Trustworthy Reinforcement Learning Against Intrinsic Vulnerabilities: Robustness, Safety, and Generalizability

MENGDI XU*, ZUXIN LIU*, and PEIDE HUANG*, Carnegie Mellon University, USA

WENHAO DING, Carnegie Mellon University, USA

ZHEPENG CEN, Carnegie Mellon University, USA

BO LI, University of Illinois Urbana-Champaign, USA

DING ZHAO, Carnegie Mellon University, USA

A trustworthy reinforcement learning algorithm should be competent in solving challenging real-world problems, including robustly handling uncertainties, satisfying safety constraints to avoid catastrophic failures, and generalizing to unseen scenarios during deployments. This study aims to overview these main perspectives of trustworthy reinforcement learning considering its intrinsic vulnerabilities on robustness, safety, and generalizability. In particular, we give rigorous formulations, categorize corresponding methodologies, and discuss benchmarks for each perspective. Moreover, we provide an outlook section to spur promising future directions with a brief discussion on extrinsic vulnerabilities considering human feedback. We hope this survey could bring together separate threads of studies together in a unified framework and promote the trustworthiness of reinforcement learning.

CCS Concepts: • Computing methodologies \rightarrow Reinforcement learning; Markov decision processes; • Security and privacy \rightarrow Social aspects of security and privacy; • Computer systems organization \rightarrow Robotics; • Hardware \rightarrow Safety critical systems.

Additional Key Words and Phrases: Trustworthy Reinforcement Learning, Robust Reinforcement Learning, Safe Reinforcement Learning, Generalizable Reinforcement Learning

ACM Reference Format:

1 Introduction

With its vast potential to tackle some of the world's most pressing problems, reinforcement learning (RL) [173] is applied to transportation [68], manufacturing [133], security[75], healthcare [211], and world hunger [44]. As RL has started to shift towards deployment on real-world problems, its rapid development is coupled with as much risk

*Authors contributed equally to this research.

Authors' addresses: Mengdi Xu, mengdixu@andrew.cmu.edu; Zuxin Liu, zuxinl@andrew.cmu.edu; Peide Huang, peideh@andrew.cmu.edu, Carnegie Mellon University, Pittsburgh, Pennsylvania, USA, 15213; Wenhao Ding, wenhaod@andrew.cmu.edu, Carnegie Mellon University, Pittsburgh, Pennsylvania, USA, 15213; Zhepeng Cen, zcen@andrew.cmu.edu, Carnegie Mellon University, Pittsburgh, Pennsylvania, USA, 15213; Bo Li, University of Illinois Urbana-Champaign, Champaign, Illinois, USA, lbo@illinois.edu; Ding Zhao, Carnegie Mellon University, Pittsburgh, Pennsylvania, USA, 15213, dingzhao@cmu.edu.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

©~2022 Association for Computing Machinery.

Manuscript submitted to ACM

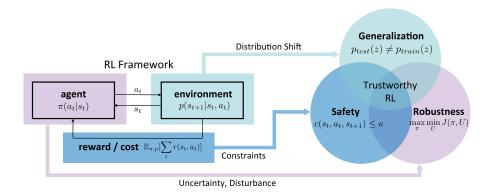


Fig. 1. Overview diagram for trustworthy RL against intrinsic vulnerabilities: robustness, safety, generalization

as reward [3, 42, 161]. Before consumers embrace RL-empowered services, researchers are tasked with proving the trustworthiness of their innovations.

Trustworthiness is conceived to maximize the benefits of AI systems while at the same time minimizing their risks [3]. It has rich meanings that go beyond its literal sense, and motivates a comprehensive framework that includes multiple principles, requirements, and criteria [3]. Recently, there has been exciting progress in the area of *trustworthy RL* [2, 5, 48, 107, 108, 121, 129, 137, 140, 145, 148, 165, 171, 201], which greatly help to advance our understanding of intrinsic vulnerabilities in RL and potential solutions in particular aspects of trustworthy RL. It is clear that the next leap toward trustworthy RL will require a holistic and fundamental understanding of the challenges of these problems, the weakness, and advantages of existing trustworthy RL approaches, and a paradigm shift of trustworthy RL based on existing work. Compared with the trustworthiness issues in traditional machine learning (ML), the problems in RL are orders of magnitude more complicated, given that RL is a multi-faceted system containing several Markov Decision Processes (MDP) components (observations, environmental dynamics, actions, and rewards) [173]. Notably, these components may be subject to diverse robustness, safety, generalization, and security concerns, which have or have not been considered in traditional ML. The concerns in the more static parts (*e.g.*, observations) can find their root in ML studies, while those more relevant to the interaction property of the system (*e.g.*, actions) are unique to RL and have been less explored. In addition, when viewing RL as two stages (model training followed by model deployment), we can study the vulnerabilities of the two stages alone and the connections between them.

To promote the advancement of the field, in this paper, we provide a holistic survey of trustworthy RL from three aspects regarding the four MDP elements and two stages including

- robustness against perturbations and uncertainties,
- safety of RL to constrain devastating costs, and

These requirements are usually simultaneously required for an intelligent agent with strong interconnections. Taking autonomous vehicles (AVs) as an example, we concretely explain the three aspects of trustworthiness. First, the observations of the AVs are susceptible to the adversaries perturbing the camera or LiDAR input, the environmental dynamics and rewards may be poisoned when training the AVs, and the actuators can also be manipulated by the adversaries when executing a deployed benign policy. Regarding the two stages and their connections, during training, safe exploration is expected so that the car would not crash; during deployment, safety constraints are also expected to Manuscript submitted to ACM

be adhered to avoid dangerous outcomes. In addition, good generalization is required from the training environment of the AVs to the testing environment. An AV should generalize between cities and different weathers and seasons.

Despite these interconnecting aspects of trustworthiness, existing surveys mainly focus on limited parts. García et al. [52], Gu et al. [62], and Brunke et al. [25] work on safe RL, while Moos et al. [121] and Kirk et al. [88] focus on robustness and generalization, respectively. They advance the field by providing concrete descriptions for one aspect of trustworthy RL, yet lack a comprehensive characterization of the intrinsic vulnerabilities of RL as we envisioned above. In this survey, we would like to provide a unified framework for the three aspects of trustworthy RL. For each aspect, we will 1) clarify the terminologies, 2) analyze their intrinsic vulnerabilities, 3) introduce methods that address these vulnerabilities, and 4) summarize popular benchmarks. As shown in Fig. 1, robustness, safety, and generalizability have strong correlations to agent, environment, reward, and cost, which correspond to the components of the MDP. To be within a scope that is easy for readers, , we constrain our survey in the MDP setting. We view our survey as the intrinsic trustworthy aspects of RL as we assume the human preference and setting of the agent and environment are pre-defined. In the outlook section, we will discuss the links between intrinsic and extrinsic trustworthiness. Particularly, we will provide an outlook on four fundamental questions.

- How to certify and evaluate for trustworthy reinforcement learning?
- What is the relation between the different aspects of trustworthiness?
- How to co-design trustworthy RL with the physical agent and environment?
- How to achieve the human-centric design for trustworthy RL?

We hope this survey will bring together disparate threads of studies together in a unified framework and spur new research for a holistic view of the intrinsic aspects of trustworthy RL.

The remainder of the survey is organized as follows. From Section 2 to Section 4, each section addresses one aspect of trustworthiness. We will explain our outlook on the three aspects of the key challenges in future directions beyond the survey in Section 5. We will conclude the survey with 10 takeaways in Section 6.

2 Robust RL against perturbation

2.1 Overview

Robust RL aims to improve the worst-case performance of algorithms deterministically or statistically in the face of uncertainties and adversarial attacks. The discrepancies between training tasks and testing tasks widely exist. For instance, in continuous control tasks, the real physical parameters may be different from those in simulation, and in autonomous driving scenarios, the surrounding agents may have novel driving behaviors. Such discrepancies motivate the development of robust RL. Furthermore, the safety-critical nature of real-world applications makes robustness an even vital feature to help avoid catastrophic failures. There is a surge of interest in researching effective attacks and defenses in RL. Detailed reviews of both fields can be found in [28, 76, 121].

In this section, we first summarize robust RL formulations in Sec. 2.2, in terms of robustness against uncertainties in different MDP components, including observations/states, actions, transitions, and rewards. We then present robust training and testing methods in Sec. 2.3 to improve the robustness against uncertainties or artificially designed attacks on each component. We summarize the robust RL methodologies in Fig. 2. Finally, we present the applications and benchmarks for testing the robustness of proposed robust RL algorithms in Sec 2.4.

2.2 Problem formulation of robust RL

We will build robust RL on the MDP setting as shown in Fig. 1. A MDP is represented by a tuple $M = (S, \mathcal{A}, R, P, \rho_0, \gamma)$. Here, S is the state space, \mathcal{A} is the action space, $R: S \times \mathcal{A} \mapsto \mathbb{R}$ is the reward function, $P: S \times \mathcal{A} \mapsto \Delta(S)$ is the transition function with $\Delta(\cdot)$ defining the probability simplex, $\rho_0 \in \Delta(S)$ is the distribution over the initial state, and $\gamma \in (0, 1]$ is the discount factor. At time step t, the agent observes its state $s_t \in S$, takes action $a_t \in \mathcal{A}$, transitions to the next state $s_{t+1} \sim P(s_t, a_t)$, and receives reward $r_t = R(s_t, a_t)$. The goal is to learn a policy $\pi: S \to \Delta(\mathcal{A})$ that maximizes the expected cumulative reward $\mathbb{E}[\sum_t \gamma^t r_t]$ with $s_0 \sim \rho_0$ and $a_t \sim \pi(s_t)$.

