptotic return as the best safe policy in the SC-MDP, yielding a complete separation between safety enforcement and performance optimization. We validate the theory on Safety Gymnasium with representative tasks and constraints, observing zero violations during training and final performance matching or exceeding unfiltered baselines. Together, these results shed light on a long-standing question in safety-filtered learning and provide a simple, principled recipe for safe RL: train and deploy RL policies with the most permissive safety filter that is available.

1 Introduction

Reinforcement learning (RL) has demonstrated outstanding performance in complex domains, yet a fundamental limitation is the lack of strict safety guarantees, both during policy training and at deployment. This poses a major barrier to deploying RL in safety-critical applications, from autonomous driving to healthcare management, where even a single safety violation would be catastrophic. One promising approach towards safety assurances in RL is to integrate a *safety filter* [1] that monitors the system and overrides any unsafe candidate actions to preempt downstream failures. In effect, the RL agent can explore and learn *within* a safe set, while the filter ensures that it never enters unsafe regions.

However, enforcing hard safety constraints during training is commonly observed to negatively interfere with the learning process and limit the asymptotic performance achievable by the learned policy [2–5]. In addition, prior studies on safety filtering in control systems [6–8] have reported that

[*]D. D. Oh and D. P. Nguyen contributed equally.
[†]Corresponding author: do9948@princeton.edu
[‡]Department of Electrical and Computer Engineering, Princeton University
[§]Department of Computer Science, Johns Hopkins University

Second Conference of the International Association for Safe and Ethical Artificial Intelligence (IASEAI'26).

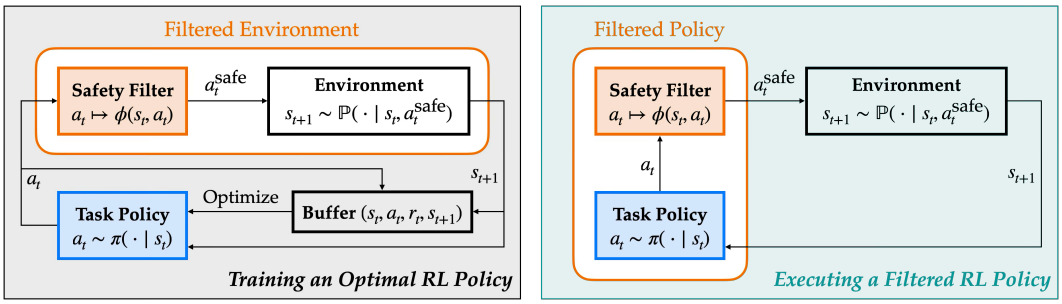

Figure 1: Our proposed framework for training and deploying optimal RL policies under safety filtering. First, we train a filter-agnostic task policy in a safety-enforcing environment where every action is passed through a safety filter. Then, we deploy the learned task policy with the same filter at runtime. The filtered RL policy provably achieves the *same asymptotic performance* as it had been trained with the safety constraints in mind.

if the task policy is unaware of the presence of a safety filter, it may repeatedly attempt unsafe actions that keep triggering filter overrides, resulting in suboptimal oscillations or "chattering" behaviors. In such cases, the separation between safety and performance is incomplete: the task policy must be made explicitly aware of the task-agnostic safety filter in order to avoid costly overrides.

These concerns raise a fundamental, unsettled question in safe RL: *Does safety filtering inevitably hinder the agent's learning and ultimately limit the achievable performance?* We provide, for the first time, a theoretical answer to this question by proving that no such entanglement between safety filtering and task policy training is necessary to obtain an optimal constrained policy. We view this as a new foundational result in safe RL, with significant practical implications as AI technology advances towards increasingly complex autonomous systems.

Contributions. *Our key insight is that safety and performance in RL can be learned separately and still converge towards provably safe and task-optimal agent behavior.* We prove that, under a safety filter that is least-restrictive yet capable of preventing all safety violations, an RL agent trained in the filtered environment converges to the *optimal* safety-constrained policy; in particular, the convergence guarantees enjoyed by RL algorithms on stationary, discounted Markov decision process (MDP)s with bounded rewards *carry over* to this setting [9–12]. Remarkably, we show—*for the first time*—that the *safety–performance separation* is complete in safe RL: the task policy can be trained entirely agnostic to the safety filter and still attain the *same* asymptotic performance as if it had been trained with the safety constraints in mind. The safety filter guarantees that the agent never leaves the safe set, wherein the RL algorithm, free of the extra burden of handling safety, optimizes the task performance. To complement our theoretical findings, we empirically validate the safe RL principle using the Safety Gymnasium benchmark [13], a modern successor to OpenAI's Safety Gym [14], across different tasks, constraints, and environments.

2 Background and Related Work

Constrained reinforcement learning. Conventionally, RL methods model safety through a constrained Markov decision process (CMDP) [15], where the agent aims to maximize its return subject to a budget on *expected cumulative costs* incurred in failure states. Prominent CMDP-based RL approaches include trust-region methods such as Constrained Policy Optimization (CPO) [16], Lagrangian/primal–dual methods (including PID–Lagrangian stabilizations) [17], reward–penalty variants such as Reward Constrained Policy Optimization (RCPO) [18], and Lyapunov-based projected policy optimization [19]. Some methods depart from the standard CMDP formulation by imposing a cap on expected cost at each individual time step [20] or considering risk measures like conditional value-at-risk (CVaR) rather than expected costs [21]. While effective at enforcing some *statistical measure* of safety, the above approaches generally *permit failures* along individual trajectories. By contrast, our formulation considers truly safety-critical systems where catastrophic failures are *categorically* unacceptable, and it consequently requires their probability to be exactly zero.[1]

[1] Our problem can be viewed as a non-generic *limiting class* of CMDPs where the allowable budget is exactly zero, or costs are infinite. The *opposite* limiting class is MDPs, where the budget is infinite or costs are zero.

Safety filters. A safety filter is an automatic process that continually monitors an agent's proposed actions and, when necessary, modifies them to avoid entering no-win scenarios from which future failures may be inescapable. These techniques are popular in robotics and control, where safety is more often formalized as all-time satisfaction of state constraints [1]. Representative classes include (i) model predictive safety filters (MPSFs), which verify system safety at runtime by forward-simulating a dynamics model or solving a trajectory optimization problem [22–24], (ii) control barrier function (CBF)-based filters, which perform runtime optimization at each control cycle to smoothly slow down any approach to the boundary of a known safe set [25–28], and (iii) reachability-based filters, which first solve a Bellman/Hamilton–Jacobi (HJ) equation to compute the largest possible safe set and then keep the agent from exiting it at runtime [29, 30]. Recently, neural approximations of the *safety value function* [31–33] have enabled scalable synthesis and deployment of reachability-based filters beyond the reach of numerical tools, from robotic walking and manipulation to air traffic control and high-speed car racing [34–38].

Safety-filtered reinforcement learning. In safety-critical settings, safety filters can be used during and after RL training to detect and modify unsafe candidate actions before the agent executes them in the environment. Although related, safety-filtered RL differs from shielded RL [39, 40]: shields are synthesized over (finite-state) abstractions, whereas safety filters are defined directly on the original (often continuous-state) MDP via controlled invariance. Prior works in safety-filtered RL have adopted CBF safety filters not only to guarantee safety but also to guide learning by constraining the set of explorable policies [41]; this idea also extends to robust CBFs that handle model uncertainty and disturbances [42]. MPSF enforces constraints via short-horizon prediction during learning, enabling zero or minimal violations in data collection and improved sample efficiency [8, 23, 24]. Reachability-based approaches include iteratively estimating the safe set (and sometimes a fallback) to override unsafe proposed actions online [43], as well as provably safe action projection using reachability analysis and polynomial zonotopes during training [44]. Across these lines, empirical results report that RL under safety filtering improves exploration efficiency by preventing safety violations and thus *accelerates* the convergence of task policies [8]. However, these works ultimately fall short of providing *optimality* guarantees for the executed (filtered) policy with respect to the safety-critical MDP. Recent works study optimality of RL under a safety monitor [45] or a safeguard [46], concepts similar to safety filters; however, they do not characterize the *maximal* controlled-invariant safe set, which prevents them from establishing that their safe RL algorithms result in optimal asymptotic performance given a safety-critical MDP. We suspect it is this lack of a formal optimality guarantee that contributes to the common misconception that safety filtering hinders the attainable task performance of RL policies [2–5]. Our results fill this gap by showing that, under a sufficiently permissive safety filter, the optimality of asymptotic task performance is preserved.

