

Safe Reinforcement Learning with Dual Robustness

Zeyang Li, Chuxiong Hu, Yunan Wang, Yujie Yang, Shengbo Eben Li

Abstract—Reinforcement learning (RL) agents are vulnerable to adversarial disturbances, which can deteriorate task performance or break down safety specifications. Existing methods either address safety requirements under the assumption of no adversary (e.g., safe RL) or only focus on robustness against performance adversaries (e.g., robust RL). Learning one policy that is both safe and robust under any adversaries remains a challenging open problem. The difficulty is how to tackle two intertwined aspects in the worst cases: feasibility and optimality. The optimality is only valid inside a feasible region (i.e., robust invariant set), while the identification of maximal feasible region must rely on how to learn the optimal policy. To address this issue, we propose a systematic framework to unify safe RL and robust RL, including the problem formulation, iteration scheme, convergence analysis and practical algorithm design. The unification is built upon constrained two-player zero-sum Markov games, in which the objective for protagonist is twofold. For states inside the maximal robust invariant set, the goal is to pursue rewards under the condition of guaranteed safety; for states outside the maximal robust invariant set, the goal is to reduce the extent of constraint violation. A dual policy iteration scheme is proposed, which simultaneously optimizes a task policy and a safety policy. We prove that the iteration scheme converges to the optimal task policy which maximizes the twofold objective in the worst cases, and the optimal safety policy which stays as far away from the safety boundary. The convergence of safety policy is established by exploiting the monotone contraction property of safety self-consistency operators, and that of task policy depends on the transformation of safety constraints into state-dependent action spaces. By adding two adversarial networks (one is for safety guarantee and the other is for task performance), we propose a practical deep RL algorithm for constrained zero-sum Markov games, called dually robust actor-critic (DRAC). The evaluations with safety-critical benchmarks demonstrate that DRAC achieves high performance and persistent safety under all scenarios (no adversary, safety adversary, performance adversary), outperforming all baselines by a large margin.

Index Terms—Reinforcement learning, zero-sum Markov game, safety, robustness.

1 INTRODUCTION

REINFORCEMENT learning (RL) has demonstrated tremendous performance in various fields, including games [1], robotics [2], [3], and autonomous driving [4], [5]. However, it remains challenging to deploy RL methods on real-world complex control tasks. The challenges are twofold. First, most real-world tasks not only require the RL agents to maximize the total rewards but also demand strict safety constraint satisfaction, since violating these constraints can lead to severe consequences [6]. Second, there are nonnegligible gaps between simulation platforms and real-world scenarios, such as model mismatches, sensory noises and environmental perturbations, which requires strong robustness of RL agents [7], [8].

Safe RL is a research area that aims at learning policies that satisfy safety constraints [6], [9]. There are mainly two categories of safety formulations in existing safe RL algorithms: trajectory safety and stepwise safety. In the trajectory safety formulation of safe RL, the objective is to find a policy that maximizes total rewards under the condition that the

expectation of trajectory costs is below a predefined threshold. Many works adopt the method of Lagrange multipliers to solve the constrained optimization problem. Ha et al. combine soft actor-critic algorithm with Lagrange multipliers and perform dual ascent on the Lagrangian function [10]. Chow et al. impose constraints for conditional value-at-risk of the cumulative cost and derive the gradient of the Lagrangian function [11]. Tessler et al. augment the reward function with additional penalty signals, which guide the policy toward a constraint-satisfying solution [12]. Trust region method is also utilized for constrained optimization, in which the objective and the constraint are both approximated with low-order functions, yielding a local analytic solution for policy improvement. Achiam et al. propose the constrained policy optimization algorithm, in which both objective and cost value constraint are approximated with linear functions [13]. Yang et al. maximize rewards using trust region optimization and project the policy back onto the constraint set [14]. In the stepwise safety formulation of safe RL, the objective is to find a policy that maximizes total rewards while ensuring strict constraint satisfaction at every state the agent visits. This line of work utilizes the rigorous definition and theoretical guarantee of safety in the safe control community [15], [16]. The crucial insight is that persistence safety is only possible for a subset of the constraint set, called invariant set [15]. Some works achieve set invariance with energy functions, such as control barrier function [16] and safety index [17]. Ma et al. jointly optimize the control policy and safety index [18]. Yang et al. jointly

- Zeyang Li, Chuxiong Hu and Yunan Wang are with the Department of Mechanical Engineering, Tsinghua University, Beijing 100084, China (email: li-zy21@mails.tsinghua.edu.cn; cxhu@tsinghua.edu.cn; wangyn22@mails.tsinghua.edu.cn).
- Yujie Yang and Shengbo Eben Li are with the School of Vehicle and Mobility, Tsinghua University, Beijing 100084, China (email: yangyj21@mails.tsinghua.edu.cn; lishbo@tsinghua.edu.cn).

(Corresponding author: Chuxiong Hu.)

optimize the control policy and control barrier function [19]. Besides energy functions, Hamilton-Jacobi reachability analysis [20] is also utilized for synthesis of invariant sets. Fisac et al. develop an RL approach for computing the safety value functions (representation of invariant sets) in Hamilton-Jacobi reachability analysis [21]. Yu et al. further utilize the learned safety value function to perform constrained policy optimization [22]. Despite the fruitful research on safe RL, previous methods rarely account for external disturbances, which may severely harm the safety-preserving capability of safe RL algorithms.

Robust RL is a research area that aims at learning policies that enjoy performance robustness against uncertainties [23]. Existing works consider uncertainties of different elements in RL, such as states [24], actions [25], transition probabilities [26] and rewards [27]. Based on the viewpoint of robust control, model mismatches, sensory noises and environmental perturbations can all be viewed as external disturbances in the system, so in this paper we focus on robustness against disturbances under the setting of two-player zero-sum Markov games, in which control inputs serve as the protagonist and disturbances serve as the adversary. Two-player zero-sum Markov game is a generalization of the standard Markov Decision Process (MDP), in which the protagonist tries to maximize the accumulated reward while the adversary tries to minimize it [28]. Pinto et al. propose the idea of robust adversarial RL, in which the protagonist policy and the adversary policy are parameterized by neural networks and jointly trained [29]. Their method improves training stability and the learned agents enjoy strong robustness in performance. Zhu et al. propose the minimax Q-network for two-player zero-sum Markov games [30]. Tessler et al. propose probabilistic action robust MDP and noisy action robust MDP, which are related to common forms of uncertainty in robotics [25]. They analyze the two forms of MDP in the tabular setting and design corresponding deep RL algorithms.

However, external disturbances not only can deteriorate the task performance (i.e., making agents unable to accomplish their goals), but also may break down safety specifications (i.e., leading to catastrophic damage on the entire system). Existing safe RL methods seldom consider external perturbations. Their agents are vulnerable to performance attacks, and the safety-preserving ability no longer holds under safety attacks. Existing robust RL methods only consider robustness against performance attacks and lack safety-preserving ability even without adversaries. To the best of our knowledge, there are no RL algorithm that can learn one policy that is robust to both safety and performance attacks. The difficulty is how to simultaneously handle two intertwined aspects, i.e., feasibility and optimality, in the worst cases. The former refers to the fact that there exists no feasible solution for a policy that can keep the system safe under worst-case safety attacks in a certain region inside the constraint set. The latter refers to attaining the highest rewards under worst-case performance attacks. The two aspects are heavily intertwined, since optimality is only valid inside a feasible region (i.e., robust invariant set), while the identification of maximal feasible region must rely on how to learn the optimal policy.

To overcome the aforementioned challenges, this paper

builds a theoretical framework to unify safe RL and robust RL. A constrained zero-sum Markov game is essentially a constrained optimization problem, the key of which is designing a twofold objective. For states inside the maximal robust invariant set, the objective is to maximize the value function while satisfying the constraints of admissible control inputs, which are specified by the safety value function. For states outside the maximal robust invariant set, the objective for these states is to reduce the degree of constraint violation, since the system will violate the safety constraints inevitably and it is meaningless to pursue rewards. The crux is to propose an iteration scheme for this constrained optimization problem, prove its convergence, and design a deep RL algorithm for practical implementation. The contributions of this paper are summarized as follows.

- We propose a dual policy iteration scheme to solve constrained zero-sum Markov games, which jointly iterates two policies: task policy for maximization of the twofold objective and safety policy for identification of the robust invariant set. We establish the self-consistency conditions of safety value functions, which are utilized to iterate the safety policy, i.e., alternating between safety policy evaluation and safety policy improvement. The task policy is optimized by alternating between task policy evaluation and task policy improvement. The latter consists of two parts, which are consistent with the twofold objective of constrained zero-sum Markov games. For states inside the current invariant set specified by the current safety policy, greedy search is performed under the constraint of safety value function. For states outside the current invariant set, the task policy copies the safety policy, for the purpose of reducing the extent of future constraint violation.
- The convergence of this iteration scheme is proved by separately discussing different behavioral properties of safety policy and task policy. For safety policy, its convergence is established by exploiting the monotone contraction property of the safety self-consistency operators. For task policy, we can establish a well-defined unconstrained zero-sum Markov game on the robust invariant set of a specified safety policy, transforming the original safety constraints into the form of state-dependent action spaces. We prove that the proposed dual policy iteration scheme converges to the optimal task policy (i.e., the policy attaining the maximal objective for the proposed constrained optimization problem) and the optimal safety policy (i.e., the policy seeking the highest safety values).
- We propose a practical deep RL algorithm for constrained zero-sum Markov games, called dually robust actor-critic (DRAC), which can learn one policy that is robust to both performance and safety attacks. Since it is intractable to conduct thorough task policy evaluation and safety policy evaluation on high-dimensional continuous spaces, a safety adversary network and a performance adversary network are additionally introduced for practical implementation. Since there are infinite constraints on the

task policy, we also introduce a Lagrange multiplier network to facilitate the constrained optimization process.