Moreover, in robust RL settings, there exists an uncertain variable U that lies within an uncertainty set $U \in \mathcal{F}_U$. The uncertain variable could be the state $s \in \mathcal{S}$, action $a \in \mathcal{A}$, reward function R or transition P. Formally, $U \in \{s, a, R, P\}$. Robust RL algorithms aim to find a policy that maximizes the worst-possible performance against the uncertain variable U. Formally, they solve the following maximin optimization problem:

$$\max_{\pi} \min_{U \in \mathcal{T}_U} J_M(\pi, U), \tag{1}$$

where $J_M(\pi, U)$ is the expected accumulated return as a function of the policy π , the MDP M, and the uncertain variable U. In this section, we discuss naturalistic uncertainties that arise in realistic applications and artificially designed adversarial attacks to different choices of U, including state observations (Sec. 2.2.1), actions (Sec 2.2.2), transitions, and rewards (Sec 2.2.3). A lot of robust RL methods motivated by naturalistic uncertainties adopt the adversarial training framework and thus assume an adversary agent conducting adversarial attacks to mimic the naturalistic uncertainties.

2.2.1 Robust RL against uncertain state observations

A majority of Robust RL methods aim to robustly handle the mismatches between observed states and actual states. The state uncertainty may come from partial state observation, due to sensor errors or limited sensor capacities. In general, the mismatch between the actual state and the state observation decreases the policy's performance or may even cause catastrophic failures in safety-critical situations. Robust RL against state observation uncertainties aims to robustly handle such state observation mismatch. Let the observed states be $\tilde{s} \in \mathcal{S}$ which serves as the input for action generations $a \sim \pi(\tilde{s})$. Let the actual states be $s \in \mathcal{S}$ for environment rollouts $s' \sim P(s, a)$. Robust RL aims to find the optimal policy π that achieves the best performance under the worst-possible state observations [221].

$$\max_{\pi} \min_{\tilde{s}} \mathbb{E}_{a_t \sim \pi(\tilde{s_t}), P, r} \left[\sum_{t=1}^{T} \gamma^{t-1} r(s_t, a_t) \right]. \tag{2}$$

The attacker adds perturbations to the state or observations to achieve certain *adversarial goals*. Let the state observation be $\tilde{s} = s + \epsilon$ where ϵ represents the perturbation, the target adversarial state be s^{adv} , and the adversarial policy be π^{adv} ; the normal reward signal be r and the adversarial reward signal be r^{adv} . For *training time attacks*, the attacker may minimize the cumulative reward (*i.e.*, $\min \mathbb{E}\left[\sum \gamma^t r_t\right]$) or maximize an adversarial cumulative reward (*i.e.*, $\max \mathbb{E}\left[\sum \gamma^t r_t^{\text{adv}}\right]$). For *testing time attacks*, the attacker corrupts the agent performance by (1) luring the agent to the target adversarial state s^{adv} , (2) misleading the action selection, i.e., $\pi(\tilde{s}) \neq \pi(s)$ or luring the agent to learn the target adversarial policy π^{adv} . Most works on attacking the observation space of the agent assume the threat model of ℓ_p -bounded attacks, extending the study on adversarial examples in the image domain [56, 174]. This threat model holds in all the following descriptions unless otherwise stated.

Attacks on state observations at *training time* are widely studied in multiple settings. ATLA [220] and PA-AD [172] consider an optimal adversary under the SA-MDP framework [221], which aims to lead to minimal value functions under Manuscript submitted to ACM

bounded state perturbations. To find this optimal adversary (*i.e.*, the optimal adversarial state perturbation), ATLA [220] proposes to train an adversary whose action space is the perturbation set in the state space, while PA-AD [172] further decouples the problem of finding state perturbations into finding policy perturbations plus finding the state that achieves the lowest value policy, thus addressing the challenge of large state space. In the *multi-agent environment* specifically, Gleave et al. [54] consider a different threat model where the adversary can control the agent's opponent. They propose to craft natural adversarial observations via training the opponent's adversarial policy w.r.t. the agent's fixed trained policy, thus achieving the adversarial goal empirically. Different from these works which minimize the cumulative reward in the original task, Tretschk et al. [185] try to maximize an adversarial cumulative reward guided by an adversarial reward signal r^{adv} . Assuming whitebox access to the trained agent, they train an Adversarial Transformer Network (ATN) [15] along with the trained agent on the adversarial task to generate the perturbations.

At testing time, to lure the agent to targeted corrupted action sequences, Lin et al. [101] first generate a preferred action sequence for luring the agent via a planning algorithm, and then craft the adversarial state perturbations that lead to the planned actions. The action sequence is dynamically adjusted after each step to increase the attack's effectiveness. In addition, non-targeted methods are effective to different degrees by applying random noise to simply interfere with the action selection [91], or adopting adversarial attacks (e.g., FGSM [56] and CW attacks [26]) to deliberately alter the probability of action selection (either reduce the probability of selecting the optimal action [17, 72, 91] or increase the probability of selecting the worst action [101, 136]). Behzadan and Munir [16] propose the policy induction attack, where a surrogate Q-network is trained in tandem with the target agent Q-network, and the adversarial state observations are computed w.r.t. the surrogate model, and then leveraged to poison the target network, exploiting the transferability of adversarial examples.

2.2.2 Robust RL against uncertain actions

Robust RL against action uncertainties focuses on the discrepancy between the actions generated by the RL agent and the conducted actions. Such action uncertainty may come from the actuator noise, limited power range, or actuator failures in the real world. Let the uncertain actions be $\tilde{a} = v(\pi(s))$, where v is the action perturbation function. Robust RL against uncertain actions aims to find the optimal policy π that achieves the best performance under the worst-possible actions. More concretely, the optimal policy is a solution to the following optimization problem:

$$\max_{\pi} \min_{\tilde{a}} \mathbb{E}_{\tilde{a}_t \sim \nu(\pi(s_t)), P, r} \left[\sum_{t=1}^{T} \gamma^{t-1} r(s_t, \tilde{a}_t) \right]. \tag{3}$$

In the existing literature, the naturalistic uncertainties in action are represented by action attackers manipulating actions directly to minimize the accumulated return. For instance, PR-MDP and NR-MDP [178] assume that the perturbation function ν is related to an adversary's policy, and characterize the action uncertainty in the presence of action perturbations.

2.2.3 Robust RL against uncertain transitions and rewards

The discrepancies between training and test environments widely exist due to sim-to-real gaps, the non-stationary nature of testing domains, training, and testing task distribution shifts, or partial observable environment transitions and rewards. The environment discrepancy is generally reflected in different transitions and rewards. Thus Robust RL against uncertain transitions and rewards aims to handle such environment mismatches robustly. In this case, the uncertain variable follows U = (P, r) or U = r.

Environment discrepancies. We summarize two mainstream robust formulations related to environment discrepancies: (1) the *robust MDP* formulation that assumes the transition and reward functions are uncertain, and (2) the *distributionally robust MDP* formulation that assumes the latent distribution for generating the transitions and rewards is uncertain. In comparison, the distributionally robust formulation encodes prior distributional information on possible transitions and rewards, and thus may help balance the performance and robustness. It is known that the distributionally robust MDP formulation help generate less conservative policies [166, 201] by balancing the nominal performances and the worst-case performances. The trade-offs between optimizing over nominal and worst-case distribution can be found in [200].

<u>Robust MDP.</u> Robust RL methods and robust MDP formulations assume no distributional information on (P, r). Robust MDP formulations directly put set constraints over environment transition probability $\tilde{p}: \mathcal{S} \times \mathcal{A} \to \mathcal{S} \subseteq \mathbb{R}^{d_{\mathcal{S}}}$ and reward distribution $\tilde{r}: \mathcal{S} \times \mathcal{A} \to \mathcal{R} \subseteq \mathbb{R}$ by assuming (\tilde{p}, \tilde{r}) takes values in the uncertainty set \mathcal{F} . It aims to derive a policy π by solving the following max-min problem

$$\max_{\pi} \min_{(\tilde{P}, \tilde{r}) \in \mathcal{F}} \mathbb{E}_{\pi, (\tilde{P}, \tilde{r})} \left[\sum_{t=1}^{T} \gamma^{t-1} \tilde{r}(s_t, a_t) \right]. \tag{4}$$

T is the planning horizon. \mathcal{F} here is a general ambiguity set without special structures.