Safe model-based learning. Orthogonal to safety-critical model-free RL, a rich line of literature focuses on learning (or refining) system *dynamics* models and certifying safety during training model-based RL policies. SafeOpt [47] ensures safety while optimizing an unknown function modeled as a Gaussian process (GP), using confidence bounds to assess the safety of unexplored decisions. In a complementary direction, Akametalu et al. [48] propose to reduce safety filter conservativeness by learning the system's unknown dynamics, treated as a disturbance input, with a GP, and integrating the relaxed safety constraints into RL. Similarly, Berkenkamp et al. [49] and Richards et al. [50] learn Lyapunov certificates (and associated regions of attraction) from data and leverage them to enforce safe exploration. Another line of work uses model predictive control (MPC) as the safety mechanism around learned dynamics—typically via (i) uncertainty-aware prediction for tightened or chance constraints, (ii) terminal conditions (invariant/robust sets and backups) for recursive feasibility, and (iii) safety certificates (Lyapunov/barrier) to justify constraint satisfaction; see Hewing et al. [51] for a comprehensive review. Koller et al. [5] couple a GP dynamics model with chance-constrained MPC to enable high-probability safe exploration while improving the policy. In summary, these safe model-based learning methods generally require a coarse dynamic model and an initial safe set with a backup controller, rely on *high-probability* (not categorical) guarantees, face scalability limits due to GP regressors, and the learned dynamics are often residual on top of a known structure. By contrast, we focus on safe model-free reinforcement learning, which bypasses these modeling requirements and directly learns task policies under safety filtering without relying on explicit dynamics models.

3 Problem Formulation

We study an RL agent operating in a stochastic environment subject to failure conditions that must never occur. We formalize this setting in two complementary ways: (i) as a safety-critical Markov decision process (SC-MDP), where only actions that keep the agent safe are allowed, and (ii) as a filtered MDP, where a safety filter minimally intervenes to correct unsafe actions. These two perspectives lay the groundwork for the theoretical result in the next section: an optimal policy learned in the filtered MDP, and subsequently deployed with the same filter, achieves the same asymptotic performance as the best policy in the SC-MDP. The assumptions we adopt in this section to formulate the safety-critical decision-making problem and prove our theoretical results are introduced in Assumption 1.

3.1 Safety-Critical Markov Decision Process

We consider an MDP defined by tuple $\mathcal{M} = (\mathcal{S}, \mathcal{A}, \mathbb{P}, r, \gamma)$, where $\mathcal{S}$ is the state space, $\mathcal{A}$ is the action space, $\mathbb{P} : \mathcal{S} \times \mathcal{A} \to \mathcal{P}(\mathcal{S})$ is the transition probability, $r : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ is the reward function, and $\gamma \in (0, 1)$ is the discount factor. We denote $\mathcal{F} \subset \mathcal{S}$ as the *failure set* encoding conditions deemed unacceptable and must be strictly avoided at all times, formally defined as:

$$\mathcal{F} \coloneqq \big\{ s \in \mathcal{S} : \; g(s) < 0 \big\}, \tag{1}$$

where g is a safety *margin* function representing $\mathcal{F}$ as its zero sublevel set.

Definition 1 (Safety-Critical MDP (SC-MDP))**.** *An* SC-MDP *is a tuple* $\mathcal{M}_{\mathrm{SC}} = (\mathcal{S}, \mathcal{A}, \mathbb{P}, r, \gamma, \mathcal{F})$. *For each* $s \in \mathcal{S}$, $\mathcal{M}_{\mathrm{SC}}$ *defines an infinite-horizon constrained optimal control problem:*

$$\max_{\pi} \quad \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t) \;\middle|\; s_0 = s,\; a_t \sim \pi(\cdot \mid s_t),\; s_{t+1} \sim \mathbb{P}(\cdot \mid s_t, a_t) \right] \tag{2a}$$

$$\text{s.t.} \quad \Pr[s_t \notin \mathcal{F},\, \forall t \mid s_0 = s] = 1, \tag{2b}$$

where the task policy π *is a stochastic kernel on* $\mathcal{A}$ *given* $\mathcal{S}$.

We now characterize what constitutes an admissible policy, *i.e.*, one that satisfies the all-time safety constraint (2b), by adopting the notion of set invariance [52]. Specifically, a *controlled-invariant* set $\Omega \subset \mathcal{S}$ guarantees the existence of a policy π that recursively keeps the state s within Ω for all time, *i.e.*, $\forall s \in \Omega$ and $\forall a \in \operatorname{supp} \pi(\cdot \mid s)$, $\operatorname{supp} \mathbb{P}(\cdot \mid s, a) \subseteq \Omega$. Denoting $\Omega^* \subseteq \mathcal{S} \setminus \mathcal{F}$ as the *maximal* controlled-invariant safe set, for each $s \in \Omega^*$, we define the *safe action set* as

$$\mathcal{A}_{\mathrm{safe}}(s) \coloneqq \{\, a \in \mathcal{A} : \; \operatorname{supp} \mathbb{P}(\cdot \mid s, a) \subseteq \Omega^* \,\}. \tag{3}$$

Now, we define the *admissible policy set* as

$$\Pi_{\mathrm{safe}} \coloneqq \Big\{ \pi : \; \operatorname{supp} \pi(\cdot \mid s) \subseteq \mathcal{A}_{\mathrm{safe}}(s),\, \forall s \in \Omega^* \Big\}. \tag{4}$$

The admissible policy set Π_{safe} is the largest set of policies that ensure all-time safety (2b) provided that the system is initialized within Ω^* (see Proposition 1 for the proof). For an admissible policy $\pi \in \Pi_{\mathrm{safe}}$, we define its *value* on $\mathcal{M}_{\mathrm{SC}}$ as

$$V^{\pi}_{\mathcal{M}_{\mathrm{SC}}}(s) \coloneqq \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t) \;\middle|\; s_0 = s,\; a_t \sim \pi(\cdot \mid s_t),\; s_{t+1} \sim \mathbb{P}(\cdot \mid s_t, a_t) \right], \tag{5}$$

and the optimal value as $V^{*}_{\mathcal{M}_{\mathrm{SC}}}(s) \coloneqq \sup_{\pi \in \Pi_{\mathrm{safe}}} V^{\pi}_{\mathcal{M}_{\mathrm{SC}}}(s)$.

Remark 1 (Probabilistic and worst-case uncertainty)**.** *The SC-MDP unifies performance-centric* probabilistic *uncertainty and safety-centric* deterministic *uncertainty within a single framework. In contrast to conventional viewpoints that treat probabilistic and deterministic uncertainty as mutually exclusive, we show that combining them is not only theoretically sound but also advantageous: it achieves the best of both worlds—optimal expected task performance under strict safety guarantees.*

Real systems are often designed around *unknown-but-bounded* uncertainty (*e.g.*, external disturbances and modeling errors), while facing hazards captured here by the failure set $\mathcal{F}$. Many "safe RL" benchmarks and baselines adopt the CMDP formalism [13, 14], in which the agent maximizes

expected return *subject to an expected cumulative constraint budget*. In that view, transient violations are permitted as long as their expected cumulative constraint cost remains below the threshold. As a result, *CMDP formulations can still permit rare but catastrophic failures* on individual trajectories, which are unacceptable in safety-critical decision-making. This also conflicts with the spirit of the operational design domain (ODD), which specifies conditions under which a system should *never* experience a specified class of failures [1]; CMDP formulations generally do not provide per-trajectory, failure-free guarantees under such conditions.

By contrast, the SC-MDP requires *categorical* avoidance of failure, yielding a zero-violation specification by construction. It is robust to tail events while still leaving task performance optimized in expectation. This allows engineers to delineate a clear operating envelope within which the system is *guaranteed* to operate safely—supporting deployment, auditability, and public trust in automated decision-making.

3.2 Filtered Markov Decision Process

To ensure that SC-MDP satisfies all-time safety constraint (2b), we leverage a *safety filter*—an automatic process that monitors the agent's decision-making and, when necessary, modifies the nominal action in order to prevent safety violations. However, constraining the agent to an overly restrictive invariant set may hinder the agent from achieving satisfactory task performance, *i.e.*, to accumulate high discounted rewards. Given the MDP $\mathcal{M}$ together with the maximal controlled-invariant safe set Ω^*, we formalize a *perfect (least-restrictive) safety filter* [1] that enforces safety by intervening if and only if the nominal action causes state s to immediately exit Ω^*.

Definition 2 (Perfect safety filter)**.** *A* perfect (least-restrictive) safety filter *is a map* $\phi : \mathcal{S} \times \mathcal{A} \to \mathcal{A}$ *such that,* $\forall s \in \Omega^*$*,*

1. $\phi(s, a) \in \mathcal{A}_{\text{safe}}(s)$*,* $\forall a \in \mathcal{A}$*,*
2. $\phi(s, a) = a$*,* $\forall a \in \mathcal{A}_{\text{safe}}(s)$*.*

For safe RL, we now define the *filtered MDP*, where every action is passed through a perfect safety filter before execution. This can be interpreted as a "bubble-wrapped" environment, where the agent remains agnostic to the safety filter.