2 BACKGROUND

2.1 Safe RL

In this work, we consider the stepwise deterministic safety specification for safe RL, which aims to ensure that the learned control policy satisfies the safety constraints on every state it visits [19], [22]. Consider an MDP with deterministic system dynamics $(\mathcal{X}, \mathcal{U}, f, r, h, \gamma)$, in which \mathcal{X} denotes the state space, \mathcal{U} denotes the control space, $f : \mathcal{X} \times \mathcal{U} \rightarrow \mathcal{X}$ denotes the system dynamics, $\pi : \mathcal{X} \rightarrow \Delta(\mathcal{U})$ denotes the control policy ($\Delta(\mathcal{U})$ represents the set of probability distributions on \mathcal{U}), γ denotes the discount factor, $r : \mathcal{X} \times \mathcal{U} \rightarrow \mathbb{R}$ denotes the reward function, and $h : \mathcal{X} \rightarrow \mathbb{R}$ denotes the constraint function. Let d denote the initial state distribution. The problem formulation is specified as follows.

$$\begin{aligned} & \max_{\pi} \mathbb{E}_{x_0 \sim d} \left\{ \sum_{t=0}^{\infty} \gamma^t r(x_t, u_t) \right\} \\ \text{s.t. } & x_{t+1} = f(x_t, u_t), u_t \sim \pi(\cdot | x_t), \\ & h(x_t) \geq 0, t = 0, 1, 2, \dots, \infty. \end{aligned}$$

2.2 Robust RL

The foundation of robust RL lies in the theory of two-player zero-sum Markov games, in which the protagonist aims to maximize the accumulated reward while the adversary tries to decrease it [28]. A two-player zero-sum Markov game can be represented by a tuple $(\mathcal{X}, \mathcal{U}, \mathcal{A}, p, r, \gamma)$, in which \mathcal{X} denotes the state space, \mathcal{U} denotes the protagonist action space, \mathcal{A} denotes the adversary action space, $p : \mathcal{X} \times \mathcal{U} \times \mathcal{A} \rightarrow \Delta(\mathcal{X})$ denotes the transition probability, $\pi : \mathcal{X} \rightarrow \Delta(\mathcal{U})$ denotes the protagonist policy, $\mu : \mathcal{X} \rightarrow \Delta(\mathcal{A})$ denotes the adversary policy, γ denotes the discount factor, and $r : \mathcal{X} \times \mathcal{U} \times \mathcal{A} \rightarrow \mathbb{R}$ denotes the reward function. Let d denote the initial state distribution. The problem formulation is specified as follows.

$$\begin{aligned} & \max_{\pi} \min_{\mu} \mathbb{E}_{x_0 \sim d} \left\{ \sum_{t=0}^{\infty} \gamma^t r(x_t, u_t, a_t) \right\} \\ \text{s.t. } & x_{t+1} = f(x_t, u_t, a_t), \\ & u_t \sim \pi(\cdot | x_t), a_t \sim \mu(\cdot | x_t). \end{aligned}$$

3 DUAL POLICY ITERATION FOR CONSTRAINED ZERO-SUM MARKOV GAMES

In this section, we develop a theoretical framework for constrained zero-sum Markov games, in which the control inputs serve as the protagonist and the external disturbances serve as the adversary. Adversarial disturbances have two kinds of effects on the system. They can either attack the safety specifications or attack the task performance. Since safety is the top priority, we must first ensure constraint satisfaction under worst-case safety adversaries, which requires identification of the maximal robust invariant set. Then we need to further impose performance robustness on the control policy. We propose a dual policy iteration scheme to accomplish this goal.

In this work, the constrained zero-sum Markov game is denoted by a tuple $\mathcal{M} = (\mathcal{X}, \mathcal{U}, \mathcal{A}, f, r, h, \gamma)$, in which \mathcal{X} represents the (finite) state space, \mathcal{U} represents the (finite) protagonist action space, \mathcal{A} represents the (finite) adversary action space, $f : \mathcal{X} \times \mathcal{U} \times \mathcal{A} \rightarrow \mathcal{X}$ represents the deterministic system dynamics, γ represents the discount factor, $r : \mathcal{X} \times \mathcal{U} \times \mathcal{A} \rightarrow \mathbb{R}$ represents the reward function, and $h : \mathcal{X} \rightarrow \mathbb{R}$ denotes the constraint function. If there are multiple safety requirements, i.e., $\{h_1(x) \geq 0, h_2(x) \geq 0, \dots\}$, the constraint function can be specified as $h(x) = \min\{h_1(x), h_2(x), \dots\}$. Let $\pi : \mathcal{X} \rightarrow \Delta(\mathcal{U})$ represent the protagonist policy and $\mu : \mathcal{X} \rightarrow \Delta(\mathcal{A})$ represent the adversary policy.

3.1 Safety Value Function

Safety is of the utmost importance. When external disturbances exist, we must ensure safety under worst-case scenarios. A fundamental fact from safe control research is that only those states inside a subset of the constraint set, called robust invariant set, can achieve persistent safety [20]. For states outside the robust invariant set, there always exist some adversarial disturbances that will drive the system to violate the safety constraints regardless of what control policy is adopted. Therefore, enforcing safety is essentially equivalent to constraining the system states inside the robust invariant set.

Hamilton-Jacobi reachability analysis [20] provides a formal mathematical tool to compute the robust invariant set of arbitrary nonlinear systems with bounded disturbances. However, it suffers from the curse of dimensionality and is intractable for high-dimensional systems. Recently, some pioneering works have migrated Hamilton-Jacobi reachability analysis to model-free RL [21], [22]. However, these works assume that there are no disturbances in the system and only compute the standard invariant set, whose persistent safety can be compromised when disturbances exist. In this work, we make use of the safety value function in Hamilton-Jacobi reachability analysis and develop RL techniques to solve for the robust invariant set.

We begin by defining the following safety values. Note that we focus on action values, which benefit model-free RL. Our results can be easily extended to state values. The notation $\min_{\lambda \sim d} f(\lambda)$ denotes the minimum value of $f(\lambda)$ on the support set of distribution d .

Definition 1 (safety value functions).

- 1) Let $\tau(x, u, a)$ denote an infinite-horizon trajectory starting from (x, u, a) , i.e.,

$$\tau(x, u, a) \triangleq \{x_0 = x, u_0 = u, a_0 = a, x_1, u_1, a_1, \dots\}.$$

The safety value of a trajectory $\tau(x, u, a)$ is defined as

$$Q_h^\tau(x, u, a) = \min_{t \in \mathbb{N}} \{h(x_t) \mid x_t \in \tau(x, u, a)\}.$$

- 2) The safety value function of a protagonist policy π_h and an adversary policy μ_h is defined as

$$Q_h^{\pi_h, \mu_h}(x, u, a) = \min_{\tau \sim \pi_h, \mu_h} \{Q_h^\tau(x, u, a)\}. \quad (1)$$

- 3) The safety value function of a protagonist policy π_h is defined as

$$\begin{aligned} Q_h^{\pi_h}(x, u, a) &= \min_{\mu_h} Q_h^{\pi_h, \mu_h}(x, u, a) \\ &= \min_{\mu_h} \min_{\tau \sim \pi_h, \mu_h} \{Q_h^\tau(x, u, a)\}. \end{aligned} \quad (2)$$

- 4) The optimal safety value function is defined as

$$\begin{aligned} Q_h^*(x, u, a) &= \max_{\pi_h} Q_h^{\pi_h}(x, u, a) \\ &= \max_{\pi_h} \min_{\mu_h} \min_{\tau \sim \pi_h, \mu_h} \{Q_h^\tau(x, u, a)\}. \end{aligned} \quad (3)$$

$Q_h^\tau(x, u, a)$ captures the most dangerous state in the trajectory, i.e., the state with the lowest constraint value. $Q_h^\tau(x, u, a) \geq 0$ implies that the whole trajectory τ satisfies the safety constraint $h(x) \geq 0$. $Q_h^{\pi_h, \mu_h}(x, u, a)$ represents the most dangerous state in the long term, when the system is driven by a specified pair of protagonist and adversary. Since both protagonist policy π_h and adversary policy μ_h may be stochastic, $Q_h^{\pi_h, \mu_h}(x, u, a)$ captures the trajectory with the lowest safety value among all possible trajectories. $Q_h^{\pi_h}(x, u, a)$ identifies the safety-preserving capability of a specified protagonist policy π_h under worst-case safety attacks. $Q_h^*(x, u, a)$ represents the best possible outcome we can get in terms of safety, starting from (x, u, a) . $Q_h^*(x, u, a) \leq 0$ implies that the system cannot guarantee safety at the state-action pair (x, u, a) . Therefore, Q_h^* identifies the maximal robust invariant set of the system. We assume that the constrained zero-sum Markov game $\mathcal{M} = (\mathcal{X}, \mathcal{U}, \mathcal{A}, f, r, h, \gamma)$ satisfies that $\max_{x \in \mathcal{X}} \max_{u \in \mathcal{U}} \min_{a \in \mathcal{A}} Q_h^*(x, u, a) \geq 0$, otherwise the system is impossible to achieve persistent safety. We formally define the robust invariant sets as follows.

Definition 2 (constraint set). The constraint set is defined as the zero-superlevel set of the constraint function $h(x)$, i.e.,

$$S_h = \{x \in \mathcal{X} \mid h(x) \geq 0\}.$$

Definition 3 (robust invariant sets).

- 1) If we have $\max_{x \in \mathcal{X}} \max_{u \in \mathcal{U}} \min_{a \in \mathcal{A}} Q_h^{\pi_h}(x, u, a) \geq 0$ for a policy π_h , the robust invariant set of this specified protagonist policy π_h is defined as

$$S_r^{\pi_h} = \left\{ x \in \mathcal{X} \mid \max_{u \in \mathcal{U}} \min_{a \in \mathcal{A}} Q_h^{\pi_h}(x, u, a) \geq 0 \right\}.$$

- 2) The maximal robust invariant set is defined as

$$S_r^* = \left\{ x \in \mathcal{X} \mid \max_{u \in \mathcal{U}} \min_{a \in \mathcal{A}} Q_h^*(x, u, a) \geq 0 \right\}.$$

Remark 1. Technically speaking, the robust invariant sets in this article are robust controlled invariant sets. For states inside these sets, there exists some control policy keeping the system safe regardless of any disturbances [20]. The term controlled is omitted for simplicity.

Remark 2. We can easily deduce that the following set inclusion relationship holds:

$$S_r^{\pi_h} \subseteq S_r^* \subseteq S_h. \quad (4)$$

To ensure safety of the system, it is not enough to only constrain the state to stay inside the constraint set S_h . Once the system state enters the region $S_h \setminus S_r^*$ (i.e., $h(x) \geq 0$ and

$\max_{u \in \mathcal{U}} \min_{a \in \mathcal{A}} Q_h^*(x, u, a) \leq 0$), there exists some adversary policy μ_h that will drive the system out of the constraint set S_h , regardless of what protagonist policy is adopted. Therefore, we must constrain the system to stay inside the maximal robust invariant set S_r^* . To impose this constraint, firstly we must figure out a way to obtain S_r^* , which is equivalent to solving for $Q_h^*(x, u, a)$.

