Safety-Polarized and Prioritized Reinforcement Learning

Ke Fan [*1] **Jinpeng Zhang** [*1] **Xuefeng Zhang** [2] **Yunze Wu** [3] **Jingyu Cao** [3] **Yuan Zhou** [4 1 5] **Jianzhu Ma** [6 7]

Abstract

Motivated by the first priority of safety in many real-world applications, we propose MAXSAFE, a chance-constrained bi-level optimization framework for safe reinforcement learning. MAXSAFE first minimizes the unsafe probability and then maximizes the return among the safest policies. We provide a tailored Q-learning algorithm for the MAXSAFE objective, featuring a novel learning process for optimal action masks with theoretical convergence guarantees. To enable the application of our algorithm to large-scale experiments, we introduce two key techniques: *safety polarization* and *safety prioritized experience replay*. Safety polarization generalizes the optimal action masking by polarizing the Q-function, which assigns low values to unsafe state-action pairs, effectively discouraging their selection. In parallel, safety prioritized experience replay enhances the learning of optimal action masks by prioritizing samples based on temporal-difference (TD) errors derived from our proposed state-action reachability estimation functions. This approach efficiently addresses the challenges posed by sparse cost signals. Experiments on diverse autonomous driving and safe control tasks show that our methods achieve near-maximal safety and an optimal reward-safety trade-off.

1. Introduction

Safety is a critical bottleneck for deploying reinforcement learning (RL) algorithms in real-world applications due to the catastrophic consequences of unsafe decisions, such as crashes in autonomous driving (Leurent & Mercat, 2019). In such scenarios, safety takes precedence over all other objectives, and RL algorithms must prioritize achieving *maximal* safety (Gu et al., 2024; García & Fernández, 2015). A widely adopted framework for safe RL is the Constrained Markov Decision Process (CMDP) (Altman, 1999; Achiam et al., 2017; Tessler et al., 2019; Ray et al., 2019; Yang et al., 2022), which enforces safety by constraining the expected cumulative safety cost to remain below a predefined budget. While this framework provides a practical way to manage safety constraints, it is inherently limited to ensuring a user-specified safety level rather than learning the optimal safety cost. Furthermore, CMDP-based algorithms typically rely on dense cost signals to guide the agent's behavior, making them ill-suited for scenarios where safety costs are sparse but highly consequential. To address these limitations, our work focuses on developing RL algorithms that maximize cumulative rewards while ensuring the unsafe probability is minimized.

There are two primary methodologies in the literature for providing frequent safety guidance to learning agents: action correction and action masking. Action correction modifies unsafe actions by either replacing them with safe ones based on a predefined backup policy or projecting them back to the safe set (Alshiekh et al., 2017; Zhang et al., 2023). However, this approach often leads to suboptimal rewards due to the over-conservatism of the backup policy or projection. In contrast, action masking excludes unsafe actions, allowing agents to explore safe options and potentially achieve optimal rewards while maintaining maximal safety. Despite its promise, designing effective action-masking procedures remains challenging. For instance, Srinivasan et al. (2020) masks actions with unsafe probabilities exceeding a fixed threshold ε. However, this uniform, state-agnostic threshold may fail to differentiate actions with varying unsafe probabilities, limiting the policy's ability to learn the safest actions and compromising safety performance.

In this work, we propose a novel chance-constrained bi-level optimization framework, namely MAXSAFE, for the

[*]Equal contribution [1]Department of Mathematical Sciences, Tsinghua University, Beijing, China [2]Institute for Artificial Intelligence, Peking University, Beijing, China [3]Department of Computer Science and Engineering, Shanghai Jiao Tong University, Shanghai, China [4]Yau Mathematical Sciences Center, Tsinghua University, Beijing, China [5]Beijing Institute of Mathematical Sciences and Applications, Beijing, China [6]Department of Electronic Engineering, Tsinghua University, Beijing, China [7]Institute for AI Industry Research, Tsinghua University, Beijing, China. Correspondence to: Yuan Zhou <yuan-zhou@tsinghua.edu.cn>, Jianzhu Ma <majianzhu@tsinghua.edu.cn>.

Proceedings of the 42^{nd} International Conference on Machine Learning, Vancouver, Canada. PMLR 267, 2025.

maximal-safety RL problem. In our framework, we mask the actions where the estimated unsafe probability is above a *state-dependent* safety threshold $\zeta(s)$ that is learned from the interaction with the environment. Our estimated unsafe probability is the *undiscounted* safety cost of the *optimal* policy, which is more aligned with the definition of the unsafe probability compared to the discounted cost used in Srinivasan et al. (2020). This undiscounted safety cost is learned by our proposed state-action dependent reachability estimation (namely the SA-REF backward induction), which is adapted from the techniques in Ganai et al. (2023). We show theoretically that our SA-REF backward induction converges to the unsafe probability of the optimal policy (Theorem 4.3). We then show that jointly learning the policy and the action mask rules (including the unsafe probability and the state-dependent safety threshold) yields the optimal policy (Theorem 4.4).

To enable applications in large-scale safety-critical RL tasks, we propose a soft masking technique called *safety polarization*, based on a pre-selected polarization function f_{pol} . This technique reduces the Q-function values used for action selection under unsafe state-action pairs, with the decrement in Q-value determined by the unsafe probability and a state-dependent threshold. To better address the sparsity of safety cost signals, we incorporate techniques inspired by Prioritized Experience Replay (PER) in Q-learning (Schulman et al., 2017), originally designed for sparse reward signals. Specifically, we use the TD error of our proposed SA-REF to prioritize samples in the replay buffer. This prioritization enables the algorithm to focus on safety-critical transitions. It enhances the learning of unsafe probabilities and improves the safety performance of the algorithm. We conduct extensive experiments on autonomous driving and safe control tasks, demonstrating that our proposed algorithms, **SPOM** and **SPOM_PER**, achieve superior safety and the best reward-safety trade-off among state-of-the-art safe RL methods (Section 6).

2. Related Work

Constrained RL by CMDP. The classical framework addressing safety constraints in RL is the CMDP (Altman, 1999), which aims to maximize the expected reward while keeping the expected cost below a predetermined threshold. Primal-dual approaches use Lagrangian relaxation to transform the original constrained optimization problem into an unconstrained one, as in PPOLag (Ray et al., 2019) and RCPO (Tessler et al., 2019). Trust region methods perform local policy improvements within the constrained region, such as CPO (Achiam et al., 2017), FOCOPS (Zhang et al., 2020), and CUP (Yang et al., 2022). Recent studies have incorporated Hamilton-Jacobi reachability (Bansal et al., 2017) into the CMDP framework to identify the largest feasible set, thereby enhancing safe policy optimization, as in RCRL (Yu et al., 2022a) and RESPO (Ganai et al., 2023).

Action-correction-based Safe RL. The first line of works replace the detected unsafe action to a safe one instructed by a backup policy, e.g., a shielding policy via formal methods (Alshiekh et al., 2017; Anderson et al., 2020), a recovery policy (Thananjeyan et al., 2020) that recovers the agent back to safe states, an intervention policy (Wagener et al., 2021), a safety editor policy (Yu et al., 2022b), etc. Another line of works project the unsafe action back to the safe set, e.g., the projection based on control barrier function (Cheng et al., 2019) with known dynamics, the Unrolling Safety Layer (Zhang et al., 2023), the Reduced Policy Optimization method (Ding et al., 2023), the Barrier Certificate method (Yang et al., 2023), etc. In general, the performance of the above methods might be sub-optimal due to the over-conservative nature of the backup policy or the projection operation.

Action Masking for Safe RL. A large body of works use specific assumptions or prior knowledge to build action masks (Krasowski et al., 2023) for safe RL. Fulton & Platzer (2018; 2019) construct the action masks based on theorem proving of differential dynamic logic specifications (Platzer, 2008), which require the knowledge of the system dynamics. Kalweit et al. (2020) propose Constrained Q-Learning where the action masking is done via a set of one-step cost constraints that query the dynamics to check whether the next state is inside the constraints. Huang & Ontanón (2022) give a more detailed analysis of the effects of action masking under the context of policy gradient algorithms. For specific tasks, there are works studying action masking for autonomous driving (Mirchevska et al., 2018; Brosowsky et al., 2021; Krasowski et al., 2022) and traffic light control (Muller & Sabatelli, 2022) to ensure safety. Most of the above works require domain knowledge that is not always accessible. In terms of learning approach, the typical example is the Safety Q-functions for RL (SafeQ) proposed in Srinivasan et al. (2020) and also in Tan et al. (2024), which is then extended to other settings, e.g., safe exploration (Bharadhwaj et al., 2021). As we will show, SafeQ might fail to achieve the minimal unsafe probability.

Hamilton-Jacobi Reachability and Reachability Estimation Function. Early works employ Hamilton-Jacobi(HJ) reachability value functions to assess state feasibility, relying on known system dynamics and numerical methods (Ganai et al., 2024). Some studies approximate unknown dynamics using Gaussian Processes (Zhao et al., 2023) or symbolic regression (Wang & Zhu, 2024). Once reachability is computed, the state space is partitioned into feasible/infeasible regions to guide policy optimization (Zheng et al., 2024). Other works focus on formal safety verification of DRL systems (Dong et al., 2024). However,

HJ-based value functions are not well-suited for stochastic MDPs during RL training. To address this, reachability estimation function(REF) estimates unsafe probabilities via backward reduction. Our work extends REF to *the state-action level*, enabling state-dependent action masking to reduce safety violations.

Prioritized Experience Replay for Off-policy RL. Prioritized Experience Replay (PER) is a widely adopted technique in RL, designed to address the inefficiency of uniform sampling in replay buffers. Initially introduced by Schulman et al. (2017), PER assigns priorities to transitions based on their TD errors, enabling agents to learn more effectively from high-impact experiences. Variants of PER have been explored, such as annealing the prioritization exponent over time (Horgan et al., 2018) and incorporating multi-step returns into the prioritization process (Hessel et al., 2018) to improve stability and learning efficiency. In our work, rather than applying prioritization to the learning of the Q-function, we tailor the prioritization process to the learning of our proposed SA-REF function, addressing the challenges posed by sparse and critical cost signals.

3. Preliminaries

In safe RL, most of the algorithms are designed under the formulation of the Constrained Markov Decision Process (CMDPs), which is defined as a tuple $\mathcal{M} = (\mathcal{S}, \mathcal{A}, P, r, c, \gamma, \rho)$, where $\mathcal{S}$ is the state space, $\mathcal{A}$ is the action space, $P : \mathcal{S} \times \mathcal{A} \to \Delta(\mathcal{S})$[1] is the transition probability function, $r : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ is the reward function, $c : \mathcal{S} \times \mathcal{A} \to \mathbb{R}^+$ is the cost function, $\gamma \in (0, 1)$ is the discount factor and ρ is the initial state distribution. Typically, the reward value function and the cost value function are defined respectively as $V_r^\pi(s) = \mathbb{E}_{\tau\sim(\pi,P)}[\sum_{t=0}^\infty \gamma^t r(s_t, a_t)]$ and $V_c^\pi(s) = \mathbb{E}_{\tau\sim(\pi,P)}[\sum_{t=0}^\infty \gamma^t c(s_t, a_t)]$, where $\tau = \{s_0 = s, a_0, s_1, \cdots\} \sim (\pi, P)$ denotes a trajectory starting from s under the given policy π and transition function P. The goal is to find a policy $\pi : \mathcal{S} \to \Delta(\mathcal{A})$ that maximizes the expected discounted cumulative reward while ensuring that the cost value function remains below a predefined safety budget d. Formally:

$$\begin{aligned} &\max_\pi \mathbb{E}_{s_0\sim\rho}[V_r^\pi(s_0)] \\ &\text{s.t. } \mathbb{E}_{s_0\sim\rho}[V_c^\pi(s_0)] \le d. \end{aligned} \tag{1}$$

The CMDP framework ensures agents operate within an expected safety budget, but it may fall short in safety-critical scenarios where a single unsafe action can cause catastrophic failure. This necessitates a stricter safety-oriented framework.

We introduce $\mathcal{S}_u \subseteq \mathcal{S}$ to be the set of unsafe states. In this paper, we assume that any unsafe state $s_u \in \mathcal{S}_u$ is an absorbing state, meaning that the episode will terminate once the agent enters this region. This assumption imposes a stricter safety constraint, motivated by the critical importance of safety in practical applications. The overall goal of our proposed MAXSAFE framework consists of chance-constrained bi-level objective which maintains unsafe probability as low as possible while maximizing the return:

$$\begin{aligned} &\max_\pi \mathbb{E}_{s_0\sim\rho}[V_r^\pi(s_0)] \\ &\text{s.t. } \pi \in \arg\min_\pi \Pr_{\substack{s_0\sim\rho,\\ \tau\sim(\pi,P)}}[\exists s_t \in \tau : s_t \in \mathcal{S}_u]. \end{aligned} \tag{2}$$

We assume that there is sufficiently large policy space with minimum unsafe probability, e.g., zero unsafe probability. Such environments are common in autonomous driving and robotic control, for instance there are many possible driving policies that allow a vehicle to operate safely on the road without collisions. Also, this assumption is already explored in the literature, e.g., Ganai et al. (2023), which assumes the existence of safest policies with zero cost.

Let π^* be an optimal policy of MAXSAFE objective (2) and let $\zeta(s)$ denote the unsafe probability at state s under π^*, which is in fact minimal,

$$\zeta(s) := \Pr_{\tau\sim(\pi^*,P)}[\exists s_t \in \tau : s_t \in \mathcal{S}_u | s_0 = s]. \tag{3}$$

Then we can define the optimal action mask $\mathcal{C}_\zeta(s)$ for MAXSAFE as the set of safest actions at state s

$$\begin{aligned} \mathcal{C}_\zeta(s) = \{b \in \mathcal{A} \mid \Pr_{\tau\sim(\pi^*,P)}[\exists s_t \in \tau : s_t \in \mathcal{S}_u | s_0 = s, \\ a_0 = b] \le \zeta(s)\}. \end{aligned} \tag{4}$$

For tabular MDPs, $\mathcal{C}_\zeta(s)$ is optimal in the sense that, if we know $\mathcal{C}_\zeta(s)$, we can define the Bellman operator $\mathcal{B}_\zeta : \mathbb{R}^{|\mathcal{S}||\mathcal{A}|} \to \mathbb{R}^{|\mathcal{S}||\mathcal{A}|}$ with optimal action masking $\mathcal{C}_\zeta$,

$$\mathcal{B}_\zeta Q(s, a) = r(s, a) + \gamma \mathbb{E}_{s'} \max_{a' \in \mathcal{C}_\zeta(s')} Q(s', a'), \tag{5}$$

which can be iteratively applied to find an optimal policy under MAXSAFE framework, as shown in the following lemma:

Lemma 3.1. *The following results hold:*

(1) (Contraction). *The Bellman operator (5) is a γ-contraction and thus has a unique fixed point, which is denoted by Q_ζ^*.*

(2) (Safety Optimality). *For all $s \in \mathcal{S}$, we define $\pi_\zeta^*(s) := \arg\max_{a\in\mathcal{C}_\zeta(s)} Q_\zeta^*(s, a)$ to be the corresponding policy induced by Q_ζ^*. Then the safety of π_ζ^* is optimal: $\pi_\zeta^* \in \arg\min_\pi \Pr_{s_0\sim\rho,\tau\sim(\pi,P)}[\exists s_t \in \tau : s_t \in \mathcal{S}_u]$.*

(3) (Reward Optimality). *Reward of π_ζ^* is optimal among the safe policies: $\mathbb{E}_{s_0\sim\rho}[V^{\pi^*}(s_0)] = \mathbb{E}_{s_0\sim\rho}[V^{\pi_\zeta^*}(s_0)]$.*

[1] $\Delta(B)$ denotes the set of probability distributions over set B.