Definition 3 (Filtered MDP)**.** *The filtered MDP is a tuple* $\mathcal{M}_\phi := (\Omega^*, \mathcal{A}, \mathbb{P}_\phi, r_\phi, \gamma)$*, where* ϕ *is a perfect safety filter,* $\mathbb{P}_\phi(\cdot \mid s, a) := \mathbb{P}\big(\cdot \mid s, \phi(s, a)\big)$ *is the filtered transition probability, and* $r_\phi(s, a) := r\big(s, \phi(s, a)\big)$ *is the filtered reward. For a measurable and stationary policy* π*, we define its value on* $\mathcal{M}_\phi$ *as*

$$V^\pi_{\mathcal{M}_\phi}(s) := \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t \, r_\phi(s_t, a_t) \;\middle|\; s_0 = s,\ a_t \sim \pi(\cdot \mid s_t),\ s_{t+1} \sim \mathbb{P}_\phi(\cdot \mid s_t, a_t)\right], \tag{6}$$

and the optimal filtered value as $V^*_{\mathcal{M}_\phi}(s) := \sup_\pi V^\pi_{\mathcal{M}_\phi}(s)$*.*

4 Safe Reinforcement Learning Theory

In this section, we present our core theoretical results. First, we show that learning in the filtered MDP $\mathcal{M}_\phi$ is provably safe, and that the standard RL convergence guarantees remain intact. Second, we prove that any policy that is optimal in $\mathcal{M}_\phi$, when executed under the same filter at deployment, is also optimal in the SC-MDP $\mathcal{M}_{\text{SC}}$. To establish these results rigorously, we begin by stating the technical assumptions required for our analysis.

Assumption 1 (Standing assumptions)**.** *We assume the following throughout this section:*

1. *(*Spaces and measurability*)* $\mathcal{S}$ *and* $\mathcal{A}$ *are Borel spaces.* Ω^* *is nonempty and closed.* $\mathbb{P}(\cdot \mid s, a)$ *and* $r(s, a)$ *are Borel-measurable in their arguments.*
2. *(*Safe initialization*) All training episodes start in* Ω^**.*
3. *(*Stationarity and boundedness*)* $\mathbb{P}(\cdot \mid s, a)$ *and* $r(s, a)$ *are time-invariant. Rewards are bounded, i.e.,* $\sup_{s,a} |r(s, a)| < \infty$*.*

4. *(*Safety filter existence*) A perfect safety filter ϕ (Definition 2) exists; ϕ is measurable and time-invariant.*

We are now ready to state our main theoretical result that formalizes the safety–performance separation principle in safe reinforcement learning.

Theorem 1 (Safe and optimal RL under a perfect safety filter)**.** *If Assumption 1 holds, then the following claims on the safety, convergence, and optimality of RL under safety filtering are true:*

1. ***Safe learning.*** *For any sequence of task policies produced by any RL algorithm during training, the filtered trajectories remain in Ω^* for all time.*
2. ***Convergence.*** *Let* ALG *be an RL algorithm that converges on stationary discounted MDPs with bounded rewards. Then, for any such MDP $\mathcal{M}$,* ALG *also converges on the corresponding filtered MDP $\mathcal{M}_\phi$.*
3. ***Optimality under safety filtering.*** *Let π_ϕ^ε denote a measurable and stationary ε-optimal policy on $\mathcal{M}_\phi$, possibly returned by* ALG*, for some $\varepsilon > 0$:*

$$V_{\mathcal{M}_\phi}^{\pi_\phi^\varepsilon}(s) \geq V_{\mathcal{M}_\phi}^*(s) - \varepsilon, \qquad \forall s \in \Omega^*. \tag{7}$$

We define the executed policy π_{exec} *as the pushforward of π_ϕ^ε by the map $a \mapsto \phi(s,a)$; i.e., for all Borel measurable sets $B \subseteq \mathcal{A}$,*

$$\pi_{\text{exec}}(B \mid s) := \pi_\phi^\varepsilon\big(\{a \in \mathcal{A} : \ \phi(s,a) \in B\} \mid s\big), \qquad \forall s \in \Omega^*. \tag{8}$$

Then, π_{exec} is a safe ε-optimal policy on $\mathcal{M}_{\text{SC}}$:

$$V_{\mathcal{M}_{\text{SC}}}^{\pi_{\text{exec}}}(s) \ \geq \ V_{\mathcal{M}_{\text{SC}}}^*(s) - \varepsilon, \qquad \forall s \in \Omega^*. \tag{9}$$

Proof. The proof is deferred to Appendix A. □

Remark 2 (Safety with optimal performance)**.** ***What Theorem 1 establishes.*** *For any $\varepsilon > 0$, any ε-optimal policy on the filtered MDP $\mathcal{M}_\phi$ induces, when executed through a perfect safety filter, a* safe ε-optimal *policy on the SC-MDP $\mathcal{M}_{\text{SC}}$. Consequently, as an RL algorithm drives suboptimality on $\mathcal{M}_\phi$ to zero (e.g., by sufficient exploration of state–action pairs and appropriate stepsize conditions), the* executed policy becomes asymptotically optimal *on $\mathcal{M}_{\text{SC}}$:*

$$\lim_{\varepsilon \to 0^+} \big(V_{\mathcal{M}_{\text{SC}}}^*(s) - V_{\mathcal{M}_{\text{SC}}}^{\pi_{\text{exec}}}(s)\big) = 0, \quad \forall s \in \Omega^*.$$

Therefore, the safety–performance separation in RL is complete*: enforcing safety with a sufficiently permissive safety filter during task policy training does not degrade asymptotic performance.*

Remark 3 (Practical recipe for Safe RL)**.** ***Equivalence.*** *Consider two ways to obtain a safe policy:*

- Oracle safe RL benchmark: *In the SC-MDP ($\mathcal{M}_{\text{SC}}$), choose any policy that is optimal among those that never violate safety (i.e., those in Π_{safe}).*
- Filter-based safe RL: *Learn in a world where every* proposed *action is passed through the safety filter ($\mathcal{M}_\phi$), and pick any policy that is optimal there. At deployment in the SC-MDP, run the learned policy through the same filter.*

The two routines achieve the same long-run return on safe trajectories in the limit as the training suboptimality on $\mathcal{M}_\phi$ vanishes (Remark 2).

Practical Recipe. *Unlike the oracle safe RL, which cannot be realized by any practical algorithm, filter-based safe RL offers an actionable framework as depicted in Figure 1: first synthesize a (task-agnostic) safety filter for the SC-MDP, then train your task policy with any standard RL algorithm while keeping the safety filter in the loop, and deploy the same policy–filter pair at runtime. The safety filter needs to be computed only once for any number of task policies, provided that the transition probability $\mathbb{P}$ and the failure set $\mathcal{F}$ stay the same. Crucially, under the perfect (least-restrictive) safety filter assumption, this approach yields formal safety guarantees during both training and deployment with asymptotic task performance* identical *to that obtained by training directly under hard safety constraints. In practice, approximate yet sufficiently permissive filters can still dramatically reduce (or eliminate) safety violations during training while matching or even surpassing the performance of baseline constrained RL methods (see subsection 5.1).*

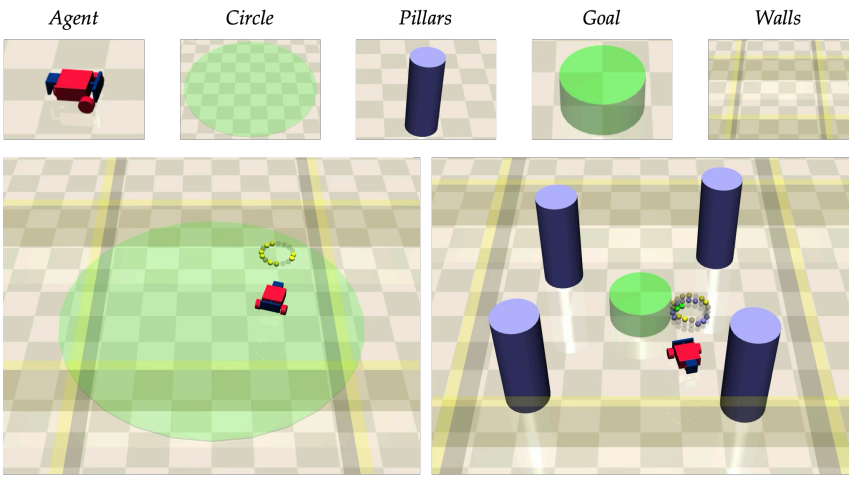

Figure 2: *Top*: environment components (Agent, Circle, Pillars, Goal, Walls). *Bottom*: example scenes for the two tasks from Safety Gymnasium [13]. We use the *Car* agent across all experiments. *Circle* task: the agent aims to track the green circle's boundary as fast as possible while remaining inside the square wall; safety is violated by crossing the wall. *Goal* task: the agent navigates to a fixed green goal while avoiding the pillars and staying inside the wall; safety is violated by crossing the wall or colliding with a pillar. In both tasks, the agent is randomly initialized inside the wall and away from the pillars and the goal.