The set inclusion relationship (4) suggests that we may perform policy iteration, which converges to an optimal protagonist policy π_h^* , such that $S_r^{\pi_h^*} = S_r^*$. First, we need the following proposition to simplify the policy iteration procedure.

Proposition 1. Given a pair of stochastic protagonist policy π_h^{sto} and adversary policy μ_h^{sto} , there exist a pair of deterministic protagonist policy π_h^{det} and adversary policy μ_h^{det} , such that $\forall x \in \mathcal{X}, u \in \mathcal{U}, a \in \mathcal{A}$,

$$Q_h^{\pi_h^{\text{sto}}, \mu_h^{\text{sto}}}(x, u, a) = Q_h^{\pi_h^{\text{det}}, \mu_h^{\text{det}}}(x, u, a).$$

Proof. Let τ_{wor} denote the trajectory with the lowest safety values among all possible trajectories when the system starting from (x, u, a) is driven by π_h^{sto} and μ_h^{sto} , i.e., $Q_h^{\pi_h^{\text{sto}}, \mu_h^{\text{sto}}}(x, u, a) = Q_h^{\tau_{\text{wor}}}(x, u, a)$. Let x_{wor} denote the state with the lowest constraint value, i.e., $h(x_{\text{wor}}) = \min_{x \in \tau_{\text{wor}}} \{h(x)\}$. The key insight is that x_{wor} is reachable from (x, u, a) under a pair of deterministic policies. Take the finite-horizon trajectory inside τ_{wor} that starts at (x, u, a) and ends at x_{wor} . We denote this transition trajectory as τ_{trans} . If there are two identical states inside τ_{trans} , we remove the parts between the two states and the remaining trajectory still satisfies the system dynamics $x' = f(x, u, a)$. Suppose that the filtered trajectory of τ_{trans} is $\{x_0 = x, u_0 = u, a_0 = a, x_1, u_1, a_1, x_2, u_2, a_2, \dots\}$. The deterministic protagonist policy π_h^{det} is defined to satisfy

$$\pi_h^{\text{det}}(x_1) = u_1, \pi_h^{\text{det}}(x_2) = u_2, \dots$$

The deterministic adversary policy μ_h^{det} is defined to satisfy

$$\mu_h^{\text{det}}(x_1) = a_1, \mu_h^{\text{det}}(x_2) = a_2, \dots$$

With this construction we have

$$Q_h^{\pi_h^{\text{sto}}, \mu_h^{\text{sto}}}(x, u, a) = Q_h^{\pi_h^{\text{det}}, \mu_h^{\text{det}}}(x, u, a).$$

□

Proposition 1 indicates that we can restrict ourselves to using deterministic policies when solving for Q_h^* without any loss of optimality. From now on, we assume that the safety protagonist policy π_h and the safety adversary policy μ_h are deterministic. Since the system dynamics is also deterministic, the whole trajectory is fixed given the starting state-action pair (x, u, a) . The safety value function of a pair of π_h and μ_h can be simplified to

$$\begin{aligned} Q_h^{\pi_h, \mu_h}(x, u, a) &= \min_{t \in \mathbb{N}} \{h(x_t)\} \\ \text{s.t. } x_0 &= x, u_0 = u, a_0 = a, \\ u_t &= \pi_h(x_t), a_t = \mu_h(x_t), \\ x_t &= f(x_{t-1}, u_{t-1}, a_{t-1}), t \geq 1. \end{aligned}$$

Since the safety value functions are defined on infinite horizon, they naturally hold a recursive structure, which we

refer to as self-consistency condition, just as the common value functions in RL. We have the following theorem.

Theorem 1 (safety self-consistency conditions). *The safety value functions satisfy the following self-consistency conditions:*

$$Q_h^{\pi_h, \mu_h}(x, u, a) = \min \{h(x), Q_h^{\pi_h, \mu_h}(x', \pi_h(x), \mu_h(x))\}, \quad (5)$$

$$Q_h^{\pi_h}(x, u, a) = \min \left\{ h(x), \min_{a' \in \mathcal{A}} Q_h^{\pi_h}(x', \pi_h(x), a') \right\}, \quad (6)$$

$$Q_h^*(x, u, a) = \min \left\{ h(x), \max_{u' \in \mathcal{U}} \min_{a' \in \mathcal{A}} Q_h^*(x', u', a') \right\}, \quad (7)$$

in which $x' = f(x, u, a)$.

Proof. First, we prove (5). From the definition of the safety value function, we have

$$\begin{aligned} Q_h^{\pi_h, \mu_h}(x, u, a) &= \min_{t \geq 0} \{h(x_t)\} \\ &= \min \left\{ h(x), \min_{t \geq 1} \{h(x_t) \mid x_1 = x'\} \right\} \\ &= \min \left\{ h(x), \min_{t \geq 0} \{h(x_t) \mid x_0 = x'\} \right\} \\ &= \min \{h(x), Q_h^{\pi_h, \mu_h}(x', \pi_h(x), \mu_h(x))\}, \end{aligned} \quad (8)$$

in which $x' = f(x, u, a)$.

The proof for (6) and (7) is similar to (8), with additional use of Bellman's principle of optimality. \square

A problem with these self-consistency conditions is that they are not contraction mappings. To apply RL techniques such as policy iteration, we introduce their discounted versions, which are defined as follows.

Definition 4 (safety operators). *Given $\gamma_h \in (0, 1)$, the safety self-consistency operator of a pair of protagonist policy π_h and adversary policy μ_h is defined as*

$$\begin{aligned} [T_h^{\pi_h, \mu_h}(Q_h)](x, u, a) &= (1 - \gamma_h)h(x) \\ &\quad + \gamma_h \min \{h(x), Q_h(x', \pi_h(x), \mu_h(x))\}. \end{aligned} \quad (9)$$

The safety self-consistency operator of a protagonist policy π_h is defined as

$$\begin{aligned} [T_h^{\pi_h}(Q_h)](x, u, a) &= (1 - \gamma_h)h(x) \\ &\quad + \gamma_h \min \left\{ h(x), \min_{a' \in \mathcal{A}} Q_h(x', \pi_h(x), a') \right\}. \end{aligned} \quad (10)$$

The safety Bellman operator is defined as

$$\begin{aligned} [T_h(Q_h)](x, u, a) &= (1 - \gamma_h)h(x) \\ &\quad + \gamma_h \min \left\{ h(x), \max_{u' \in \mathcal{U}} \min_{a' \in \mathcal{A}} Q_h(x', u', a') \right\}. \end{aligned} \quad (11)$$

The following theorem indicates that the three safety operators in Definition 4 are monotone contractions.

Theorem 2 (monotone contraction of safety operators).

1) Given any $Q_h, \tilde{Q}_h \in \mathbb{R}^{|\mathcal{X}| \cdot |\mathcal{U}| \cdot |\mathcal{A}|}$, we have

$$\begin{aligned} \|T_h^{\pi_h, \mu_h}(Q_h) - T_h^{\pi_h, \mu_h}(\tilde{Q}_h)\|_\infty &\leq \gamma_h \|Q_h - \tilde{Q}_h\|_\infty, \\ \|T_h^{\pi_h}(Q_h) - T_h^{\pi_h}(\tilde{Q}_h)\|_\infty &\leq \gamma_h \|Q_h - \tilde{Q}_h\|_\infty, \\ \|T_h(Q_h) - T_h(\tilde{Q}_h)\|_\infty &\leq \gamma_h \|Q_h - \tilde{Q}_h\|_\infty. \end{aligned}$$

2) Suppose $Q_h(x, u, a) \geq \tilde{Q}_h(x, u, a)$ holds for all $x \in \mathcal{X}, u \in \mathcal{U}, a \in \mathcal{A}$, then we have

$$\begin{aligned} [T_h^{\pi_h, \mu_h}(Q_h)](x, u, a) &\geq [T_h^{\pi_h, \mu_h}(\tilde{Q}_h)](x, u, a), \\ [T_h^{\pi_h}(Q_h)](x, u, a) &\geq [T_h^{\pi_h}(\tilde{Q}_h)](x, u, a), \\ [T_h(Q_h)](x, u, a) &\geq [T_h(\tilde{Q}_h)](x, u, a). \end{aligned} \quad (5)$$

Proof. We only prove the monotonicity and contraction of T_h . The proof for $T_h^{\pi_h}$ and $T_h^{\pi_h, \mu_h}$ is similar.

Since the max and min operation contained in T_h is monotone, T_h is also monotone. Given Q_h and \tilde{Q}_h , we have

$$\begin{aligned} &[T_h(Q_h)](x, u, a) - [T_h(\tilde{Q}_h)](x, u, a) \\ &= \gamma_h \min \left\{ h(x), \max_{u' \in \mathcal{U}} \min_{a' \in \mathcal{A}} Q_h(x', u', a') \right\} \\ &\quad - \gamma_h \min \left\{ h(x), \max_{u' \in \mathcal{U}} \min_{a' \in \mathcal{A}} \tilde{Q}_h(x', u', a') \right\}, \end{aligned}$$

in which $x' = f(x, u, a)$. Utilizing the relationship

$$\begin{aligned} &\left| \max_x \min_y F(x, y) - \max_x \min_y G(x, y) \right| \\ &\leq \max_x \max_y |F(x, y) - G(x, y)|, \end{aligned}$$

we have

$$\begin{aligned} &\|T_h(Q_h) - T_h(\tilde{Q}_h)\|_\infty \\ &\leq \gamma_h \max_{u' \in \mathcal{U}} \max_{a' \in \mathcal{A}} |Q_h(x', u', a') - \tilde{Q}_h(x', u', a')| \\ &\leq \gamma_h \|Q_h - \tilde{Q}_h\|_\infty. \end{aligned}$$

\square

Since the three operators in Definition 4 are contraction mappings, they all have unique fixed points. These fixed points serve as approximations of the original safety value functions in Definition 1. The following proposition shows that as the discount factor γ_h goes to 1, these fixed points converge to the original safety values.