The proof of Lemma 3.1 can be found in Appendix B.1. Grounded by Lemma 3.1, in the following section, we focus on designing an algorithm that learns $\mathcal{C}_\zeta(s)$ on the fly which then guides the Q-function to select the action with optimal reward among the safest actions and thus ensure maximal safety and reward to solve our MAXSAFE objective (2).

4. Learning the Optimal Action Masks

In this section, we focus on learning state-dependent optimal action masks $\mathcal{C}_\zeta(s)$. Ganai et al. (2023) proposed a reachability estimation function (REF) to capture the probability of constraint violation at state s under a given policy. In our work, we extend the definition of REF to be state-action dependent. As shown in Section 4.1, this extended REF can serve as an estimate of the unsafe probability for a given state-action pair. It is then utilized to construct our action masks, forming the foundation of our learning solution to the MaxSafe objective (2).

4.1. State-Action Reachability Estimation Functions

Definition 4.1. Given a policy π, we define the state-action reachability estimation function (SA-REF) $\psi^\pi : \mathcal{S} \times \mathcal{A} \to [0, 1]$ as follows,

$$\psi^\pi(s, a) := \mathbb{E}_{\tau\sim(\pi,P)}\Big[\max_{s_t\in\tau} \mathbb{I}[s_t \in \mathcal{S}_u] \mid s_0 = s, a_0 = a\Big], \tag{6}$$

$\psi^\pi(s, a)$ represents the unsafe probability at state s and action a under the current policy π, that is, $\psi^\pi(s, a) = \Pr_{\tau\sim(\pi,P)}[\exists s_t \in \tau : s_t \in \mathcal{S}_u | s_0 = s, a_0 = a]$. Expressing in the form of Equation (6) provides a convenient way to compute $\psi^\pi(s, a)$ via backward induction.

Lemma 4.2. *We have the following Bellman backup for the SA-REF ψ^π:*

$$\begin{aligned}\psi^\pi(s, a) =& \max\{\mathbb{I}[s \in \mathcal{S}_u], \\ & \mathbb{E}_{s'\sim P(\cdot|s,a),a'\sim\pi(\cdot|s')}[\psi^\pi(s', a')]\}.\end{aligned} \tag{7}$$

The proof of Lemma 4.2 is in Appendix B.3. For convenience, we define the optimal SA-REF $\psi^* := \psi^{\pi^*}$ where π^* is our optimal policy of MAXSAFE. Based on our SA-REF formulation, we can rewrite the definition in Equation (3) as

$$\zeta(s) = \mathbb{E}_{a\sim\pi^*(s)}[\psi^*(s, a)] = \min_{a\in\mathcal{A}} \psi^*(s, a) \tag{8}$$

since π^* reaches the minimum unsafe probability, and the optimal action mask becomes

$$\mathcal{C}_\zeta(s) = \{b \in \mathcal{A} \mid \psi^*(s, b) \le \zeta(s) = \min_{a\in\mathcal{A}} \psi^*(s, a)\}. \tag{9}$$

Therefore, to find the optimal action mask $\mathcal{C}_\zeta$, we would like to learn the function ψ^*. Fortunately, this can be done via backward induction by mimicking π^* to take actions with the minimum unsafe probability as follows.

Theorem 4.3. (Optimal SA-REF backward induction) *For tabular MDPs, with $\psi_0 = 0$, the update*

$$\psi_{t+1}(s, a) = \max\{\mathbb{I}[s_t \in \mathcal{S}_u], \mathbb{E}_{s'\sim P(\cdot|s,a)} \min_{a'\in\mathcal{A}} \psi_t(s', a')\} \tag{10}$$

will converge to the optimal SA-REF ψ^* as $t \to +\infty$.

The proof of Theorem 4.3 is deferred to Appendix B.3.

4.2. Safe Q-Learning with Optimal Action Masks

With Theorem 4.3, we provide an update of ψ_t in the finite-sample form and incorporate this update with the Q-learning algorithm to obtain a safe Q-learning algorithm with theoretical convergence-to-optimality guarantees.

The ψ update, starting from $\psi_0 = 0$, goes as follows

$$\begin{aligned}\psi_{t+1}(s_t, a_t) =& (1 - \beta_t)\psi_t(s_t, a_t)+ \\ & \beta_t \max\{\mathbb{I}[s_t \in \mathcal{S}_u], \min_{a'\in\mathcal{A}} \psi_t(s_{t+1}, a')\},\end{aligned} \tag{11}$$

where $0 < \beta_t \le 1$ is the step size, each $a_t \sim \pi_b(\cdot|s_t)$ is sampled from a behavior policy π_b. Following Equation (9) we then define the learned optimal action masks as

$$\mathcal{C}_{\zeta_t}(s) := \{b \in \mathcal{A} \mid \psi_t(s, b) \le \zeta_t(s)\}, \tag{12}$$

where for a small enough constant $\kappa > 0$, $\zeta_t(s) := \min_{a\in\mathcal{A}} \psi_t(s, a) + \kappa$. Here κ is needed in order to tolerate the stochastic approximation error during the learning process. Note that the threshold function $\zeta_t(s)$ is state-dependent, which is the key for the action masks $\mathcal{C}_{\zeta_t}$ to optimality.

We now propose our safe Q-learning update with optimal action masks:

$$\begin{aligned}Q_{t+1}(s_t, a_t) \leftarrow& (1 - \beta_t)Q_t(s_t, a_t)+ \\ & \beta_t(r(s_t, a_t) + \gamma \max_{a'\in\mathcal{C}_{\zeta_t}(s_{t+1})} Q_t(s_{t+1}, a')),\end{aligned} \tag{13}$$

where the difference from the original Q-learning algorithm is that the action maximizing Q at the next state s' is chosen only from the learned optimal action mask during iteration t as $\mathcal{C}_{\zeta_t}(s')$. Now the learned policy is of the form

$$\pi_t(s) := \arg\max_{a\in\mathcal{C}_{\zeta_t}(s)} Q_t(s, a). \tag{14}$$

The following convergence analysis is based on the fact that $\mathcal{C}_{\zeta_t}(s')$ will be the same as the optimal $\mathcal{C}_\zeta(s')$ for t large enough and after that the Q-update (13) will stably converge.

Theorem 4.4. *For tabular MDPs, suppose the following conditions hold:*

(1) *each state-action pair (s, a) is infinitely visited;*

(2) *the step size sequence $\{\beta_t\}_{t\ge0}$ satisfies $0 < \beta_t \le 1$ and $\sum_{t\ge0} \beta_t = +\infty, \sum_{t\ge0} \beta_t^2 < +\infty$;*

Algorithm 1 **S**afety-**P**olarized **O**ptimal action **M**asks with **P**rioritized **E**xperience **R**eplay (**SPOM_PER**)

Require: a polarization function f_{pol}, safety prioritization exponent α, importance sampling exponent θ
Initialize: a safety prioritized replay buffer $\mathcal{D}$, Q-network Q, ψ-network ψ, along with target networks $\bar{Q}$ and $\bar{\psi}$
for each time step **do**
  Take action a_t based on $\pi(s_t)$ (cf. Equation (17)) combined with any exploration strategy, e.g., ϵ-greedy
  Store the collected sample (s_t, a_t, r_t, s_{t+1}) into $\mathcal{D}$ with maximal priority $p_t = \max_{i<t} p_i$
  for each update step **do**
    Sample a mini-batch of data (s_i, a_i, r_i, s_{i+1}) from replay buffer according to their priorities $i \sim P(i)$ (cf. Equation (19))
    Compute importance sampling weight based w_i (cf. Equation (20))
    Compute

$$a^Q_{i+1} = \arg\max_{a\in\mathcal{A}} Q(s_{i+1}, a) + f_{\text{pol}}(1 - \psi(s_{i+1}, a))$$

    Compute Q targets $y^Q_i = r_i + \gamma\bar{Q}(s_{i+1}, a^Q_{i+1})$
    Minimize MSE loss between $Q(s_i, a_i)$ and y^Q_i
    Compute $a^\psi_{i+1} = \arg\min_{a\in\mathcal{A}} \psi(s_{i+1}, a)$
    Compute ψ targets

$$y^\psi_i = \max\{\mathbb{I}[s_i \in \mathcal{S}_u], \gamma\bar{\psi}(s_{i+1}, a^\psi_{i+1})\}$$

    and ψ-TD errors $\delta_i = y^\psi_i - \psi(s_i, a_i)$
    Minimize weighted MSE loss between $\psi(s_i, a_i)$ and y^ψ_i based on w_i (cf. Equation (20))
    Update transition priority p_i based on ψ-TD errors
    Update target networks $\bar{Q}$ and $\bar{\psi}$
  end for
end for

(3) *$\kappa > 0$ is small enough to identify the gap between $\mathcal{C}_\zeta(s)$ and $\mathcal{A} \setminus \mathcal{C}_\zeta(s)$, i.e., $\forall b \in \mathcal{C}_\zeta(s), e \in \mathcal{A} \setminus \mathcal{C}_\zeta(s)$,*

$$\psi^*(s, b) + 2\kappa < \psi^*(s, e), \tag{15}$$

then the ψ_t update in Equation (11) will converge to the optimal SA-REF ψ^*, and our safe Q-learning update in Equation (13) will converge to Q^*_ζ. Thus, the learned policy $\pi_t(s)$ will converge to an optimal policy π^*_ζ under our MAXSAFE *objective (2).*

The proof of Theorem 4.4 is deferred to Appendix B.4.

5. Deep Q-Learning with Safety Polarization and Safety Prioritized Experience Replay

In this section, we mainly focus on how we combine modern deep Q-learning algorithms to solve practical safety-critical RL tasks. In Section 5.1, we focus on a practical implementation that transfers our proposed masking strategy using the polarization function. In Section 5.2, we adopt a technique inspired by prioritized experience replay to further help maximize safety in long-horizon sparse cost signal scenarios.

5.1. Safety Polarization for Q-functions

To fully implement our learned action masking in a deep RL algorithm, we define the gating operator as

$$\Gamma_\eta(x) = \begin{cases} 0, x \le \eta \\ 1, \text{otherwise.} \end{cases}$$

Our action masking $\mathcal{C}_{\zeta_t}(s)$ defined in Equation 12 can be viewed as adding $-\infty$ to Q_t at the masked action while adding 0 to Q_t at other action. Specifically, for actions where $\Gamma_{\zeta_t(s)}(\psi(s, a)) = 1$, we penalize Q_t during training. We implement the following class of polarization function to combine with the learned Q-function. Formally, we define the polarization function class as follows:

Definition 5.1. Define the *polarization function class* $\mathcal{F}_{\text{pol}}$ as the set of all polarization functions $f_{\text{pol}} : [0, 1] \to [-\infty, 0]$ which is a monotonically increasing function satisfying $f_{\text{pol}}(1) = 0, f_{\text{pol}}(0) := \lim_{x\to 0+} f_{\text{pol}}(x) = -\infty$.

Examples of polarization functions include $c \cdot \log(x), 1 - \frac{1}{x^c}$ for a constant value $c > 0$. Together with the gating operator Γ_η and our learned optimal action mask $\mathcal{C}_{\zeta_t}(s)$, we derive our learned MaxSafe policy can be rewritten as

$$\begin{aligned} \pi_t(s) &= \arg\max_{a\in\mathcal{C}_\zeta(s)} Q_t(s, a) \\ &= \arg\max_{a\in\mathcal{A}} \left[Q_t(s, a) + f_{\text{pol}}(1 - \Gamma_{\zeta(s)}(\psi_t(s, a)))\right]. \end{aligned} \tag{16}$$

During implementation, we further get rid of the hard masking rule induced by the gating operator Γ, and compose the Q-function Q_t with the SA-REF ψ_t through f as follows:

$$\pi_t(s) := \arg\max_{a\in\mathcal{A}} Q_t(s, a) + f(1 - \psi_t(s, a)). \tag{17}$$

Intuitively, if a state-action pair (s, a) is measured to be unsafe, $1 - \psi_t(s, a)$ will be close to 0, then $f(1 - \psi_t(s, a))$ will be close to $-\infty$ and the action a is likely not to be chosen at state s since $Q_t(s, a) + f_{\text{pol}}(1 - \psi_t(s, a))$ is very low. Therefore the actions below the state-dependent threshold $\zeta_t(s)$ are more likely to be selected. The choice of the polarization function f_{pol} provides a balance between exploration and safety: we focus more on safety if f_{pol} converges faster to $-\infty$ as $x \to 0+$, we encourages more exploration otherwise.

5.2. Safety Prioritized Experience Replay

Prioritized Experience Replay (PER) enhances off-policy RL by assigning priorities to transitions based on their TD errors, emphasizing those with the highest learning potential. This approach improves sample efficiency and accelerates convergence, particularly in scenarios with sparse reward signals. In our work, to address the challenge of sparse cost signals, we leverage PER to enhance the learning of the ψ network. The motivation behind this approach is similar to that of PER in traditional Q-learning, where transitions with larger TD errors are prioritized to accelerate learning. Furthermore, importance sampling is employed to mitigate the distribution shift introduced by the prioritization process.

Specifically, we introduce a prioritization technique tailored for the ψ network. The priority of a transition i is determined by the ψ-TD error, defined as

$$\delta_i = \max\{\mathbb{I}[s_i \in \mathcal{S}_u], \min_{a\in\mathcal{A}} \psi_t(s_{i+1}, a)\} - \psi_t(s_i, a_i), \tag{18}$$

where the priority p_i is calculated as $p_i = |\delta_i| + \epsilon$. Here, ϵ is a small positive constant added to ensure that transitions are not excluded from sampling when their error becomes zero. The sampling probability for each transition is then computed using the prioritization heuristic:

$$P(i) = \frac{p_i^\alpha}{\sum_{k=1}^N p_k^\alpha}, \tag{19}$$

where $\alpha \in [0, 1]$ controls the degree of prioritization. Unlike traditional PER in Q-learning, our prioritization scheme is explicitly tied to the learned ψ network, reflecting the specific requirements of safety-critical scenarios.