<u>Distributionally robust MDP.</u> In contrast, distributionally Robust MDPs (DR-MDP) assume additional distributional information on (P, r). DR-MDP puts set constraints on μ , where $(P, r) \sim \mu$. Formally, the transition probability \tilde{p} and the reward \tilde{r} are unknown parameters and obey a joint probability distribution μ within an ambiguity set \mathcal{F} . Similar to the robust MDP settings, let T be the planning horizon. More concretely, the DR-MDP objective is

$$\max_{\pi} \min_{\mu \in \mathcal{F}_{\mathcal{S}}} \mathbb{E}_{\mu} \left[\mathbb{E}_{\pi, (\tilde{p}, \tilde{r}) \sim \mu} \sum_{t=1}^{T} \gamma^{t-1} \tilde{r}(s_t, a_t) \right]. \tag{5}$$

It is known that solving the maximin problem in Eq. (4) or Eq. (5) with general ambiguity set \mathcal{F} is NP-hard [194]. Hence, several choices of the ambiguity set that facilitate solving robust MDPs are proposed. $\mathcal{F}_{s,a}$ is a state-action-wise independent ambiguity set following "(s,a)"-rectangularity [129], where $\mathcal{F} = \bigotimes_{(s,a) \in \mathcal{S} \times \mathcal{A}} \mathcal{F}_{s,a}$. \mathcal{F}_s is a state-wise independent ambiguity set [194], where $\mathcal{F} = \bigotimes_{(s) \in \mathcal{S}} \mathcal{F}_s$. The k-rectangular ambiguity set proposed in [116] is a set such that its projection onto $\mathcal{S}' \subset \mathcal{S}$ is one of at most k different possible sets. The r-rectangular ambiguity set proposed in [58] assumes the transition is a linear function of a factor matrix within a factor matrix uncertainty set. Both k-rectangular and r-rectangular ambiguity sets are coupled sets that model the correlation of transitions between different states. $\mathcal{F}_{s,a}$ assumes that the transition probability corresponding to each (s,a) pair are unrelated to any other state-action pairs. One can use robust value iteration to get a stationary and deterministic optimal policy with $\mathcal{F}_{s,a}$. \mathcal{F}_s generalizes $\mathcal{F}_{s,a}$ by assuming that the transitions corresponding to each state are independent of other states. The optimal robust policy under \mathcal{F}_s could be stationary but not necessarily deterministic. The k-rectangular ambiguity set [116] is proposed to address the conservative issue of $\mathcal{F}_{s,a}$ and $\mathcal{F}_{s,a}$ when modeling uncertainty by modeling the correlations between transitions corresponding to different states.

In addition to the rectangularity assumption, the discrepancy measure is another vital setting to define the ambiguity set \mathcal{F} . Possible choices include Wasserstein distance [1, 37, 94, 165], KL divergence [115, 167] or moments [29, 201, 207, 214]. A comprehensive review of ambiguity sets is in [144]. Choosing a proper structure of the ambiguity set and dynamically updating the ambiguity set with streaming data are still open problems.

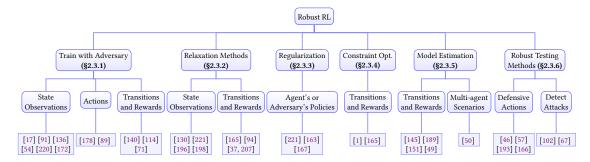


Fig. 2. Categories of robust RL methods

Beyond the MDP formulations, the robust and distributionally robust setting in [127] can be further extended to settings with partial observations, such as robust POMDP formulations in [132, 146, 156] and distributionally robust POMDP formulation in [127].

Adversarial attacks on rewards. Adversarial attacks on rewards distort the rewards at each time step to lure the agent into learning a bad policy. Let the corrupted rewards be $\tilde{r} = v(r)$, where v manifests a reward Poisoning strategy [73, 112, 223] or equivalently a reward corruption function in the corrupted reward MDP formulation [47, 222]. Huang and Zhu [73] analyze the impact of falsified reward (cost) signal on the convergence of Q-learning algorithm. The authors propose a robust region regarding the falsified cost within which the adversary will consistently fail and provide conditions that the agent will learn the target adversarial policy. Similarly, Zhang et al. [223] categorize the reward-poisoning attacks into three categories depending on the attack budget. A threshold has to be reached such that the attacks become feasible. Further increasing the threshold afterward gives rise to another two categories: non-adaptive attacks and adaptive attacks [223].

2.3 Methodologies towards robust RL

In this section, we will summarize commonly-used training methods once the robust goals are defined in the previous section. Particularly, people would like to address the following issues in adversarial training: (1) How to set up the goal of the adversaries? (2) How to make the adversarial training computationally feasible with relaxation? (3) How to regulate the capabilities of adversaries such that the policy will not be over-conservative? We will introduce corresponding methods in the sections below.

2.3.1 Robust training with adversaries

Adversarial training [56, 113] has been recognized as one of the most effective approaches in traditional supervised learning tasks in *training* time defenses. In this subsection, we will discuss one type of adversarial training strategy which is training RL agents with a lower bound of the loss (upper bound of the accumulated reward) obtained with local adversarial attacks. The local attacks can be either unparameterized attacks or attacks from a parameterized adversary agent. Training with an adversary agent can naturally be formulated as a zero-sum game between the adversary agent and the RL agent [140].

State observations Robust RL against state observations uncertainties directly adds perturbations to state observations or gradient updates of the RL agent. Such methods are based on a Q function to quantify the optimality of a state-action pair and search for the worst-possible perturbations. In the Atari Pong environment with discrete actions and image-based observations, the adversarial training on noisy observations and FGSM attacks improves the resilience

of the DQN [17] and A3C algorithm [91]. Instead of storing perturbed images based on FGSM attacks, Pattanaik et al. [136] propose to store states perturbed by stronger gradient-based attacks, which helps improve DQN and DDPG's robustness over dynamics uncertainties. In contrast to the FSGM attacks in [91] that decrease the probability of selecting the optimal actions given the current state, the gradient-based attacks in [136] encourage the probability of selecting the worst possible action. In addition to adding un-parameterized perturbations, it is possible to have a parameterized adversary agent that perturbs the state observations and minimizes the RL agent's cumulative reward. The adversary agent could be either trained in parallel [172, 220] with the RL agent or be a pre-trained adversary opponent [54].

Actions Existing adversarial training strategies against action perturbations mainly focus on local adversarial attacks [89, 178]. They adopt the game-theoretic formulation of robust RL, treating the RL agent as the protagonist agent trained against an adversary agent. For instance, Tessler et al. [178] propose two criteria of robustness to action uncertainty: the Probabilistic Action Robust MDP (PR-MDP) criterion and the Noisy Action Robust MDP (NR-MDP) criterion which consider the occasional adversarial action, and the constant adversarial perturbations, respectively. They formulate the robust learning problem by finding the optimal solution to the proposed MDP formulations and develop policy iteration methods for both criteria in the tabular setting. In a deep RL setting, Tessler et al. [178] proposes a robust deep deterministic policy gradient algorithm to train an adversary agent along with the protagonist agent. Similar to the PR-MDP setting, Klima et al. [89] introduce an adversary agent to shrink the action space of the protagonist agent in critical domains. They build on TD-learning with a modified TD error, which is a convex combination of Q values for the protagonist and the adversary agents.

Transitions and rewards Robust adversarial RL (RARL) [140] is one of the most popular frameworks to handle environment uncertainties. RARL models the environment discrepancy as extra forces/disturbances exerted by an adversary player. It formulates the policy learning as a zero-sum, minimax objective, and proposes to alternatively optimize the adversary's and the protagonist's policy until convergence. This can be viewed as optimizing the conditional value at risk (CVaR) [176] for the worst possible cases. A contemporary work ARPL [114] leverages active computation of adversarial examples in terms of dynamics noise, process noise, or observation noise during training to enable training robust policies that transfer across domains. The method gains robustness by learning from the naturally occurring adversarial scenarios that are generated on the fly, which constantly expose the flaws in the agent's decision. Recently, Huang et al. [71] proposes the RRL-stack formulation which treats robust RL as a general-sum Stackelberg game with the protagonist agent as the leader and the adversary agent as the follower. RRL-stack naturally encodes the sequential nature and provides a general formulation for robust training. With a proposed Stackelberg Policy Gradient algorithm, the RRL-stack formulation has better training stability compared with RARL in robotics and autonomous driving scenarios.

2.3.2 Robust training with relaxation

In contrast to methods in Sec. 2.3.1 that assume an adversary agent, another line of work trains RL agents with an upper bound of the loss (lower bound of the accumulated reward) which can be obtained with relaxation methods.

State observations It is possible to derive a lower bound of the accumulated reward under adversarial perturbations and directly maximize the lower bound instead of the standard RL objective. For instance, Oikarinen et al. [130] constructs the loss with adversarial examples that lead to the maximized training loss. Zhang et al. [221] propose State-Adversarial MDP where the perturbation is on state observations. Given a policy π , they derive the upper bound on performance drop between the value for a non-adversarial MDP and the value against the optimal adversary, leveraging convex relaxations of neural networks. The performance drop upper bound is in general equivalent to robustness guarantee and is closely related to the certifiable robust RL literature [196, 198].

Transitions and rewards It is possible to derive the policy performance gap under environment transition discrepancies by leveraging the nature of the ambiguity set. Lecarpentier and Rachelson [94] define a novel non-stationary MDPs (NS-MDPs) which assumes the transition dynamics and reward functions are changing slowly over time. Mathematically, They bound the evolution rate with Lipschitz Continuity and use the Wasserstein distance as the metric Mankowitz et al. [115] aim to improve the robustness of state-of-the-art continuous control RL algorithms and propose a robust version of maximum a-posteriori Policy Optimization (R-MPO). They utilize domain knowledge to select parameters to perturb and define the parameter-wise range of uncertainty set in training and testing.

With a Wasserstein-based ambiguity set and finite state action spaces, Yang [207] constructs the lower bound of the inner value minimization problem using Kantorovich duality to solve the Bellman equations. The Wasserstein distributionally robust MDPs (DRWMDP) setting proposed in [37] bears some resemblances to [207]. Derman and Mannor [37] establish the connection between DRWMDP and regularizations and provides a regularized value function lower bound that guarantee robustness w.r.t. the Wasserstein ball in linear approximation case. In addition to Wasserstein-based ambiguity sets, the ambiguity set in [201] is represented by a sequence of nested sets, and each set is associated with a confidence level. [214] later extends [201] to a more general uncertainty set formulation that can model both nested and disjoint sets. [29] generalizes the results in [214] with a new class of uncertainty set with the Wasserstein-based ambiguity set and the general-moment-based ambiguity set as two special cases.