Proposition 2. *As γ_h goes to 1, the fixed point of operator $T_h^{\pi_h, \mu_h}$ converges to the safety value function defined in (1), the fixed point of operator $T_h^{\pi_h}$ converges to the safety value function defined in (2), and the fixed point of operator T_h converges to the safety value function defined in (3).*

Proof. The key of proof is defining a discounted formulation of the safety values. Recall that $Q_h^{\pi_h, \mu_h}(x, u, a) = \min_{t \in \mathbb{N}} \{h(x_t)\}$, in which $h(x_t)$ denote the constraint values of the trajectory driven by π_h and μ_h . Define the following discounted value

$$D = (1 - \gamma_h)h(x_0) + \gamma_h \{ \min \{h(x_0), (1 - \gamma_h)h(x_1) + \gamma_h \{ \min \{h(x_1), (1 - \gamma_h)h(x_2) + \dots \} \} \} \}. \quad (12)$$

It can be verified that D is the explicit formulation of the fixed point of $T_h^{\pi_h, \mu_h}$, i.e., $D = T_h^{\pi_h, \mu_h}(D)$. Taking the limit of (12) as $\gamma_h \rightarrow 1$, we obtain

$$\lim_{\gamma_h \rightarrow 1} D = \min_{t \in \mathbb{N}} \{h(x_t)\} = Q_h^{\pi_h, \mu_h}(x, u, a),$$

which indicates that the fixed point of $T_h^{\pi_h, \mu_h}$ converges to the original safety value function defined in (1). The proof for $T_h^{\pi_h}$ and T_h is similar. \square

From now on, we assume that the chosen γ_h is sufficiently close to 1 and the fixed points of safety operators represent the original safety values.

3.2 Problem Formulation

In this subsection, we define a proper problem formulation for constrained zero-sum Markov games. Since safety is the top priority, the pursuit of high total rewards must be carried out under the condition that safety is guaranteed even with worst-case safety attacks. Given a state $x \in \mathcal{X}$, there are two cases. First, $x \notin S_r^*$. The safety constraint $h(x) \geq 0$ will be violated sooner or later under the worst-case safety adversary. Therefore, it is pointless to optimize the total rewards of this kind of states. We should execute the safest action (i.e., optimize its safety value) and drive the system back to S_r^* as soon as possible. Second, $x \in S_r^*$. We should maximize the total rewards and ensure persistent safety (i.e., do not choose those actions that could drive the system out of S_r^*). We have the following definition and proposition.

Definition 5 (invariant policy sets).

- 1) The invariant policy set $\Pi_s^{\pi_h}$ specified by a robust invariant set $S_r^{\pi_h}$ is defined as

$$\Pi_s^{\pi_h} = \{\pi \mid \forall x \in S_r^{\pi_h}, \pi(u \mid x) = 0, \text{ if } u \notin U_s^{\pi_h}(x)\},$$

in which $U_s^{\pi_h}(x)$ contains the admissible action to maintain persistent safety at state $x \in S_r^{\pi_h}$, i.e.,

$$U_s^{\pi_h}(x) = \left\{ u \in \mathcal{U} \mid \min_{a \in \mathcal{A}} Q_h^{\pi_h}(x, u, a) \geq 0 \right\}.$$

- 2) The optimal invariant policy set Π_s^* is defined as

$$\Pi_s^* = \{\pi \mid \forall x \in S_r^*, \pi(u \mid x) = 0, \text{ if } u \notin U_s^*(x)\},$$

in which $U_s^*(x)$ contains the admissible action to maintain persistent safety at state $x \in S_r^*$, i.e.,

$$U_s^*(x) = \left\{ u \in \mathcal{U} \mid \min_{a \in \mathcal{A}} Q_h^*(x, u, a) \geq 0 \right\}.$$

Proposition 3.

- 1) Given $x \in S_r^{\pi_h}$, if the protagonist policy $\pi \in \Pi_s^{\pi_h}$, the system will stay in $S_r^{\pi_h}$ no matter what adversary policy μ is adopted.
- 2) Given $x \in S_r^*$, if the protagonist policy $\pi \in \Pi_s^*$, the system will stay in S_r^* no matter what adversary policy μ is adopted.

Proof. We only prove the first claim in Proposition 3. The proof for the second claim is similar.

Given $x \in S_r^{\pi_h}$, if the protagonist policy $\pi \in \Pi_s^{\pi_h}$, we have

$$Q_h^{\pi_h}(x, u, a) \geq 0, u \sim \pi(\cdot \mid x), \forall a \in \mathcal{A}.$$

Utilizing the self-consistency condition (6) for $Q_h^{\pi_h}$, we have

$$Q_h^{\pi_h}(x, u, a) = \min \left\{ h(x), \min_{a' \in \mathcal{A}} Q_h^{\pi_h}(x', \pi_h(x), a') \right\}.$$

Therefore $\min_{a' \in \mathcal{A}} Q_h^{\pi_h}(x', \pi_h(x), a') \geq 0$ holds, which implies that $\max_{u' \in \mathcal{U}} \min_{a' \in \mathcal{A}} Q_h^{\pi_h}(x', u', a') \geq 0$. Based on the definition of $S_r^{\pi_h}$, we have $x' \in S_r^{\pi_h}$. Utilizing this procedure recursively,

we conclude that starting from $x \in S_r^{\pi_h}$, if the protagonist policy $\pi \in \Pi_s^{\pi_h}$, the system state will always be in $S_r^{\pi_h}$ regardless of what adversary actions are taken. \square

Proposition 3 indicates that any policy belonging to an invariant policy set will keep the system inside the corresponding robust invariant set, thus achieving persistent safety. Essentially, the infinite-horizon safety constraint $h(x_t) \geq 0, t \in \mathbb{N}$ is transformed into state-dependent constraints on the action space \mathcal{U} specified by invariant policy sets. To capture the performance optimality inside the robust invariant sets, we define the following induced zero-sum Markov games.

Definition 6 (induced zero-sum Markov games).

- 1) Given the original constrained zero-sum Markov game $\mathcal{M} = (\mathcal{X}, \mathcal{U}, \mathcal{A}, f, r, h, \gamma)$ and a robust invariant set $S_r^{\pi_h}$, the induced zero-sum Markov game is defined as $\mathcal{M}^{\pi_h} = \left(S_r^{\pi_h}, \bigcup_x U_s^{\pi_h}(x), \mathcal{A}, f, r, \gamma \right)$.
- 2) Given the original constrained zero-sum Markov game $\mathcal{M} = (\mathcal{X}, \mathcal{U}, \mathcal{A}, f, r, h, \gamma)$ and the optimal robust invariant set S_r^* , the optimal induced zero-sum Markov game is defined as $\mathcal{M}^* = \left(S_r^*, \bigcup_x U_s^*(x), \mathcal{A}, f, r, \gamma \right)$.

The notation $\bigcup_x U_s^{\pi_h}(x)$ means that for each state x , the admissible state-dependent action space is $U_s^{\pi_h}(x)$. Based on Proposition 3, it is easy to check that these games are well-defined. Also note that the induced zero-sum Markov games are unconstrained. As discussed before, it is only meaningful to optimize total rewards inside the maximal robust invariant set. We present the definitions of value functions as follows.

Definition 7 (value functions).

- 1) The value function of a protagonist policy π and an adversary policy μ is defined as

$$Q^{\pi, \mu}(x, u, a) = \sum_{t=0}^{\infty} \gamma^t r(x_t, u_t, a_t), \quad (13)$$

in which $x_0 = x, u_0 = u, a_0 = a$ and for $t \geq 1, x_t = f(x_{t-1}, u_{t-1}, a_{t-1}), u_t \sim \pi(\cdot \mid x_t), a_t \sim \mu(\cdot \mid x_t)$.

- 2) The value function of a protagonist policy π is defined as

$$Q^\pi(x, u, a) = \min_\mu Q^{\pi, \mu}(x, u, a). \quad (14)$$

- 3) For the optimal induced zero-sum Markov game $\mathcal{M}^* = \left(S_r^*, \bigcup_x U_s^*(x), \mathcal{A}, f, r, \gamma \right)$, the optimal value function is defined as

$$\begin{aligned} Q^*(x, u, a) &= \max_{\pi \in \Pi_s^*} Q^\pi(x, u, a) \\ &= \max_{\pi \in \Pi_s^*} \min_\mu Q^{\pi, \mu}(x, u, a). \end{aligned} \quad (15)$$

As shown in (15), the optimal value function is defined for states inside the maximal robust invariant set S_r^* (i.e., for the optimal induced zero-sum Markov games) and the optimal policy is searched inside the optimal invariant policy set Π_s^* , which contains all the policies that are able to maintain persistent safety of the system under worst-case safety attacks. Let d denote the initial state distribution.

Based on the definitions of value functions and safety value functions, we formally specify the problem formulation for constrained zero-sum Markov games, in which the objective for protagonist is twofold.

$$\begin{aligned} \max_{\pi} \mathbb{E}_{x_0 \sim d} \{V^\pi(x_0) \cdot \mathbb{1}_{S_r^*}(x_0) + V_h^\pi(x_0) \cdot \mathbb{1}_{\mathcal{X} \setminus S_r^*}(x_0)\} \\ \text{s.t. } x_{t+1} = f(x_t, u_t, a_t), u_t \sim \pi(\cdot | x_t), t \geq 0, \\ \min_{a \in \mathcal{A}} Q_h^*(x_t, u_t, a) \geq 0, \forall x_t \in S_r^*. \end{aligned} \quad (16)$$

Q_h^* represents the fixed point of the safety Bellman operator T_h , as defined in (11). $\mathbb{1}_{S_r^*}$ represents the indicator function, i.e., $\mathbb{1}_{S_r^*}(x) = 1$ when $x \in S_r^*$ and otherwise $\mathbb{1}_{S_r^*}(x) = 0$. V^π and V_h^π represent the state-value function and the safety state-value function, i.e.,

$$\begin{aligned} V^\pi(x) &= \sum_{u \in \mathcal{U}} \pi(u | x) \min_{a \in \mathcal{A}} Q^\pi(x, u, a), \\ V_h^\pi(x) &= \max_{u \in \mathcal{U}} \min_{a \in \mathcal{A}} Q_h^\pi(x, u, a). \end{aligned}$$

The value functions in Definition 7 are defined on infinite horizon, so they naturally hold a recursive structure, which we refer to as the self-consistency condition. We present the self-consistency conditions of value functions in the following theorem and define the corresponding operators.