5 Experiments

To empirically validate our theoretical results, we build on the Safety Gymnasium benchmark [13], a modern successor to OpenAI's Safety Gym [14]. Safety Gymnasium provides a unified reinforcement learning environment for evaluating different training algorithms, emphasizing safety-critical tasks with standardized metrics. Importantly, it retains the original design philosophy of Safety Gym: testing whether agents can train an optimal task policy without sacrificing safety during training, with enhanced environment and task diversity, simulation fidelity, and compatibility with modern RL training pipelines such as Stable-Baselines3 [53]. By situating our experiments in this benchmark, we ensure that our evaluation (i) directly validates the safety–performance separation as stated in Theorem 1, (ii) is grounded in a standardized and widely used testbed for constrained RL, and (iii) provides consistent results across diverse tasks and environments. We expect that our safe RL framework will serve as a state-of-the-art baseline for future work on safe reinforcement learning; because it is directly deployable within the Safety Gymnasium benchmark, it can be used out of the box to conveniently evaluate and compare new methods.

5.1 Experiment Setup

In this section, we introduce the setup of our experiments, including the agent, environment, safety filters used for training and deployment, and baseline comparisons. Full details of the experimental setup, including model architecture, safety specifications, and filter implementation, can be found in Appendix B.

Agent. We use the *Car* agent from Safety Gymnasium. It has continuous control inputs and is implemented as a differential-drive platform with two independently driven parallel wheels and a free-rolling rear wheel, exactly as provided in the benchmark suite [13, 14].

Tasks, rewards, and safety constraints. We evaluate safe RL with the following representative tasks from Safety Gymnasium [13], each with distinct reward functions and safety constraints.

- **Goal:** The robot navigates to a fixed goal position while avoiding contact with cylindrical obstacles and remaining inside a square wall (see Figure 2). Specifically, the reward is proportional to the decrement of the distance between the robot and the goal over a single

timestep, and an additional reward is provided when the robot reaches the goal. The robot is constrained within a square wall that contains four cylindrical obstacles (*i.e.*, *pillars*); the position of the square wall and the pillars remain fixed. Crossing the square wall or colliding with a pillar is considered a safety violation. The robot is randomly initialized inside the square wall, sufficiently far from the pillars and the goal.

- **Circle:** The robot aims to track the boundary of the green circle as fast as possible while remaining inside a square wall (see Figure 2). Specifically, the reward function consists of two factors: (i) a tracking reward that increases as the robot stays closer to the circle boundary, and (ii) a velocity factor that rewards larger tangential velocity along the circle. The robot is constrained within a square wall with a fixed position; crossing the wall is considered a safety violation. The robot is randomly initialized inside the square wall.

Safety filters. We use two types of safety filters in our experiments: one with value-based monitoring [1, Section 3.1] and the other with rollout-based monitoring [1, Section 3.3]. For both filters, we learn a best-effort fallback policy and the associated safety value function with model-free, RL-based HJ reachability analysis [31, 38]. Specifically:

- **Value-based filter** queries, at each timestep, the learned value function to determine whether the current state is safe. If the proposed action sampled from the task policy is deemed safe, we execute it; otherwise, we override it with the best-effort safety fallback.
- **Rollout-based filter** simulates, at each timestep, a state trajectory based on a proposed action sampled from the task policy, followed by a series of actions from the best-effort fallback policy for a fixed horizon. If the state reaches a known terminal safe state (or controlled-invariant safe set) within that horizon, we may execute an action sampled from the task policy; otherwise, we apply the safety fallback action.

While a valid (resolution-complete) safety filter with value-based monitoring can be obtained by solving the safety Bellman equation with dynamic programming (DP) [30], this method does not scale to more than 6 continuous state dimensions. Therefore, we adopt RL-based reachability analysis to obtain *neural approximations* of a perfect safety filter for our experiments, which is, to the best of our knowledge, the only purely *model-free* method that *scalably* approximates the *maximal* controlled-invariant safe set and a fallback policy using a simulator. This makes RL-based reachability analysis the best-suited safety filter synthesis method for Safety Gymnasium benchmark, which provides more than 40 continuous observation dimensions and no information regarding the system dynamics. In subsection 5.2, we show empirically that despite the lack of formal guarantees, a value-based neural safety filter can significantly reduce the frequency of safety violations during training. Furthermore, the rollout-based monitoring, unlike the value-based counterpart, leads to a provably *valid safety filter* for the benchmarking tasks in Safety Gymnasium, since coming to a stop is a known terminal safe state for the *Car* agent. Indeed, as shown in subsection 5.2, this approach achieves *zero safety violations*. Moreover, the task policies trained under both filters converge to the same—and, in some cases, higher—episodic return when compared to the baselines.

Baselines. We compare against two baselines for constrained RL:

- **CPO [16]:** The agent optimizes its return subject to a budget on the environment's constraint costs. We use the same implementation provided by Safety Gymnasium exactly as released.
- **Standard soft actor–critic (SAC):** The agent optimizes its return with no explicit constraints or costs other than episode termination, which occurs immediately after a safety violation. Because episode termination prevents the agent from accruing positive rewards in the future, this setup implicitly encourages the agent to learn policies that satisfy the safety constraints. This is a commonly used heuristic for RL under soft safety constraints [38, 54].

RL algorithms. All task policies are trained with SAC [12] (except the CPO baseline) using the Stable-Baselines3 [53] implementation. Training budgets (environment steps per run), model capacity, and evaluation protocols are kept identical across all methods to ensure a fair comparison.

Reproducibility. Each method is run with *five* independent random seeds. We report the mean and standard error across seeds under a fixed training step budget and a shared evaluation protocol (periodic evaluation episodes with safety violations and episodic returns logged). All code and experimental scripts necessary to reproduce our results will be publicly released upon publication.

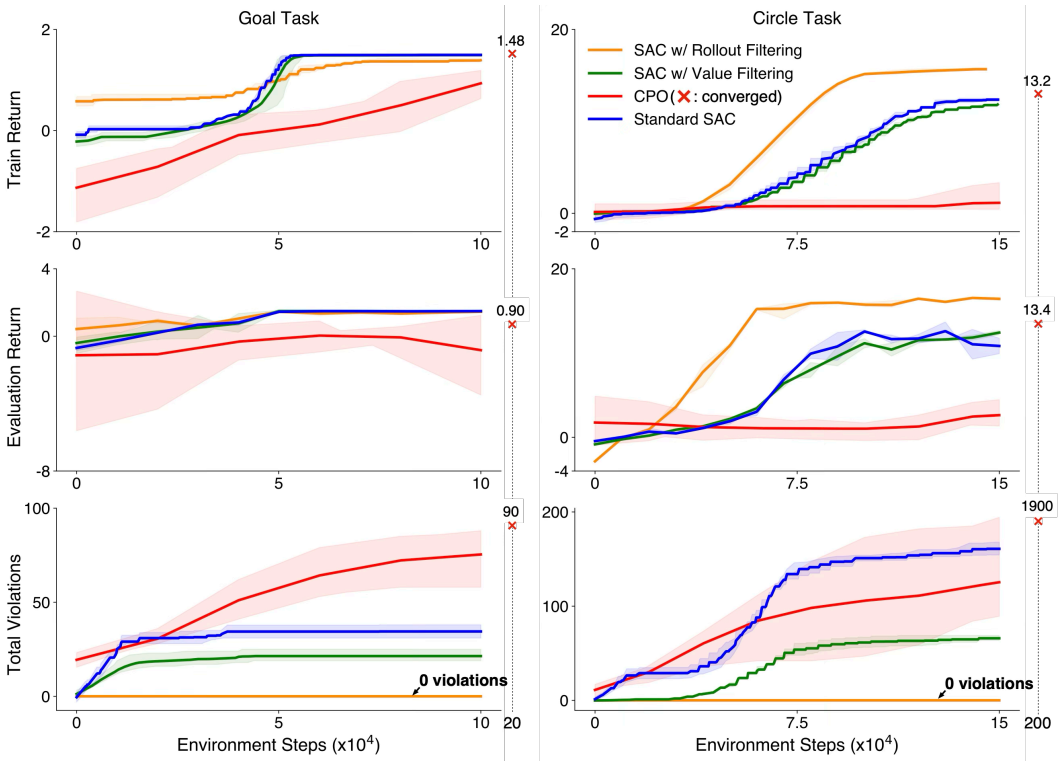

Figure 3: Experimental results on the *Goal* (left) and *Circle* (right) tasks in Safety Gymnasium. We compare our safe RL framework, implemented with an SAC-based task policy and a neural safety filter with rollout- and value-based monitoring, against CPO and standard SAC baselines. *Top*: mean episodic training return versus the number of environment steps. *Middle*: mean episodic evaluation return versus the number of environment steps. *Bottom*: cumulative safety violations versus the number of environment steps. Our method achieves **zero safety violations** when using a valid (rollout-based) safety filter while **converging to the same or higher return than the baselines**. Even with an error-prone approximate (value-based) filter, the number of safety violations is significantly reduced compared to the baselines.