Since safety prioritized experience replay introduces sampling bias by altering the transition distribution, we correct this bias using importance-sampling (IS) weights during the ψ update. The IS weight for a sampled transition i is defined as:

$$w_i = \left(\frac{1}{N} \cdot \frac{1}{P(i)}\right)^\theta, \tag{20}$$

where N is the size of the replay buffer, and $\theta \in [0, 1]$ controls the degree of correction. Then the update of our ψ network will become

$$\begin{aligned}\psi_t(s_t, a_t) \leftarrow &(1 - w_t\beta_t)\psi_t(s_t, a_t) + \\ &w_t\beta_t \max\{\mathbb{I}[s_t \in \mathcal{S}_u], \min_{a'\in\mathcal{A}} \psi_t(s_{t+1}, a')\}.\end{aligned} \tag{21}$$

By incorporating safety-prioritized experience replay, our approach effectively enhances the learning of the ψ network, particularly in environments with long-horizon tasks and sparse cost signals. This integration improves both safety and sample efficiency.

5.3. Practical Implementation

For large-scale tasks with discrete action space $\mathcal{A}$ based on DQN (Mnih et al., 2015), we use neural networks Q and ψ to approximate the optimal Q-function and the SA-REF, respectively (the corresponding target networks are $\bar{Q}$ and $\bar{\psi}$). We select the Sigmoid function as the activation for the output layer of the ψ-network to ensure that its output remains bounded within the range $[0, 1]$. The pseudo-code is presented in Algorithm 1.

When computing the targets for Q and ψ updates, we use a double-Q-learning-style implementation (Van Hasselt et al., 2015). Specifically, at lines 9 and 10 of Algorithm 1, the target for Q is computed by querying the action a^Q_{i+1}, which combines the max Q-value with the safety polarization function penalty at the next state s_{i+1}. This is done using the current policy π, composed of the current Q and ψ. Similarly, the target for ψ (computed at lines 12 and 13 of Algorithm 1) follows the same principle. Note that for the ψ target at line 12, we multiply $\bar{\psi}$ by the discount factor γ to reduce the long-term variance in ψ backward induction. This approach is analogous to the multiplication of $\bar{Q}$ by γ in DQN.

For the polarization function f_{pol} used in our experiments, we select $10 \cdot \log(x)$ which demonstrates the best empirical performance. Additionally, we conduct an ablation study to analyze the effects of different polarization functions (see Section 6.3).

6. Experiments

6.1. Experiment Setup

Benchmarks. Our evaluation adopts the following four tasks: TwoWay, Merge, Roundabout, and Intersection. These tasks are from the highway-env environment (Leurent, 2018; Leurent & Mercat, 2019), designed for simulated autonomous driving with diverse objectives that require intricate behaviors to safely achieve the corresponding goals. Additionally, our evaluation includes classical safe control tasks such as Adaptive Cruise Control (ACC) (Anderson et al., 2020) and Circle (Achiam et al., 2017). We highlight that the cost functions used for the CMDP-based methods are designed as $c = 1$ when a crash happens, and 0 otherwise. More details about the environments, e.g., state spaces, action spaces, reward functions, etc., can be found in Appendix C.1.

Baselines. Our base unconstrained RL is **DQN** which, in our implementation, uses the double Q-learning technique (Van Hasselt et al., 2015) by default. Other safe RL baselines include the following:

- **Reward Shaping.** The reward is shaped to -10 when

Table 1. **SPOM_PER** achieves the best safety-reward tradeoff in terms of SWU score, while **SPOM** performs the best among the remaining safe-RL baselines.

SWU Score ↑	TwoWay	Merge	Roundabout	Intersection	ACC	Circle	Overall
SPOM_PER (ours)	**0.98**	0.94	**1.22**	0.68	**1.07**	**0.96**	**0.98**
SPOM (ours)	0.88	**0.95**	0.40	**0.96**	0.87	0.37	0.74
SafeQ	0.73	0.84	0.63	0.75	0.50	0.15	0.60
Recovery	0.34	0.66	0.81	0.66	0.85	0.25	0.60
RCDQN	0.41	0.81	0.48	0.57	0.46	0.14	0.48
RewsDQN	0.33	0.92	0.48	0.50	0.47	0.11	0.47
DQN	0.37	0.56	0.48	0.58	0.46	0.13	0.43

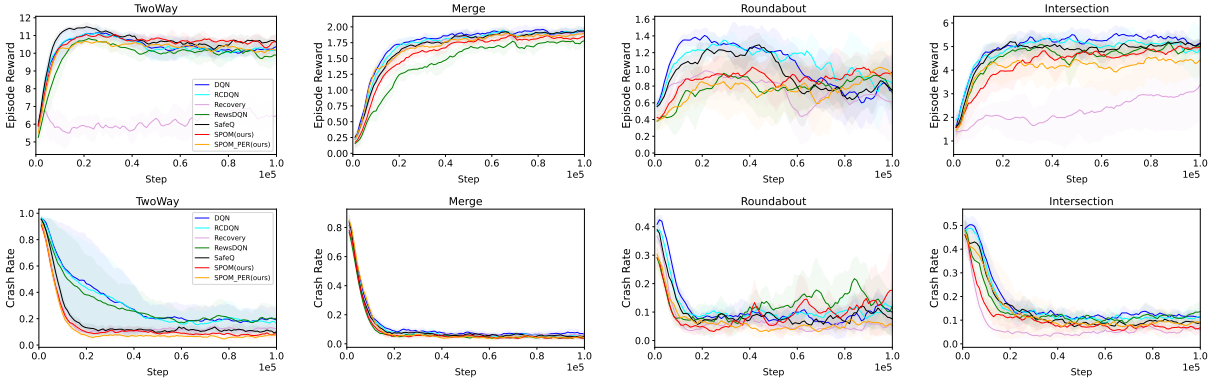


Figure 1. Training curves of **SPOM**, **SPOM_PER** and baselines in the four tasks from highway-env. The x-axis represents the number of steps, and the y-axis: the first row is episode reward (higher is better); the second row is safety measured in crash rate (lower is better). All experiments are run over 6 random seeds and the shaded areas are 95% confidence interval.

a crash occurs, which is a strong penalty, and we label the reward-shaped DQN variant as **RewsDQN**.

- **Direct ε-Masking.** This is the **SafeQ** (Srinivasan et al., 2020) as discussed before, which masks out the actions whose estimated unsafe probability by a safety critic is greater than ε. The original actor-critic based algorithm in Srinivasan et al. (2020) is modified to be DQN-based.
- **Recovery RL.** Originated from Thananjeyan et al. (2020), labeled as **Recovery** here, this corresponds to the action correction approach, where a task policy selects an action, but if the estimated unsafe probability from a safety critic is greater than ε, the action will be corrected by a learned recovery policy. The task policy and recovery policy are implemented via DQN.
- **CMDP-based RL.** We choose the reward constrained approach in Tessler et al. (2019) which constrains the reward through $r - \lambda c$, where λ is the Lagrange multiplier and c is the cost signal corresponding to crash. Then we use DQN to learn upon this constrained reward. We denote this baseline by **RCDQN**.

For **SafeQ** and **Recovery**, the parameter ε is chosen to be 0.1 throughout experiments. Note that the above baselines are implemented via DQN for fair competition, while the common CMDP-based RL is built on an actor-critic framework. For completeness and clarity of presentation, we add additional CMDP-based approaches **RESPO** (Ganai et al., 2023), **PPOLag** (Ray et al., 2019), which use Proximal Policy Optimization (PPO) (Schulman et al., 2017) as the base RL algorithm, in Appendix C.3, where we find that they fail to optimize under our environments since the cost signal corresponding to a crash is very sparse. The detailed implementations and hyperparameters of these baselines and our algorithm **SPOM** and its safety prioritized experience replay version **SPOM_PER** are provided in Appendix C.2. **Evaluation Metric.** To clearly quantify the trade-off between rewards and safety, we follow Yu et al. (2022b) and compute the *safety-weighted-utility* (SWU) score as the final evaluation metric. The SWU score is defined as:

$$\mathrm{SWU} := \min\left\{1, \frac{\mathrm{UnsafeRateTarget}}{\mathrm{UnsafeRate}}\right\} \cdot \frac{\mathrm{Utility}}{\mathrm{Utility}_{\mathrm{BaseRL}}},$$

where BaseRL refers to the unconstrained RL algorithm **DQN** in our experiments. We choose the $\mathrm{Utility}$ to be the episode reward, and $\mathrm{UnsafeRate}$ is the crash rate. The $\mathrm{UnsafeRateTarget}$ is chosen as the minimal crash rate among the compared methods for each environment, since we aim at achieving the maximal safety. To reduce variance, $\mathrm{Utility}$, $\mathrm{UnsafeRate}$ and $\mathrm{UnsafeRateTarget}$ are averaged

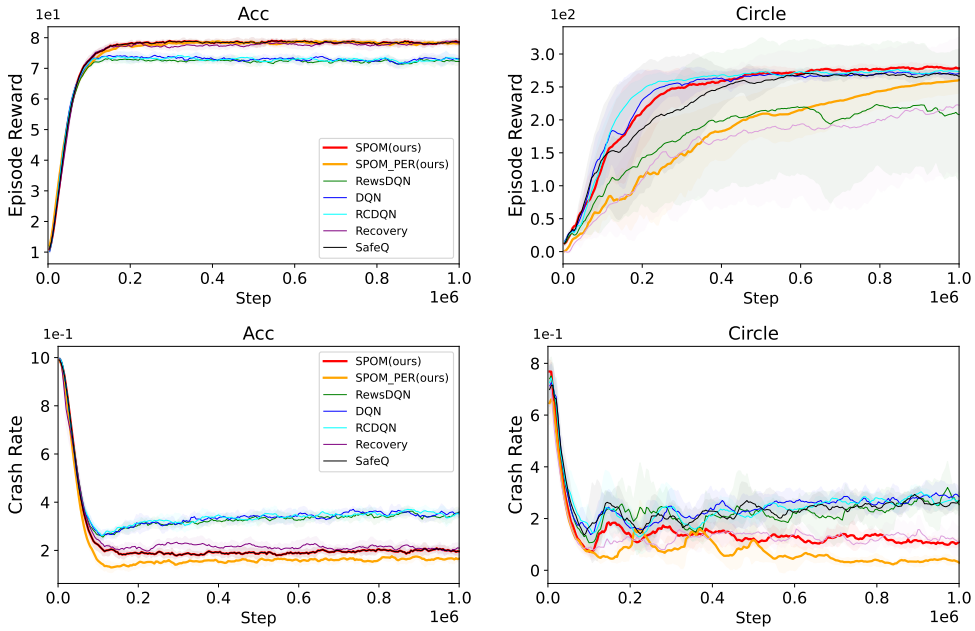


Figure 2. Training curves of **SPOM**, **SPOM_PER** and baselines in the classical safe control tasks: ACC and Circle. The x-axis represents the number of steps, and the y-axis: the first row is episode reward (higher is better); the second row is safety measured in crash rate (lower is better).

over the last $\frac{1}{10}$ training steps (Yu et al., 2022b).[2]

6.2. Main Results

First, our proposed algorithms, **SPOM_PER** and **SPOM**, consistently achieve the best performance across all tasks, demonstrating their effectiveness in balancing safety and rewards. As shown in Table 1, **SPOM_PER** achieves the highest average SWU score of 0.90, followed by **SPOM** with a score of 0.79, outperforming all other baselines by a significant margin.

For the classical safe control tasks, ACC and Circle, **SPOM_PER**, thanks to the prioritization mechanism in the replay buffer for the learning of the ψ network, achieves significantly lower crash rates compared to other methods while ensuring convergence to higher episode rewards. In comparison, **SPOM** also performs competitively in ACC and Circle but slightly lags behind **SPOM_PER** in terms of crash rate reduction. Baselines such as **Recovery** perform reasonably well in ACC, but their inherent conservativeness leads to suboptimal rewards, and they struggle significantly in Circle.

For the four tasks from highway-env, both **SPOM** and **SPOM_PER** demonstrate leading performance. **SPOM_PER** achieves the highest SWU scores in Merge and Roundabout, while **SPOM** leads in TwoWay and Intersection. The performance gap between **SPOM** and **SPOM_PER** is relatively small in these tasks, which we attribute to the shorter episode lengths in the highway environment, reducing the impact of the prioritization mechanism. Compared to baselines, **SafeQ** performs moderately well in Merge and Intersection but struggles to achieve competitive SWU scores in other tasks, failing to effectively balance safety and rewards. Methods like **Recovery** and **RCDQN** exhibit more conservative behavior, achieving lower crash rates at the cost of significantly reduced rewards. **DQN** and **RewsDQN**, on the other hand, fail to manage safety effectively, resulting in much lower SWU scores across all tasks.

Overall, the results validate that **SPOM_PER** and **SPOM** are highly effective in optimizing the trade-off between safety and rewards, with **SPOM_PER** excelling in long-horizon tasks and maintaining the highest overall SWU score.

6.3. Ablation Studies

We conduct ablation studies on the usage of different polarized function f_{pol} and apply the optimal action masks directly in Intersection and TwoWay. The results are shown in Figure 3. Observe that, the direct optimal action masking "OAM" is significantly conservative in Intersection, while in TwoWay, its crash rate has little improvement but reward is lower. This shows the hurt brought by optimal action masking on an immature ψ network in the early stage of training, which is also the reason of suboptimality of the strong polarization effect given by "xp", since masking is a

[2]Code for the experiments is available at https://github.com/FrankSinatral/Safety-PP.git.

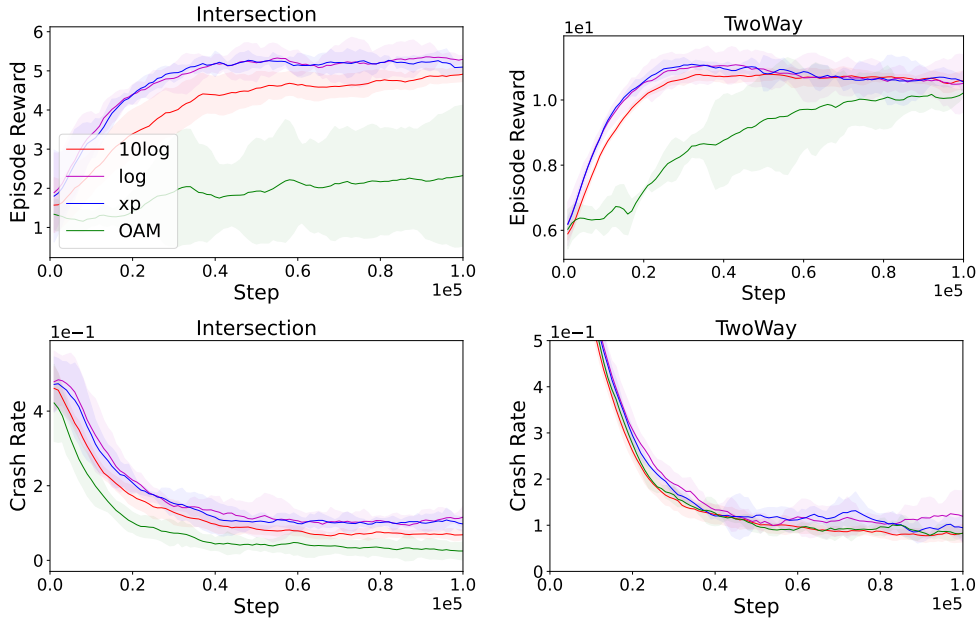


Figure 3. Ablation studies, where "log", "xp" represent using polarization function $\log(x)$ and $1 - \frac{1}{x}$, respectively, "$10\log$" is our default choice $10 \cdot \log(x)$, and "OAM" means applying optimal action masks directly. The x-axis represents the number of steps, and the y-axis: the first row is episode reward (higher is better); the second row is safety measured in crash rate (lower is better).

special case of safety polarization. Finally, for "$\log$" its polarization effect is weak and thus cannot be safe enough like our default "$10\log$". The same phenomena can be observed in other tasks. Full details are presented in Appendix C.3.