Theorem 3 (performance self-consistency conditions). *The value functions satisfy the following self-consistency conditions:*

$$\begin{aligned} Q^{\pi, \mu}(x, u, a) &= r(x, u, a) \\ &+ \gamma \sum_{u' \in \mathcal{U}} \pi(u' | x') \sum_{a' \in \mathcal{A}} \mu(a' | x') Q^{\pi, \mu}(x', u', a'), \end{aligned} \quad (17)$$

$$\begin{aligned} Q^\pi(x, u, a) &= r(x, u, a) \\ &+ \gamma \sum_{u' \in \mathcal{U}} \pi(u' | x') \min_{a' \in \mathcal{A}} Q^\pi(x', u', a'), \end{aligned} \quad (18)$$

$$\begin{aligned} Q^*(x, u, a) &= r(x, u, a) \\ &+ \gamma \max_{\pi(\cdot | x') \in \Pi_s^*} \sum_{u' \in \mathcal{U}} \pi(u' | x') \min_{a' \in \mathcal{A}} Q^*(x', u', a'), \end{aligned} \quad (19)$$

in which $x' = f(x, u, a)$. (19) only applies to the optimal induced zero-sum Markov game \mathcal{M}^* .

Proof. (17) and (18) are the same as the standard self-consistency conditions in zero-sum Markov games [28]. Since the optimal induced zero-sum Markov game \mathcal{M}^* is well-defined, (19) follows from considering the Bellman equation on \mathcal{M}^* . \square

Definition 8 (performance operators). *The self-consistency operator of a pair of protagonist policy π_h and adversary policy μ_h is defined as*

$$\begin{aligned} [T^{\pi, \mu}(Q)](x, u, a) &= r(x, u, a) \\ &+ \gamma \sum_{u' \in \mathcal{U}} \pi(u' | x') \sum_{a' \in \mathcal{A}} \mu(a' | x') Q(x', u', a'). \end{aligned} \quad (20)$$

The self-consistency operator of a protagonist policy π_h is defined as

$$\begin{aligned} [T^\pi(Q)](x, u, a) &= r(x, u, a) \\ &+ \gamma \sum_{u' \in \mathcal{U}} \pi(u' | x') \min_{a' \in \mathcal{A}} Q(x', u', a'). \end{aligned} \quad (21)$$

The Bellman operator on the optimal induced zero-sum Markov game \mathcal{M}^* is defined as

$$\begin{aligned} [T(Q)](x, u, a) &= r(x, u, a) \\ &+ \gamma \max_{\pi(\cdot | x') \in \Pi_s^*} \sum_{u' \in \mathcal{U}} \pi(u' | x') \min_{a' \in \mathcal{A}} Q(x', u', a'). \end{aligned} \quad (22)$$

3.3 Dual Policy Iteration

A key issue of problem (16) is that we do not know the maximal robust invariant set S_r^* or the optimal safety value function Q_h^* in advance. To address this challenge, we propose a dual policy iteration scheme, which simultaneously optimizes two policies: the task policy and the safety policy. The safety policy seeks the highest safety values. The task policy seeks the highest rewards inside the robust invariant set specified by the safety policy and seeks the highest safety value outside the robust invariant set. The pseudo-code of dual policy iteration is presented in Algorithm 1. Note that for the first time of task policy evaluation, we check that $\max_{x \in \mathcal{X}} \max_{u \in \mathcal{U}} \min_{a \in \mathcal{A}} Q_h^{\pi_h}(x, u, a) \geq 0$ holds, otherwise go back to safety policy evaluation.

Algorithm 1: Dual Policy Iteration

```

Input: initial task policy  $\pi$ , initial safety policy  $\pi_h$ .
for  $m$  times do
  for  $n$  times do
    (safety policy evaluation)
    Solve for  $Q_h^{\pi_h}$  such that  $T_h^{\pi_h}(Q_h^{\pi_h}) = Q_h^{\pi_h}$ .
    (safety policy improvement)
    for each  $x \in \mathcal{X}$  do
       $\pi_h(x) \leftarrow \arg \max_{u \in \mathcal{U}} \left\{ \min_{a \in \mathcal{A}} Q_h^{\pi_h}(x, u, a) \right\}$ .
    end
  end
  (task policy evaluation)
  Solve for  $Q^\pi$  such that  $T^\pi(Q^\pi) = Q^\pi$ .
  (task policy improvement)
  for each  $x \in S_r^{\pi_h}$  do
     $\pi(\cdot | x) \leftarrow \arg \max_{\pi(\cdot | x) \in \Pi_s^{\pi_h}} \min_{a \in \mathcal{A}} \sum_{u \in \mathcal{U}} \pi(u | x) Q^\pi(x, u, a)$ .
  end
  for each  $x \notin S_r^{\pi_h}$  do
     $\pi(x) \leftarrow \pi_h(x)$ .
  end
end

```

Remark 3. The task policy improvement on state $x \in S_r^{\pi_h}$ is equivalent to the following linear programming problem:

$$\begin{aligned} &\max_{(\pi(\cdot | x), c)} c \\ \text{s.t. } &\sum_{u \in \mathcal{U}} \pi(u | x) Q^\pi(x, u, a) \geq c, \forall a \in \mathcal{A}, \\ &\sum_{u \in \mathcal{U}} \pi(u | x) = 1, \\ &\pi(u | x) \geq 0, \forall u \in U_s^{\pi_h}(x), \\ &\pi(u | x) = 0, \forall u \notin U_s^{\pi_h}(x). \end{aligned} \quad (23)$$

Note that Q^π in (23) is a constant, which is obtained by previous task policy evaluation.

Remark 4. In dual policy iteration, the task policy can be stochastic while the safety policy is restricted to the class of deterministic policies. As shown in Proposition 1, only considering deterministic policies when solving for Q_h^* does not lose any optimality. In task policy improvement, for states inside the robust invariant set, it is equivalent to finding the Nash equilibrium of a matrix game, as shown in Remark 3, which requires the task policy to be stochastic to guarantee the existence of a Nash equilibrium. For states outside the robust invariant set, we only optimize their safety values, so we directly copy the deterministic safety policy to the task policy: $\pi(x) \leftarrow \pi_h(x)$.

Remark 5. The hyperparameter n in Algorithm 1 controls the relative update frequency between the task policy and the safety policy. An extreme case is that by choosing an n sufficiently large, we have optimal safety value Q_h^* and the maximal robust invariant set S_r^* precomputed before optimizing the task policy. This case works well in the tabular setting. However, its corresponding deep RL algorithm suffers from the distribution mismatch issue, i.e., the pretrained Q_h and the task policy π are on different data basis, which may lead to poor algorithm performance.

We prove that both policies converge to the optimal ones in the following theorem.

Theorem 4 (convergence of dual policy iteration). *By choosing an m sufficiently large in Algorithm 1, the task policy π converges to the solution of problem (16) and the safety value $Q_h^{\pi_h}$ of the safety policy π_h converges to the fixed point of safety Bellman operator T_h .*

Proof. First we prove the convergence of safety policy π_h . Q_h is regarded as a vector in Euclidean space, i.e., $Q_h \in \mathbb{R}^{|\mathcal{X}| \cdot |\mathcal{U}| \cdot |\mathcal{A}|}$. Let k denote the iteration number of the safety policy. We set out to prove the following relationship:

$$Q_h^{\pi_h^k} \leq T_h(Q_h^{\pi_h^k}) \leq Q_h^{\pi_h^{k+1}} \leq Q_h^*. \quad (24)$$

Based on the definition of T_h and safety policy improvement, we have

$$T_h(Q_h^{\pi_h^k}) = T_h^{\pi_h^{k+1}}(Q_h^{\pi_h^k}) \geq T_h^{\pi_h^k}(Q_h^{\pi_h^k}) = Q_h^{\pi_h^k}.$$

Since $Q_h^{\pi_h^k} \leq T_h^{\pi_h^{k+1}}(Q_h^{\pi_h^k})$, using the monotone contraction of $T_h^{\pi_h^{k+1}}$, we have

$$\begin{aligned} Q_h^{\pi_h^k} &\leq T_h(Q_h^{\pi_h^k}) = T_h^{\pi_h^{k+1}}(Q_h^{\pi_h^k}) \\ &\leq \left(T_h^{\pi_h^{k+1}}\right)^\infty(Q_h^{\pi_h^k}) = Q_h^{\pi_h^{k+1}}. \end{aligned}$$

Since $T_h(Q_h^{\pi_h^{k+1}}) \geq T_h^{\pi_h^{k+1}}(Q_h^{\pi_h^{k+1}}) = Q_h^{\pi_h^{k+1}}$, using the monotone contraction of T_h , we obtain

$$Q_h^* = (T_h)^\infty(Q_h^{\pi_h^{k+1}}) \geq \dots \geq Q_h^{\pi_h^{k+1}}.$$

So we conclude that (24) holds. The safety value sequence $\{Q_h^{\pi_h^k}\}$ is monotone and bounded, so it converges. After the safety policy convergences, we have $Q_h^{\pi_h^k} = T_h(Q_h^{\pi_h^k}) =$

$Q_h^{\pi_h^{k+1}}$, indicating that the sequence $\{Q_h^{\pi_h^k}\}$ converges to the fixed point of T_h . The monotonicity of the safety value sequence also implies that the robust invariant set $S_r^{\pi_h}$ is expanding.

Next we prove the convergence of task policy π . Let j denote the iteration number of the task policy. The task policy improvement for $x \in S_r^{\pi_h}$ is equivalent to a standard policy improvement on the induced zero-sum Markov game $\mathcal{M}^{\pi_h} = \left(S_r^{\pi_h}, \bigcup_x U_s^{\pi_h}(x), \mathcal{A}, f, r, \gamma\right)$. Therefore, we have

$$Q^{\pi_{j+1}}(x, u, a) \geq Q^{\pi_j}(x, u, a), \forall x \in S_r^{\pi_h}, u \in U_s^{\pi_h}(x), a \in \mathcal{A}.$$

As the iteration of safety policy π_h goes on, $S_r^{\pi_h}$ and the corresponding $U_s^{\pi_h}(x)$ are expanding and converging to S_r^* and $U_s^*(x)$. We can conclude that the value function Q^π on S_r^* will converge to the fixed point of the Bellman operator on the optimal induced zero-sum Markov game \mathcal{M}^* (22). For states outside the maximal robust invariant set, i.e., $x \notin S_r^*$, the task policy directly copies the safety policy, so its convergence is the same as the safety policy. In summary, for states inside the maximal robust invariant set, the converged task policy seeks the highest total rewards under the condition of persistent safety is guaranteed; for states outside the maximal robust invariant set, the converged task policy seeks the highest constraint values as the converged safety policy does. Therefore, the iteration of task policy converges to the solution of problem (16). \square

Remark 6. We highlight that during the iteration, the robust invariant set of the task policy is also non-shrinking. Since the task policy improvement searches for a new task policy π in the invariant policy set $\Pi_s^{\pi_h}$, the robust invariant set S_r^π of the task policy π is at least the same size as $S_r^{\pi_h}$, i.e., $x \in S_r^{\pi_h} \rightarrow x \in S_r^\pi$. This merit contributes to the stability of the training process of our proposed DRAC.

