

Robust Safe Reinforcement Learning under Adversarial Disturbances

Zeyang Li, Chuxiong Hu, Shengbo Eben Li, Jia Cheng, Yunan Wang

Abstract—Safety is a primary concern when applying reinforcement learning to real-world control tasks, especially in the presence of external disturbances. However, existing safe reinforcement learning algorithms rarely account for external disturbances, limiting their applicability and robustness in practice. To address this challenge, this paper proposes a robust safe reinforcement learning framework that tackles worst-case disturbances. First, this paper presents a policy iteration scheme to solve for the robust invariant set, i.e., a subset of the safe set, where persistent safety is only possible for states within. The key idea is to establish a two-player zero-sum game by leveraging the safety value function in Hamilton-Jacobi reachability analysis, in which the protagonist (i.e., control inputs) aims to maintain safety and the adversary (i.e., external disturbances) tries to break down safety. This paper proves that the proposed policy iteration algorithm converges monotonically to the maximal robust invariant set. Second, this paper integrates the proposed policy iteration scheme into a constrained reinforcement learning algorithm that simultaneously synthesizes the robust invariant set and uses it for constrained policy optimization. This algorithm tackles both optimality and safety, i.e., learning a policy that attains high rewards while maintaining safety under worst-case disturbances. Experiments on classic control tasks show that the proposed method achieves zero constraint violation with learned worst-case adversarial disturbances, while other baseline algorithms violate the safety constraints substantially. Our proposed method also attains comparable performance as the baselines even in the absence of the adversary.

I. INTRODUCTION

Reinforcement learning (RL) [1] has achieved remarkable success in various fields, such as games [2], robotics [3], and autonomous driving [4]. RL aims at finding an optimal policy that maximizes the expected cumulative rewards. However, many real-world control tasks require not only maximization of rewards but also strict satisfaction of safety constraints, as violating these constraints can have catastrophic consequences [5], [6]. To tackle this problem, safe RL has emerged as a research area that aims at learning a safe optimal policy with zero constraint violation [7], [8].

There are mainly two categories of safety formulation in existing safe RL methods. The first one deals with safety in the framework of constrained Markov decision process (CMDP) [9], where an auxiliary cost signal indicates the state-action pairs that violate the safety constraint. The objective is to find a policy maximizing rewards under the

condition that the cost value is below a certain threshold. Algorithms for solving CMDPs include Lagrangian methods [10], [11], [12], penalty methods [13] and projection methods [14]. However, they cannot achieve zero constraint violation since the cost value is defined in expectation, and the potential danger at one state is averaged to the whole trajectory. The second category developed more recently, aims to ensure that the agent satisfies safety constraints at every state, and is referred to as state-wise safe RL [15]. This line of work adopts the rigorous definition and theoretical guarantee of safety in safe control research [5], [6], in which a key insight is that persistent safety can only be achieved within a subset of the safe set, termed invariant set [16], [17]. Outside this set, the safety constraints will be violated inevitably in the future even though they are temporarily satisfied, regardless of what actions the agent takes. Algorithms belonging to state-wise safe RL adopt energy functions such as control barrier function [6], [18] and safety index [19], [20] to ensure persistent safety and identify invariant sets, or utilize Hamilton-Jacobi reachability analysis [21], [22] which provides the theoretical maximal invariant set. These methods typically synthesize the invariant set and use it to conduct constrained policy optimization, achieving zero violation test results in several benchmark environments.

However, simulators used for safe RL training are inevitably imperfect approximations for real-world systems. There can always be uncertainties, such as model errors, sensory noises and environmental perturbations, which may lead to severe violations of safety constraints when transferring policies from simulators to real-world control tasks [23]. These uncertainties can be viewed as extra disturbances in the system [24]. For example, higher frictions at test time might be modeled as extra forces at contact points. Therefore, robustness against external disturbances is a crucial requirement for applying safe RL to physical systems. Nevertheless, most existing methods do not account for such robustness.

To address this challenge, this paper proposes a safe reinforcement learning framework that learns an optimal safe policy under worst-case disturbances. This research follows the second category of safe RL, i.e., ensuring state-wise safety. Just like the common cases without disturbances, persistent safety under worst-case disturbances can only be achieved in a subset of the safe set, named robust invariant set [17], [25] which is smaller than the standard invariant set due to the presence of disturbances. To attain optimal performance as well as ensure safety, the maximal robust invariant set must be identified. We propose a policy iteration scheme that converges monotonically to the maximal robust invariant set. The key idea is to establish a two-player

Zeyang Li, Chuxiong Hu, Jia Cheng 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; chengjia@tsinghua.edu.cn; wang-yn22@mails.tsinghua.edu.cn).

Shengbo Eben Li is with the School of Vehicle and Mobility, Tsinghua University, Beijing 100084, China (email: lishbo@tsinghua.edu.cn).

Corresponding author: Chuxiong Hu.

zero-sum game by leveraging the safety value function in Hamilton-Jacobi reachability analysis [21], in which the protagonist (i.e., control inputs) aims to maintain safety and the adversary (i.e., external disturbances) tries to break down safety. However, this policy iteration scheme only obtains the safest policy that seeks to stay far away from the boundary of the safe set. We further integrate it into a constrained reinforcement learning algorithm that simultaneously synthesizes the robust invariant set and uses it for constrained policy optimization. This algorithm attains one policy that satisfies both safety and optimality.

Our contributions are summarized as follows.

- We propose a policy iteration scheme for synthesizing the maximal robust invariant set. Robust invariant sets are represented by the safety value functions inspired by Hamilton-Jacobi reachability analysis, and a two-player zero-sum game is established. The self-consistency conditions of safety value functions is proved. Furthermore, we prove that the proposed policy iteration converges to the maximal robust invariant set monotonically, as well as the safest policy that seeks the highest safety value.
- We propose a constrained reinforcement learning algorithm that learns an optimal safe policy under worst-case disturbances. This algorithm simultaneously synthesizes the robust invariant set and uses it for constrained policy optimization which is solved by the Lagrangian multiplier method.
- Experiments on classic control tasks show that our method achieves zero constraint violation with learned worst-case adversarial disturbances while baseline algorithms violate the safety constraints substantially. Our method also attains comparable performance as the baselines even in the absence of the adversary.