5.2 Results

In this section, we report results in training task policies under a value-based filter and a rollout-based filter, and compare them to the baselines. Specifically, we focus on (i) safety violations during task policy training, (ii) convergence of the RL algorithm, and (iii) optimality of the trained task policy.

Safety during RL training. The number of safety violations accumulated up to a given environment step during task policy training is shown in the last row of Figure 3. Remarkably, our *rollout-based filter recorded zero safety violations* in both tasks across five random seeds. This is expected since the rollout-based filter is a *valid safety filter* with strict guarantees, and it directly supports our theoretical claim of *safe learning* in Theorem 1: given a valid safety filter, system trajectories remain safe throughout task policy training. Although our value-based filter produced a non-zero violation count due to neural approximation error, it nevertheless resulted in fewer violations on both tasks compared to the two baselines. Standard SAC incurred the second-most violations on the *Goal* task and the most on the *Circle* task. We also observe that the cumulative violation curves plateau with training, indicating that although standard SAC cannot enforce safety during training, the learned task policy increasingly avoids violations as the algorithm converges. CPO recorded the most total violations on *Goal* and the second-most on *Circle*. For completeness, we additionally mark CPO's post-convergence metric (symbol "×" in Figure 3), since CPO converges much later than the other methods. Our primary comparisons, however, use counts at a common training budget—100k environment steps for *Goal* and 150k for *Circle*—for all methods.

Convergence of RL training under safety filtering. From Figure 3, we conclude that SAC under both rollout- and value-based filtering, as well as standard SAC, converged well within our training budget. For these three methods, the episodic training return and the evaluation return plateau within the budget on both tasks, and the variance of these metrics remains very small across seeds, indicating convergence over repeated runs. This supports our theoretical claim of *convergence* in Theorem 1: if an RL algorithm converges on an (unconstrained) MDP, then the same algorithm converges on its filtered counterpart. By contrast, CPO takes significantly more environment steps to converge, and its episodic returns exhibit substantially larger variance than the other filtered RL methods.

In the *Circle* task, we observe that training under the *rollout-based* filter converges substantially faster (in environment steps) than the other methods. This acceleration is consistent with prior reports that safety filtering improves sample efficiency by pruning unsafe exploration [8, 23, 24]. A plausible explanation is task geometry and reward shaping: *Circle* rewards encourage high tangential speed while staying close to the circle boundary, which lies inside the square wall. Pushing for higher reward inevitably increases the chance of wall contact under unconstrained exploration; the rollout-based filter preemptively removes those unsafe proposals, concentrating data on productive (safe) trajectories and speeding convergence. By contrast, the effect is less pronounced in the *Goal* task. Depending on the random initialization, a shortest path to the goal can often avoid obstacles and the wall even without interventions, so there is less unsafe exploration to prune. Consequently, while the safety filter still prevents violations, the gain in exploration efficiency—and thus in convergence rate—is smaller on *Goal* than on *Circle*. Characterizing and optimizing the convergence rate for reinforcement learning under safety filtering is beyond the scope of this paper and remains an important direction for future work.

Performance of the task policy. As shown in Figure 3, on both tasks the policies trained under rollout- and value-based filtering attain final performance that matches—or exceeds—standard SAC and CPO. This empirically supports our most significant claim of *optimality under safety filtering* (Theorem 1, Remark 2): enforcing safety with a sufficiently permissive safety filter during training does not degrade the asymptotic performance. In the *Goal* task, training under rollout- and value-based filtering yields nearly identical episodic training and evaluation returns, while CPO achieves a substantially lower return under the same training budget. As a sanity check, we run CPO until convergence and observe that its return eventually matches that of the other methods. In the *Circle* task, training with the rollout-based filter achieves significantly higher return than both the value-based filter and standard SAC; CPO records little meaningful return within budget and, even post-convergence, reaches a level comparable to value-based filtering and standard SAC, yet still below the policy trained with the rollout-based filter. Taken together, these results demonstrate that task policy training under safety filtering attains the same (and, in some cases, higher) task performance as unconstrained baselines, providing concrete empirical evidence for the *safety–performance separation* in safe RL.

6 Conclusion

In this paper, we have established formal convergence and optimality results for reinforcement learning under safety filtering (safe RL). Our theoretical analysis proves that, under an idealized safety filter that is minimally restrictive yet capable of preventing all unsafe actions, reinforcement learning achieves complete safety–performance separation, yielding the same asymptotic performance as direct optimization under hard safety constraints. This result resolves a longstanding misconception in reinforcement learning that enforcing safety inevitably limits the agent's attainable performance. It shows that safety filtering provides a principled mechanism for maintaining both formal safety guarantees and optimal long-term behavior in RL, independent of the specific RL algorithm used for optimizing task performance. Empirical validation on Safety Gymnasium benchmarks further supports our theory, demonstrating that, in practice, the proposed safe RL framework achieves zero safety violations with a valid safety filter, while converging to a policy that matches or surpasses baseline performance. Taken together, these findings provide a rigorous paradigm for safe reinforcement learning: learn your safety filter once, train any RL algorithm with the filter in the loop, and deploy the same filter–policy pair to achieve strict safety and optimal performance in tandem.

Acknowledgments and Disclosure of Funding

This work has been partially supported by NSF CAREER Award 2340851.

References

[1] K. C. Hsu, H. Hu, and J. F. Fisac. "The Safety Filter: A Unified View of Safety-Critical Control in Autonomous Systems". *Annual Review of Control, Robotics, and Autonomous Systems* 7.1 (2024), pp. 47–72.

[2] S. Gros, M. Zanon, and A. Bemporad. "Safe reinforcement learning via projection on a safe set: How to achieve optimality?" *IFAC-PapersOnLine* 53.2 (2020), pp. 8076–8081.

[3] G. Anderson, A. Verma, I. Dillig, and S. Chaudhuri. "Neurosymbolic reinforcement learning with formally verified exploration". *Advances in neural information processing systems* 33 (2020), pp. 6172–6183.

[4] W.-C. Yang, G. Marra, G. Rens, and L. De Raedt. "Safe reinforcement learning via probabilistic logic shields". *Thirty-Second International Joint Conference on Artificial Intelligence*. 2023, pp. 5739–5749.

[5] T. Koller, F. Berkenkamp, M. Turchetta, et al. "Learning-based Model Predictive Control for Safe Exploration and Reinforcement Learning". *arXiv preprint arXiv:1906.12189* (2019).

[6] K. Leung, E. Schmerling, M. Zhang, et al. "On infusing reachability-based safety assurance within planning frameworks for human–robot vehicle interactions". *The International Journal of Robotics Research* 39.10-11 (2020), pp. 1326–1345.

[7] H. Hu, D. Isele, S. Bae, and J. F. Fisac. "Active uncertainty reduction for safe and efficient interaction planning: A shielding-aware dual control approach". *The International Journal of Robotics Research* 43.9 (2024), pp. 1382–1408.

[8] F. P. Bejarano, L. Brunke, and A. P. Schoellig. "Safety Filtering While Training: Improving the Performance and Sample Efficiency of Reinforcement Learning Agents". *IEEE Robotics and Automation Letters* 10.1 (2025), pp. 788–795.

[9] C. J. Watkins and P. Dayan. "Q-learning". *Machine learning* 8 (1992), pp. 279–292.

[10] J. N. Tsitsiklis. "Asynchronous stochastic approximation and Q-learning". *Machine learning* 16.3 (1994), pp. 185–202.

[11] V. Konda and J. Tsitsiklis. "Actor-critic algorithms". *Advances in neural information processing systems* 12 (1999).

[12] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine. "Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor". *International conference on machine learning*. PMLR. 2018, pp. 1861–1870.

[13] J. Ji, B. Zhang, J. Zhou, et al. "Safety gymnasium: A unified safe reinforcement learning benchmark". *Advances in Neural Information Processing Systems* 36 (2023), pp. 18964–18993.

[14] OpenAI. "Safety Gym". `https://openai.com/index/safety-gym/`. 2019.

[15] E. Altman. "Constrained Markov Decision Processes". 1st ed. Routledge, 1999.

[16] J. Achiam, D. Held, A. Tamar, and P. Abbeel. "Constrained policy optimization". *International conference on machine learning*. PMLR. 2017, pp. 22–31.

[17] A. Stooke, J. Achiam, and P. Abbeel. "Responsive safety in reinforcement learning by PID Lagrangian methods". *International Conference on Machine Learning*. PMLR. 2020, pp. 9133–9143.

[18] C. Tessler, D. J. Mankowitz, and S. Mannor. "Reward constrained policy optimization". *arXiv preprint arXiv:1805.11074* (2018).

[19] Y. Chow, O. Nachum, E. Duéñez-Guzmán, and M. Ghavamzadeh. "A Lyapunov-based Approach to Safe Reinforcement Learning". *Advances in Neural Information Processing Systems*. Vol. 31. 2018, pp. 8092–8101.