7. Conclusions

We propose MAXSAFE, a novel chance-constrained bi-level optimization framework for safe RL, motivated by the prioritization of safety in real-world applications. We address the MAXSAFE objective by learning optimal action masks and introduce the technique of safety polarization as a practical generalization of optimal action masks. Additionally, we propose safety-prioritized experience replay, designed to accelerate the learning of optimal action masks, especially when cost signals are sparse. Extensive experiments demonstrate that our method achieves an optimal trade-off between reward and safety, delivering near-maximal safety.

Impact Statement

This paper presents work whose goal is to advance the field of Machine Learning. There are many potential societal consequences of our work, none which we feel must be specifically highlighted here.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China Grant 52494974, China's Village Science and Technology City Key Technology funding, and Wuxi Research Institute of Applied Technologies.

References

Achiam, J., Held, D., Tamar, A., and Abbeel, P. Constrained policy optimization. In *International Conference on Machine Learning (ICML)*, 2017.

Alshiekh, M., Bloem, R., Ehlers, R., Könighofer, B., Niekum, S., and Topcu, U. Safe reinforcement learning via shielding. In *AAAI Conference on Artificial Intelligence (AAAI)*, 2017.

Altman, E. *Constrained Markov Decision Processes*. Chapman and Hall/CRC, 1999.

Anderson, G., Verma, A., Dillig, I., and Chaudhuri, S. Neurosymbolic reinforcement learning with formally verified exploration. In *Neural Information Processing Systems (NeurIPS)*, 2020.

Bansal, S., Chen, M., Herbert, S., and Tomlin, C. J. Hamilton-jacobi reachability: A brief overview and recent advances. In *IEEE Conference on Decision and Control (CDC)*, 2017.

Bharadhwaj, H., Kumar, A., Rhinehart, N., Levine, S., Shkurti, F., and Garg, A. Conservative safety critics for exploration. In *International Conference on Learning Representations (ICLR)*, 2021.

Borkar, V. S. *Stochastic Approximation: A Dynamical Systems Viewpoint*. Cambridge University Press, 2008.

Brosowsky, M., Keck, F., Ketterer, J., Isele, S. T., Slieter, D., and Zöllner, J. M. Safe deep reinforcement learning for adaptive cruise control by imposing state-specific safe sets. In *IEEE Intelligent Vehicles Symposium (IV)*, 2021.

Cheng, R., Orosz, G., Murray, R. M., and Burdick, J. W. End-to-end safe reinforcement learning through barrier functions for safety-critical continuous control tasks. In *AAAI Conference on Artificial Intelligence (AAAI)*, 2019.

Ding, S., Wang, J., Du, Y., and Shi, Y. Reduced policy optimization for continuous control with hard constraints. In *Neural Information Processing Systems (NeurIPS)*, 2023.

Dong, Y., Zhao, X., Wang, S., and Huang, X. Reachability verification based reliability assessment for deep reinforcement learning controlled robotics and autonomous systems. *IEEE Robotics and Automation Letters*, 9(4): 3299–3306, 2024.

Fulton, N. and Platzer, A. Safe reinforcement learning via formal methods: Toward safe control through proof and learning. In *AAAI Conference on Artificial Intelligence (AAAI)*, 2018.

Fulton, N. and Platzer, A. Verifiably safe off-model reinforcement learning. In *International Conference on Tools and Algorithms for Construction and Analysis of Systems (TACAS)*, 2019.

Ganai, M., Gong, Z., Yu, C., Herbert, S. L., and Gao, S. Iterative reachability estimation for safe reinforcement learning. In *Neural Information Processing Systems (NeurIPS)*, 2023.

Ganai, M., Gao, S., and Herbert, S. L. Hamilton-jacobi reachability in reinforcement learning: A survey. *IEEE Open Journal of Control Systems*, 3:310–324, 2024.

García, J. and Fernández, F. A comprehensive survey on safe reinforcement learning. *Journal of Machine Learning Research*, 16:1437–1480, 2015.

Gu, S., Yang, L., Du, Y., Chen, G., Walter, F., Wang, J., Yang, Y., and Knoll, A. A review of safe reinforcement learning: Methods, theory and applications. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46 (12):11216–11235, 2024.

Hessel, M., Modayil, J., van Hasselt, H., Schaul, T., Ostrovski, G., Dabney, W., Horgan, D., Piot, B., Azar, M., and Silver, D. Rainbow: Combining improvements in deep reinforcement learning. In *AAAI Conference on Artificial Intelligence (AAAI)*, 2018.

Horgan, D., Quan, J., Budden, D., Barth-Maron, G., Hessel, M., van Hasselt, H., and Silver, D. Distributed prioritized experience replay. In *International Conference on Learning Representations (ICLR)*, 2018.

Huang, S. and Ontanón, S. A closer look at invalid action masking in policy gradient algorithms. In *International Florida Artificial Intelligence Research Society Conference (FLAIRS)*, 2022.

Kalweit, G., Huegle, M., Werling, M., and Boedecker, J. Deep constrained q-learning, 2020. URL https://arxiv.org/abs/2003.09398.

Krasowski, H., Zhang, Y., and Althoff, M. Safe reinforcement learning for urban driving using invariably safe braking sets. In *International Conference on Intelligent Transportation Systems (ITSC)*, 2022.

Krasowski, H., Thumm, J., Müller, M., Schäfer, L., Wang, X., and Althoff, M. Provably safe reinforcement learning: Conceptual analysis, survey, and benchmarking. *Transactions on Machine Learning Research*, 2023.

Leurent, E. An environment for autonomous driving decision-making, 2018. URL https://github.com/eleurent/highway-env.

Leurent, E. and Mercat, J. Social attention for autonomous decision-making in dense traffic. In *Machine Learning for Autonomous Driving Workshop at NeurIPS*, 2019.

Mirchevska, B., Pek, C., Werling, M., Althoff, M., and Boedecker, J. High-level decision making for safe and reasonable autonomous lane changing using reinforcement learning. In *International Conference on Intelligent Transportation Systems (ITSC)*, 2018.

Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., Graves, A., Riedmiller, M. A., Fidjeland, A. K., Ostrovski, G., Petersen, S., Beattie, C., Sadik, A., Antonoglou, I., King, H., Kumaran, D., Wierstra, D., Legg, S., and Hassabis, D. Human-level control through deep reinforcement learning. *Nature*, 518:529–533, 2015.

Muller, A. and Sabatelli, M. Safe and psychologically pleasant traffic signal control with reinforcement learning using action masking. In *International Conference on Intelligent Transportation Systems (ITSC)*, 2022.

Platzer, A. Differential dynamic logic for hybrid systems. *Journal of Automated Reasoning*, 41:143–189, 2008.

Puterman, M. L. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. John Wiley and Sons, Inc., USA, 1st edition, 1994. ISBN 0471619779.

Ray, A., Achiam, J., and Amodei, D. Benchmarking safe exploration in deep reinforcement learning. In *International Conference on Machine Learning (ICML)*, 2019.

Schulman, J., Wolski, F., Dhariwal, P., Radford, A., and Klimov, O. Proximal policy optimization algorithms, 2017. URL https://arxiv.org/abs/1707.06347.

Srinivasan, K. P., Eysenbach, B., Ha, S., Tan, J., and Finn, C. Learning to be safe: Deep rl with a safety critic, 2020. URL https://arxiv.org/abs/2010.14603.

Tan, D. C., McCarthy, R., Acero, F., Delfaki, A. M., Li, Z., and Kanoulas, D. Safe value functions: Learned critics as hard safety constraints. In *IEEE International Conference on Automation Science and Engineering*, 2024.

Tessler, C., Mankowitz, D. J., and Mannor, S. Reward constrained policy optimization. In *International Conference on Learning Representations (ICLR)*, 2019.

Thananjeyan, B., Balakrishna, A., Nair, S., Luo, M., Srinivasan, K. P., Hwang, M., Gonzalez, J. E., Ibarz, J., Finn, C., and Goldberg, K. Recovery rl: Safe reinforcement learning with learned recovery zones. *IEEE Robotics and Automation Letters*, 6:4915–4922, 2020.

Van Hasselt, H., Guez, A., and Silver, D. Deep reinforcement learning with double q-learning. In *AAAI Conference on Artificial Intelligence (AAAI)*, 2015.

Wagener, N., Boots, B., and Cheng, C.-A. Safe reinforcement learning using advantage-based intervention. In *International Conference on Machine Learning (ICML)*, 2021.

Wang, Y. and Zhu, H. Safe exploration in reinforcement learning by reachability analysis over learned models. In Gurfinkel, A. and Ganesh, V. (eds.), *Computer Aided Verification (CAV)*, volume 14683 of *Lecture Notes in Computer Science*, pp. 190–213. Springer, Cham, 2024.

Yang, L., Ji, J., Dai, J., Zhang, L., Zhou, B., Li, P., Yang, Y., and Pan, G. Constrained update projection approach to safe policy optimization. In *Neural Information Processing Systems (NeurIPS)*, 2022.

Yang, Y., Jiang, Y., Liu, Y., Chen, J., and Li, S. E. Model-free safe reinforcement learning through neural barrier certificate. *IEEE Robotics and Automation Letters*, 8(3): 1295–1302, 2023.

Yu, D., Ma, H., Li, S., and Chen, J. Reachability constrained reinforcement learning. In *International Conference on Machine Learning (ICML)*, 2022a.

Yu, H., Xu, W., and Zhang, H. Towards safe reinforcement learning with a safety editor policy. In *Neural Information Processing Systems (NeurIPS)*, 2022b.

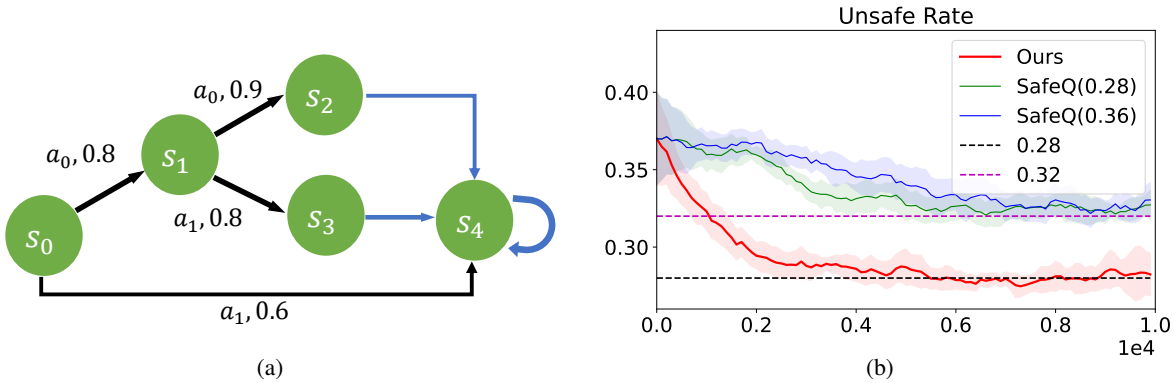
Zhang, L., Zhang, Q., Shen, L., Yuan, B., Wang, X., and Tao, D. Evaluating model-free reinforcement learning toward safety-critical tasks. In *AAAI Conference on Artificial Intelligence (AAAI)*, 2023.

Zhang, Y., Vuong, Q., and Ross, K. First order constrained optimization in policy space. In *Neural Information Processing Systems (NeurIPS)*, 2020.

Zhao, W., He, T., and Liu, C. Probabilistic safeguard for reinforcement learning using safety index guided gaussian process models. In *Learning for Dynamics and Control Conference (L4DC)*, 2023.

Zheng, Y., Li, J., Yu, D., Yang, Y., Li, S. E., Zhan, X., and Liu, J. Safe offline reinforcement learning with feasibility-guided diffusion model. In *International Conference on Learning Representations (ICLR)*, 2024.

A. Why State-agnostic Masking Threshold is not Enough



(a) (b)

Figure 4. (a) The MDP structure: the reward is always 0. Taking action $a \in \mathcal{A} = \{a_0, a_1\}$ transitions to the next state with probability p or to a fixed unsafe state (termination) with probability $1 - p$; blue arrows indicate transitions to the pointed state regardless of action. (b) Learning curves of unsafe rates using our algorithm and SafeQ over 10 seeds. The x-axis represents the number of steps. Lower is better.

Our proposed optimal action masking (*state-dependent*) enjoys a convergence guarantee as shown above. In contrast, we will show that the direct ε-masking method used by SafeQ, which masks actions with an estimated unsafe probability exceeding a fixed, state-agnostic threshold ε, may fail to achieve the minimum unsafe probability $\varepsilon_{\min}$, even when ε is set to $\varepsilon_{\min}$. To be more specific, we write SafeQ in the form of Q-learning and use the SA-REF ψ to be its unsafe probability estimator

$$\begin{aligned} Q_{t+1}(s_t, a_t) \leftarrow (1 - \beta_t) Q_t(s_t, a_t) + \\ \beta_t (r_t + \gamma \max_{a' \in \mathcal{C}_\varepsilon(s_{t+1})} Q_t(s_{t+1}, a')), \end{aligned} \tag{22}$$

where $\mathcal{C}_\varepsilon(s) := \{b \in \mathcal{A} \mid \psi_t(s, b) \leq \varepsilon\}$ is the action mask based on ε, and the ψ-update of SafeQ is

$$\begin{aligned} \psi_{t+1}(s_t, a_t) \leftarrow (1 - \beta_t) \psi_t(s_t, a_t) + \\ \beta_t \max\{\mathbb{I}[s_t \in \mathcal{S}_u], \psi_t(s_{t+1}, \tilde{\pi}_t(s_{t+1}))\}, \end{aligned} \tag{23}$$

where $\tilde{\pi}_t(s) := \arg\max_{a \in \mathcal{C}_\varepsilon(s)} Q_t(s, a)$ is the learned policy by SafeQ.