II. PRELIMINARIES

A. Hamilton-Jacobi Reachability Analysis

Hamilton-Jacobi reachability analysis is a formal verification method for ensuring safety of general continuous nonlinear systems [21], [26]. It also includes formal treatment of external disturbances. Suppose the system dynamics is given by

$$\dot{x} = f(x, u, a), \quad x \in \mathcal{S}, u \in \mathcal{U}, a \in \mathcal{A}, \quad (1)$$

in which \mathcal{S} denotes the state space, \mathcal{U} denotes the input space, \mathcal{A} denotes disturbance space. The state flow map under the given control policy $\pi: \mathcal{S} \rightarrow \mathcal{U}$ and disturbance policy $\mu: \mathcal{S} \rightarrow \mathcal{A}$ is defined as

$$\Phi_{f, \pi, \mu}(x, t) = x + \int_0^t f(x(\tau), \pi(x(\tau)), \mu(x(\tau))) d\tau. \quad (2)$$

The safety constraint for the system is $h(x) \geq 0$. The key of Hamilton-Jacobi reachability analysis is to define a safety value function:

$$V_h^*(x, t) = \max_{\pi(\cdot)} \min_{\mu(\cdot)} \min_{\tau \in [t, T]} \{h(\Phi_{f, \pi, \mu}(x, \tau))\}, \quad (3)$$

which corresponds to the worst constraint value in the time horizon $[t, T]$ starting at current state x , under the best

possible control inputs and worst-case disturbances. The safety value function satisfies the following partial differential equation (PDE) [21]:

$$\min \left\{ h(x) - V_h^*(x, t), \frac{\partial V_h^*}{\partial t} + \max_{u \in \mathcal{U}} \min_{a \in \mathcal{A}} \frac{\partial V_h^*}{\partial x^T} f(x, u, a) \right\} = 0, \quad (4)$$

and the boundary condition is given by $V_h^*(x, T) = h(x)$.

B. State-wise Safe RL

State-wise safe RL [15] aims to ensure that the learned control policy satisfies the safety constraints on every state it visits. The problem is formulated as

$$\begin{aligned} \max_{\pi} \mathbb{E} \left\{ \sum_{t=0}^{\infty} \gamma^t r(x_t, u_t) \right\} \\ \text{s.t. } x_{t+1} = f(x_t, u_t), u_t = \pi(x_t), \\ h(x_t) \geq 0, t = 0, 1, 2, \dots, \infty. \end{aligned} \quad (5)$$

Note that the system in (5) is discrete-time, which is different from the setting in Hamilton-Jacobi reachability analysis. It is important to constrain the system state inside the invariant set for state-wise safe RL. Some works adopt energy functions such as control barrier function [6], [18] and safety index [19], [20] to ensure persistent safety and identify invariant sets. Others utilize Hamilton-Jacobi reachability analysis [22], which provides the theoretical maximal invariant set.

III. POLICY ITERATION FOR ROBUST INVARIANT SET

Despite the rigorous theoretical formulation of Hamilton-Jacobi reachability analysis, the computational complexity scales exponentially with respect to system dimension, referred to as the curse of dimensionality [26]. To overcome this challenge, recent works utilize RL techniques to approximate the safety value function. Fisac et al. propose a time-discounted modification of Hamilton-Jacobi reachability and solve it using RL algorithms [27]. Yu et al. propose a reachability constrained reinforcement learning algorithm that jointly synthesizes the optimal safe control policy and the invariant set [22]. However, these works assume that there is no disturbance.

This paper proposes a policy iteration scheme that converges monotonically to the maximal robust invariant set, i.e., states that allow for persistent safety under worst-case disturbances. The key idea is to utilize the safety value function (3). Note that Hamilton-Jacobi PDE (4) is for continuous-time systems with finite horizon T . To adapt it for infinite horizon RL on discrete-time systems, adjustments need to be made.

Consider a discrete system as

$$x_{t+1} = f(x_t, u_t, a_t), \quad x \in \mathcal{S}, u \in \mathcal{U}, a \in \mathcal{A}, \quad (6)$$

in which \mathcal{S} denotes the state space, \mathcal{U} denotes the action space of control inputs, \mathcal{A} denotes the action space of external disturbances. The control input u follows policy $\pi: \mathcal{S} \rightarrow \mathcal{U}$ (called protagonist) and disturbance a follows

policy $\mu : \mathcal{S} \rightarrow \mathcal{A}$ (called adversary). We define the safe set and safety value functions as follows.

Definition 1 (safe set): The safe set S_h is defined as the zero-superlevel set of the constraint function $h(x)$, i.e.,

$$S_h = \{x \mid h(x) \geq 0\}. \quad (7)$$

Definition 2 (safety value functions): Let $\{x_t^{\pi, \mu}\}$ denote the state trajectory when the system is driven by a protagonist policy π and an adversary policy μ .

The safety value function for a protagonist policy π and an adversary policy μ is defined as

$$V_h^{\pi, \mu}(x) = \min_{t \in \mathbb{N}} \{h(x_t^{\pi, \mu} \mid x_0 = x)\}. \quad (8)$$

The safety value function for a protagonist policy π is defined as

$$V_h^\pi(x) = \min_{\mu(\cdot)} \min_{t \in \mathbb{N}} \{h(x_t^{\pi, \mu} \mid x_0 = x)\}. \quad (9)$$

The optimal safety value function is defined as

$$V_h^*(x) = \max_{\pi(\cdot)} \min_{\mu(\cdot)} \min_{t \in \mathbb{N}} \{h(x_t^{\pi, \mu} \mid x_0 = x)\}. \quad (10)$$

The safety value functions quantify the riskiness of violating the safety constraints. $V_h^{\pi, \mu}(x)$ denotes the lowest constraint value (the most dangerous scenario) in the long term, when the system is driven by specified protagonist π and adversary μ . It varies if either the protagonist π or the adversary μ changes. $V_h^\pi(x)$ denotes the lowest constraint value when the system is driven by specified protagonist π and its worst-case adversary. It evaluates the safety-keeping ability of the policy π and varies if the policy π changes. $V_h^*(x)$ denotes the best safety value we can get where we choose the policy maximizing the constraint value (thus minimizing the constraint violation) against the worst-case adversary.