[20] J. Li, D. Fridovich-Keil, S. Sojoudi, and C. J. Tomlin. "Augmented lagrangian method for instantaneously constrained reinforcement learning problems". *IEEE Conference on Decision and Control (CDC)*. IEEE. 2021, pp. 2982–2989.

[21] Y. Chow, M. Ghavamzadeh, L. Janson, and M. Pavone. "Risk-constrained reinforcement learning with percentile risk criteria". *Journal of Machine Learning Research* 18.167 (2018), pp. 1–51.

[22] K. P. Wabersich and M. N. Zeilinger. "Linear model predictive safety certification for learning-based control". *IEEE Conference on Decision and Control (CDC)*. IEEE. 2018, pp. 7130–7135.

[23] K. P. Wabersich and M. N. Zeilinger. "A predictive safety filter for learning-based control of constrained nonlinear dynamical systems". *Automatica* 129 (2021), p. 109597.

[24] O. Bastani. "Safe reinforcement learning with nonlinear dynamics via model predictive shielding". *American control conference (ACC)*. IEEE. 2021, pp. 3488–3494.

[25] A. D. Ames, X. Xu, J. W. Grizzle, and P. Tabuada. "Control barrier function based quadratic programs for safety critical systems". *IEEE Transactions on Automatic Control* 62.8 (2016), pp. 3861–3876.

[26] A. D. Ames, S. Coogan, M. Egerstedt, et al. "Control barrier functions: Theory and applications". *European control conference (ECC)*. IEEE. 2019, pp. 3420–3431.

[27] A. Robey, H. Hu, L. Lindemann, et al. "Learning control barrier functions from expert demonstrations". *IEEE Conference on Decision and Control (CDC)*. IEEE. 2020, pp. 3717–3724.

[28] D. D. Oh, D. Lee, and H. J. Kim. "Safety-critical control under multiple state and input constraints and application to fixed-wing UAV". *IEEE Conference on Decision and Control (CDC)*. IEEE. 2023, pp. 1748–1755.

[29] I. M. Mitchell, A. M. Bayen, and C. J. Tomlin. "A time-dependent Hamilton-Jacobi formulation of reachable sets for continuous dynamic games". *IEEE Transactions on automatic control* 50.7 (2005), pp. 947–957.

[30] S. Bansal, M. Chen, S. Herbert, and C. J. Tomlin. "Hamilton-Jacobi reachability: A brief overview and recent advances". *IEEE 56th Annual Conference on Decision and Control (CDC)*. IEEE. 2017, pp. 2242–2253.

[31] J. F. Fisac, N. F. Lugovoy, V. Rubies-Royo, et al. "Bridging Hamilton-Jacobi Safety Analysis and Reinforcement Learning". *International Conference on Robotics and Automation (ICRA)*. IEEE. 2019, pp. 8550–8556.

[32] S. Bansal and C. J. Tomlin. "Deepreach: A deep learning approach to high-dimensional reachability". *IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2021, pp. 1817–1824.

[33] K.-C. Hsu, D. P. Nguyen, and J. F. Fisac. "ISAACS: Iterative Soft Adversarial Actor-Critic for Safety". *Annual Learning for Dynamics and Control Conference (L4DC)*. PMLR. 2023, pp. 90–103.

[34] D. P. Nguyen, K.-C. Hsu, W. Yu, et al. "Gameplay Filters: Robust Zero-Shot Safety through Adversarial Imagination". *Conference on Robot Learning*. PMLR. 2025, pp. 387–407.

[35] J. Wang, H. Hu, D. P. Nguyen, and J. F. Fisac. "MAGICS: Adversarial RL with Minimax Actors Guided by Implicit Critic Stackelberg for Convergent Neural Synthesis of Robot Safety". *Workshop on the Algorithmic Foundations of Robotics (WAFR)*. 2024.

[36] K. Nakamura, L. Peters, and A. Bajcsy. "Generalizing Safety beyond Collision-Avoidance via Latent-Space Reachability Analysis". *Robotics: Science and Systems (RSS)*. 2025.

[37] J. J. Choi, J. J. Aloor, J. Li, et al. "Resolving Conflicting Constraints in Multi-Agent Reinforcement Learning with Layered Safety". *Robotics: Science and Systems (RSS)*. 2025.

[38] D. D. Oh, J. Lidard, H. Hu, et al. "Safety with Agency: Human-Centered Safety Filter with Application to AI-Assisted Motorsports". *Robotics: Science and Systems (RSS)*. 2025.

[39] M. Alshiekh, R. Bloem, R. Ehlers, et al. "Safe reinforcement learning via shielding". *AAAI conference on artificial intelligence*. Vol. 32. 1. 2018.

[40] S. Carr, N. Jansen, S. Junges, and U. Topcu. "Safe reinforcement learning via shielding under partial observability". *AAAI conference on artificial intelligence*. Vol. 37. 12. 2023, pp. 14748–14756.

[41] R. Cheng, G. Orosz, R. M. Murray, and J. W. Burdick. "End-to-end safe reinforcement learning through barrier functions for safety-critical continuous control tasks". *AAAI conference on artificial intelligence*. Vol. 33. 01. 2019, pp. 3387–3395.

[42] Y. Emam, G. Notomista, P. Glotfelter, et al. "Safe Reinforcement Learning Using Robust Control Barrier Functions". *IEEE Robotics and Automation Letters* 10.3 (2025), pp. 2886–2893.

[43] M. Ganai, Z. Gong, C. Yu, et al. "Iterative reachability estimation for safe reinforcement learning". *Advances in Neural Information Processing Systems* 36 (2023), pp. 69764–69797.

[44] N. Kochdumper, H. Krasowski, X. Wang, et al. "Provably safe reinforcement learning via action projection using reachability analysis and polynomial zonotopes". *IEEE Open Journal of Control Systems* 2 (2023), pp. 79–92.

[45] N. Hunt, N. Fulton, S. Magliacane, et al. "Verifiably safe exploration for end-to-end reinforcement learning". *International Conference on Hybrid Systems: Computation and Control*. 2021, pp. 1–11.

[46] H. Markgraf, S. Sawant, H. Krasowski, et al. "Safe Reinforcement Learning using Action Projection: Safeguard the Policy or the Environment?" *arXiv preprint arXiv:2509.12833* (2025).

[47] Y. Sui, A. Gotovos, J. Burdick, and A. Krause. "Safe exploration for optimization with Gaussian processes". *International conference on machine learning*. PMLR. 2015, pp. 997–1005.

[48] A. K. Akametalu, J. F. Fisac, J. H. Gillula, et al. "Reachability-based safe learning with Gaussian processes". *IEEE Conference on Decision and Control*. IEEE. 2014, pp. 1424–1431.

[49] F. Berkenkamp, M. Turchetta, A. Schoellig, and A. Krause. "Safe model-based reinforcement learning with stability guarantees". *Advances in neural information processing systems* 30 (2017).

[50] S. M. Richards, F. Berkenkamp, and A. Krause. "The Lyapunov neural network: Adaptive stability certification for safe learning of dynamical systems". *Conference on robot learning*. PMLR. 2018, pp. 466–476.

[51] L. Hewing, K. P. Wabersich, M. Menner, and M. N. Zeilinger. "Learning-based model predictive control: Toward safe learning in control". *Annual Review of Control, Robotics, and Autonomous Systems* 3.1 (2020), pp. 269–296.

[52] F. Blanchini. "Set invariance in control". *Automatica* 35.11 (1999), pp. 1747–1767.

[53] A. Raffin, A. Hill, A. Gleave, et al. "Stable-Baselines3: Reliable reinforcement learning implementations". *Journal of machine learning research* 22.268 (2021), pp. 1–8.

[54] A. Faust, K. Oslund, O. Ramirez, et al. "PRM-RL: Long-range robotic navigation tasks by combining reinforcement learning and sampling-based planning". *IEEE international conference on robotics and automation (ICRA)*. IEEE. 2018, pp. 5113–5120.

A Theoretical Results and Proofs

Lemma 1 (No all-time safety outside the maximal invariant set)**.** *Let $\Omega^* \subseteq \mathcal{S} \setminus \mathcal{F}$ be the maximal controlled-invariant safe set. Then, for any stationary policy π and any $s \in \Omega^{*c} \cap (\mathcal{S} \setminus \mathcal{F})$,*

$$\Pr_{\pi,\mathbb{P}}\big[s_t \notin \mathcal{F},\, \forall t \;\big|\; s_0 = s\big] \;<\; 1.$$

In words, starting outside Ω^*, no stationary policy can satisfy the all-time safety constraint with probability 1.