Our bad example is based on the intrinsic shortcoming of SafeQ: if $\exists a \neq b \in \mathcal{C}_\varepsilon(s)$ while $Q_t(s, a) = Q_t(s, b)$, then SafeQ policy $\tilde{\pi}_t$ could not identify which action is better. The MDP structure of the bad example is presented in Figure 4(a) where s_0 is the fixed initial state and the reward function is set to 0. Clearly, the optimal policy is $\pi^*(s_0) = a_0$ and $\pi^*(s_1) = a_0$, achieving the minimum unsafe probability $\varepsilon_{\min} = 0.28$. The sub-optimal policy $\pi_{\text{sub}}(s_0) = a_0$ and $\pi_{\text{sub}}(a_0|s_1) = \pi_{\text{sub}}(a_1|s_1) = \frac{1}{2}$ has unsafe probability 0.32. We run our method and SafeQ in this example and find that, even setting $\varepsilon = 0.28$ for SafeQ (the SafeQ(0.28) in Figure 4(b)), it only learns the sub-optimal policy π_{sub} with unsafe probability 0.32. This is because at state s_1, the unsafe probability of executing action a_0 or a_1 is smaller than $\varepsilon_{\min} = 0.28$ and the action mask $\mathcal{C}_\varepsilon(s_1)$ with $\varepsilon = 0.28$ could not distinguish the difference between a_0 and a_1 since $Q_t \equiv 0$ by our MDP. Thus, SafeQ may only uniformly choose actions at s_1. The same argument holds for larger ε, e.g., $\varepsilon = 0.36$ (SafeQ(0.36) in Figure 4(b)) which is the unsafe probability of policy $\pi(s_0) = a_0, \pi(s_1) = a_1$. In contrast, our safe Q-learning with optimal action masks converges to a minimum unsafe probability of 0.28, which aligns with our Theorem 4.4.

B. Theortical Proofs

B.1. Proofs of Lemma 3.1

Proof: For the contraction property, $\forall(s,a) \in \mathcal{S} \times \mathcal{A}$, we have

$$\begin{aligned}
&\Big|\mathcal{B}_\zeta Q_1(s,a) - \mathcal{B}_\zeta Q_2(s,a)\Big| \\
&= \Big|r(s,a) + \gamma \mathbb{E}_{s'\sim P(\cdot|s,a)} \max_{a'\in\mathcal{C}_\zeta(s')} Q_1(s',a') - r(s,a) - \gamma \mathbb{E}_{s'\sim P(\cdot|s,a)} \max_{a'\in\mathcal{C}_\zeta(s')} Q_2(s',a')\Big|, \\
&= \Big|\gamma \mathbb{E}_{s'\sim P(\cdot|s,a)}\Big[\max_{a'\in\mathcal{C}_\zeta(s')} Q_1(s',a') - \max_{a'\in\mathcal{C}_\zeta(s')} Q_2(s',a')\Big]\Big| \\
&\le \gamma \mathbb{E}_{s'\sim P(\cdot|s,a)}\Big[\max_{a'\in\mathcal{C}_\zeta(s')} \Big|Q_1(s',a') - Q_2(s',a')\Big|\Big] \quad \Big(\Big|\max_a Q_1 - \max_a Q_2\Big| \le \max_a \Big|Q_1 - Q_2\Big|\Big) \\
&\le \gamma \max_{s',a'} \Big|Q_1(s',a') - Q_2(s',a')\Big|,
\end{aligned}$$

and thus

$$\|\mathcal{B}_\zeta Q_1 - \mathcal{B}_\zeta Q_2\|_\infty \le \gamma \|Q_1 - Q_2\|_\infty,$$

where $\|Q\|_\infty := \max_{s,a} |Q(s,a)|$ is the L_∞-norm. Therefore $\mathcal{B}_\zeta$ is a γ-contraction (with respect to the L_∞-norm).

Let Q^*_ζ be the unique fixed point of $\mathcal{B}_\zeta$ and $\pi^*_\zeta(s) := \arg\max_{a\in\mathcal{C}_\zeta(s)} Q^*_\zeta(s,a)$ be the corresponding policy induced by Q^*_ζ.

For the safety optimality of π^*_ζ, since $\pi^*_\zeta(s)$ always chooses actions among the optimal action mask $\mathcal{C}_\zeta(s), \forall s \in \mathcal{S}$, which masks out the action with non-minimal unsafe probability, the unsafe probability of π^*_ζ is the same as π^* and thus safety optimal, i.e.,

$$\pi^*_\zeta \in \arg\min_\pi \Pr_{\substack{s_0\sim\rho,\\ \tau\sim(\pi,P)}} [\exists s_t \in \tau : s_t \in \mathcal{S}_u].$$

For the reward optimality, this follows from the fact that π^* will only select actions in $\mathcal{C}_\zeta(s)$ since $\pi^* \in \arg\min_\pi \Pr_{\substack{s_0\sim\rho,\\ \tau\sim(\pi,P)}} [\exists s_t \in \tau : s_t \in \mathcal{S}_u]$. Therefore the Bellman backup of Q^{π^*} is the same as that of Q^*_ζ, which implies that

$$Q^{\pi^*}(s,a) = Q^*_\zeta(s,a), \forall s \in \mathcal{S}, a \in \mathcal{C}_\zeta(s).$$

As a result, π^*_ζ is reward optimal.

□

B.2. Proof of Lemma 4.2

Proof: First, if $s \in \mathcal{S}_u$, then $\psi^\pi(s,a) = 1$ by definition and the result holds clearly. Therefore we assume $s \notin \mathcal{S}_u$. Let $\tau = \{s_0 = s, a_0 = a, s_1, \cdots, s_t, a_t, \cdots\} \sim (\pi, P)$ be a sampling trajectory starting from state $s_0 = s$ and action $a_0 = a$ using policy π with transition function P. Then we have

$$\begin{aligned}
\psi^\pi(s,a) &= \mathbb{E}_{\tau\sim(\pi,P)}\Big[\max_{s_t\in\tau} \mathbb{I}[s_t \in \mathcal{S}_u] \;\Big|\; s_0 = s, a_0 = a\Big] \\
&= \mathbb{E}_{s'\sim P(\cdot|s,a), a'\sim\pi(\cdot|s')}\Big[\mathbb{E}_{\tau_1\sim(\pi,P)} \max_{s_t\in\tau_1} \mathbb{I}[s_t \in \mathcal{S}_u] \;\Big|\; s_1 = s', a_1 = a'\Big] \\
&= \mathbb{E}_{s'\sim P(\cdot|s,a), a'\sim\pi(\cdot|s')}\Big[\psi^\pi(s',a')\Big]
\end{aligned}$$

where $\tau_1 = \{s_1 = s', a_1 = a', \ldots, s_t, \ldots\} \sim (\pi, P)$ is the sampling trajectory starting from state s' and action a'. This completes the proof.

□

B.3. Proof of Theorem 4.3

Proof: We convert the update of ψ_t to an equivalent *undiscounted* Q-function update as follows. Consider the following *undiscounted* infinite horizon MDP $\mathcal{M}_u := (\mathcal{S}, \mathcal{A}, r_u, P, \rho)$ where

$$r_u(s,a) = \begin{cases} -1, & s \in \mathcal{S}_u, \\ 0, & \text{otherwise.} \end{cases}$$

Let $Q_u^\pi(s,a) := \lim_{N\to\infty} Q_N^\pi(s,a)$, where

$$Q_N^\pi(s,a) := \mathbb{E}_\pi\Big[\sum_{i=0}^{N-1} r_u(s_i,a_i) \mid s_0 = s, a_0 = a\Big].$$

Since $\mathcal{S}_u$ consists of absorbing states, $Q_u^\pi(s) > -\infty, \forall \pi, s$. Clearly, $\Pr_{\tau\sim(\pi,P)}[\exists s_t \in \tau : s_t \in \mathcal{S}_u | s_0 = s, a_0 = a] = -Q_u^\pi(s,a)$. Then $\psi^\pi(s,a) = -Q_u^\pi(s,a)$ and $\psi^*(s,a) = -Q_u^*(s,a)$, where $Q_u^*(s,a)$ is the optimal undiscounted Q-value in $\mathcal{M}_u$. Furthermore,

$$\begin{aligned} &\psi_{t+1}(s,a) = \max\{\mathbb{I}[s \in \mathcal{S}_u], \mathbb{E}_{s'\sim P(\cdot|s,a)} \min_{a'\in\mathcal{A}} \psi_t(s',a')\} \\ \Leftrightarrow \quad &Q_{u,t+1}(s,a) = r_u(s,a) + \mathbb{E}_{s'\sim P(\cdot|s,a)} \max_{a'\in\mathcal{A}} [Q_{u,t}(s',a')] =: \mathcal{L}Q_{u,t}(s,a). \end{aligned}$$

According to Theorem 7.3.10. of Puterman (1994), starting from $Q_{u,0} = 0$, the iterative update $Q_{u,t+1} = \mathcal{L}Q_{u,t}$ converges monotonically to Q_u^*, which is equivalent to the convergence of ψ_t to ψ^* starting from $\psi_0 = 0$. In this way we complete the proof.

□

B.4. Proof of Theorem 4.4

Proof. First consider the ψ update,

$$\begin{aligned} \psi_{t+1}(s_t,a_t) &= (1-\beta_t)\psi_t(s_t,a_t) + \beta_t \max\{\mathbb{I}[s_t \in \mathcal{S}_u], \min_{a'\in\mathcal{A}} \psi_t(s_{t+1},a')\} \\ &= \psi_t(s_t,a_t) + \beta_t\Big(F(\psi_t; s_t, a_t) - \psi_t(s_t,a_t) + M_{t+1}\Big). \end{aligned} \tag{24}$$

where $F : \mathbb{R}^{|\mathcal{S}||\mathcal{A}|} \to \mathbb{R}^{|\mathcal{S}||\mathcal{A}|}$ is the ψ backup operator defined by

$$F(\psi; s, a) := \max\{\mathbb{I}[s \in \mathcal{S}_u], \mathbb{E}_{s'\sim P(\cdot|s,a)} \min_{a'\in\mathcal{A}} \psi(s',a')\} \tag{25}$$

and the noise term

$$M_{t+1} := \max\{\mathbb{I}[s_t \in \mathcal{S}_u], \min_{a'\in\mathcal{A}} \psi_t(s_{t+1},a')\} - \max\{\mathbb{I}[s_t \in \mathcal{S}_u], \mathbb{E}_{s'\sim P(\cdot|s_t,a_t)} \min_{a'\in\mathcal{A}} \psi_t(s',a')\}. \tag{26}$$

The associated ordinary differential equation (ODE), with initial condition $\psi(0) = \psi_0 = 0$, is

$$\dot{\psi} = F(\psi) - \psi. \tag{27}$$

The following lemma is borrowed from Borkar (2008), Page 127, Theorem 4.

Lemma B.1. (Borkar, 2008). *Consider a stochastic approximation algorithm*

$$X_{t+1} = X_t + \beta_t[F(X_t) - X_t + M_{t+1}], \quad t \geq 0,$$

where $X_t \in \mathbb{R}^d$, $F : \mathbb{R}^d \to \mathbb{R}^d$. Let the associated ODE be $\dot{x}(t) = F(x(t)) - x(t)$ starting from $x(0) = X_0$. Define the L_∞-norm $\|X\|_\infty := \max_{i\in[d]} |X_i|$. Assume:

(A1) (Non-expansiveness of F). $\forall X, Y \in \mathbb{R}^d$, $\|F(X) - F(Y)\|_\infty \leq \|X - Y\|_\infty$.

(A2) (Boundedness). $\|\sup_{t\geq 0} X_t\|_\infty < +\infty$ *almost surely.*

(A3) (Diminishing step sizes). *The sequence* $\{\beta_t\}_{t\geq 0}$ *satisfies* $0 < \beta_t \leq 1$ *and* $\sum_{t\geq 0} \beta_t = +\infty, \quad \sum_{t\geq 0} \beta_t^2 < +\infty$.

(A4) (Martingale difference noise). *Let* $\mathcal{F}_t$ *be the* σ*-algebra generated by* $(X_0, M_0, \ldots, X_t, M_t)$. *There exists* $K > 0$ *such that* $\forall t$, $\mathbb{E}[M_{t+1}|\mathcal{F}_t] = 0$ *and* $\mathbb{E}[\|M_{t+1}\|_\infty^2|\mathcal{F}_t] \leq K(1 + \|X_t\|_\infty^2)$.

Under the above assumptions, if $H := \{X \in \mathbb{R}^d \mid F(X) = X\}$ *is non-empty, then* $X_t \to X^*$ *a single point in* H *(depending on* X_0*).*

Claim. The ψ backup operator F in Equation (25) is non-expansive.

- Proof of Claim. This is due to, for any $\psi, \phi \in \mathbb{R}^{|\mathcal{S}||\mathcal{A}|}$, and for any $s \in \mathcal{S}, a \in \mathcal{A}$,

$$\begin{aligned}
&\Big|F(\psi; s, a) - F(\phi; s, a)\Big| \\
&= \Big|\max\{\mathbb{I}[s \in \mathcal{S}_u], \mathbb{E}_{s'\sim P(\cdot|s,a)} \min_{a'\in\mathcal{A}} \psi(s', a')\} - \max\{\mathbb{I}[s \in \mathcal{S}_u], \mathbb{E}_{s'\sim P(\cdot|s,a)} \min_{a'\in\mathcal{A}} \phi(s', a')\}\Big| \\
&\leq \Big|\mathbb{E}_{s'\sim P(\cdot|s,a)} \min_{a'\in\mathcal{A}} \psi(s', a') - \mathbb{E}_{s'\sim P(\cdot|s,a)} \min_{a'\in\mathcal{A}} \phi(s', a')\Big| \\
&\leq \max_{s'\in\mathcal{S}, a'\in\mathcal{A}} \Big|\psi(s', a') - \phi(s', a')\Big| \\
&= \|\psi - \phi\|_\infty.
\end{aligned}$$

Claim. ψ_t is bounded: $\forall t \geq 0$, $\|\psi_t\|_\infty \leq 1$.

- Proof of Claim. We prove this by induction. First $\psi_0 = 0$ and the claim holds at $t = 0$. Assume now the claim holds at t. For $t + 1$, if $(s, a) \neq (s_t, a_t)$ then $|\psi_{t+1}(s, a)| = |\psi_t(s, a)| \leq 1$; if $(s, a) = (s_t, a_t)$, then, since $0 < \beta_t \leq 1$ we have,

$$\begin{aligned}
|\psi_{t+1}(s_t, a_t)| &\leq (1 - \beta_t)|\psi_t(s_t, a_t)| + \beta_t|\max\{\mathbb{I}[s_t \in \mathcal{S}_u], \min_{a'\in\mathcal{A}} \psi_t(s_{t+1}, a')\}| \\
&\leq (1 - \beta_t) + \beta_t = 1.
\end{aligned}$$

  Therefore $\|\psi_{t+1}\|_\infty \leq 1$ and then the claim holds for all $t \geq 0$.

Claim. The noise term M_{t+1} in Equation (26) satisfies: $\forall t$, $\mathbb{E}[M_{t+1}|\mathcal{F}_t] = 0$ and $\mathbb{E}[\|M_{t+1}\|_\infty^2|\mathcal{F}_t] \leq 4$.