One can easily deduce that the maximal robust invariant set is the zero-superlevel set of the optimal safety value function V_h^* . States out of this set will violate the safety constraint in the future inevitably under worst-case disturbances (see Figure 1 for illustration). We formally define robust invariant sets in the following definition.

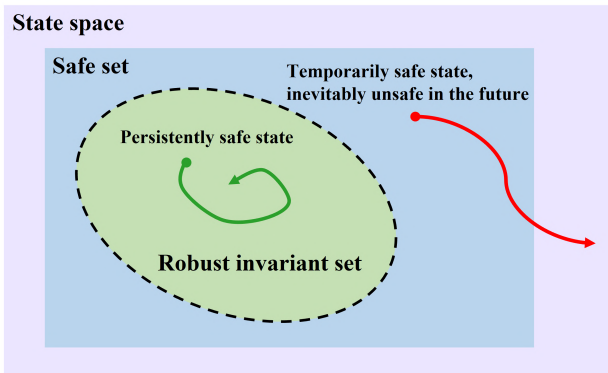


Fig. 1. Illustration of robust invariant sets. Purple region: state space. Blue region: safe set. Green region: robust invariant set. Persistent safety is only possible for states inside the robust invariant set. States outside the robust invariant set will inevitably violate the safety constraints under worst-case disturbances, even though they are temporarily safe.

Definition 3 (robust invariant sets): The robust invariant set of a given policy π is the zero-superlevel set of its safety value:

$$S_r^\pi = \{x \mid V_h^\pi(x) \geq 0\}. \quad (11)$$

The maximal robust invariant set is the zero-superlevel set of the optimal safety value:

$$S_r^* = \{x \mid V_h^*(x) \geq 0\}. \quad (12)$$

We have $S_r^\pi \subseteq S_r^* \subseteq S_h$ for any π .

Remark 1 (clarification): The invariant sets (standard or robust) in this article are actually controlled invariant sets. For states inside a robust controlled invariant set, there exists a control policy that keeps the system safe under any disturbances. We omit the term controlled for simplicity. Xue et al. propose RL approaches (value iteration, policy iteration) for estimating the robust invariant sets of perturbed discrete-time systems [28]. The systems they consider only have disturbances and there are no control inputs. The systems in this article are more general, containing both control inputs and external disturbances, which compete against each other.

Utilizing dynamic programming, we show that the safety value functions satisfy the following self-consistency conditions.

Theorem 1 (self-consistency conditions): Suppose x' is the successive state of x . The following self-consistency conditions hold for safety value functions, i.e.,

$$V_h^{\pi, \mu}(x) = \min \{h(x), V_h^{\pi, \mu}(x')\}, \quad (13)$$

$$V_h^\pi(x) = \min \left\{ h(x), \min_{a \in \mathcal{A}} \{V_h^\pi(x')\} \right\}, \quad (14)$$

$$V_h^*(x) = \min \left\{ h(x), \max_{u \in \mathcal{U}} \min_{a \in \mathcal{A}} \{V_h^*(x')\} \right\}. \quad (15)$$

The proof of theorems and propositions is provided in Appendix.

Note that x' in (13) satisfies $x' = f(x, \pi(x), \mu(x))$, x' in (14) is dependent on the disturbance a , and x' in (15) is dependent on both the control input u and disturbance a .

Unlike the common Bellman equation or self-consistency condition in RL, self-consistency conditions for safety value functions in Theorem 1 are not contraction mappings. We introduce contraction properties by modifying the original formulation with a discount factor γ_h .

Definition 4 (safety self-consistency operators): Suppose $0 < \gamma_h < 1$. The safety self-consistency operators are defined as

$$[T_h^{\pi, \mu}(V_h)](x) = (1 - \gamma_h)h(x) + \gamma_h \min \{h(x), V_h(x')\}, \quad (16)$$

$$[T_h^\pi(V_h)](x) = (1 - \gamma_h)h(x) + \gamma_h \min \left\{ h(x), \min_{a \in \mathcal{A}} V_h(x') \right\}, \quad (17)$$

$$[T_h(V_h)](x) = (1 - \gamma_h)h(x) + \gamma_h \min \left\{ h(x), \max_{u \in \mathcal{U}} \min_{a \in \mathcal{A}} V_h(x') \right\}. \quad (18)$$

Note that a specified V_h can be viewed as a vector in the Euclidean space $\mathbb{R}^{|\mathcal{S}|}$. The outputs of the safety self-consistency operators are also vectors and the expressions

in Definition 4 are element-wise. We show in the following theorem that the three safety self-consistency operators are all monotone contractions, just like the Bellman operator in standard RL. This lays the foundation for applying RL techniques such as policy iteration.

Theorem 2 (monotone contraction): Let T denote any of $T_h^{\pi, \mu}, T_h^\pi, T_h$.

1) Given any $V_h, \tilde{V}_h \in \mathbb{R}^{|\mathcal{S}|}$, we have

$$\|T(V_h) - T(\tilde{V}_h)\|_\infty \leq \gamma_h \|V_h - \tilde{V}_h\|_\infty. \quad (19)$$

2) Suppose $V_h(x) \geq \tilde{V}_h(x)$ holds for any $x \in \mathcal{S}$. Then we have

$$[T(V_h)](x) \geq [T(\tilde{V}_h)](x), \quad \forall x \in \mathcal{S}. \quad (20)$$

The fixed points of the proposed self-consistency operators serve as approximations for their original values in Definition 2. The following proposition shows that these fixed points converges to the optimal ones as γ_h goes to 1.