Proof. We prove by contradiction. Assume there exists $s \in \Omega^{*c} \cap (\mathcal{S} \setminus \mathcal{F})$ and a stationary policy π such that $\Pr_{\pi,\mathbb{P}}[s_t \notin \mathcal{F},\, \forall t \mid s_0 = s] = 1$. Define

$$R_\pi \;:=\; \Big\{x \in \mathcal{S} \setminus \mathcal{F} :\; \Pr_{\pi,\mathbb{P}}\big[s_t \notin \mathcal{F},\, \forall t \;\big|\; s_0 = x\big] = 1\Big\}.$$

By the Markov property, if $x \in R_\pi$, then the set of next possible states rendered by π and $\mathbb{P}$ should be a subset of R_π. More formally, for all $a \in \operatorname{supp}\pi(\cdot \mid x)$, $\operatorname{supp}\mathbb{P}(\cdot \mid x, a) \subseteq R_\pi$. Thus, R_π is a controlled-invariant subset of $\mathcal{S} \setminus \mathcal{F}$ under π, and since there exists $s \in R_\pi \setminus \Omega^*$, the set $\Omega^* \cup R_\pi$ is a strictly larger controlled-invariant safe set than Ω^*. This contradicts the maximality of Ω^*. Therefore, no such s exists. □

Proposition 1 (Maximality of the admissible policy set)**.** *Let $\Omega^* \subseteq \mathcal{S} \setminus \mathcal{F}$ be the maximal controlled-invariant safe set, and define $\mathcal{A}_{\text{safe}}(s) := \{a \in \mathcal{A} : \operatorname{supp}\mathbb{P}(\cdot \mid s, a) \subseteq \Omega^*\}$ for all $s \in \Omega^*$. Consider the set of measurable, stationary policies*

$$\Pi_{\text{safe}} \coloneqq \Big\{\pi :\; \operatorname{supp}\pi(\cdot \mid s) \subseteq \mathcal{A}_{\text{safe}}(s),\, \forall s \in \Omega^*\Big\}.$$

Then Π_{safe} is the largest collection of policies that satisfy the all-time safety constraint from every initial state in Ω^*.

Proof. (Π_{safe} *policies are admissible.*) Let $\pi \in \Pi_{\text{safe}}$. For any $s \in \Omega^*$ and any $a \in \operatorname{supp}\pi(\cdot \mid s)$, we have $\operatorname{supp}\mathbb{P}(\cdot \mid s, a) \subseteq \Omega^*$. By induction, we get $\Pr_{\pi,\mathbb{P}}\big[s_t \notin \mathcal{F},\, \forall t \;\big|\; s_0 = s\big] = 1$.

(*Policies not in Π_{safe} are not admissible.*) Let $\pi \notin \Pi_{\text{safe}}$. Then there exist $s \in \Omega^*$ and $a \in \operatorname{supp}\pi(\cdot \mid s)$ such that $a \notin \mathcal{A}_{\text{safe}}(s)$, so $\operatorname{supp}\mathbb{P}(\cdot \mid s, a) \nsubseteq \Omega^*$. Hence, with positive probability, the next state rendered by π and $\mathbb{P}$ lies in Ω^{*c}. By Lemma 1, the overall probability of ever entering $\mathcal{F}$ starting from s under π is positive. Thus π violates the all-time safety constraint from s.

Maximality of Π_{safe} follows immediately: any stationary policy assigning positive mass to an action outside $\mathcal{A}_{\text{safe}}(s)$ at some $s \in \Omega^*$ cannot satisfy the all-time safety constraint from that state. □

Theorem (Restatement of Theorem 1)**.** *If Assumption 1 holds, then the following claims on the safety, convergence, and optimality of RL under safety filtering are true:*

1. ***Safe learning.*** *For any sequence of task policies produced by any RL algorithm during training, the filtered trajectories remain in Ω^* for all time.*
2. ***Convergence.*** *Let* ALG *be an RL algorithm that converges on stationary discounted MDPs with bounded rewards. Then, for any such MDP $\mathcal{M}$,* ALG *also converges on the corresponding filtered MDP $\mathcal{M}_\phi$.*
3. ***Optimality under safety filtering.*** *Let π_ϕ^ε denote a measurable and stationary ε-optimal policy on $\mathcal{M}_\phi$, possibly returned by* ALG*, for some $\varepsilon > 0$:*

   $$V_{\mathcal{M}_\phi}^{\pi_\phi^\varepsilon}(s) \geq V_{\mathcal{M}_\phi}^*(s) - \varepsilon, \qquad \forall s \in \Omega^*.$$

   We define the executed policy π_{exec} *as the pushforward of π_ϕ^ε by the map $a \mapsto \phi(s, a)$; i.e., for all Borel measurable sets $B \subseteq \mathcal{A}$,*

   $$\pi_{\text{exec}}(B \mid s) \coloneqq \pi_\phi^\varepsilon\big(\{a \in \mathcal{A} :\; \phi(s, a) \in B\} \mid s\big), \qquad \forall s \in \Omega^*.$$

   Then, π_{exec} is a safe ε-optimal policy on $\mathcal{M}_{\text{SC}}$:

   $$V_{\mathcal{M}_{\text{SC}}}^{\pi_{\text{exec}}}(s) \;\geq\; V_{\mathcal{M}_{\text{SC}}}^*(s) - \varepsilon, \qquad \forall s \in \Omega^*.$$

Proof. (*Safe learning*) Since all training episodes begin in Ω^* (Assumption 1) and $\phi(s,a)$ renders Ω^* invariant for all $a \in \mathcal{A}$ produced by any task policy (Definition 2, Assumption 1), all filtered trajectories remain in Ω^* for all time.

(*Convergence*) From Assumption 1 and Definition 3, both $\mathbb{P}_\phi$ and r_ϕ are time-invariant and r_ϕ is bounded. Thus, the filtered MDP $\mathcal{M}_\phi$ is a stationary discounted MDP with bounded rewards. The iterates of ALG depend only on the executed tuples

$$(s_t, a_t, r_t, s_{t+1}) \quad \text{with} \quad r_t = r_\phi(s_t, a_t), \quad s_{t+1} \sim \mathbb{P}_\phi(\cdot \mid s_t, a_t),$$

and on the stepsize/exploration schedule of ALG. Every step in the convergence proof of ALG on $\mathcal{M}$ applies to $\mathcal{M}_\phi$ with the substitution $(\mathbb{P}, r) \mapsto (\mathbb{P}_\phi, r_\phi)$. Therefore ALG converges on $\mathcal{M}_\phi$. This carry-over presumes the same algorithmic side conditions—e.g., stepsize conditions and ergodicity/coverage under the executed policy—also hold in $\mathcal{M}_\phi$. It covers, for example, almost-sure convergence of tabular Q-learning [9, 10], two-timescale actor–critic [11], and soft policy-iteration convergence for SAC [12].

(*Optimality under safety filtering*) We begin with showing the equivalence between $V^*_{\mathcal{M}_\phi}$ and $V^*_{\mathcal{M}_{\mathrm{SC}}}$. We define the optimal Bellman operators for the filtered MDP $\mathcal{M}_\phi$ and the SC-MDP $\mathcal{M}_{\mathrm{SC}}$, respectively:

$$\begin{aligned}
(\mathcal{T}^*_{\mathcal{M}_\phi} V)(s) &= \sup_{a \in \mathcal{A}} \Big\{ r_\phi\big(s,a\big) + \gamma\, \mathbb{E}_{s' \sim \mathbb{P}_\phi(\cdot|s,a)}[V(s')] \Big\} \\
&= \sup_{a \in \mathcal{A}} \Big\{ r\big(s, \phi(s,a)\big) + \gamma\, \mathbb{E}_{s' \sim \mathbb{P}(\cdot|s,\phi(s,a))}[V(s')] \Big\}, \\
(\mathcal{T}^*_{\mathcal{M}_{\mathrm{SC}}} V)(s) &= \sup_{a \in \mathcal{A}_{\mathrm{safe}}(s)} \Big\{ r(s,a) + \gamma\, \mathbb{E}_{s' \sim \mathbb{P}(\cdot|s,a)}[V(s')] \Big\}.
\end{aligned}$$

From Definition 2, the image of the map $a \mapsto \phi(s,a)$ is exactly $\mathcal{A}_{\mathrm{safe}}(s)$ and ϕ is the identity on that image. Taking the supremum over $a \in \mathcal{A}$ post-composition by ϕ equals the supremum over $a \in \mathcal{A}_{\mathrm{safe}}(s)$. Therefore, we have

$$\mathcal{T}^*_{\mathcal{M}_\phi} = \mathcal{T}^*_{\mathcal{M}_{\mathrm{SC}}}, \qquad \text{pointwise on } \Omega^*.$$

Since both operators are γ-contractions on $(\mathcal{B}_b(\Omega^*), \|\cdot\|_\infty)$—the Banach space of bounded Borel-measurable value functions $V : \Omega^* \to \mathbb{R}$ endowed with the sup norm—they have the same unique fixed point:

$$V^*_{\mathcal{M}_\phi}(s) = V^*_{\mathcal{M}_{\mathrm{SC}}}(s), \qquad \forall s \in \Omega^*.$$