2.3.3 Robust training with regularization

In Robust RL, a regularizer can be applied to the policy to enhance robustness. The regularizer can help encourage the smoothness of the learned policy, e.g., $\min_{\theta} \max_{\tilde{s}} \max_{\tilde{s}} D(\pi(\cdot|\tilde{s}), \pi(\cdot|\tilde{s}))$. Thus it prevents the drastic change of the policy under perturbations on state observations [163, 221]. Regularization-based methods [61, 69, 206] have also been investigated in standard neural network training to improve the robustness of the trained models. Beyond policy smoothness, Smirnova et al. [167] propose to put a set constraint on the RL agent's policy to ensure safe exploration during the learning process. They propose a distributionally robust policy iteration scheme and employ an adaptive KL-based uncertainty set with the uncertainty level related to the sample size. Huang et al. [71] propose to adaptively-regularize the adversary's policy using the highest possible return of the protagonist given the current environment generated by the adversary. The regularized adversary generates challenging yet solvable environments to improve the robustness of the protagonist agents.

2.3.4 Robust training with constrained optimization

With an explicitly defined ambiguity set, it is possible to solve the inner minimization problem in Eq.(4) using techniques in constrained optimization [1, 165]. With a Wasserstein ambiguity set, Sinha et al. [165] present a case study considering distributional robustness of deep Q-learning with Q value modeled by an overparameterized feed-forward neural net. With a Lagrangian penalty formulation, it can solve the worst perturbation over the transition dynamics with gradient descent when the penalty is sufficiently large. Abdullah et al. [1] propose Wasserstein Robust RL (WR²L), which considers an RL setting that assumes an environment adversary adding perturbations within an average Wasserstein ball \mathcal{P}_0 . WR²L conducts gradient ascents of the dynamics parameters until (almost) convergence. To guarantee that the updated dynamics parameters satisfy the ambiguity set constraint, WR²L calculates the Wasserstein distance based on Monte-Carlo estimation which scales to high-dimensional non-convex or non-concave settings.

2.3.5 Robust training with model estimation

Transitions. Ensemble learning methods aim to improve predictive performance by leveraging on a set of learning

Manuscript submitted to ACM

algorithms or models. Ensemble learning is combined with adversarial training in EPOpt [145]. EPOpt iterates between two procedures: (1) modeling the source domain with an ensemble of models and finding a robust policy for the source domain, and (2) refining the ensemble of models given the data collected from the target domain.

Rewards. Wang et al. [189] consider the setting of a finite reward set and model the reward corruption via a confusion matrix. Through dynamically refining the estimation of the confusion matrices with aggregated rewards, they manage to approximate the true reward signal. Romoff et al. [151] propose to use an estimator for reward estimation in the scenario of corrupted stochastic reward. Basically, they learn the value functions and the reward functions simultaneously. When updating the value estimation, they leverage the estimated reward rather than the sampled reward, reducing the variance propagated to the value function. AIRL [49] is an adversarial learning based inverse adversarial learning algorithm that aims to learn the disentangled reward function that is invariant to changing dynamics by using a state-action level discriminator restricted to a reward approximator plus a shaping term.

Multi-agent Scenarios. Gallego et al. [50] augment the MDPs with interfering adversaries and introduce the Threatened MDPs (TMDPs), explicitly modeling the beliefs that the agent has about the adversary's strategy. The level-k scheme [170] is adopted such that the agent and adversary alternatively model each other's belief and thus become a player of a higher level. A level-2 learner has been demonstrated to be significantly better than an independent learner (that does not consider the adversary) when playing against the adversary.

2.3.6 Robust testing methods

With a pre-trained policy, robust testing methods aim to enhance the robustness of the agent during deployment. We identify two types of testing time robust methods, including (1) passively selecting defensive actions and (2) actively detecting potential attacks.

Selecting defense actions. At test-time, one natural way to improve robustness is selecting actions under the worst possible perturbations. Considering potential perturbations on state observations, CARRL [46] proposes to select the action under the worst observation perturbations, by passing the worst observation to a trained DNN model. It computes a lower bound for Q-value based on the neural net verification methods which are related to model architecture and weights. Other methods derive lower bounds of the Q values under the worst-case perturbations based on neural network verification [57, 193]. Beyond the uncertainty in states, Sinha et al. [166] consider the uncertainty in the beliefs about the opponents' behaviors at test-time. They propose a distributionally robust bandit optimization method and dynamically adjust the risk aversion level. Therefore, the RL agent selects the optimal policy assuming the worst-possible beliefs within the ambiguity set.

Detecting potential attacks. In contrast to the above methods which are specifically designed to achieve action sub-optimality, directly *detecting* adversarial examples and predicting action based on the recovered state or reward information is rather generic. This type of method first detects the adversarial attacks and/or predicts the benign states/reward functions, and then provides suggestions on actions based on the already trained model. With *state observation* uncertainties, the prediction can be based on a frame prediction model (visual foresight module [102]) or a master policy in the meta-learning hierarchy MLAH [67].

2.4 Application benchmarks and resources

Deep robust RL algorithms are widely tested in MuJoCo control tasks [184] with physical parameter discrepancies between training and testing tasks. For instance, the 2D Hopper benchmark [45], and the HalfCheetah benchmark [191]

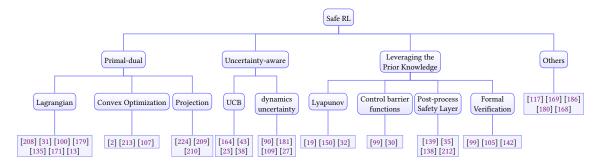


Fig. 3. Categories of safe RL methods

vary the agent's mass, ground friction, joint damping, and armature according to Gaussian distributions. Both benchmarks are later adopted in [145]. Pinto et al. [140] later propose to test in a wider range of tasks including Inverted-Pendulum, HalfCheetah, Swimmer, Hopper, Walker2d, and Ant with different mass and frictions. In addition to the MuJoCo-based benchmark, there are also robust benchmarks based on the OpenAI Gym environment [24]. For instance, Huang et al. [71] propose to test in a LunarLander benchmark with different action delays. Mankowitz et al. [115] present experiments on dexterous hand tasks which aim to rotate a cube into a target orientation using the Shadow hand domain. Robust RL methods are also tested in Atari games with noise added to the image inputs such as [17, 91].

Beyond robotics continuous control tasks and simulated games, robust RL is also tested in mobile robot tasks and autonomous driving scenarios. Liu et al. [108] propose a safe and robust benchmark containing mobile robot tasks based on Bullet safety gym [60] environments. Jaafra et al. [77] propose to test in CARLA simulator [41] with different conditions, including the traffic density, such as the number of dynamic objects, and visual effects such as weather and lightening conditions. Huang et al. [71] propose a highway merge benchmark based on the highway-env environment [97] with different surrounding vehicles' initial positions.

3 Safe learning to avoid devastating costs

3.1 Overview

Safety is also another major concern when deploying them to real-world safety-critical applications, such as self-driving cars. Traditional reinforcement learning only aims to maximize the task reward received from the environment, which lacks the guarantees of satisfying safety constraints. A safe policy should explicitly consider the safety constraints during training and prevent an RL agent from causing devastating costs or being in dangerous states. For instance, if an RL algorithm is deployed on a real robot arm, the safety constraints would be avoiding hitting fragile objects and surrounding people that may break valuable property or cause injury. Similarly, an RL agent in self-driving applications should obey traffic rules as well as avoid collisions with surrounding obstacles. Therefore, it is important to develop safe reinforcement learning algorithms for real-world applications, which allow them to complete tasks while satisfying certain safety constraints.

Safe reinforcement learning, which is also named constrained reinforcement learning, aims to learn policies that maximize the expected task reward while satisfying safety constraints. Depending on the safety requirements and the training requirements, safe RL has different formulations. In terms of the safety requirement, there are *trajectory-wise* safety constraint and state-wise safety constraint. From the training requirement perspective, we can categorize safe RL into two groups: safe deployment and safe exploration, where the former aims to act safely after the model training but

Manuscript submitted to ACM

may violate constraints during training, and the latter one refers to providing safety guarantees during training so that the whole training procedure could be finished without causing catastrophic unsafe behaviors.

Note in the setting of RL, agents need to learn from failures, similarly in the safe RL setting. One may argue that we should always avoid any safety-critical failures with domain knowledge rather than active learning. In this survey, however, we find in many cases, a precise description of the environment risk is unavailable. This is also the reason why we should pay attention to both safe deployment and safe exploration to manage the risks. Mathematically, the consequence is described as safety constraints. In the following section, we extend the traditional MDP to model them.

3.2 Problem formulation of safe reinforcement learning

Safe RL is usually modeled under the constrained Markov decision processes (CMDPs) [6] framework, where the agents are enforced with restrictions on auxiliary safety constraint violation costs. A CMDP is defined by a tuple $(S, \mathcal{A}, R, P, \gamma, \rho_0, C)$, where S is the state space, \mathcal{A} is the action space, $R: S \times \mathcal{A} \times S \mapsto \mathbb{R}$ is the reward function, $P: S \times \mathcal{A} \mapsto \Delta(S)$ is the transition function, $P: S \mapsto \Delta(S)$ is the distribution over the initial state. The first 6 elements are the same as the original MDP. The last element $C = \{c^i: S \times \mathcal{A} \times S \mapsto \mathbb{R}_{\geq 0}, i = 1, ..., m\}$ is unique to CMDPs, which is a set of costs for violating P safety constraints. For simplicity, we consider only one constraint P in the following definitions, though they could be extended to multiple constraints as well.