Proposition 1: Let T denote any of $T_h^{\pi, \mu}, T_h^\pi, T_h$, and V_h^d denote the fixed point of operator T , i.e., $T(V_h^d) = V_h^d$. Let V_h denote the corresponding original safety value function in Definition 2. Then we have $\lim_{\gamma_h \rightarrow 1} V_h^d(x) = V_h(x)$.

From now on, we assume that a γ_h sufficiently close to 1 is chosen. Since the self-consistency operators are monotone contractions, we can utilize policy iteration techniques in RL to compute their fixed points, i.e., perform policy evaluation and policy improvement alternatively. The pseudo-code is shown in Algorithm 1. Note that the proposed policy iteration is more like the policy iteration in robust RL (see [29], [30]).

Algorithm 1: Policy iteration for robust invariant sets

Input: initial policy π_0 .

for each iteration k do

(policy evaluation)

Solve for $V_h^{\pi_k}$ such that $V_h^{\pi_k} = T_h^{\pi_k}(V_h^{\pi_k})$.

(policy improvement)

$\pi_{k+1} = \arg \max_{\pi} \{T_h^\pi(V_h^{\pi_k})\}$.

end

The following theorem shows that the proposed policy iteration scheme converges monotonically to the optimal safety value (thus identifying the maximal robust invariant set).

Theorem 3 (monotone convergence of policy iteration):

The sequence $\{V_h^{\pi_k}\}$ generated by Algorithm 1 converges monotonically to the fixed point V_h^* of T_h , i.e., $T_h(V_h^*) = V_h^*$.

Algorithm 1 is developed for the tabular case, i.e., state and action spaces with finite elements. In continuous state and action spaces, we can develop a deep RL version for the proposed policy iteration scheme, in which the policies and safety values are approximated with neural networks. From now on, we denote the protagonist policy maintaining safety as π_h , to distinguish it from the policy π that tackles both safety and optimality in Section IV. We utilize the standard actor-critic framework with state-action value functions. We have the state-action safety value function

$Q_h(x, u, a)$, which satisfies $Q_h(x, \pi_h(x), \mu(x)) = V_h(x)$. The self-consistency conditions and operators also apply to Q_h , just like the common RL cases. We denote the parameterized safety value function as $Q_h(x, u, a; \psi)$, the parameterized protagonist policy as $\pi_h(x; \phi)$ and the parameterized adversary policy as $\mu(x; \beta)$. For a set \mathcal{D} of collected samples, the loss function for $Q_h(x, u, a; \psi)$ is

$$L_{Q_h}(\psi) = \mathbb{E}_{(x, u, a, h, x') \sim \mathcal{D}} \left\{ (Q_h(x, u, a; \psi) - \hat{Q}_h)^2 \right\}, \quad (21)$$

where

$$\begin{aligned} \hat{Q}_h &= (1 - \gamma_h)h(x) \\ &+ \gamma_h \min \{h(x'), Q_h(x', \pi_h(x'; \phi), \mu(x'; \beta); \psi)\}. \end{aligned} \quad (22)$$

The loss function for $\pi_h(x; \phi)$ is

$$L_{\pi_h}(\phi) = -\mathbb{E}_{x \sim \mathcal{D}} \{Q_h(x, \pi_h(x; \phi), \mu(x; \beta); \psi)\}. \quad (23)$$

The loss function for $\mu(x; \beta)$ is

$$L_\mu(\beta) = \mathbb{E}_{x \sim \mathcal{D}} \{Q_h(x, \pi_h(x; \phi), \mu(x; \beta); \psi)\}. \quad (24)$$

Algorithm 2 provides the pseudo-code of the actor-critic algorithm for robust invariant sets.

Algorithm 2: Actor-critic algorithm for robust invariant sets

Input: network parameters ψ, ϕ and β , target network parameter $\tilde{\psi} \leftarrow \phi$, learning rate η , target smoothing coefficient τ , replay buffer $\mathcal{D} \leftarrow \emptyset$.

for each iteration do

for each system step do

Sample control input $u_t \sim \pi_h(x_t; \phi)$;

Sample disturbance $a_t \sim \mu(x_t; \beta)$;

Observe next state x_{t+1} , constraint value h_t ;

Store transition $\mathcal{D} \leftarrow \mathcal{D} \cup \{(x_t, u_t, a_t, h_t, x_{t+1})\}$.

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 protagonist policy

$\phi \leftarrow \phi - \eta \nabla_\phi L_{\pi_h}(\phi)$;

Update adversary policy $\beta \leftarrow \beta - \eta \nabla_\beta L_\mu(\beta)$;

Update target network $\tilde{\psi} \leftarrow \tau \psi + (1 - \tau) \tilde{\psi}$.

end

end

IV. JOINT SYNTHESIS OF OPTIMAL SAFE POLICY AND MAXIMAL ROBUST INVARIANT SET

The maximal robust invariant set can be solved with the proposed policy iteration scheme (Algorithm 1 and 2). However, the policy we obtain always seeks the highest constraint values, i.e., stays away from the safety boundary as far as possible. In the safe RL scenario, the goal is to find a policy that maximizes rewards and maintains safety at the same time. So how do we pursue reward optimality

in the condition of safety is guaranteed? The key insight is that to maintain safety, we do not need to choose inputs such that $u = \operatorname{argmax}_u Q_h(x, u, a)$. Instead, we only need to choose inputs u satisfying $Q_h(x, u, a) \geq 0$, i.e., the state-action safety value function identifies the admissible inputs range at each state for maintaining safety (staying inside the robust invariant set). Motivated by this fact, we propose a constrained RL algorithm that simultaneously synthesizes the robust invariant set and uses it to constrain the optimal policy. This algorithm converges to both the maximal robust invariant set and the optimal safe policy, i.e., one policy that tackles both safety and optimality. Note that the optimality here is with respect to reward maximization. The policy iteration scheme in Section III converges to the optimal protagonist policy and the maximal robust invariant set, in which the optimality is with respect to constraint value maximization.