Remark 7. Our proposed dual policy iteration also yields a sound solution for the common safe RL problem, when disturbances do not exist. The robust invariant sets degenerate to the standard invariant sets. Both safety policy and task policy also converge to the optimal ones in the no-disturbance case.

4 DUALLY ROBUST ACTOR-CRITIC

Based on the proposed dual policy iteration scheme, in this section, we present dually robust actor-critic (DRAC), a deep RL algorithm that can learn one policy that is safe and simultaneously robust to both performance and safety attacks. Our algorithm is built on top of soft actor-critic (SAC) [31], a well-known model-free off-policy RL algorithm.

We denote the task policy network as $\pi(x; \theta)$ and the safety policy network as $\pi_h(x; \phi)$. Since the min operations in both task policy evaluation and safety policy evaluation are hard to conduct for continuous state and action spaces, we train two adversary networks: the performance adversary network $\mu(x; \beta)$ and the safety adversary network $\mu_h(x; \xi)$. The task policy and performance adversary are stochastic, while the safety policy and safety adversary are deterministic. We follow the double Q-network design in SAC and make use of two performance value networks, denoted as $Q(x, u, a; \omega_1)$ and $Q(x, u, a; \omega_2)$. The safety value network is denoted as $Q_h(x, u, a; \psi)$.

For a set \mathcal{D} of collected samples, the loss functions of performance value networks are

$$L_Q(\omega_i) = \mathbb{E}_{(x,u,a,r,x') \sim \mathcal{D}} \left\{ \left(Q(x, u, a; \omega_i) - \hat{Q} \right)^2 \right\},$$

where $i = 1, 2$ and

$$\hat{Q} = r(x, u, a) + \gamma (Q(x', u', a'; \hat{\omega}_j) - \alpha \log \pi(u' | x'; \theta)),$$

in which j is randomly chosen from $\{1, 2\}$, $\hat{\omega}_j$ denotes the target nework parameters, $u' \sim \pi(\cdot | x'; \theta)$, $a' = \mu(\cdot | x'; \beta)$ and α denotes the temperature. The loss function for the temperature α is

$$L(\alpha) = \mathbb{E}_{x \sim \mathcal{D}} \{-\alpha \log \pi(u | x; \theta) - \alpha \mathcal{H}\},$$

in which \mathcal{H} represents the target entropy and $u \sim \pi(\cdot | x; \theta)$. The loss function of safety value network is

$$L_{Q_h}(\psi) = \mathbb{E}_{(x,u,a,h,x') \sim \mathcal{D}} \left\{ \left(Q_h(x, u, a; \psi) - \hat{Q}_h \right)^2 \right\},$$

where

$$\hat{Q}_h = (1 - \gamma_h)h(x) + \gamma_h \min \left\{ h(x), Q_h(x', u', a'; \hat{\psi}) \right\},$$

in which $u' = \pi_h(x'; \phi)$, $a' = \mu_h(x'; \xi)$ and $\hat{\psi}$ denote the target network parameters. The loss functions of the safety policy and the safety adversary are

$$L_{\pi_h}(\phi) = -\mathbb{E}_{x \sim \mathcal{D}} \{Q_h(x, u, a; \psi)\},$$

$$L_{\mu_h}(\xi) = \mathbb{E}_{x \sim \mathcal{D}} \{Q_h(x, u, a; \psi)\},$$

in which $u = \pi_h(x; \phi)$ and $a = \mu_h(x; \xi)$. The loss function of the performance adversary is

$$L_\mu(\beta) = \mathbb{E}_{x \sim \mathcal{D}} \{Q(x, u, a; \omega_i)\},$$

in which $u \sim \pi(\cdot | x; \phi)$, $a \sim \mu(\cdot | x; \xi)$ and j is randomly chosen from $\{1, 2\}$.

To ensure persistent safety under worst-case safety attacks, the task policy π must belong to the invariant policy set $\Pi_s^{\pi_h}$ identified by the safety value function Q_h , i.e., $Q_h(x, u, a; \psi) \geq 0$ for $u \sim \pi(\cdot | x; \phi)$ and $a = \mu_h(x; \xi)$. We utilize the method of Lagrange multipliers to carry out the constrained policy optimization on π . For continuous state and action spaces, there are infinite constraints on the task policy π , so we adopt a Lagrange multiplier network $\lambda(x; \zeta)$ to facilitate the learning process [32]. The Lagrangian is formulated as

$$\mathcal{L}(\theta, \zeta) = \mathbb{E}_{x \in \mathcal{X}} \{Q(x, u, a_1; \omega_j) + \lambda(x; \zeta)Q_h(x, u, a_2; \psi)\}, \quad (25)$$

in which $u \sim \pi(\cdot | x; \theta)$, $a_1 \sim \mu(x; \beta)$, $a_2 = \mu_h(x; \xi)$ and j is randomly chosen from $\{1, 2\}$. We solve for the saddle point of the Lagrangian using dual ascent. For states outside the maximal robust invariant set S_r^* , their safety values are always smaller than zero. Therefore, the corresponding Lagrange multipliers $\lambda(x; \zeta)$ will go to $+\infty$. In this case, the second term becomes dominant in (25), which indicates that we only optimize the safety value functions for states outside S_r^* , as in dual policy iteration. For practical implementations, we set an upper bound λ_{max} for the outputs of $\lambda(x; \zeta)$. The loss function of the task policy is

$$L_\pi(\theta) = \mathbb{E}_{x \sim \mathcal{D}} \{\alpha \log \pi(u | x; \theta) - Q(x, u, a_1; \omega_j)\} - \mathbb{E}_{x \sim \mathcal{D}} \{\lambda(x; \zeta)Q_h(x, u, a_2; \psi)\}, \quad (26)$$

in which $u \sim \pi(\cdot | x; \theta)$, $a_1 \sim \mu(x; \beta)$, $a_2 = \mu_h(x; \xi)$ and j is randomly chosen from $\{1, 2\}$. The loss function of the Lagrange multiplier network is composed of two parts, i.e., $L_\lambda(\zeta) = L_\lambda^A(\zeta) + L_\lambda^B(\zeta)$. For state x inside the current robust invariant set $S_r^{\pi_h}$, i.e., $Q_h(x, \pi_h(x; \phi), \mu_h(x; \xi); \psi) \geq 0$, the loss function is

$$L_\lambda^A(\zeta) = \mathbb{E}_{x \in \mathcal{D}} \{\lambda(x; \zeta)Q_h(x, u, a_2; \psi)\}. \quad (27)$$

For state x outside the current robust invariant set $S_r^{\pi_h}$, i.e., $Q_h(x, \pi_h(x; \phi), \mu_h(x; \xi); \psi) < 0$, the loss function is

$$L_\lambda^B(\zeta) = \mathbb{E}_{x \in \mathcal{D}} \{(\lambda(x; \zeta) - \lambda_{max})^2\}. \quad (28)$$

We should focus on optimizing the safety values of states outside the current robust invariant set, therefore (28) provides strong supervised signals for these states (the second term of (26) becomes dominant). We summarize the overall procedure of DRAC in Algorithm 2.

Algorithm 2: Dually Robust Actor-Critic

```

Input: network parameters  $\theta, \phi, \beta, \xi, \omega_1, \omega_2, \psi, \zeta$ ,
target network parameters  $\psi \leftarrow \psi, \bar{\omega}_1 \leftarrow \omega_1,$ 
 $\bar{\omega}_2 \leftarrow \omega_2$ , temperature  $\alpha$ , learning rate  $\eta$ ,
target smoothing coefficient  $\tau$ , replay buffer
 $\mathcal{D} \leftarrow \emptyset$ 

for each iteration do
  for each system step do
    Sample control input  $u_t \sim \pi(x_t; \theta)$ ;
    Sample disturbance  $a_t \sim \mu(\cdot | x_t; \beta)$  or
     $a_t = \mu_h(x_t; \xi)$  (random choice);
    Observe next state  $x_{t+1}$ , reward  $r_t$ , constraint
    value  $h_t$ ;
    Store transition
     $\mathcal{D} \leftarrow \mathcal{D} \cup \{(x_t, u_t, a_t, r_t, h_t, x_{t+1})\}$ .
  end
  for each gradient step do
    Sample a batch of data from  $\mathcal{D}$ ;
    Update safety value function
     $\psi \leftarrow \psi - \eta \nabla_\psi L_{Q_h}(\psi)$ ;
    Update value functions
     $\omega_i \leftarrow \omega_i - \eta \nabla_{\omega_i} L_Q(\omega_i)$  for  $i \in \{1, 2\}$ ;
    Update task policy  $\theta \leftarrow \theta - \eta \nabla_\theta L_\pi(\theta)$ ;
    Update safety policy  $\phi \leftarrow \phi - \eta \nabla_\phi L_{\pi_h}(\phi)$ ;
    Update performance adversary
     $\beta \leftarrow \beta - \eta \nabla_\beta L_\mu(\beta)$ ;
    Update safety adversary  $\xi \leftarrow \xi - \eta \nabla_\xi L_{\mu_h}(\xi)$ ;
    Update Lagrange multiplier
     $\zeta \leftarrow \zeta + \eta \nabla_\zeta L_\lambda(\zeta)$ ;
    Update temperature  $\alpha \leftarrow \alpha - \eta \nabla_\alpha L(\alpha)$ ;
    Update target networks  $\bar{\psi} \leftarrow \tau \psi + (1 - \tau) \psi$ ,
     $\bar{\omega}_i \leftarrow \tau \omega_i + (1 - \tau) \bar{\omega}_i$  for  $i \in \{1, 2\}$ .
  end
end

```

5 EXPERIMENTS

In this section, we evaluate our algorithm DRAC on safety-critical benchmark environments. We compare our algorithm with state-of-the-art safe RL and robust RL algorithms.

5.1 Environments

All algorithms are tested on three environments: CartPole, RacingCar and Walker2D.