We denote $V_r^{\pi}(\rho_0) = \mathbb{E}_{\pi}[\sum_t \gamma^t r(s_t, a_t, s_{t+1})]$ as the reward value function, which is the expected cumulative rewards under the policy $\pi \in \Pi : \mathcal{S} \mapsto \Delta(\mathcal{A})$ and the initial state distribution ρ_0 . Similarly, we denote $V_c^{\pi}(\rho_0) = \mathbb{E}_{\pi}[\sum_t c(s_t, a_t, s_{t+1})]$ as the cost value function. There are multiple formulations and definitions of the safe RL problem, depending on the level of safety requirements, though they share the same principle: maximizing the task performance while satisfying the constraint requirement. We begin by introducing the most commonly used **trajectory-wise safety constraint**.

Trajectory-wise safety constraint. The constraint aims to ensure that the episodic cost under the policy $\tilde{\pi}$ from the initial states to the terminal states is under the threshold κ .

$$\pi^* = \arg\max_{\pi} V_r^{\pi}(\mu_0), \quad s.t. V_c^{\tilde{\pi}}(\mu_0) \le \kappa$$
 (6)

where $\tilde{\pi}$ depends on the safety requirement of different tasks. For instance, $\tilde{\pi} = \pi^*$ for **safe deployment** applications, since we only require the optimized policy satisfying safety constraints. In contrast, for **safe exploration** applications, we require $\tilde{\pi} = \{\pi_1, \pi_2, ..., \pi_K\}$, where $\pi_k, k \in \{1, 2, ..., K\}$ is the policy for the k-th optimization iteration and K is the total number of policy updates. Another similar formulation of safe RL studies the **state-wise safety constraint**, which aims to ensure that the cost of each state along the policy trajectory is under the threshold κ .

$$\pi^* = \arg\max_{\pi} V_r^{\pi}(\mu_0), \quad s.t. \forall (s_t, a_t, s_{t+1}) \sim \pi, s_0 \sim \mu_0, c(s_t, a_t, s_{t+1}) \le \kappa$$
 (7)

We could observe that the **safe exploration** formulation and the **state-wise** formulation have stricter safety requirements, since the constraint should be satisfied either all the time during the training or for every state. As a result, these formulations are more challenging and have less literature. Unless otherwise stated, we will by default assume the **safe deployment** formulation with **trajectory-wise** safety constraint in the following subsections.

To better describe the unique properties of a safe RL problem, we provide the feasibility, optimality, and temptation definitions following the previous work [108]. Their figure illustrations for one CMDP are presented in Fig. 4.

Definition 1. Feasibility. The feasible policy class is the set of policies that satisfies the constraint with threshold κ : $\Pi^{\kappa} := \{\pi(a|s) : V_c^{\pi}(\mu_0) \le \kappa, \pi \in \Pi\}$. A feasible policy should satisfy $\pi \in \Pi^{\kappa}$.

Definition 2. Optimality. A policy π^* is optimal in the safe RL context if 1) it is feasible: $\pi^* \in \Pi^K$; 2) no other feasible policy has higher reward return than it: $\forall \pi \in \Pi^K, V_r^{\pi^*}(\mu_0) \geq V_r^{\pi}(\mu_0)$.

We denote π^* as the optimal policy. Note that the optimality is defined w.r.t. the reward return within the feasible policy class Π^{κ} rather than the full policy class space Π , which means that policies that have a higher reward return than π^* may exist in a safe RL problem due to the constraint, and we formally define them as tempting policies because they are rewarding but unsafe:

Definition 3. Temptation. We define the tempting policy class as the set of policies that have a higher reward re-

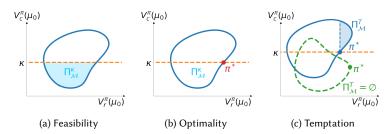


Fig. 4. Illustration of definitions via a mapping from the policy space to the metric plane $\Pi \to \mathbb{R}^2$, where the x-axis is the reward return and the y-axis is the cost return. A point on the metric plane denotes corresponding policies, i.e., the point (v_r, v_c) represents the policies $\{\pi \in \Pi | V_r^\pi(\mu_0) = v_r, V_c^\pi(\mu_0) = v_c\}$. The blue and green circles denote the policy space of two safe RL problems.

turn than the optimal policy: $\Pi^T := \{\pi(a|s) : V_r^{\pi}(\mu_0) > V_r^{\pi^*}(\mu_0), \pi \in \Pi\}$. A tempting safe RL problem has a non-empty tempting policy class: $\Pi^T \neq \emptyset$.

The existence of tempting policies is a unique feature and one of the major challenges of safe RL, since the agent needs to update the policy carefully to prevent being tempted when maximizing the reward. Suppose that a safe RL problem has no tempting policies, then one can simply maximize the reward to obtain the optimal policy, which is equivalent to solving a standard RL problem without considering the safety constraints.

3.3 Methodologies towards safe reinforcement learning

In this section, we describe how to solve the safe RL problem from different perspectives: optimization, planning, and control theory.

3.3.1 Primal-dual-based method

The primal-dual framework is commonly used to solve constrained optimization problems, which introduces an additional Lagrange multiplier λ to penalize constraint violations and gives rise to the weighted optimization objective:

$$L(\pi,\lambda) = V_r^{\pi}(\mu_0) + \lambda(\kappa - V_c^{\pi}(\mu_0)). \tag{8}$$

Manually selecting λ is a straightforward solution to the Lagrangian dual problem and is applier in early researches [20, 22]. However, hand-turning λ is challenging and can lead to suboptimal solutions, as the constraint satisfaction performance is sensitive to λ . Recent works focus on adaptively tuning the multiplier such that the penalty weight for constraint violation could be dynamically adjusted based on the policy performance and the reward scale. We will first introduce the Lagrangian-based approaches, which is the most commonly used framework to learn the Lagrange multiplier, and then present other types of approaches that utilize additional constraints, approximations, or variational inference to overcome the limitations of the Lagrangian-based methods.

Lagrangian-based method. As a simple and effective framework, Lagrangian-based methods update the primal variable (i.e., policy parameter) and dual variable (i.e., Lagrange multiplier) in turn by solving the following min-max optimization problem:

$$\min_{\lambda \ge 0} \max_{\theta} V_r^{\pi_{\theta}}(\mu_0) + \lambda(\kappa - V_c^{\pi_{\theta}}(\mu_0)), \tag{9}$$

where we denote θ as the policy parameters. Alternating between the maximizing over θ via any unconstrained reinforcement learning algorithms and minimizing over the Lagrange multiplier λ yields a series of Lagrangian-based methods to solve the safe deployment problem [208]. Chow et al. [31] propose PDO to update both primal parameters and dual variables by performing gradient descent based on on-policy estimations of the reward and cost value functions $V_r^{\pi_{\theta}}(\mu_0)$ and $J_c(\pi_{\theta})$. Liang et al. [100] accelerate the dual variable learning process by incorporating an off-policy evaluation of $V_c^{\pi\theta}(\mu_0)$ to learn the Lagrange multiplier in the dual problem, while keeping on-policy training for the primal update of the policy parameters. Tessler et al. [179] reformulate the Lagrangian dual problem and merged the cost into reward directly in each transition step to guide the policy, which can also handle mean value constraints. Paternain et al. [135] prove strong duality holds in the Lagrangian dual problem of CMDP under mild assumptions and primal-dual algorithms can converge to a saddle point without a duality gap. Stooke et al. [171] observe that the large phase shift between the constraint function and the Lagrangian multiplier could lead to unstable training and oscillation behaviors of the training curves, so they apply the idea of PID control to the dual variable learning phase such that the oscillation and overshoot behaviors could be mitigated. The augmented Lagrangian multiplier is also adopted in recent works, which can achieve superior safety performance [13]. The Lagrangian-based method is a type of generic algorithm to learn a constraint satisfaction policy and can be easily incorporated into the classic RL method. However, approximately solving the minimization (dual problem) can lead to suboptimal dual variables for each iteration, and the non-stationary cost penalty term involving λ will make the policy gradient step in the primal problem hard to optimize.

Convex optimization for the dual variable. To deal with the performance instability during optimization, CPO [2] extends the idea in TRPO [160] that adds additional trust region constraint during policy updating, and proposes to solve the dual variable via convex optimization after approximating the policy parameter space via second-order Taylor expansion. Similar to TRPO, it also reconciles the updating policy to the safe set via line search when violating the constraint. Yu et al. [213] also propose to replace the non-convex constrained objective with an approximated convex quadratic function, and thus transform the original problem into a sequence of surrogate convex constrained optimization problems. Though the dual variables could be solved efficiently via convex optimization, the approximation errors can lead to poor constraint satisfaction performance in practice. Liu et al. [107] further propose a variational inference-based approach – CVPO – to decompose the policy learning to a convex optimization phase with a non-parametric variational distribution and a supervised learning phase to fit the parameterized policy. CVPO solves the safe RL problem via a novel Expectation-Maximization style to naturally incorporate constraints during policy optimization, which provides both optimality and robustness guarantees of the learned policy.