We combine the proposed policy iteration scheme for robust invariant set with soft-actor critic (SAC) [31], a well-known model-free off-policy RL algorithm. SAC concurrently learns a policy and two state-action value functions, which we denote as $\pi(x; \theta)$, $Q(x, u, a; \omega_1)$ and $Q(x, u, a; \omega_2)$. Note that $\pi(x; \theta)$ is different from $\pi_h(x; \phi)$. The loss functions for $Q_h(x, u, a; \psi)$, $\pi_h(x; \phi)$, and $\mu(x; \beta)$ are the same as in Section III. The loss function for Q -functions is

$$L_Q(\omega_i) = \mathbb{E}_{(x, u, a, r, x') \sim \mathcal{D}} \left\{ (Q(x, u, a; \omega_i) - \hat{Q})^2 \right\}, (i = 1, 2) \quad (25)$$

where

$$\begin{aligned} \hat{Q} &= r(x, u, a) \\ &+ \gamma \left(\min_{j=1,2} Q(x', u', \mu(x'; \beta); \hat{\omega}_j) - \alpha \log \pi(u'|x'; \theta) \right), \end{aligned} \quad (26)$$

in which $u' \sim \pi(\cdot|x'; \theta)$, $\hat{\omega}_j$ denotes the corresponding target network and α denotes the temperature. The loss function of the temperature α is

$$L(\alpha) = \mathbb{E}_{x \sim \mathcal{D}} \{ -\alpha \log \pi(u|x; \theta) - \alpha \mathcal{H} \}, \quad (27)$$

where $u \sim \pi(\cdot|x; \theta)$ and \mathcal{H} is the target entropy. The loss function of the policy is

$$L_\pi(\theta) = \mathbb{E}_{x \sim \mathcal{D}} \left\{ \alpha \log \pi(u|x; \theta) - \min_{j=1,2} Q(x, u, \mu(x; \beta); \omega_j) \right\}, \quad (28)$$

where $u \sim \pi(\cdot|x; \theta)$. To ensure safety under worst-case disturbances, we must constrain the policy π such that its outputs satisfy the requirement of state-action safety value function:

$$Q_h(x, u, \mu(x; \beta); \psi) \geq 0, \quad u \sim \pi(\cdot|x; \theta). \quad (29)$$

We utilize the Lagrange multiplier method to solve this constrained policy optimization problem. The Lagrangian is formulated as

$$\mathcal{L}(\theta, \lambda) = L_\pi(\theta) - \lambda \mathbb{E}_{x \in \mathcal{D}} \{ Q_h(x, u, \mu(x; \beta); \psi) \}, \quad (30)$$

in which $u \sim \pi(\cdot|x; \theta)$. We update policy $\pi(x; \theta)$ and multiplier λ using dual ascent for the dual problem:

$$\min_{\theta} \max_{\lambda \geq 0} \mathcal{L}(\theta, \lambda). \quad (31)$$

The pseudo-code of the overall algorithm is summarized in Algorithm 3.

Algorithm 3: Soft actor-critic with robust invariant set (SAC-RIS)

Input: network parameters $\psi, \phi, \beta, \omega_1, \omega_2, \theta$, target network parameters $\bar{\psi} \leftarrow \psi, \bar{\omega}_1 \leftarrow \omega_1, \bar{\omega}_2 \leftarrow \omega_2$, Lagrange multiplier λ , temperature α , learning rate η , target smoothing coefficient τ , 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 $a_t \sim \mu(x_t; \beta)$;
 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 policy $\theta \leftarrow \theta - \eta \nabla_{\theta} \mathcal{L}(\theta, \lambda)$;
 Update protagonist policy
 $\phi \leftarrow \phi - \eta \nabla_{\phi} L_{\pi_h}(\phi)$;
 Update adversary policy $\beta \leftarrow \beta - \eta \nabla_{\beta} L_{\mu}(\beta)$;
 Update temperature $\alpha \leftarrow \alpha - \eta \nabla_{\alpha} L(\alpha)$;
 Update target networks $\bar{\psi} \leftarrow \tau \psi + (1 - \tau) \bar{\psi}$,
 $\bar{\omega}_i \leftarrow \tau \omega_i + (1 - \tau) \bar{\omega}_i$ for $i \in \{1, 2\}$;
 Update multiplier $\lambda \leftarrow \lambda + \eta \nabla_{\lambda} \mathcal{L}(\theta, \lambda)$.

end

end

V. EXPERIMENTS

A. Learning the maximal robust invariant set

We test the proposed policy iteration scheme on the double integrator with state constraints, whose invariant sets have analytic expressions, to examine whether the learned robust invariant set matches the ground truth. The double integrator has the following dynamics:

$$\begin{bmatrix} x_{t+1} \\ v_{t+1} \end{bmatrix} = \begin{bmatrix} 1 & 0.005 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_t \\ v_t \end{bmatrix} + \begin{bmatrix} 0 \\ 0.005 \end{bmatrix} (u_t + a_t), \quad (32)$$

where $u_t \in [-1, 1]$ and $a_t \in [-0.5, 0.5]$. The safety constraints are $|x| \leq 2$ and $|v| \leq 2$. The constraint function is $h(x, v) = \min\{x + 2, 2 - x, v + 2, 2 - v\}$. The policies and safety value function are parameterized with multilayer perceptrons.

We solve the safety value functions for two cases: with or without disturbances. The results are shown in Figure 2. The invariant sets are zero-superlevel sets of the safety value functions. We can see that the learned invariant sets (denoted by pink lines) match well with the true invariant sets (denoted

by black lines), which validates the effectiveness of our algorithm. Besides, we can see that the robust invariant set is smaller than the standard invariant set due to the presence of disturbances. To ensure safety under external disturbances, it is crucial to solve for the robust invariant set and constrain the system state to stay inside it.