CartPole is a classic control task based on MuJoCo [33], as illustrated in Fig. 1a. The goal is to push the cart to a target position as fast as possible. The state of the system includes cart position x , cart velocity v , pole angle θ and pole angular velocity ω . The safety constraint is imposed on the pole angle: $|\theta| \leq 0.2$. Both control inputs and external disturbances are level forces applied on the cart. The protagonist action space is $\mathcal{U} = [-1, 1]$ and the adversary action space is $\mathcal{A} = [-0.5, 0.5]$.

RacingCar is a safe RL benchmark based on PyBullet [34], as shown in Fig. 1b. The four-wheeled car needs to track the edge of the blue region accurately and quickly. The state space $\mathcal{X} \subset \mathbb{R}^7$. The safety constraint is staying inside the region between the two yellow boundaries. The protagonist action space is $\mathcal{U} = [-1, 1]^2$ and the adversary action space is $\mathcal{A} = [-0.25, 0.25]^2$.

Walker2D is a safe RL benchmark based on MuJoCo [33], as shown in Fig. 1c. The agent is a two-dimensional two-legged figure with a state space $\mathcal{X} \subset \mathbb{R}^{17}$. The goal is to move as far as possible with minimal control efforts. The safety constraint is imposed on the angle θ and height z of the torso: $|\theta| \leq 0.25, 1.0 \leq z \leq 1.8$. The control inputs $u \in \mathbb{R}^6$ are torques applied on the hinge joints. The external disturbances $a \in \mathbb{R}^6$ are forces applied on the torso and both feet.

During training, we do not terminate the episodes when safety constraints are violated, since the early termination trick may confuse the reward-seeking and safety-preserving capability of safe RL algorithms, leading to vague results. Safe RL algorithms should figure out ways of maintaining safety purely based on the signals of constraint function $h(x)$. For algorithms without safety considerations, we equip them with the reward shaping trick (adding a bonus to the reward when no constraint is violated) for fair comparison.

5.2 Baselines

We compare DRAC to the following baselines.

Soft Actor-Critic with Reward Shaping (SAC-Rew, [31]): The standard SAC algorithm with additional bonuses added to the original rewards for encouragement of constraint satisfaction.

Robust Soft Actor-Critic with Reward Shaping (RSAC-Rew, [29]): The SAC version of robust adversarial reinforcement learning (RARL), a state-of-the-art robust RL algorithm, with additional bonuses added to the original rewards for encouragement of constraint satisfaction.

Soft Actor-Critic with Lagrange Multiplier (SAC-Lag, [10]): The combination of SAC and the method of Lagrange multipliers, a state-of-the-art safe RL algorithm.

Reachable Actor-Critic (RAC, [22]): The combination of SAC and the reachability constraint, which makes use of Hamilton-Jacobi reachability analysis (under the assumption of no disturbances), a state-of-the-art safe RL algorithm.

Soft Actor-Critic with Robust Invariant Set (SAC-RIS, ours): A weaker implementation of our proposed DRAC, which only focuses on robustness against safety attacks. The performance adversary in DRAC is removed.

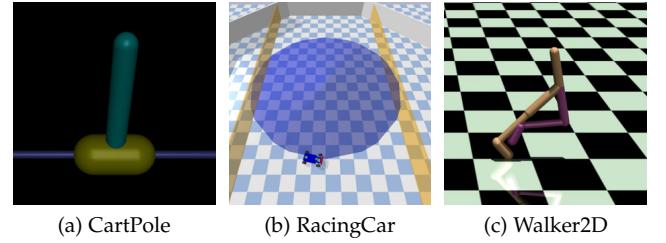


Fig. 1. Snapshots of three safety-critical environments.

5.3 Results

We adopt two evaluation metrics for each algorithm in the training process. (1) **episode return** indicates the performance of the learned agents. (2) **episode constraint violation** identifies the safety-preserving capability of the learned agents.

All algorithms are tested under three scenarios. (1) **no adversary**: There are no external disturbances in the environments. (2) **safety adversary**: The task policies are evaluated under the attacks of the learned safety adversary from our algorithm DRAC. This scenario compares the safety-preserving robustness of different algorithms. (3) **performance adversary**: The task policies are evaluated under the attacks of the learned performance adversary from our algorithm DRAC. This scenario compares the reward-seeking robustness of different algorithms.

The learning curves for three environments are shown in Fig. 2. Since safety is the top priority, it is meaningless to attain high rewards when safety constraints are violated. SAC-RIS and DRAC are the only two algorithms that can maintain persistent safety under all scenarios. SAC-Rew and RSAC-Rew violate the safety constraints heavily, even in the case of no adversary. This is due to the lack of safety-preserving design in their mechanisms. Although the reward shaping trick encourages constraint satisfaction, it is far from enough to achieve persistent safety. SAC-Lag and RAC can maintain safety when no disturbances exist, but their safety-preserving capabilities are compromised under safety or performance attacks. Despite the fact that RAC-RIS is not trained with performance adversary, it exhibits excellent safety-preserving capability under the unseen performance attacks. This further justifies the concept of robust invariant sets, inside which the system can maintain safety under any form of safety adversaries. Compared to SAC-RIS, DRAC achieves considerably higher performance, which justifies the effectiveness of considering performance adversary. The policies learned by DRAC are robust to both performance and safety attacks. Furthermore, they attain higher rewards than SAC-Lag and RAC even in the absence of adversary.

6 CONCLUSION

In this paper, we propose a systematic framework to unify safe RL and robust RL, including the problem formulation, iteration scheme, convergence analysis and practical algorithm design. The unification is built upon constrained two-player zero-sum Markov games, in which a twofold objective is designed. We propose a dual policy iteration

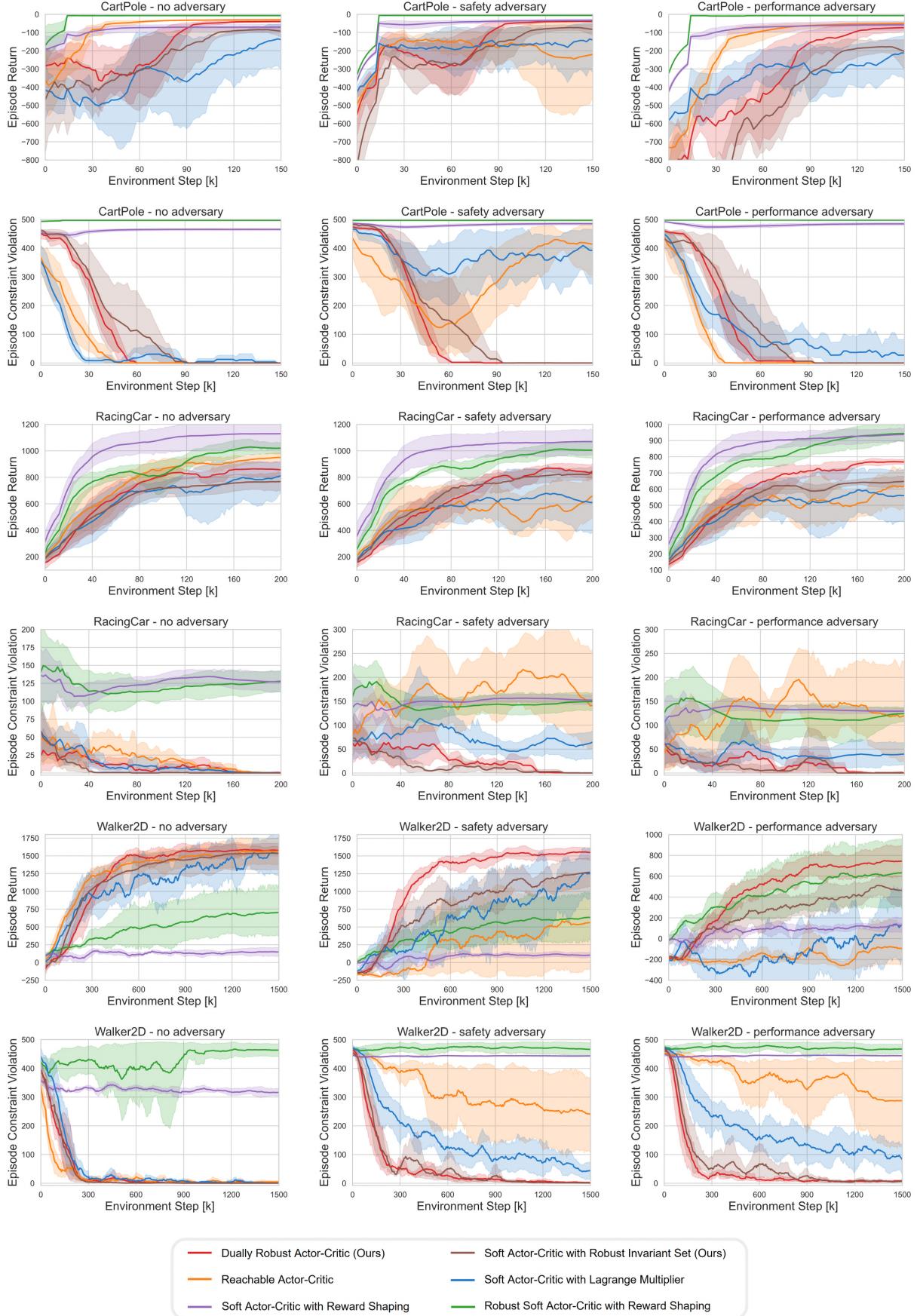


Fig. 2. Training curves on three environments. For each environment, the first row corresponds to episode return and the second row corresponds to episode constraint violation. The solid lines correspond to the mean and the shaded regions correspond to 95% confidence interval over five seeds.

scheme that jointly optimizes task policy and safety policy. Safety value function is utilized to characterize the riskiness of states under disturbances. The safety policy seeks the highest safety value for each state, identifying the robust invariant set, which serves as constraint for the task policy. We prove that the proposed scheme converges to the optimal task policy and the optimal safety policy. Furthermore, we propose dually robust actor-critic (DRAC), a deep RL algorithm that is robust to both safety and performance attacks. Experimental results demonstrate the effectiveness of our algorithm.