Projection-based. Another type of method adds a projection step based on primal-dual algorithms to facilitate policy learning. Zhang et al. [224] propose a first-order gradient descent method with trust region constraint which first calculate the optimal updated policy by closed-form solution and then projects it back into parametric policy space, which is more computationally efficient compared to other trust region method. PCPO [209] first ignores cost and solves unconstrained RL and then projects the immediate policy into the safe set, which is obtained by approximation on cost constraint. Yang et al. [210] add a new step which projection immediate policy into the neighbor set of a baseline policy to accelerate learning. Although these methods improve safety during training effectively, constraint violation

is still almost inevitable because of the extrapolation error of value function parameterized by neural network and approximation error in safe set estimation.

3.3.2 Uncertainty-aware method

We have introduced a series of Lagrangian-based methods that solve the safe RL problem from the optimization perspective, now we present approaches that consider from the data collection or action sampling perspective. Instead of directly interacting with the environment based on the policy output, some approaches explicitly take uncertainty estimation into account to minimize the constraint violations. We generally categorize them into two groups: upper confidence bound-based approaches and dynamics uncertainty-based methods, where the former ones are usually model-free while the latter ones are usually model-based approaches.

Upper confidence bound (UCB). UCB-based methods leverage the uncertainty estimation to balance the exploration and exploitation during training and are common to explore tabular environment efficiently with bounded constraint violation. Singh et al. [164] and Efroni et al. [43] apply UCB on reward, cost and transition function to accelerate sampling on tabular CMDP; their methods bound both reward regret and cost constraint violation via linear program or other primal-dual optimization approaches. Different from linear constraint, Brantley et al. [23] propose UCB-based planning to deal with concave-convex and hard constraints setting for episodic tabular CMDP. Ding et al. [38] extended tabular constrained RL to CMDP with linear transition kernel and provide a provably efficient method in function approximation setting.

Dynamics uncertainty. Safe MPC [90] utilizes Gaussian Processes to estimate the dynamics uncertainty and explore the environment safely to learn the dynamics. SAVED [181] and RCE [109] both use an ensemble of neural networks as the dynamics model to estimate the dynamics prediction uncertainty and solve the constrained optimization problem via a model-predictive-control fashion, where the former formulates chance constraints to ensure safety from a probabilistic perspective, while the latter one considers the worst case unsafe scenario. CASRL [27] further extends previous approaches from stationary environments to non-stationary environments by modeling the non-stationary disturbances as probabilistic latent variables.

3.3.3 Leveraging the prior knowledge for safe exploration

Different from previous approaches that focus on safety after the policy training, some researchers also care about safety during the training process, i.e., providing certain safety guarantees during the exploration. Apparently, it is hard to guarantee the policy always safety certain safety constraints during the training without any prior knowledge of the system, safety constraints, and the environment. Therefore, some studies leverage prior or domain knowledge of specific tasks to achieve safe exploration under certain assumptions. Most approaches among them borrow ideas from the control community, such as the Lyapunov theory and control barrier functions, to leverage the prior knowledge of the system dynamics. Another type of approach utilizes the structure of safety constraints to achieve safe exploration.

Lyapunov-based method: Lyapunov theory can provide useful insights into the stability and safety of control systems. In control theory, safety is usually defined through the region of attraction that could be computed via Lyapunov functions. The key idea of using Lyapunov functions is to show the stability of the system, which is similar to showing the stability of gradient descent on strictly quasiconvex functions. If one can show that applying the policy and dynamics to the state could result in strictly smaller values on the Lyapunov function, then the state eventually converges to the equilibrium point at the origin. Berkenkamp et al. [19] utilize this principle to achieve safe exploration during the learning time with Lipschitz continuity assumption of the dynamics and policy classes. Richards et al. [150] further extend the previous work by constructing a neural network Lyapunov function. The proposed safe training

algorithm could adapt the safety certificates to the shape of the largest safe region in the state space, while relying only on knowledge of inputs and outputs of the dynamics, rather than on any specific model structure. In contrast to restricting the exploration space, Chow et al. [32] propose a Lyapunov-based safe policy gradient algorithm to solve the constrained optimization problem with unknown dynamics and constraints.

Control barrier functions-based method: Control barrier functions are another useful tool in control theory, which provides a continuous function for penalizing unsafe states or actions to infinite cost. Li and Belta [99] propose a temporal logic-guided approach with Lyapunov functions and control barrier functions to safeguard the policy exploration and deployment, which allows users to specify task objectives and constraints in different forms and at various levels. Cheng et al. [30] propose a controller architecture that combines (1) a model-free RL-based controller with (2) model-based controllers utilizing control barrier functions (CBFs) and (3) online learning of the unknown system dynamics, to ensure safety during learning. The proposed framework leverages the success of RL algorithms to learn high-performance controllers, while the CBF-based controllers both guarantee safety and guide the learning process by constraining the set of explorable policies.

Post-process safety layer: Post-process safety layer is another type of method to guarantee safe exploration. It corrects all unsafe actions generated from policy and ensures state-wise safety, which exceeds other safe RL methods only with expected or cumulative constraints. Pham et al. [139] propose an optimization layer (OptLayer) to calculate the safe action closest to the original unsafe action. OptLayer consists of a full quadratic solver, and it will run the iterative interior point method by a forward propagation for every unsafe action. Safety layer [35] provides a closed-form solution to a similar optimization objective instead and simplifies computation effectively by assuming that at most one constraint is active at each time. Although these methods obtain zero constraint violations in certain environments, their safety guarantee relies on the precise prediction of single-step state transition and thus is only applicable to linear systems or others with relatively simple dynamics. Peng et al. [138] propose to use a safe expert policy to guide the learning policy when the action is away from the expert's. Yu et al. [212] propose to perform action corrections by learning a safety editor policy simultaneously from the replay buffer, which can achieve efficient off-policy training.

Formal methods: Formal verification provides a rigorous way to provide safety for control systems. Temporal logic is a common formal method to evaluate a temporal behavior of the system, such as the multi-modal sensing information of a robot. Users could provide their domain knowledge to the temporal logic specifications that are described mathematically, precisely, and unambiguously, so that the behavior of a system could be formally defined. Li and Belta [99], Liu et al. [105] propose a temporal logic guided reinforcement learning algorithm to achieve safe exploration. The proposed framework allows users to specify task objectives and constraints in different mathematical forms and at various levels. Puranic et al. [142] also utilize signal temporal logic to learn control policies while satisfying pre-defined safety requirements.

3.3.4 Other approaches

There are also some research solving the safe RL problem from other perspectives. Mehta et al. [117] introduce a curriculum learning structure to learn a safe policy under the supervision of a safety teacher that could guide the agent back to safe zones when it violates constraints. Srinivasan et al. [169] pretrain a safety critic to estimate the future failure probability of a safety-constrained policy, and later fine-tunes the first-stage policy to the target tasks using the learned safety precautions. Turchetta et al. [186] propose an adaptive supervisor to prevent agents from taking dangerous policies by choosing and activating reset controllers with a learnable strategy. Thananjeyan et al. [180] propose to learn the constraint violating zones from offline data before the policy training, and recover the agent from Manuscript submitted to ACM

unsafe states to safe zones by using a separate recovery policy. Sootla et al. [168] wrap the environment by augmenting the state space such that the modified MDP could be solved by any RL algorithms and can achieve almost surely safety performance.

3.4 Benchmarks and resources

Safe RL tasks and environments. The AI Safety Gridworlds environment [96] provides a testing ground for tabular safe RL algorithms. Safety Gym [148] is a Mujoco-based safe RL environment, which provides several high-dimensional continuous control tasks with different difficulty levels. Safety Gym attracts much attention in the safe RL community and has become one of the most popular public benchmarks for safe RL. The Bullet Safety Gym [60] is a PyBullet simulation engine-based variant of safety gym, which provides more robots and safe RL tasks, while the Safe Control Gym [217] provides several classical control-based environments.

Safe RL algorithm resources. Safety Gym [148] provides several constrained deep RL algorithms to establish baselines that future work can build on, as well as the Bullet Safety Gym [60]. SafePO [81] is a public safe policy optimization algorithm benchmark that provides a good number of on-policy model-free safe RL method implementations.

4 Generalization in Reinforcement Learning

4.1 Overview

Generalization in RL focuses on designing algorithms to produce policies that can transfer or adapt to a variety of environments, without being overfitting to the training environments. This capability is crucial to the real-life deployment of RL agents since the environments at the testing time are often different from the training environments or inherently dynamic. There are several existing surveys that categorize RL generalization studies by their methodologies [88, 188, 225]. However, in this survey, we take a different approach to categorization by their evaluation variation.

The structure of this section is as follows: we first discuss the formalization of the contextual MDP on which the definition of generalizable RL is based in Section 4.2.1. Then we describe the two dimensions of evaluation variation in Section 4.2.2 and 4.2.3 respectively. Finally, we discuss the methodology categorization of generalizable RL in Section 4.3.