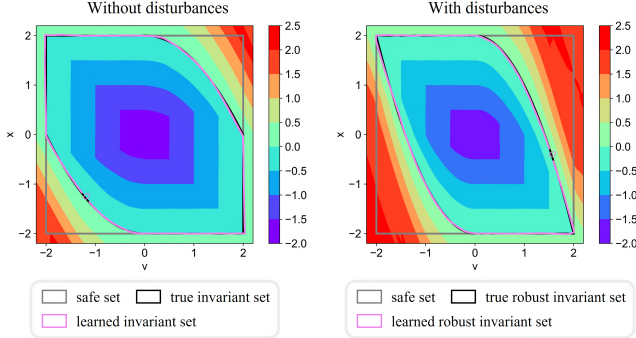


Fig. 2. Heat maps and zero contours of learned safety value functions. The grey lines are boundaries of safe sets. The pink lines are boundaries of learned invariant sets. The black lines are boundaries of true (maximal) invariant sets for the double integrator system.

B. Learning the optimal safe policy under worst-case disturbances

We test our algorithm SAC-RIS and other baselines on two safety-critical control tasks.

Cart-pole is a task based on MuJoCo [32], as shown in Figure 3(a). The objective is to move the cart to a target position as quickly as possible. The system state includes position x and velocity v of the cart, angle θ and angular velocity ω of the pole. The control inputs $u \in [-1, 1] \subset \mathbb{R}$ and external disturbances $a \in [-0.5, 0.5] \subset \mathbb{R}$ are horizontal forces applied on the cart. The safety constraints are $|\theta| \leq 0.2$, i.e., keeping the pole nearly upright. The constraint function is $h(\theta) = \min\{\theta + 0.2, 0.2 - \theta\}$.

Quadrotor is a 2D quadrotor trajectory tracking task in safe-control-gym [33], as shown in Figure 3(b). The goal is to track the circle trajectory as accurately as possible. The system state includes the horizontal position x , vertical position z , the pitch angle θ and their time derivatives. The control inputs $u \in [-1, 1]^2 \subset \mathbb{R}^2$ and external disturbances $a \in [-0.5, 0.5]^2 \subset \mathbb{R}^2$ are motor thrusts applied on the quadrotor. The safety constraints are $z - 0.5 > 0$ and $1.5 - z > 0$, i.e., maintaining its vertical position z between $[-0.5, 1.5]$. The constraint function is $h(z) = \min\{z - 0.5, 1.5 - z\}$.

We adopt two baseline safe RL algorithms for comparison. SAC Lagrangian (SAC-L) follows the CMDP formulation and takes a weighted sum of the value and the cost value as the objective of policy optimization [34]. Reachable Actor-Critic (RAC) utilizes Hamilton-Jacobi reachability analysis to characterize the invariant set and uses it for constrained policy optimization [22]. The neural networks are multilayer perceptrons. The shared hyperparameters of all three algorithms are the same. We use two metrics for evaluation: episode return and episode constraint violation, reflecting the optimality and safety of the policy, respectively.

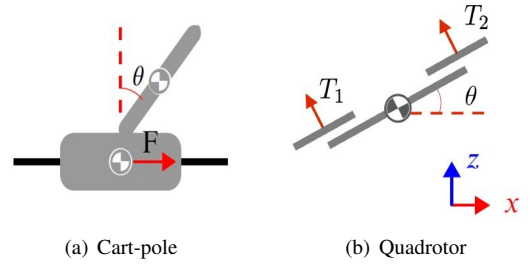


Fig. 3. Illustration of two safety-critical control tasks, taken from [33].

We evaluate all algorithms under two scenarios. In the first scenario, the policies are tested with the learned adversarial policy of our algorithm SAC-RIS, which represents the worst-case disturbances. The training curves are shown in Figure 4. As demonstrated by the second row, SAC-L and RAC violate the safety constraint substantially under worst external disturbances. Their episode returns are also affected, especially in Cart-pole, since the consequence of violating the constraints is catastrophic, i.e., the pole falls down. Our algorithm SAC-RIS learns policies that achieve both high performance and zero violation of safety constraints. This justifies the design of our algorithm and the effectiveness of synthesizing robust invariant sets.

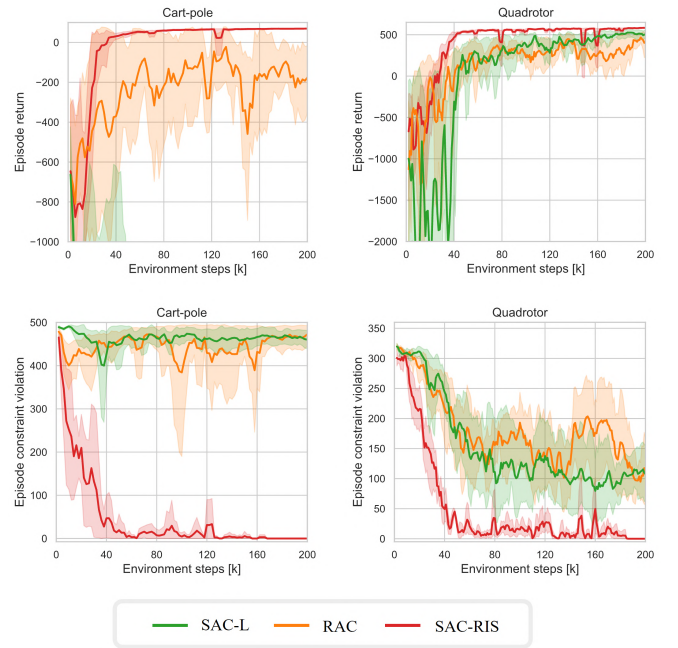


Fig. 4. Training curves on two safety-critical control tasks. The policies are tested with the learned adversarial policy from our algorithm SAC-RIS. The first row corresponds to episode return and the second row corresponds to episode constraint violation. The solid lines correspond to the mean and the shaded regions correspond to 95% confidence interval over five seeds.