- Proof of Claim. First, using $|\psi_t(s, a)| \leq 1$ and considering whether $s_t \in \mathcal{S}_u$ or not, one can easily see that

$$\begin{aligned}
\mathbb{E}[M_{t+1}|\mathcal{F}_t] &= \mathbb{E}_{s_{t+1}\sim P(\cdot|s_t,a_t)}\Big[\max\{\mathbb{I}[s_t \in \mathcal{S}_u], \min_{a'\in\mathcal{A}} \psi_t(s_{t+1}, a')\Big] \\
&\quad - \max\Big\{\mathbb{I}[s_t \in \mathcal{S}_u], \mathbb{E}_{s'\sim P(\cdot|s_t,a_t)}\Big[\min_{a'\in\mathcal{A}} \psi_t(s', a')\Big]\Big\} \\
&= 0.
\end{aligned}$$

  Second, again by $\|\psi_t\|_\infty \leq 1$,

$$\begin{aligned}
&\mathbb{E}[\|M_{t+1}\|_\infty^2|\mathcal{F}_t] \\
&= \mathbb{E}_{s_{t+1}\sim P(\cdot|s_t,a_t)}\Big[\Big|\max\{\mathbb{I}[s_t \in \mathcal{S}_u], \min_{a'\in\mathcal{A}} \psi_t(s_{t+1}, a') - \max\Big\{\mathbb{I}[s_t \in \mathcal{S}_u], \mathbb{E}_{s'\sim P(\cdot|s_t,a_t)}\Big[\min_{a'\in\mathcal{A}} \psi_t(s', a')\Big]\Big\}\Big|^2\Big] \\
&\leq 2\mathbb{E}_{s_{t+1}\sim P(\cdot|s_t,a_t)}\Big[\Big|\max\{\mathbb{I}[s_t \in \mathcal{S}_u], \min_{a'\in\mathcal{A}} \psi_t(s_{t+1}, a')\Big|^2 \\
&+ \Big|\max\Big\{\mathbb{I}[s_t \in \mathcal{S}_u], \mathbb{E}_{s'\sim P(\cdot|s_t,a_t)}\Big[\min_{a'\in\mathcal{A}} \psi_t(s', a')\Big]\Big\}\Big|^2\Big] \\
&\leq 4.
\end{aligned}$$

  This proves the claim.

As a result, the ψ update in Equation (24) satisfies the Assumption (A1)-(A4) in Lemma B.1. Therefore, according to the Lemma B.1, ψ_t converges to a single point $a \in H = \{F(\psi) = \psi\}$. Since $\psi(0) = 0$, our ODE in Equation (27) converges to $a = \psi^*$ by Theorem 4.3. Therefore the ψ update in Equation (24) will converge: $\psi_t \to a = \psi^*$, starting from $\psi_0 = 0$.

Next, we consider the Q update. We expand the Equation (13) as follows:

$$\begin{aligned}
Q_{t+1}(s_t, a_t) &= Q_t(s_t, a_t) + \beta_t\Big(r_t + \gamma \max_{a \in \mathcal{C}_{\zeta_t}(s_{t+1})} Q_t(s_{t+1}, a) - Q_t(s_t, a_t)\Big) \\
&= Q_t(s_t, a_t) + \beta_t\Big(r_t + \gamma \max_{a \in \mathcal{C}_{\zeta}(s_{t+1})} Q_t(s_{t+1}, a) - Q_t(s_t, a_t) \\
&+ \gamma \max_{a \in \mathcal{C}_{\zeta_t}(s_{t+1})} Q_t(s_{t+1}, a) - \gamma \max_{a \in \mathcal{C}_{\zeta}(s_{t+1})} Q_t(s_{t+1}, a)\Big) \\
&= Q_t(s_t, a_t) + \beta_t\Big(r_t + \gamma \mathbb{E}_{s'}\Big[\max_{a \in \mathcal{C}_{\zeta}(s')} Q_t(s', a)\Big] - Q_t(s_t, a_t) \\
&+ \gamma \max_{a \in \mathcal{C}_{\zeta}(s_{t+1})} Q_t(s_{t+1}, a) - \gamma \mathbb{E}_{s'}\Big[\max_{a \in \mathcal{C}_{\zeta}(s')} Q_t(s', a)\Big] \\
&+ \gamma \max_{a \in \mathcal{C}_{\zeta_t}(s_{t+1})} Q_t(s_{t+1}, a) - \gamma \max_{a \in \mathcal{C}_{\zeta}(s_{t+1})} Q_t(s_{t+1}, a)\Big)
\end{aligned} \tag{28}$$

where we recall that

$$\mathcal{C}_{\zeta}(s) = \{b \in \mathcal{A} \mid \psi^*(s, b) \le \zeta(s) = \min_{a \in \mathcal{A}} \psi^*(s, a)\}$$

and $\mathcal{C}_{\zeta_t}(s)$ is the set of minimizers of $\psi_t(s_t, \cdot)$:

$$\mathcal{C}_{\zeta_t}(s) := \{b \in \mathcal{A} \mid \psi_t(s, b) \le \zeta_t(s) = \min_{a \in \mathcal{A}} \psi_t(s, a) + \kappa\},$$

where $\kappa > 0$ is a small enough constant satisfying $\psi^*(s, b) + 2\kappa < \psi^*(s, e)$ for any $s \in \mathcal{S}$, $b \in \mathcal{C}_{\zeta}(s)$ and $e \in \mathcal{A} \setminus \mathcal{C}_{\zeta}(s)$.

Since ψ_t converges to ψ^* as $t \to +\infty$, we have for any $\epsilon > 0$, there exists some $T > 0$ such that when $t > T$, $|\psi_t(s, a) - \psi^*(s, a)| < \epsilon, \forall s \in \mathcal{S}, a \in \mathcal{A}$. Taking $\epsilon = \frac{\kappa}{2}$, for any $b \in \mathcal{C}_{\zeta}(s)$ and $e \in \mathcal{A} \setminus \mathcal{C}_{\zeta}(s)$, we have

$$\psi_t(s, b) + \kappa < \psi^*(s, b) + \epsilon + \kappa \le \psi^*(s, e) - 2\kappa + \epsilon + \kappa \le \psi_t(s, e) + 2\epsilon - \kappa = \psi_t(s, e). \tag{29}$$

Claim. for any $b \in \mathcal{C}_{\zeta}(s)$, we have $b \in \mathcal{C}_{\zeta_t}(s)$, and for any $e \in \mathcal{A} \setminus \mathcal{C}_{\zeta}(s)$, we have $e \in \mathcal{A} \setminus \mathcal{C}_{\zeta_t}(s)$.

- Proof of Claim. Since $\zeta(s) = \psi^*(s, b)$, we also have $\psi_t(s, b) \in [\zeta(s) - \epsilon, \zeta(s) + \epsilon]$. Then

  $$\zeta_t(s) = \min_{a \in \mathcal{A}} \psi_t(s, a) + \zeta = \min_{b \in \mathcal{C}_{\zeta}(s)} \psi_t(s, b) + \zeta \ge \zeta(s) - \epsilon + \zeta = \zeta(s) + \epsilon,$$

  where the second equality is due to Equation (29) and the final equality is due to $\epsilon = \frac{\zeta}{2}$. Thus $\psi_t(s, b) \le \zeta(s) + \epsilon \le \zeta_t(s)$. This shows that $\mathcal{C}_{\zeta}(s) \subset \mathcal{C}_{\zeta_t}(s)$.

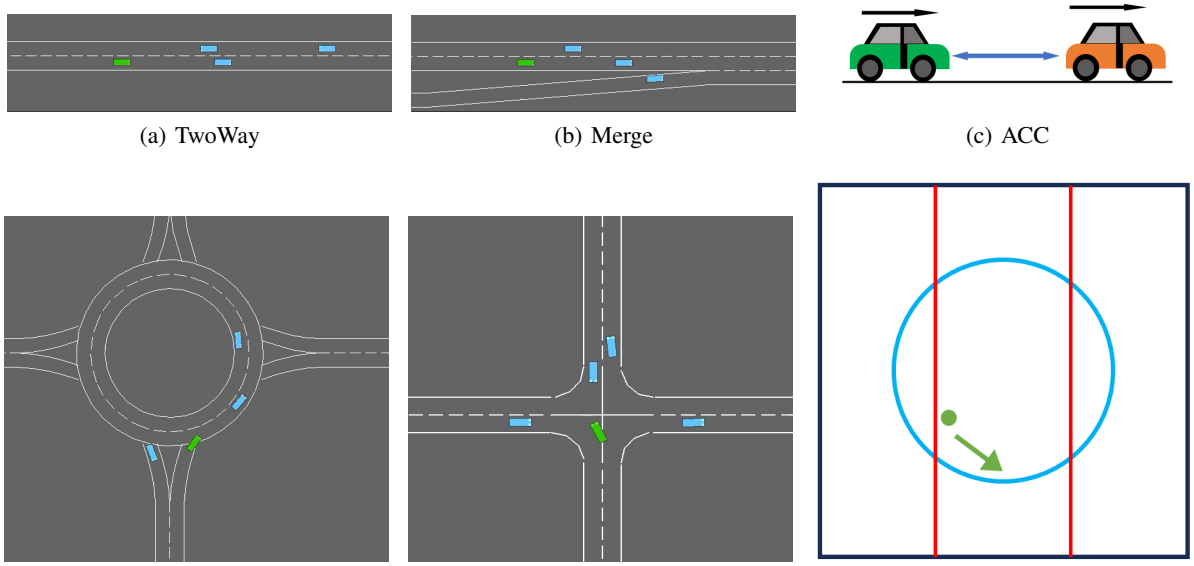
  On the other hand, again by Equation (29), $\psi_t(s, e) > \psi_t(s, b) + \zeta \ge \min_{a \in \mathcal{A}} \psi_t(s, a) + \zeta = \zeta_t(s)$. This shows that $\mathcal{A} \setminus \mathcal{C}_{\zeta}(s) \subset \mathcal{A} \setminus \mathcal{C}_{\zeta_t}(s)$.

Therefore $\mathcal{C}_{\zeta_t}(s)$ will be the same as $\mathcal{C}_{\zeta}(s)$ after $t > T$. Taking $t > T$, the Equation (28) becomes

$$\begin{aligned}
Q_{t+1}(s_t, a_t) = Q_t(s_t, a_t) + \beta_t\Big(&r_t + \gamma \mathbb{E}_{s'}\Big[\max_{a \in \mathcal{C}_{\zeta}(s')} Q_t(s', a)\Big] - Q_t(s_t, a_t) \\
&+ \gamma \max_{a \in \mathcal{C}_{\zeta}(s_{t+1})} Q_t(s_{t+1}, a) - \gamma \mathbb{E}_{s'}\Big[\max_{a \in \mathcal{C}_{\zeta}(s')} Q_t(s', a)\Big]\Big),
\end{aligned} \tag{30}$$

whose convergence to Q^*_{ζ} is guaranteed by classical Q-learning theory (Borkar, 2008). We complete the proof.

□



(a) TwoWay (b) Merge (c) ACC

(d) Roundabout (e) Intersection (f) Circle

Figure 5. The visualization of our benchmarks.

C. Experiment Details

C.1. Benchmarks

In highway-env (Leurent, 2018), the state space can either be of kinetics type, which describes a list of nearby vehicles by a set of location and velocity features, or be of time-to-collision type, which represents the predicted time-to-collision of observed vehicles on the same road as the ego car. In TwoWay and Merge, we select the time-to-collision type for state space, while in Roundabout and Intersection, we select the kinetics type. The action space consists of discrete meta actions, which can change the target lane and speed that are used as setpoints for some low-level controllers, so that the ego car can automatically follow the road at the desired speed. The rewards encourage agents to move fast and achieve their corresponding goals, which are:

- **TwoWay.** The goal is to drive on a two-way lane as fast as possible with incoming traffic.
- **Merge.** The goal of ego car is to maintain a high speed while making room for the incoming cars to the access ramp so that they can merge into the traffic safely.
- **Roundabout.** Facing the complex traffic flow, the ego car need to pass a roundabout safely following a pre-defined route with high speed.
- **Intersection.** In the dense traffic, the ego car should cross a four-way intersection while avoiding crashing with other cars.

For the other two safe control tasks, we give their detailed environment setup here:

- **ACC.** (Anderson et al., 2020) The ego car aims to follow a lead car as closely as possible without crashing into it. The state $s \in \mathcal{S}$ consists of the relative distance d, the speed of ego car v_e and the speed of lead car v_ℓ. There are three actions: -1 (decelerate), 0 (idle), 1 (accelerate). The acceleration Δv_ℓ of the lead car follows from a Gaussian distribution $\Delta v_\ell \sim \mathcal{N}(0, 0.2)$. We do not allow the lead car and ego car to go backward. The lead car is taken as reference. When $d \leq -10$ (staying too far) or $d \geq 0$ (crashing), the episode is terminated and agent receives crash reward 0; otherwise the reward $r = 10 + d$. The maximal episode length is 100.
- **Circle.** (Achiam et al., 2017; Wagener et al., 2021) A point robot gets rewards for fast circular movement, but is constrained to stay inside the area restricted by two walls which is narrower than the target circle. The state

Table 2. Hyperparameters for all DQN-based algorithms

Algorithm	Parameter	Value
Shared	optimizer	Adam
	discount factor	0.99
	Q-network learning rate	$5 \cdot 10^{-4}$
	batch size	64
	update every	1
	initial ϵ-greedy exploration rate	1
	ϵ decay	0.995
	ϵ min	0.01
	number of random seeds	6
SPOM (ours)	SA-REF ψ learning rate	$5 \cdot 10^{-4}$
	polarization function	$f_{\text{pol}}(x) = 10\log(x)$
SPOM_PER (ours)	priority exponent α	0.6
	importance sampling exponent θ	0.4
SafeQ and **Recovery**	safety critic Q_{risk} learning rate	$5 \cdot 10^{-4}$
	$\varepsilon_{\text{risk}}$	0.1
	γ_{risk}	0.99
RCDQN	Lagrange multiplier learning rate	0.001
RewsDQN	shaped crash reward	-10

Table 3. Hyperparameters for PPO-based algorithms

Algorithm	Parameter	Value
Shared	optimizer	Adam
	discount factor	0.99
	learning rates of actor and critic	$3 \cdot 10^{-4}$
	GAE parameter	0.97
	clip ratio	0.2
RESPO	REF learning rate	$1 \cdot 10^{-4}$
	Lagrange multiplier learning rate	$5 \cdot 10^{-5}$
PPOLag	Lagrange multiplier learning rate	0.001

$s = (x, y, \dot{x}, \dot{y})$, where (x, y) is the xy-coordinate of the agent, $\dot{x}$ and $\dot{y}$ are the speeds at x-direction and y-direction respectively. The action space consists of one idle action and the following eight directions for acceleration: $(\cos(k\pi/4), \sin(k\pi/4)), k = 0, 1, \ldots, 7$. After choosing an action a, the corresponding acceleration vector is $a_{\text{const}} \cdot a$, where $a_{\text{const}} > 0$ is a constant acceleration scalar. The maximal episode length is 100. The rewards encourage circular movement of radius R^* and the unsafe state set $\mathcal{S}_u$ restricts agent to stay inside the region $|x| \leq x_{\max}$:

$$\mathcal{S}_u = \{s = (x, y, \dot{x}, \dot{y}) \in \mathcal{S} \mid |x| \leq x_{\max}\},$$

$$r(s, a) = \begin{cases} \dfrac{(\dot{x}, \dot{y}) \cdot (-y, x)}{1 + |\ \|(x, y)\|_2 - R^*\ |}, & \text{if } s \notin \mathcal{S}_u, \\ -1, & \text{otherwise.} \end{cases}$$

For our experiments, we take $a_{\text{const}} = 1$, $x_{\max} = 2.5$ and $R^* = 5$.