REFERENCES

- [1] D. Silver, J. Schrittwieser, K. Simonyan, I. Antonoglou, A. Huang, A. Guez, T. Hubert, L. Baker, M. Lai, A. Bolton *et al.*, "Mastering the game of go without human knowledge," *Nature*, vol. 550, no. 7676, pp. 354–359, 2017.
- [2] J. Hwangbo, J. Lee, A. Dosovitskiy, D. Bellicoso, V. Tsounis, V. Koltun, and M. Hutter, "Learning agile and dynamic motor skills for legged robots," *Science Robotics*, vol. 4, no. 26, p. eaau5872, 2019.
- [3] C. Huang, G. Wang, Z. Zhou, R. Zhang, and L. Lin, "Reward-adaptive reinforcement learning: Dynamic policy gradient optimization for bipedal locomotion," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022.
- [4] S. Feng, H. Sun, X. Yan, H. Zhu, Z. Zou, S. Shen, and H. X. Liu, "Dense reinforcement learning for safety validation of autonomous vehicles," *Nature*, vol. 615, no. 7953, pp. 620–627, 2023.
- [5] Y. Guan, Y. Ren, Q. Sun, S. E. Li, H. Ma, J. Duan, Y. Dai, and B. Cheng, "Integrated decision and control: Toward interpretable and computationally efficient driving intelligence," *IEEE Transactions on Cybernetics*, vol. 53, no. 2, pp. 859–873, 2023.
- [6] J. Garcia and F. Fernández, "A comprehensive survey on safe reinforcement learning," *Journal of Machine Learning Research*, vol. 16, no. 1, pp. 1437–1480, 2015.
- [7] H. Ju, R. Juan, R. Gomez, K. Nakamura, and G. Li, "Transferring policy of deep reinforcement learning from simulation to reality for robotics," *Nature Machine Intelligence*, vol. 4, pp. 1–11, 2022.
- [8] J. Li, S. E. Li, J. Duan, Y. Lyu, W. Zou, Y. Guan, and Y. Yin, "Relaxed policy iteration algorithm for nonlinear zero-sum games with application to h-infinity control," *IEEE Transactions on Automatic Control*, 2023.
- [9] S. E. Li, *Reinforcement Learning for Sequential Decision and Optimal Control*. Springer, 2023.
- [10] S. Ha, P. Xu, Z. Tan, S. Levine, and J. Tan, "Learning to walk in the real world with minimal human effort," in *Proceedings of the 2020 Conference on Robot Learning*, vol. 155. PMLR, 2021, pp. 1110–1120.
- [11] Y. Chow, M. Ghavamzadeh, L. Janson, and M. Pavone, "Risk-constrained reinforcement learning with percentile risk criteria," *The Journal of Machine Learning Research*, vol. 18, no. 1, pp. 6070–6120, 2017.
- [12] C. Tessler, D. J. Mankowitz, and S. Mannor, "Reward constrained policy optimization," in *International Conference on Learning Representations*, 2019.
- [13] J. Achiam, D. Held, A. Tamar, and P. Abbeel, "Constrained policy optimization," in *International Conference on Machine Learning*. PMLR, 2017, pp. 22–31.
- [14] T.-Y. Yang, J. Rosca, K. Narasimhan, and P. J. Ramadge, "Projection-based constrained policy optimization," in *International Conference on Learning Representations*, 2020.
- [15] M. Korda, D. Henrion, and C. N. Jones, "Convex computation of the maximum controlled invariant set for polynomial control systems," *SIAM Journal on Control and Optimization*, vol. 52, no. 5, pp. 2944–2969, 2014.
- [16] 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*, vol. 62, no. 8, pp. 3861–3876, 2016.
- [17] T. Wei and C. Liu, "Safe control algorithms using energy functions: A unified framework, benchmark, and new directions," in *2019 IEEE 58th Conference on Decision and Control (CDC)*. IEEE, 2019, pp. 238–243.
- [18] H. Ma, C. Liu, S. E. Li, S. Zheng, and J. Chen, "Joint synthesis of safety certificate and safe control policy using constrained reinforcement learning," in *Learning for Dynamics and Control Conference*. PMLR, 2022, pp. 97–109.
- [19] Y. Yang, Y. Jiang, Y. Liu, J. Chen, and S. E. Li, "Model-free safe reinforcement learning through neural barrier certificate," *IEEE Robotics and Automation Letters*, 2023.
- [20] K. Margellos and J. Lygeros, "Hamilton-jacobi formulation for reach-avoid differential games," *IEEE Transactions on Automatic Control*, vol. 56, no. 8, pp. 1849–1861, 2011.
- [21] J. F. Fisac, N. F. Lugovoy, V. Rubies-Royo, S. Ghosh, and C. J. Tomlin, "Bridging hamilton-jacobi safety analysis and reinforcement learning," in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 8550–8556.
- [22] D. Yu, H. Ma, S. Li, and J. Chen, "Reachability constrained reinforcement learning," in *International Conference on Machine Learning*. PMLR, 2022, pp. 25636–25655.
- [23] J. Moos, K. Hansel, H. Abdulsamad, S. Stark, D. Clever, and J. Peters, "Robust reinforcement learning: A review of foundations and recent advances," *Machine Learning and Knowledge Extraction*, vol. 4, no. 1, pp. 276–315, 2022.
- [24] H. Zhang, H. Chen, C. Xiao, B. Li, M. Liu, D. Boning, and C.-J. Hsieh, "Robust deep reinforcement learning against adversarial perturbations on state observations," in *Advances in Neural Information Processing Systems*, vol. 33, 2020, pp. 21024–21037.
- [25] C. Tessler, Y. Efroni, and S. Mannor, "Action robust reinforcement learning and applications in continuous control," in *International Conference on Machine Learning*. PMLR, 2019, pp. 6215–6224.
- [26] A. Nilim and L. Ghaoui, "Robustness in markov decision problems with uncertain transition matrices," in *Advances in Neural Information Processing Systems*, vol. 16. MIT Press, 2003.
- [27] J. Wang, Y. Liu, and B. Li, "Reinforcement learning with perturbed rewards," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, 2020, pp. 6202–6209.
- [28] J. Perolat, B. Scherrer, B. Piot, and O. Pietquin, "Approximate dynamic programming for two-player zero-sum markov games," in *Proceedings of the 32nd International Conference on Machine Learning*, vol. 37. PMLR, 2015, pp. 1321–1329.
- [29] L. Pinto, J. Davidson, R. Sukthankar, and A. Gupta, "Robust adversarial reinforcement learning," in *International Conference on Machine Learning*. PMLR, 2017, pp. 2817–2826.
- [30] Y. Zhu and D. Zhao, "Online minimax q network learning for two-player zero-sum markov games," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 33, no. 3, pp. 1228–1241, 2020.
- [31] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine, "Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor," in *International Conference on Machine Learning*. PMLR, 2018, pp. 1861–1870.
- [32] H. Narasimhan, A. Cotter, Y. Zhou, S. Wang, and W. Guo, "Approximate heavily-constrained learning with lagrange multiplier models," *Advances in Neural Information Processing Systems*, vol. 33, pp. 8693–8703, 2020.
- [33] E. Todorov, T. Erez, and Y. Tassa, "Mujoco: A physics engine for model-based control," in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2012, pp. 5026–5033.
- [34] E. Coumans and Y. Bai, "Pybullet, a python module for physics simulation for games, robotics and machine learning," <http://pybullet.org>, 2016–2021.



Zeyang Li received the B.S. degree in mechanical engineering in 2021, from the School of Mechanical Engineering, Shanghai Jiao Tong University, Shanghai, China. He is currently working toward the M.S. degree in mechanical engineering with the Department of Mechanical Engineering, Tsinghua University, Beijing, China. His research interests include reinforcement learning and optimal control.



Chuxiong Hu (Senior Member, IEEE) received his B.S. and Ph.D. degrees in Mechatronic Control Engineering from Zhejiang University, Hangzhou, China, in 2005 and 2010, respectively. He is currently an Associate Professor (tenured) at Department of Mechanical Engineering, Tsinghua University, Beijing, China. From 2007 to 2008, he was a Visiting Scholar in mechanical engineering with Purdue University, West Lafayette, USA. In 2018, he was a Visiting Scholar in mechanical engineering with University

of California, Berkeley, CA, USA. His research interests include precision motion control, high-performance multiaxis contouring control, precision mechatronic systems, intelligent learning, adaptive robust control, neural networks, iterative learning control, and robot. Prof. Hu was the recipient of the Best Student Paper Finalist at the 2011 American Control Conference, the 2012 Best Mechatronics Paper Award from the ASME Dynamic Systems and Control Division, the 2013 National 100 Excellent Doctoral Dissertations Nomination Award of China, the 2016 Best Paper in Automation Award, the 2018 Best Paper in AI Award from the IEEE International Conference on Information and Automation, and 2022 Best Paper in Theory from the IEEE/ASME International Conference on Mechatronic, Embedded Systems and Applications. He is now an Associate Editor for the IEEE Transactions on Industrial Informatics and a Technical Editor for the IEEE/ASME Transactions on Mechatronics.



Yunan Wang (Graduate Student Member, IEEE) received the B.S. degree in mechanical engineering, in 2022, from the Department of Mechanical Engineering, Tsinghua University, Beijing, China. He is currently working toward the Ph.D. degree in mechanical engineering. His research interests include optimal control, trajectory planning, toolpath planning, and precision motion control. He was the recipient of the Best Conference Paper Finalist at the 2022 International Conference on Advanced Robotics and Mechatronics, and 2021 Top Grade Scholarship for Undergraduate Students of Tsinghua University.



Yujie Yang received his B.S. degree in automotive engineering from the School of Vehicle and Mobility, Tsinghua University, Beijing, China, in 2021. He is currently pursuing his Ph.D. degree in the School of Vehicle and Mobility, Tsinghua University, Beijing, China. His research interests include decision and control of autonomous vehicles and safe reinforcement learning.



Shengbo Eben Li (Senior Member, IEEE) received his M.S. and Ph.D. degrees from Tsinghua University in 2006 and 2009. Before joining Tsinghua University, he has worked at Stanford University, University of Michigan, and UC Berkeley. His active research interests include intelligent vehicles and driver assistance, deep reinforcement learning, optimal control and estimation, etc. He is the author of over 130 peer-reviewed journal/conference papers, and the co-inventor of over 30 patents. He is the recipient of best (student) paper awards of IEEE ITSC, ICCAS, IEEE ICUS, CCCC, etc. His important awards include National Award for Technological Invention of China (2013), Excellent Young Scholar of NSF China (2016), Young Professor of ChangJiang Scholar Program (2016), National Award for Progress in Sci & Tech of China (2018), Distinguished Young Scholar of Beijing NSF (2018), Youth Sci & Tech Innovation Leader from MOST (2020), etc. He also serves as Board of Governor of IEEE ITS Society, Senior AE of IEEE OJ ITS, and AEs of IEEE ITSM, IEEE Trans ITS, Automotive Innovation, etc.