In the second scenario, the policies are tested without any disturbances. The training curves are shown in Figure 5. Although the policies of SAC-RIS are trained under the presence of adversarial disturbances, they generalize well to the no-disturbance case. The policies of SAC-RIS achieve

comparable performance as the baselines, as well as zero constraint violation.

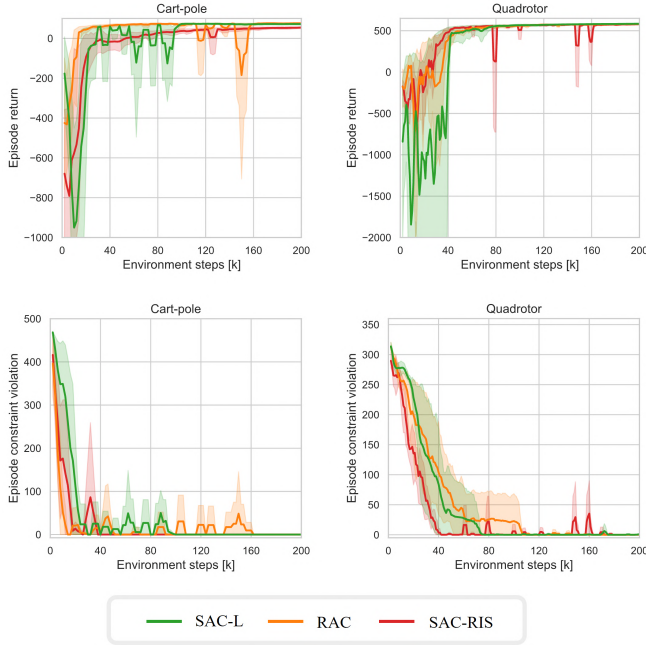


Fig. 5. Training curves on two safety-critical control tasks. The policies are tested without disturbances.

Remark 2 (limitation): As shown in Figure 4 and 5, the constraint violations in training are considerable. Therefore, our proposed algorithm is not suitable for direct online deployment. The learning process needs to be carried out in simulators. Once the training is completed, the policy is ready to be implemented online.

VI. CONCLUSION

In this paper, a safe reinforcement learning framework has been proposed to learn an optimal safe policy under worst-case disturbances. We have proposed a policy iteration scheme that converges monotonically to the maximal robust invariant set. We have designed a constrained reinforcement learning algorithm that simultaneously synthesizes the robust invariant set and uses it for constrained policy optimization. Experiments on classic control tasks have validated the effectiveness of the proposed scheme.

ACKNOWLEDGMENTS

The authors would like to thank Dongjie Yu, Yujie Yang and Runzhong Li for their valuable suggestions on the code implementation in this work.

APPENDIX

A. Proof of Theorem 1

We only prove (13), while the proof for (14) and (15) is similar. From the definition of the safety value function, we

have

$$\begin{aligned}
 V_h^{\pi, \mu}(x) &= \min_{t \geq 0} \{h(x_t^{\pi, \mu} \mid x_0 = x)\} \\
 &= \min \left\{ h(x), \min_{t \geq 1} \{h(x_t^{\pi, \mu} \mid x_0 = x)\} \right\} \\
 &= \min \left\{ h(x), \min_{t \geq 1} \{h(x_t^{\pi, \mu} \mid x_1 = x')\} \right\} \\
 &= \min \left\{ h(x), \min_{t \geq 0} \{h(x_t^{\pi, \mu} \mid x_0 = x')\} \right\} \\
 &= \min \{h(x), V_h^{\pi, \mu}(x')\}.
 \end{aligned} \tag{33}$$

B. Proof of Theorem 2

We only prove the monotone contraction for T_h , while the proof for T_h^π and $T_h^{\pi, \mu}$ is similar. We have

$$\begin{aligned}
 &[T_h(V_h)](x) - [T_h(\tilde{V}_h)](x) \\
 &= \gamma_h \min \left\{ h(x), \max_{u \in \mathcal{U}} \min_{a \in \mathcal{A}} V_h(x') \right\} \\
 &\quad - \gamma_h \min \left\{ h(x), \max_{u \in \mathcal{U}} \min_{a \in \mathcal{A}} \tilde{V}_h(x') \right\}.
 \end{aligned} \tag{34}$$

We have

$$\begin{aligned}
 &\|T_h(V_h) - T_h(\tilde{V}_h)\|_\infty \\
 &\leq \gamma_h \max_{u \in \mathcal{U}} \max_{a \in \mathcal{A}} |V_h(f(x, u, a)) - \tilde{V}_h(f(x, u, a))| \\
 &\leq \gamma_h \|V_h - \tilde{V}_h\|_\infty.
 \end{aligned} \tag{35}$$

The first inequality in (35) follows from the relationship as

$$\begin{aligned}
 &\left| \max_x \min_y f(x, y) - \max_x \min_y g(x, y) \right| \\
 &\leq \max_x \max_y |f(x, y) - g(x, y)|.
 \end{aligned} \tag{36}$$

The monotonicity of T_h follows from the monotonicity of the max and min operations contained in T_h .

C. Proof of Proposition 1

The proof is similar to that of the Proposition 1 in [27].