We now show the equivalence between $V^{\pi^\varepsilon_\phi}_{\mathcal{M}_\phi}$ and $V^{\pi_{\mathrm{exec}}}_{\mathcal{M}_{\mathrm{SC}}}$. We unroll the expectation and apply the definition of the filtered MDP $\mathcal{M}_\phi$ (Definition 3):

$$\begin{aligned}
V^{\pi^\varepsilon_\phi}_{\mathcal{M}_\phi}(s) &= \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t\, r_\phi(s_t, a_t) \;\middle|\; s_0 = s,\ a_t \sim \pi^\varepsilon_\phi(\cdot \mid s_t),\ s_{t+1} \sim \mathbb{P}_\phi(\cdot \mid s_t, a_t) \right] \\
&= \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t\, r\big(s_t, \phi(s_t, a_t)\big) \;\middle|\; s_0 = s,\ a_t \sim \pi^\varepsilon_\phi(\cdot \mid s_t),\ s_{t+1} \sim \mathbb{P}\big(\cdot \mid s_t, \phi(s_t, a_t)\big) \right].
\end{aligned}$$

Let $\tilde{a}_t \coloneqq \phi(s_t, a_t)$, where $a_t \sim \pi^\varepsilon_\phi(\cdot \mid s_t)$. The executed policy π_{exec} is defined as the pushforward of the ε-optimal policy π^ε_ϕ by the map $a \mapsto \phi(s,a)$. Therefore, $\tilde{a}_t \sim \pi_{\mathrm{exec}}(\cdot \mid s_t)$, and we have

$$\begin{aligned}
V^{\pi^\varepsilon_\phi}_{\mathcal{M}_\phi}(s) &= \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t\, r(s_t, \tilde{a}_t) \;\middle|\; s_0 = s,\ \tilde{a}_t \sim \pi_{\mathrm{exec}}(\cdot \mid s_t),\ s_{t+1} \sim \mathbb{P}(\cdot \mid s_t, \tilde{a}_t) \right] \\
&= V^{\pi_{\mathrm{exec}}}_{\mathcal{M}}(s), \qquad \forall s \in \Omega^*.
\end{aligned}$$

By the perfect safety filter property (Definition 2) and the pushforward definition of π_{exec}, we have $\operatorname{supp} \pi_{\mathrm{exec}}(\cdot \mid s) \subseteq \mathcal{A}_{\mathrm{safe}}(s)$ for all $s \in \Omega^*$. Therefore, $\pi_{\mathrm{exec}} \in \Pi_{\mathrm{safe}}$, and this gives

$$V^{\pi_{\mathrm{exec}}}_{\mathcal{M}}(s) = V^{\pi_{\mathrm{exec}}}_{\mathcal{M}_{\mathrm{SC}}}(s), \qquad \forall s \in \Omega^*.$$

Finally, combining the equalities yields

$$V^{\pi_{\mathrm{exec}}}_{\mathcal{M}_{\mathrm{SC}}}(s) = V^{\pi^\varepsilon_\phi}_{\mathcal{M}_\phi}(s) \;\geq\; V^*_{\mathcal{M}_\phi}(s) - \varepsilon = V^*_{\mathcal{M}_{\mathrm{SC}}}(s) - \varepsilon, \qquad \forall s \in \Omega^*.$$

This proves the ε-optimality and the safety of π_{exec} on $\mathcal{M}_{\mathrm{SC}}$. □

B Implementation Details

Model architecture. The SAC task policy and safety policy have the same architecture, with the actor and critic policies implemented by a fully connected feedforward neural network with 2 hidden layers of 256 neurons. The CPO task policy has 2 hidden layers of 64 neurons. All policies use *ReLU* activations for SAC-based architectures and *Tanh* activations for CPO.

The safety policy is trained with a learning rate 1×10^{-5}, replay buffer size of 2×10^5, batch size of 256, discount factor $\gamma = 0.995$, and soft update coefficient $\tau = 0.01$, for a total of 2×10^6 steps.

The task policy SAC is trained with a learning rate 3×10^{-4}, replay buffer size of 1×10^5, batch size of 256, $\gamma = 0.99$, and $\tau = 0.01$.

The CPO policy is trained with a learning rate of 3×10^{-4}, $\gamma = 0.99$, and KL-divergence step size of $\delta_{\text{KL}} = 0.01$.

Safety filter implementation. For rollout filtering, we adopt the rollout evaluation procedure described in Nguyen et al. [34]. We use a finite rollout horizon of $H = 100$ and define the target margin function $l(s)$ and stop policy π_{stop} as

$$l(s) = \eta - \sqrt{v_x^2 + v_y^2}, \qquad \pi_{\text{stop}}(a \mid s) = \mathbf{0}$$

where η denotes the target safety velocity threshold. Although the safety policy was not explicitly trained to bring the robot to a complete stop, it implicitly learned to reduce the robot's velocity near obstacles, either by slowing down to a complete stop or rotating in place until the obstacles are out of sight. This emergent behavior motivates the design of the above target margin function and stop policy as the fallback policy when $l(s) > 0$. We choose $\eta = 0.01$ in our experiments.

For value-based filtering, we run the experiment across a sweep of ϵ values, and choose the ϵ with the highest return and lowest total violations. Specifically, we choose $\epsilon_{\text{goal}} = 0.4$, and $\epsilon_{\text{circle}} = 0.1$.

State and action spaces. The *Circle* task and the *Goal* tasks each employ ego-centric proprioceptive observations, consisting of accelerometer, velocity, angular rate, magnetic field, rear ball rotation, and local LiDAR readings. The observation space has 40 and 72 dimensions, respectively, while the continuous action space is similar across both tasks:

$$\begin{aligned}
s_{\textit{Goal}} &:= \Big[a_{x,y,z}, v_{x,y,z},\ \omega_{x,y,z}, m_{x,y,z}\ \dot{b}_{x,r}, \dot{b}_{y,r}, \dot{b}_{z,r},\ q_{b,r}^{(3\times3)},\ \ell_{1:16}^{\text{goal}},\ \ell_{1:16}^{\text{pillar}},\ \ell_{1:16}^{\text{wall}}\Big], \\
s_{\textit{Circle}} &:= \Big[a_{x,y,z}, v_{x,y,z},\ \omega_{x,y,z}, m_{x,y,z}\ \dot{b}_{x,r}, \dot{b}_{y,r}, \dot{b}_{z,r},\ q_{b,r}^{(3\times3)},\ \ell_{1:16}^{\text{sigwall}}\Big], \\
a &:= \Big[\tau_L, \tau_R\Big],
\end{aligned}$$

where $(a_{x,y,z})$, $(v_{x,y,z})$, and $(\omega_{x,y,z})$ denote accelerometer, velocimeter, and gyroscope readings; $(m_{x,y,z})$ the magnetometer readings; $\dot{b}_{(\cdot),r}$ the rear ball angular velocity; $q_{b,r}^{(3\times3)}$ the rear ball orientation matrix; and $\ell_{1:16}^{(\cdot)}$ the 16-beam LiDAR signals for different object classes (goal, wall, pillar). The action vector $a = [\tau_L, \tau_R]$ applies continuous torques to the left and right wheels, each bounded within $[-1, 1]$.

Goal task margin function. Let $z_i^{\text{pillar}}, z_i^{\text{wall}} \in [0, 1]$ denote 16-beam LiDAR *proximities* (1 = very close, 0 = far) for pillars and walls, respectively. With LiDAR max range $R > 0$ and safety clearance $\delta > 0$, convert to distances

$$d_i^{\text{pillar}} = (1 - z_i^{\text{pillar}})\, R, \qquad d_i^{\text{wall}} = (1 - z_i^{\text{wall}})\, R,$$

and define the minimum distance to safety-critical objects

$$d_{\min}(s) \ = \ \min_i \Big\{\, d_i^{\text{pillar}},\, d_i^{\text{wall}} \,\Big\}.$$

The margin is

$$g_{\text{goal}}(s) \;=\; d_{\min}(s) \;-\; \delta.$$

In our experiments we use $R = 3.0\,\text{m}$ and $\delta = 0.02\,\text{m}$. The episode terminates if $g_{\text{goal}}(s) < 0$ or upon explicit collision.

Circle task margin function. Let $x(s) \in \mathbb{R}^2$ be the robot planar position and $\mathcal{W} = \{W_k\}$ the set of signal-wall line segments (axis-aligned). Define the point–segment distance

$$d(x, W_k) \;=\; \min_{p \in W_k} \|x - p\|_2, \qquad d_{\min}(s) \;=\; \min_k d\big(x(s), W_k\big).$$

The margin is

$$g_{\text{circle}}(s) \;=\; d_{\min}(s) \;-\; \delta.$$

We use $R = 6.0\,\text{m}$ and $\delta = 0.02\,\text{m}$. The episode terminates if $g_{\text{circle}}(s) < 0$ or upon explicit collision.