4.2 Problem formulation for generalizable RL

4.2.1 Definition of generalizable RL

To discuss generalization in a unified framework, we need to first formalize the concepts of a collection of environments. The formalization we adopt here is the Contextual Markov Decision Process (Contextual MDP), first proposed in Hallak et al. [65], and used by a recent survey in the generalization of RL [88]. A Contextual Markov Decision Process (MDP) extends the standard single-task MDP to a multi-task setting. In this work, we consider discounted infinite-horizon CMDPs, represented as a tuple $M = (S, Z, \mathcal{A}, R, P, p_0, \rho, \gamma)$. Here, S is the state space, Z is the context space, R is the action space, $R: S \times \mathcal{A} \times \mathcal{Z} \mapsto \mathbb{R}$ is the context-dependent reward function, $P: S \times \mathcal{A} \times \mathcal{Z} \mapsto \Delta(S)$ is the context-dependent transition function, $p_0: \mathcal{Z} \mapsto \Delta(S)$ is the context-dependent variable in the case of non-stationary environment distribution.

To sample a trajectory $\tau := \{(s_t, a_t, r_t)\}_{t=0}^{\infty}$ in Contextual MDPs, the context $z \sim \rho(\cdot)$ is randomly generated by the environment at the beginning of each episode. Here each z defines an environment (i.e. task). With the initial state

 $s_0 \sim p_0(\cdot \mid z)$, at each time step t, the agent follows a policy π to select an action $a_t \sim \pi(s_t)$ and receives a reward $R(s_t, a_t, z)$. Then the environment transits to the next state $s_{t+1} \sim P(\cdot \mid s_t, a_t, z)$. In some special cases, z is sampled at every time step so it changes throughout the episode.

Note that for most of the works studied in this survey, we do not emphasize the difference between the state and observation. However, we attempt to introduce the context-dependent emission function o = O(s|z) whenever the evaluation is focused on the generalization against apparent variation on the observation but not the underlying state of the environment. Examples would be only changing the color of objects in robotic manipulation tasks [66] or the background distractors in control tasks with pixel observation [219]. With the formalization of contextual MDP, we are ready to define the generalization tasks.

Definition of Generalizable RL: assuming the agent is trained in the environment distribution $\rho_{\text{train}}(z)$ for N_{train} environment steps, and the objective is to maximize the performance metric P measured over a target task distribution $\rho_{\text{test}}(z)$ after N_{test} environment steps:

$$\begin{split} \max_{\pi_{\text{test}}^{(N_{\text{test}})}} & \mathbb{E}_{\pi,z \sim \rho_{\text{test}}(z)} P_{\text{test}}^{N_{\text{test}}}(\pi,z) \\ s.t. & \pi_{\text{train}}^{(N_{\text{train}})} = \arg\max_{\pi} \mathbb{E}_{\pi,z \sim \rho_{\text{train}}(z)} P_{\text{train}}^{N_{\text{train}}}(\pi,z), \\ & \pi_{\text{test}}^{(0)} = \pi_{\text{train}}^{(N_{\text{train}})} \end{split}$$

where $P^N(\pi, z)$ represents the performance metric (that can be different for training and testing) of the policy π on the task context z after N environment steps, $\pi^{(N)}$ represents the agent's policy after N environment steps. Note that the word "testing" here does not necessarily mean the agent cannot be updated, rather refers to the "target" evaluation environment. Only in the zero-shot setting, the agent cannot be updated during testing.

4.2.2 Relationship between the training and testing distribution

In this section, we will describe the categorization of the relationship between the training distribution $\rho_{\text{train}}(z)$ and testing distribution $\rho_{\text{test}}(z)$. Here we have three categories: IID, OOD, and non-stationary.

Independent-and-Identical Distribution (IID): The training and testing environments are drawn from the same distribution, i.e., $\rho_{\text{train}}(z) = \rho_{\text{test}}(z)$. Note that although we are describing the relationship between the distributions, the RL agent is usually trained or tested on a set of sample MDPs drawn from the distribution. Therefore, even though the training and testing are drawn from the same distribution, it is still a non-trivial problem since the agent may never experience the exact same environment during training. Qualitatively, with a smaller number of distinct contexts seen during training, the IID generalization becomes more difficult since the training environments are sparsely sampled and cannot represent the true distribution well.

Out-of-Distribution (OOD) (stationary): The training and testing environments are drawn from the different distributions, i.e., $\rho_{\text{train}}(z) \neq \rho_{\text{test}}(z)$. This mismatch could be due to many reasons. For example, the exact testing distribution is unknown or difficult to model. One of the motivations for studying OOD generalization is to enable Sim-to-Real transfer. Due to the low data efficiency of RL, it is a common practice to train agents in simulators and then transfer the agents to the physical world. However, even the state-of-the-art simulators cannot capture the real world perfectly, and mismatches between the simulated and real environments can cause catastrophic failures during deployment. Another motivation is combinatorial generalization, where the testing contexts take values seen during training *independently*, but in unseen *combinations*. For example, during training, the agent has seen context combinations (a_0, b_0) , (a_0, b_1) , (a_1, b_0) , while it is tested on (a_1, b_1) .

Table 1. Categorization of generalizable RL. Each row represents the MDP components variation during evaluation, and each column represents the distributional variation during evaluation. We use colors to represent different methodologies of generalizable RL. red: generalize by environment design; blue: generalize by online adaptation; green: generalize by system learning. If a method uses two or more approaches, it will be colored with corresponding colors.

Evaluation	$\mathbf{IID}\left(p_{\mathbf{train}}(z) = p_{\mathbf{test}}(z)\right)$	OOD $(p_{train}(z) \neq p_{test}(z))$	Non-Stationary
Variation			$(\{p_{test,1}(z),p_{test,2}(z),\ldots\})$
Observation	Homomorphic MLP [187],	SODA [66], PLR [82], DR-	Benna-Fusi-RL [85]
emission	DBC [219]	Sim2Real [183], ViewInvari-	
function		antServo [154]	
o = O(s z)			
Initial state	ALP-GMM [141]	SODA [66], DR-	Benna-Fusi-RL [85]
distribution		Sim2Real [183], DR-	
$\rho_0(s z)$		RDPG [137]	
Dynamics	RARL [140], RRL-Stack [71],	ActiveDR [118], Au-	DPGP-RL [203], UP-
P(s, a z)	ActiveDR [118], ALP-	toDR [5], BayRn [124],	OSI [215], GrBAL [125],
	GMM [141], ICIL [21], HB-	PAIRED [36], DR-	PG-ELLA [7], Benna-Fusi-
	MTRL [195], ML-GP [155],	RDPG [137], UP-OSI [215],	RL [85]
	UP-OSI [215], OSC-RL [86],	GrBAL [125], EPOpt [145],	
	Agile-Quad [177], VAE-	T-MCL [162]	
	MAML [11], GrBAL [125],		
	EPOpt [145]		
Reward func-	MAML [48], meta-	GRADER [39]	Benna-Fusi-RL [85]
tion $R(s, a z)$	ADR [117], HB-MTRL [195]		
	MISA [218]		

Non-stationary: The testing environments are drawn from time-variant, non-stationary distributions. Although there is some overlap between the OOD and non-stationary, studies categorized into non-stationary focus on dealing with the non-stationarity explicitly, i.e., life-long or continuous RL [182]. In this setting, there exists a (often infinite and unknown) sequence of testing distributions, $\{\rho_{\text{test},1}(z), \rho_{\text{test},2}(z), \ldots\}$. The agents may have to learn how to leverage past experience, identify new distribution, and avoid catastrophic forgetting. To evaluate these properties, we can examine, for example, whether the agent experiences a significant performance drop when encountering the transitions of testing distribution.

4.2.3 MDP components variation of the generalizable RL

With the relationship between training and testing distribution described above, we now examine what component(s) of MDP the context controls, which constitutes another dimension of evaluation variation. There are four context-dependent components of MDP: initial state distribution, dynamics, reward function, and observation emission function.

Observation emission function o = O(s|z): The context-dependent emission function outputs the observation based on the state and the context. Since most studies do not explicitly differentiate between state and observation, we only introduce this term when the context creates a variation on the observation but not the underlying state of the environment. Examples: changing the color of the object or the view angle of the camera when using the pixel images as inputs to the policy.

Initial state distribution $\rho_0(s|z)$: The context-dependent initial state distribution outputs the initial state distribution based on the context. When the context changes the underlying state of the system, for example, the initial

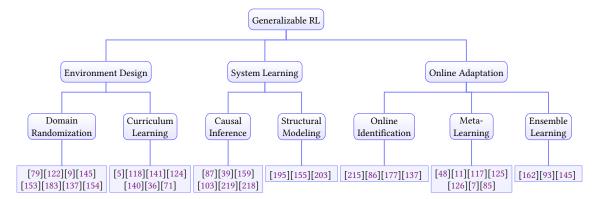


Fig. 5. Categorization of generalizable RL methods

positions of objects for the robotic manipulation tasks, the shape, and the location of obstacles for robotic locomotion tasks.

Dynamics P(s|s, a, z): The context-dependent dynamics outputs the distribution over the next state based on the previous state, the action, and the context. Examples: physical parameters such as gravity constant, friction coefficient, or perturbation forces applied to the robot.

Reward function R(s, a|z): The context-dependent reward function outputs a scalar reward based on the state, the action, and the context. The goal-conditioned MDP [8, 83, 158] can be considered as a MDP with a context-dependent reward function. Here, the goal is equivalent to the context.

With the two dimensions of the evaluation variation described in Section 4.2.2 and 4.2.3, we classify a list of papers into Table 1. Note that classifying one paper into one category does not necessarily imply the method itself can only be applied to that evaluation variation. Instead, we consider the evaluation actually appearing in the paper. From Table 1, we observe that most of the papers focus on the dynamics variation for IID, OOD, and non-stationary. This phenomenon might result from the popular applications in robotic Sim-to-Real transfer, which is one of the main motivations of generalizable RL.