D. Proof of Theorem 3

We will prove the following relationship:

$$V_h^{\pi_k} \leq T_h(V_h^{\pi_k}) \leq V_h^{\pi_{k+1}} \leq V_h^* \tag{37}$$

First, using the definition of T_h and policy improvement, we have

$$T_h(V_h^{\pi_k}) = T_h^{\pi_{k+1}}(V_h^{\pi_k}) \geq T_h^{\pi_k}(V_h^{\pi_k}) = V_h^{\pi_k}. \tag{38}$$

Since $V_h^{\pi_k} \leq T_h^{\pi_{k+1}}(V_h^{\pi_k})$, using the monotonicity and contraction of $T_h^{\pi_{k+1}}$, we have

$$V_h^{\pi_k} \leq T_h(V_h^{\pi_k}) = T_h^{\pi_{k+1}}(V_h^{\pi_k}) \leq (T_h^{\pi_{k+1}})^\infty(V_h^{\pi_k}) = V_h^{\pi_{k+1}}. \tag{39}$$

Since $T_h(V_h^{\pi_{k+1}}) \geq T_h^{\pi_{k+1}}(V_h^{\pi_{k+1}}) = V_h^{\pi_{k+1}}$, using the monotonicity and contraction of T_h , we have

$$V_h^* = (T_h)^\infty(V_h^{\pi_{k+1}}) \geq \dots \geq V_h^{\pi_{k+1}}. \tag{40}$$

Therefore, (37) holds. The sequence $\{V_h^{\pi_k}\}$ is monotone and bounded, so it converges. After convergence, we have $V_h^{\pi_k} = T_h(V_h^{\pi_k}) = V_h^{\pi_{k+1}}$, which indicates that the convergence point is the fixed point of T_h .

REFERENCES

- [1] S. E. Li, *Reinforcement Learning for Sequential Decision and Optimal Control*. Springer, 2023.
- [2] 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.
- [3] O. Kroemer, S. Niekum, and G. Konidaris, “A review of robot learning for manipulation: Challenges, representations, and algorithms,” *The Journal of Machine Learning Research*, vol. 22, no. 1, pp. 1395–1476, 2021.
- [4] B. R. Kiran, I. Sobh, V. Talpaert, P. Mannion, A. A. Al Sallab, S. Yogamani, and P. Pérez, “Deep reinforcement learning for autonomous driving: A survey,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 6, pp. 4909–4926, 2021.
- [5] 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.
- [6] A. D. Ames, S. Coogan, M. Egerstedt, G. Notomista, K. Sreenath, and P. Tabuada, “Control barrier functions: Theory and applications,” in *2019 18th European Control Conference (ECC)*. IEEE, 2019, pp. 3420–3431.
- [7] 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.
- [8] L. Brunke, M. Greeff, A. W. Hall, Z. Yuan, S. Zhou, J. Panerati, and A. P. Schoellig, “Safe learning in robotics: From learning-based control to safe reinforcement learning,” *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 5, pp. 411–444, 2022.
- [9] E. Altman, *Constrained markov decision processes: stochastic modeling*. Routledge, 1999.
- [10] 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.
- [11] A. Stooke, J. Achiam, and P. Abbeel, “Responsive safety in reinforcement learning by pid lagrangian methods,” in *International Conference on Machine Learning*. PMLR, 2020, pp. 9133–9143.
- [12] C. Tessler, D. J. Mankowitz, and S. Mannor, “Reward constrained policy optimization,” in *International Conference on Learning Representations*, 2019.
- [13] 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, 2022.
- [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] W. Zhao, T. He, R. Chen, T. Wei, and C. Liu, “State-wise safe reinforcement learning: A survey,” *arXiv preprint arXiv:2302.03122*, 2023.
- [16] F. Blanchini, “Set invariance in control,” *Automatica*, vol. 35, no. 11, pp. 1747–1767, 1999.
- [17] F. Blanchini and S. Miani, *Set-theoretic methods in control*. Springer, 2008, vol. 78.
- [18] 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.
- [19] C. Liu and M. Tomizuka, “Control in a safe set: Addressing safety in human-robot interactions,” in *Dynamic Systems and Control Conference*, vol. 46209. American Society of Mechanical Engineers, 2014, p. V003T42A003.
- [20] 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.
- [21] 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.
- [22] D. Yu, H. Ma, S. Li, and J. Chen, “Reachability constrained reinforcement learning,” in *International Conference on Machine Learning*. PMLR, 2022, pp. 25 636–25 655.
- [23] W. Zhao, J. P. Queralta, and T. Westerlund, “Sim-to-real transfer in deep reinforcement learning for robotics: a survey,” in *2020 IEEE Symposium Series on Computational Intelligence (SSCI)*. IEEE, 2020, pp. 737–744.
- [24] T. Başar and P. Bernhard, *H-infinity optimal control and related minimax design problems: a dynamic game approach*. Springer Science & Business Media, 2008.
- [25] E. C. Kerrigan, “Robust constraint satisfaction: Invariant sets and predictive control,” Ph.D. dissertation, University of Cambridge UK, 2001.
- [26] S. Bansal, M. Chen, S. Herbert, and C. J. Tomlin, “Hamilton-jacobi reachability: A brief overview and recent advances,” in *2017 IEEE 56th Annual Conference on Decision and Control (CDC)*. IEEE, 2017, pp. 2242–2253.
- [27] 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.
- [28] B. Xue and N. Zhan, “Robust invariant sets computation for discrete-time perturbed nonlinear systems,” *IEEE Transactions on Automatic Control*, vol. 67, no. 2, pp. 1053–1060, 2021.
- [29] S. D. Patek, “Stochastic and shortest path games: theory and algorithms,” Ph.D. dissertation, Massachusetts Institute of Technology, 1997.
- [30] J. Perolat, B. Scherrer, B. Piot, and O. Pietquin, “Approximate dynamic programming for two-player zero-sum markov games,” in *International Conference on Machine Learning*. PMLR, 2015, pp. 1321–1329.
- [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] 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.
- [33] Z. Yuan, A. W. Hall, S. Zhou, L. Brunke, M. Greeff, J. Panerati, and A. P. Schoellig, “safe-control-gym: A unified benchmark suite for safe learning-based control and reinforcement learning in robotics,” *IEEE Robotics and Automation Letters*, vol. 7, no. 4, pp. 11 142–11 149, 2022.
- [34] 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.