C.2. Algorithms

C.2.1. Implementation

The original SafeQ (Srinivasan et al., 2020) and Recovery RL (Thananjeyan et al., 2020) are based on actor-critic frameworks. Here we give detailed descriptions on their adaption to the DQN implementations.

- **SafeQ.** (Srinivasan et al., 2020) A Q-network Q and a safety critic Q_{risk} are required with their corresponding target networks $\bar{Q}$ and $\bar{Q}_{\text{risk}}$. The safety critic Q_{risk} is used to predict the future *discounted* risk for safety violation, with discounted risk factor $\gamma_{\text{risk}} \in (0, 1)$. **SafeQ** performs direct $\varepsilon_{\text{risk}}$-masking:

$$\tilde{\pi}(s) = \arg \max_{a \in \mathcal{C}_{\varepsilon_{\text{risk}}}(s)} Q(s, a),$$

  where $\mathcal{C}_{\varepsilon_{\text{risk}}}(s) = \{b \in \mathcal{A} \mid Q_{\text{risk}}(s, b) \leq \varepsilon_{\text{risk}}\}$ and $\varepsilon_{\text{risk}} > 0$ is a fixed state-agnostic threshold. The target of safety critic is computed through

$$y_t^{\text{risk}} = \mathbb{I}[s_t \in \mathcal{S}_u] + (1 - \mathbb{I}[s_t \in \mathcal{S}_u])\gamma_{\text{risk}} \bar{Q}_{\text{risk}}(s_{t+1}, \tilde{\pi}(s_{t+1})),$$

  to estimate the unsafe risk under current masked policy $\tilde{\pi}$, and the target of Q-network is also based on $\tilde{\pi}$:

$$y_t^Q = r_t + \gamma \bar{Q}(s_{t+1}, \tilde{\pi}(s_{t+1})).$$

- **Recovery.** (Thananjeyan et al., 2020) Here we also require a Q-network and a safety critic Q_{risk}. But the final policy π is composed of a task policy π_{task} and a recovery policy π_{rec}, where

$$\pi_{\text{task}}(s) = \arg \max_{a \in \mathcal{A}} Q(s, a), \quad \pi_{\text{rec}}(s) = \arg \min_{a \in \mathcal{A}} Q_{\text{risk}}(s, a),$$

  and then π is obtained via an intervention-based scheme: if the action a^{task} chosen by the task policy π_{task} has unsafe risk greater than $\varepsilon_{\text{risk}}$, then it will be overtaken by the recovery policy π_{rec} which will choose the action with minimal risk:

$$\pi(s) = \begin{cases} a^{\text{task}} = \pi_{\text{task}}(s), & \text{if } Q_{\text{risk}}(s, a^{\text{task}}) \leq \varepsilon_{\text{risk}}, \\ \pi_{\text{rec}}(s), & \text{otherwise.} \end{cases}$$

  Note that according to Thananjeyan et al. (2020), the task policy is trained on the task buffer $(s_t, a_t^{\text{task}}, r_t, s_{t+1})$, and the Q target is computed by

$$y_t^Q = r_t + \gamma \bar{Q}(s_{t+1}, \pi_{\text{task}}(s_{t+1})),$$

  while the safety critic Q_{risk} is trained on the real buffer (s_t, a_t, r_t, s_{t+1}) and the Q_{risk} target is

$$y_t^{\text{risk}} = \mathbb{I}[s_t \in \mathcal{S}_u] + (1 - \mathbb{I}[s_t \in \mathcal{S}_u])\gamma_{\text{risk}} \bar{Q}_{\text{risk}}(s_{t+1}, \pi(s_{t+1})),$$

  in order to estimate the unsafe risk under current policy π.

Since we focus on the persistent safety, for both **SafeQ** and **Recovery**, we will take $\gamma_{\text{risk}} = \gamma$ the discount factor of the MDP.

- **RCDQN.** Following Tessler et al. (2019), we constrains the reward through $r - \lambda c$, where λ is the Lagrange multiplier and c is the cost signal corresponding to crash, and then we use DQN to learn upon this constrained reward. Note that the Lagrange multiplier is updated via the on-policy samples, which relates to the safety of current policy more closely.
- **RewsDQN.** This reward-shaping-based DQN tries to inform the agent to avoid crashing via a strong penalty when a crash occurs. The shaped crash reward is set to be -10 uniformly.

For completeness, we also choose some of the Proximal Policy Optimization (PPO) (Schulman et al., 2017) based based algorithms under CMDP framework to our tasks. We include the following representative ones:

- **RESPO.** Ganai et al. (2023) propose the Reachability Estimation Safe Policy Optimization, where a state-dependent reachability estimation function (REF) is learned to estimate the unsafe probability under current policy. Then **RESPO** optimizes the rewards in safe regions while maintaining safety through Lagrangian methods, and optimizes the costs in unsafe regions.
- **PPOLag.** (Ray et al., 2019) A classical primal-dual method, which uses a Lagrangian relaxation to transfer the original CMDP-based constrained optimization problem to an unconstrained one and then adapts to PPO.
- **PPOBarrier.** (Yang et al., 2023) An extension of PPO that incorporates control-theoretic barrier certificates to ensure policy safety by enforcing constraint satisfaction during both training and execution.

Table 4. The SWU scores of all algorithms, including **RESPO** and **PPOLag**.

SWU Score↑	TwoWay	Merge	Roundabout	Intersection	ACC	Circle	Overall
SPOM_PER (ours)	**0.98**	0.94	**1.04**	0.68	**1.07**	**0.96**	**0.95**
SPOM (ours)	0.88	**0.95**	0.34	**0.96**	0.87	0.37	0.73
SafeQ	0.73	0.84	0.54	0.75	0.50	0.15	0.59
Recovery	0.34	0.66	0.69	0.66	0.85	0.25	0.58
RCDQN	0.41	0.81	0.41	0.57	0.46	0.14	0.47
RewsDQN	0.33	0.92	0.41	0.50	0.47	0.11	0.46
DQN	0.37	0.56	0.41	0.58	0.46	0.13	0.42
RESPO	0.06	0.09	0.09	0.11	0.06	0.01	0.07
PPOLag	0.06	0.03	0.13	0.11	0.11	0.02	0.08

C.2.2. HYPERPARAMETERS

Network Architectures. In Roundabout and Intersection, to deal with the kinetics type of state, which contains a list of features about nearby vehicles, we use the ego-attention-based architecture proposed by Leurent & Mercat (2019), which can handle the permutation of the list input and thus is suitable for these two tasks.

In TwoWay and Merge, we use two-layer multi-layer perceptron (MLP) networks, with hidden layer of size 128. In ACC and Circle, we also use two-layer MLP, with 256 hidden units.

Parameter Settings for DQN-based Algorithms. The exploration strategy is unified to be ϵ-greedy. In highway-env, we use replay buffers of size 15000 and hard update for target networks with hard update interval 512, following the default configuration in Leurent & Mercat (2019); while in ACC and Circle, we use the replay buffers of size 10^5 and soft update with soft update coefficient $5 \cdot 10^{-3}$ and interval 1, to ensure stable and fast learning. Other hyperparameters are unified across tasks and can be found in Table 2.

Parameter Settings for PPO-based Algorithms. According to Ganai et al. (2023), **RESPO** should maintain that the Lagrange multiplier learning $<$ the REF learning rate $<$ the actor and critic learning rates, in order to ensure a stable convergence to a local optimum, and so we use their default parameters. For **PPOLag**, the Lagrange multiplier learning rate is set to be larger than that of **RESPO**, to bias more towards constraint violation in the original primal-dual formulation. The detailed hyperparameters are listed in Table 3.

C.3. Full Results

C.3.1. MAIN EXPERIMENTS

The full training curves of all baselines (DQN-based and PPO-based) and **SPOM**, **SPOM_PER** are provided in Figure 6 and 7. We find that **RESPO** and **PPOLag** fail to optimize under our environments since the cost signal corresponding to a crash is very sparse. The full SWU scores are presented in Table 4, where **SPOM_PER** achieves the highest overall SWU score of 0.90, and **SPOM** secures the second-best score of 0.79. Both algorithms demonstrate significant improvement over baselines, with **SPOM_PER** achieving over 50% improvement compared to most baselines.

We also list the average crash rates and episode rewards over the last $\frac{1}{10}$ training steps and all random seeds, per algorithm and per task, in Table 5. **SPOM_PER** along with **SPOM** achieves the lowest or one of the lowest crash rates across almost all tasks while maintaining competitive or superior episode rewards.

Comparison with direct ε-masking approach **SafeQ**: Across all tasks, **SPOM_PER** and **SPOM** outperform **SafeQ** both in terms of crash rates and rewards. This again validates our insight that a state-agnostic masking threshold is not enough to ensure maximal safety.

Comparison with action-correction-based method **Recovery**: In ACC, **SPOM_PER** achieves the best SWU score of 1.07, outperforming **Recovery**, which sacrifices rewards to reduce crash rates. Similarly, in Circle, **SPOM_PER** and **SPOM** achieve significantly lower crash rates than **Recovery** while maintaining higher rewards. In Intersection, **Recovery** trains to be safe faster but converges to lower rewards compared to **SPOM_PER**. For other tasks, **SPOM_PER** and **SPOM** achieve a better trade-off between safety and rewards while being safer than **Recovery**.

Table 5. The crash rates and episode rewards (**mean and std**), averaged over the last $\frac{1}{10}$ training steps and six random seeds, of all compared algorithms. **SPOM** achieves the lowest or one of the lowest crash rates while maintaining competitive episode rewards.

		TwoWay	Merge	Roundabout	Intersection	ACC	Circle
Crash Rate ↓	SPOM_PER (ours)	**0.073**±0.011	0.043±0.008	0.055±0.027	0.086±0.025	**0.163**±0.011	**0.038**±0.010
	SPOM (ours)	0.085±0.019	**0.042**±0.006	0.167±0.105	0.069±0.014	0.200±0.013	0.104±0.016
	SafeQ	0.104±0.018	0.050±0.007	0.090±0.056	0.091±0.020	0.351±0.013	0.255±0.021
	Recovery	0.134±0.016	0.063±0.007	0.062±0.031	**0.068**±0.023	0.205±0.011	0.119±0.029
	RCDQN	0.176±0.061	0.052±0.008	0.120±0.029	0.113±0.028	0.350±0.009	0.268±0.018
	RewsDQN	0.209±0.097	0.042±0.008	0.122±0.067	0.130±0.027	0.344±0.018	0.350±0.043
	DQN	0.198±0.079	0.074±0.014	0.115±0.047	0.118±0.012	0.351±0.008	0.288±0.035
	RESPO	0.704±0.391	0.228±0.247	**0.047**±0.104	0.403±0.169	0.932±0.062	0.239±0.017
	PPOLag	0.509±0.354	0.406±0.276	0.140±0.184	0.355±0.209	0.729±0.043	0.148±0.022
Episode Reward ↑	SPOM_PER (ours)	10.124±0.370	1.883±0.035	**0.985**±0.324	4.433±0.601	**78.393**±0.558	**279.432**±2.865
	SPOM (ours)	10.520±0.389	1.823±0.028	0.947±0.218	4.978±0.210	78.337±0.551	279.432±2.865
	SafeQ	**10.636**±0.163	1.929±0.043	0.799±0.111	5.115±0.112	73.013±0.551	268.774±4.754
	Recovery	6.392±0.279	1.897±0.028	0.637±0.413	3.208±1.351	78.296±0.320	216.606±91.921
	RCDQN	10.217±0.567	**1.935**±0.049	0.818±0.256	4.844±0.818	73.026±0.818	273.679±1.308
	RewsDQN	9.775±0.472	1.764±0.075	0.827±0.177	4.932±0.286	72.571±1.084	209.573±85.724
	DQN	10.284±0.250	1.919±0.040	0.789±0.292	**5.125**±0.121	73.278±1.077	270.404±5.609
	RESPO	6.099±3.568	0.900±0.384	0.071±0.103	3.294±1.476	22.396±7.868	7.179±1.662
	PPOLag	4.860±3.870	0.535±0.690	0.298±0.352	3.024±1.812	37.752±2.766	25.753±4.506

Comparison with CMDP-based approach **RCDQN**, **RESPO**, and **PPOLag**: The DQN-based **RCDQN** has stable performance but fails to guide the policy effectively in terms of safety due to sparse cost signals. The PPO-based algorithms, **RESPO** and **PPOLag**, suffer from high variance and fail to optimize either safety or rewards effectively. In highway-env, their performance shows inconsistency between crash rates and rewards, and in ACC and Circle, they converge prematurely to suboptimal solutions. Overall, CMDP-based methods struggle with sparse cost signals in our tasks, leading to subpar performance compared to **SPOM_PER** and **SPOM**.

Comparison with reward-shaping approach **RewsDQN**: While **RewsDQN** achieves competitive crash rates in some tasks like Merge, its episode rewards are significantly lower. In Circle, **RewsDQN** fails to achieve comparable crash rates and rewards as **SPOM_PER** and **SPOM**, demonstrating the difficulty of balancing safety through reward shaping alone.

Note on vanilla **DQN**: Vanilla **DQN** can reduce crash rates to a certain level due to early termination in tasks. However, its performance in balancing crash rates and rewards falls far behind **SPOM_PER** and **SPOM**, which consistently achieve better safety and reward trade-offs.

C.3.2. Additional Ablation Studies

We provide the training curves of our ablation studies for the remaining four tasks in Figure 8.

Once again, we can find the over-conservative phenomena of direct optimal action masking "OAM" in Merge and Roundabout (the first two rows in Figure 8), where "OAM" shares similar crash rates as others while it significantly sacrifices the rewards to achieve the same level of safety. This shows the hurt brought by optimal action masking on an immature ψ network in the early stage of training, which is also the reason of suboptimality of the strong polarization effect given by "xp". And the weak polarization effect given by "log" cannot ensure to be safe enough. But in ACC and Circle, all ablation algorithms have competitive safety and reward performance (the last two rows in Figure 8).

C.3.3. Additional Results Compared to PPOBarrier

We present the training curves on the four highway benchmarks, comparing our methods against the baseline **PPOBarrier**, in Figure 9.

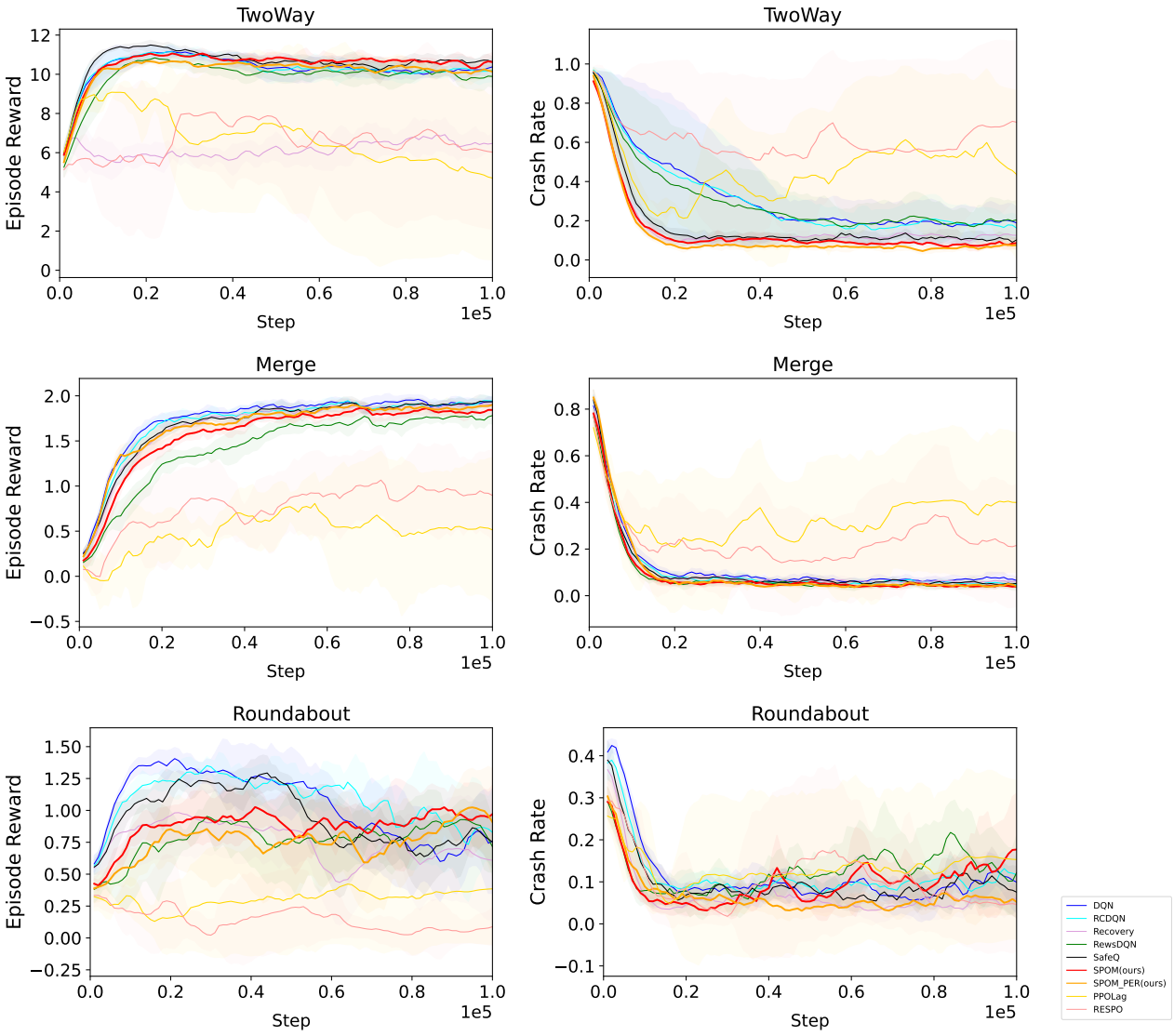


Figure 6. Training curves across all environments. Each row shows the episode reward (left) and crash rate (middle) for a specific environment. The rightmost panel in the third row shows the legend shared across all plots. Higher reward and lower crash rate are better.

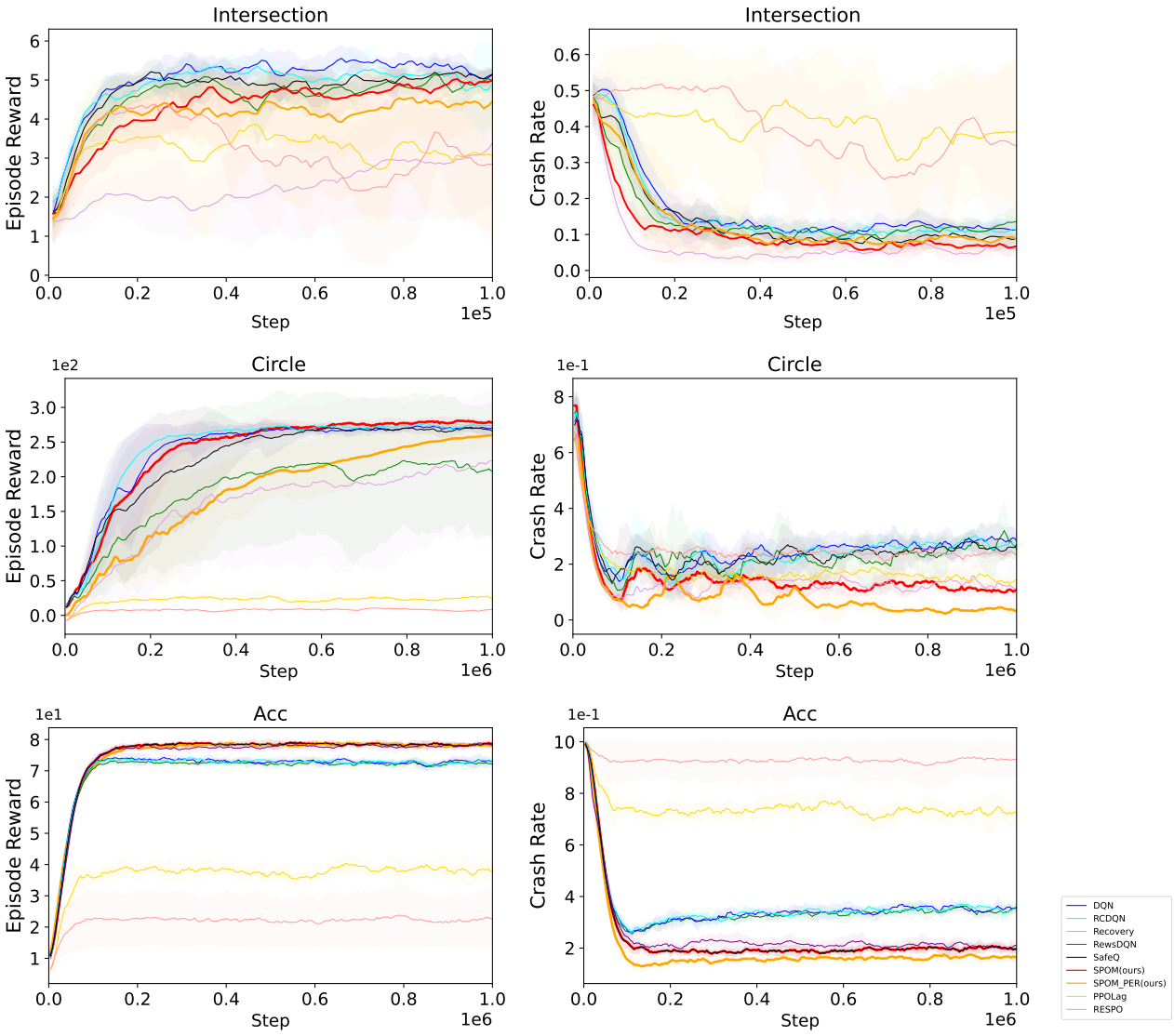


Figure 7. All training curves in Intersection, ACC, Circle. The x-axis represents the number of steps, and the y-axis: the first row is episode reward (higher is better); the second row is safety measured in crash rate (lower is better).

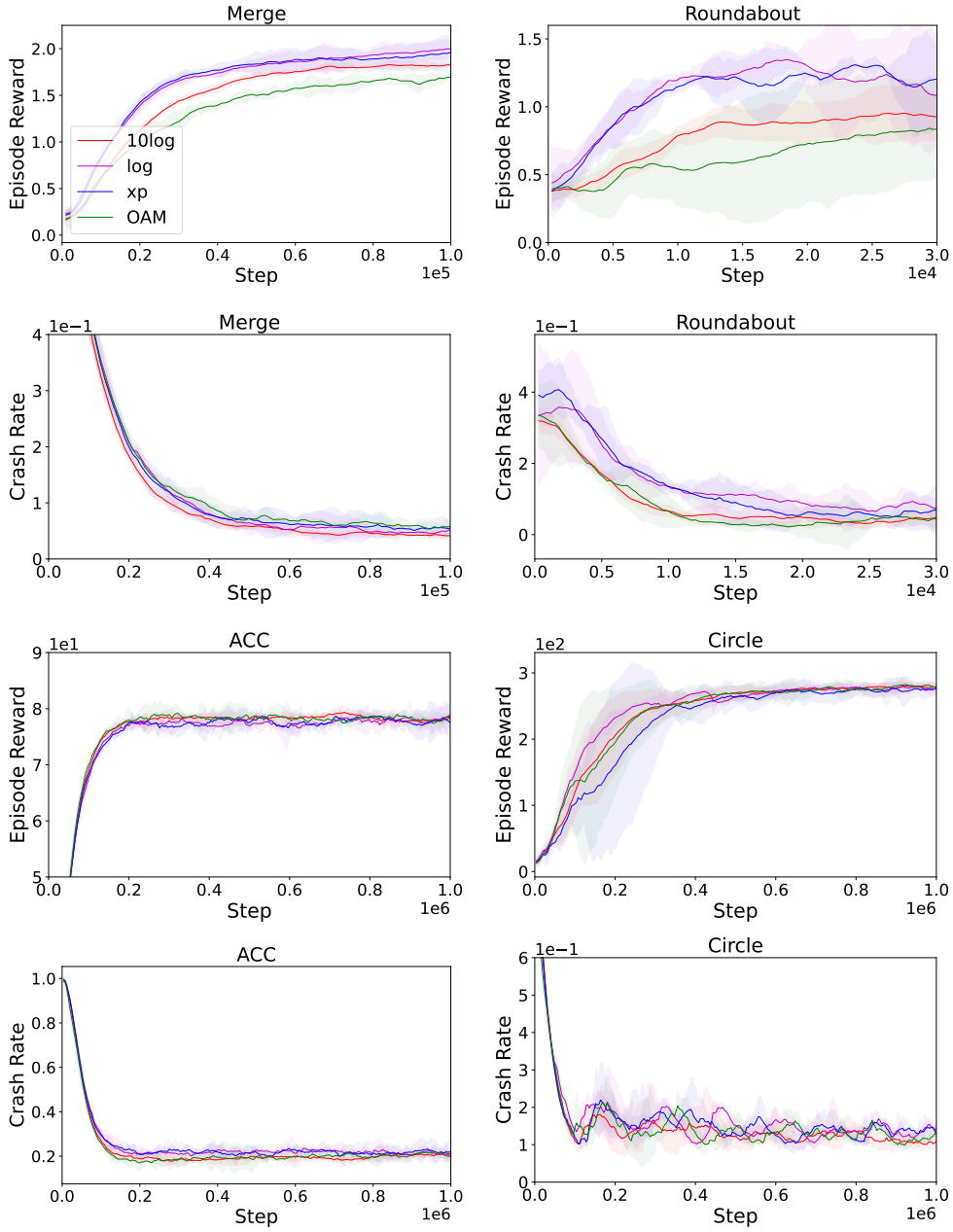


Figure 8. Training curves of other ablation studies, where "log", "xp" represent using polarization function $\log(x)$ and $1 - \frac{1}{x}$, respectively, "$10\log$" is our default choice $10 \cdot \log(x)$, and "OAM" means applying optimal action masks directly. The x-axis represents the number of steps, and the y-axis: the odd rows are episode reward (higher is better); the even rows are safety measured in crash rate (lower is better).

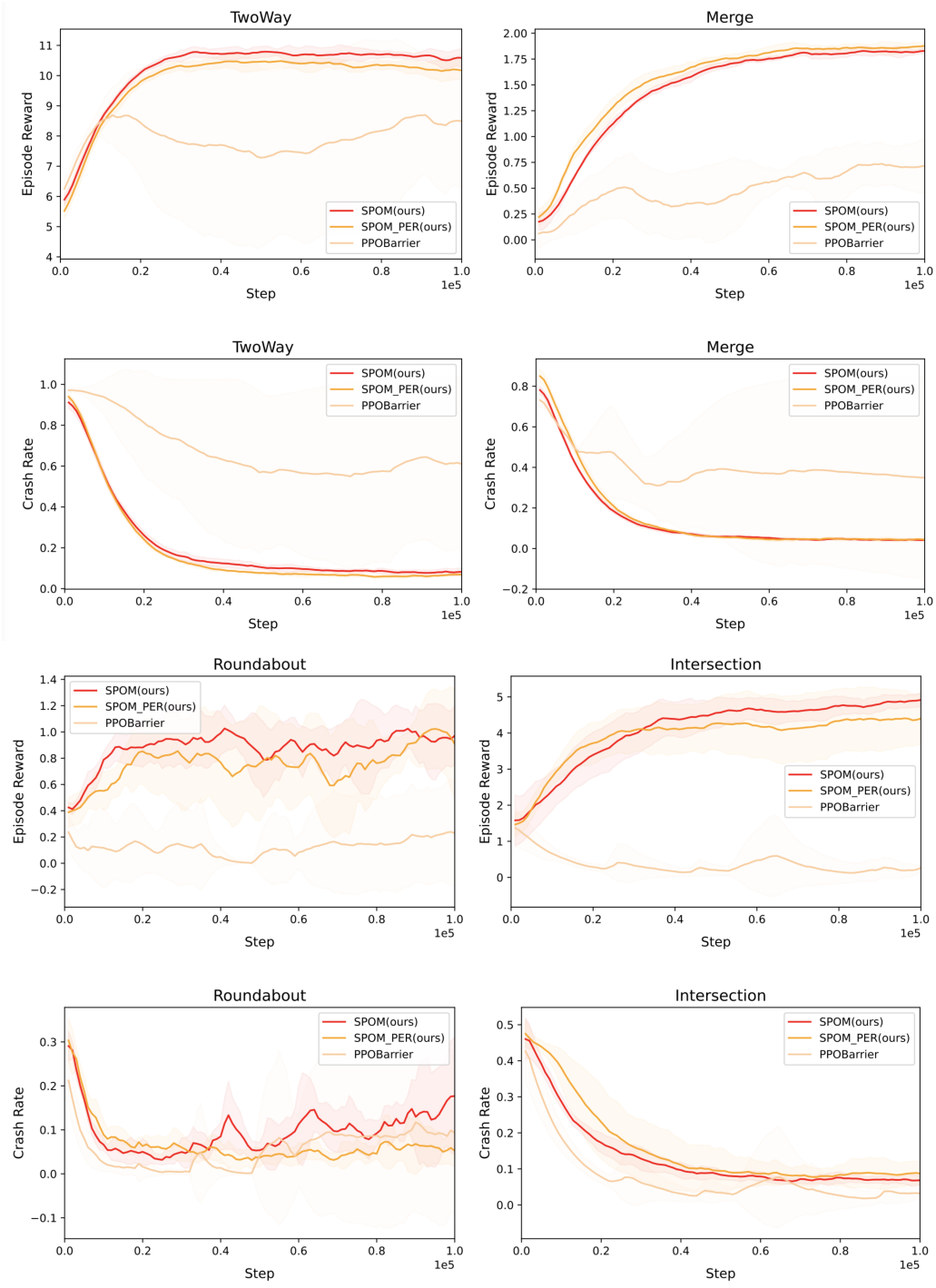


Figure 9. Training curves compared to **PPOBarrier** on four highway tasks.