4.3 Methodologies toward generalizable reinforcement learning

To improve the generalization, one can approach from three different perspectives: designing environments that guide training (**Environment Design**), learning the features of environments (**System Learning**), or designing a learning algorithm that can adapt fast to diverse tasks during testing (**Online Adaptation**). Next, we will introduce the three main approaches.

4.3.1 Environment design

Since the challenges to generalizable RL often come from the discrepancy between the training and testing environments, it is natural to ask such a question: how to design/generate training environments in order to improve the testing performance?

Domain Randomization. Domain randomization is a technique to diversify the variability of simulation environments during the training time so that the policy is able to generalize to the real-world environment during testing time. Early works use randomized model parameters that are not critical to the robot's performance [79], training Manuscript submitted to ACM

on perturbed dynamic models [9, 122, 145] and randomized vision features [153], before the terminology *domain* randomization was first coined in [183]. The simulator features that could be randomized include but are not limited to:

- Controller properties such as noise of torque, control frequency and delay, controller gain, and damping [137].
- Physics parameters such as mass, gravity, and friction [137].
- Visual inputs such as lighting, textures of objects, position, orientation, and field of view of the cameras [154, 183].

Curriculum Learning. The aforementioned works rely on handpicking randomized features and values, which inevitably requires some degrees of parameter tuning to find the proper values or distributions of the randomized features. In order to minimize human labor, there is an increasing interest recently in developing methods for automatic domain randomization and curriculum learning. Active/Automatic domain randomization uses the performance of models as a feedback signal to update the distribution of training contexts. Akkaya et al. [5] proposes Automatic Domain randomization which gradually increases the difficulty of training environments only when a minimum level of performance is achieved. Mehta et al. [118] propose Active Domain Randomization to learn a sampling strategy that produces the most informative environment by leveraging the performance difference between the randomized and reference environments. Portelas et al. [141] learns a difficulty level and an order of environments to set as training environments by iteratively sampling and fitting a Gaussian mixture model so that it maximizes the efficiency of learning. Muratore et al. [124] uses Bayesian optimization to search the space of source domain distribution parameters for learning a robust policy that maximizes real-world performance.

Adversarial training is another common formulation of curriculum learning. RARL [140] formulates the curriculum learning as a two-player zero-sum game between the agent and an adversary applying perturbation forces. Empirical studies find this kind of adversarial training can improve the generalization of the RL agent. Differently from the aforementioned zero-sum simultaneous game setting, PAIRED [36] assumes the adversary is maximizing the regret, which is defined as the performance gap between the optimal agent and the current agent. RRL-Stack [71] proposes to use a general-sum Stackelberg game formulation to address the potential problems of unsolvable environments and unstable training.

One important assumption of curriculum learning is the access and control of the environment context since the algorithms need to actively change the distributions of training environments. Compared with other methods discussed in later sections of our survey, curriculum RL does require more information and control over the environment. However, in most cases, this is not considered restrictive since RL agents are usually trained in simulators therefore access is not a major constraint. As an emerging area, the empirical evaluations of curriculum RL have attracted a lot of attention recently, but there is very little theoretical understanding of how or why it works in practice. We believe there will be more research in this area soon.

4.3.2 System learning

Causality and Invariant Feature Discovery. There usually exist underlying causality behind the dynamic system and tasks, which determines the mechanism of how things happen or how one object influence another object. Discovering such causality is a popular topic in RL, where the methods can be generally categorized into explicit learning and implicit learning. Explicit learning methods estimate the graph structure by either differentiable optimization [87], which learns a soft adjacent matrix with interventional data, or by statistic independent test [39], which iteratively updates the world causal model and planning policy. Active intervention methods are also explored in Scherrer et al. [159] to increase the efficiency of data collection and decrease the cost of conducting intervention [103]. Implicit learning

focus on extracting invariant feature from multiple environments that share the same underlying causality [218, 219]. This kind of feature is assumed to be the only important factor that influences the dynamics. Therefore, different systems can be represented by assigning different values to this variable.

Structural Modeling. In addition to learning the causality hidden in the environments, structural modeling aims to establish an explicit relationship between the environments. A common choice for such a relationship is to maintain a hierarchical probabilistic model. In Wilson et al. [195], the authors propose Hierarchical Bayesian Multi-Task RL (HB MTRL), which uses hierarchical Bayesian model classes of MDPs. Each class of MDPs shares some common structure and the shared knowledge can be transferred among MDPs of the same class. Sæmundsson et al. [155] uses a hierarchical latent variable model with Gaussian Process to model dynamics and infers the relationship between tasks automatically from the collected data. Similarly, Xu et al. [203] uses Dirichlet-Process-Gaussian-Process (DPGP) to model the MDPs distribution and maintains a mixture of experts to handle the non-stationarity.

4.3.3 Online adaptation

Online Identification. Online Identification takes a different approach from a reversed perspective of most generalizable RL. Instead of learning a policy that performs well over a wide range of environments, online identification identifies parameters of a dynamic model in the real world and then use them as information for the controller. Yu et al. [215] trains an Online System Identification (OIS) to predict the dynamics model parameters and feeds them to the controller along with system states. Kaspar et al. [86] executes predefined actions on the real robot to collect trajectories and optimizes the simulator parameters to align the simulated trajectories with real trajectories. Tan et al. [177] also uses collected real trajectories to fine-tune the simulator and match trajectories. Online Identification can also be incorporated into the policy by using a recurrent model conditioned on a latent variable representing the history of states and actions [137].

Meta-Learning. Meta-Learning, or learning-to-learn, aims to learn from past experience so that it can adapt fast to the new unseen testing tasks. Model-Agnostic Meta-Learning (MAML) is first introduced in Finn et al. [48] to train a model on a variety of learning tasks using gradient-based methods, such that it can adapt to new learning tasks using only a few training samples. Arndt et al. [11] applies MAML with a task-specific trajectory generation model to facilitate exploration and deploy trained policy on a real robot. Mehta et al. [117] considers the importance of task distributions in gradient-based Meta-learning and proposes meta-ADR to find a curriculum of tasks to avoid various problems associated with meta-learning.

Nagabandi et al. [125, 126] uses meta-learning to train a dynamics model prior such that this prior can be rapidly adapted to the testing environments. Ammar et al. [7] proposes a multi-task policy gradient method to learn and transfer knowledge between tasks to accelerate learning with theoretical guarantees. Kaplanis et al. [85] provides another perspective to equip RL agents with a synaptic model that can mitigate catastrophic forgetting.

Ensemble Learning. Ensemble learning aims to learn a collection of models and combine them to improve performance during testing. Different from Structural Modelling, Ensemble learning does not focus on maintaining an explicit (mostly hierarchical) relationship between models. Instead, it often assumes a flat structure of models and focuses on policy adaptation during testing time. From a model-based perspective, Seo et al. [162] learns an ensemble of dynamic models by updating the most accurate model during training and incorporating an online adaptive mechanism by extracting contextual information using the sampled trajectory. From a model-free perspective, SMERL [93] aims to learn a diverse set of policies so that it can generalize to the unseen environment by adopting the best solutions.

Rajeswaran et al. [145] learns a robust policy based on an ensemble of the training environments and keeps adapting the distribution over the training environments using the data from the testing environments.

4.4 Application benchmarks and resources

There exist many benchmark environments for generalizable RL. In general, the benchmark environments for generalization can be differentiated by their application domains: control/robotics (e.g. RoboSuite [226] and DMC [59]), or game-like arcade (e.g. Arcade Learning Environment game modes [18] and Phy-Q [205]). Note that it is challenging to assess the difficulty of the environment just by its application domain. A grid-world-like environment may impose greater challenges in generalization than continuous control tasks, depending on the underlying experimental settings.

Most of the benchmark environments provide the users with great flexibility to control the training and testing contexts. Therefore, many of them are not tied to a specific evaluation variation, such as IID, OOD, or non-stationary. It also highlights the difficulty of creating a standard training/testing protocol for generalizable RL, in contrast to the supervised learning domain.

5 Outlook and future research

In this section, we will provide our outlook for promising research directions. The first upfront task is that we should be able to evaluate and certify RL regarding trustworthiness requirements. Second, we should understand whether there would be some fundamental trade-offs between the trustworthy aspects, as well as the trade-off between the requirements of trustworthiness and RL's nominal performance. Third, besides the intrinsic "digital trustworthy RL", how "physical trustworthy RL" would be influenced by the agent's physical design and operational environment setting? Finally, we will discuss the "societal trustworthy RL" comprising human modeling, privacy, trust, ethics, and fairness.

5.1 How to certify and evaluate trustworthy reinforcement learning?

5.1.1 How to design trustworthy RL with certification?

It is challenging and important to provide certification for the robustness, safety, or even generalization of reinforcement learning models and systems. Although there are some existing works providing certified *robustness* for RL under different settings [92, 110, 190, 197, 198], the constraints for the attacks are limited (e.g., ℓ_p norm bounded input state perturbation) and the certified goal such as action consistency is not practical enough for safe deployment. In particular, the dynamic nature of RL and complicated environment constraints and uncertainties that cannot be modeled accurately contribute to the high complexity in this direction. Thus, more realistic certification goals such as robust action dynamics which take the temporal property RL into account are promising and more practical.

From the *safety* perspective, which focuses on a set of specified safety constraints, the certification goal is more specific considering these constraints. However, additional challenges are also introduced given the fact that 1) these constraints are usually hard to characterize in an end-to-end fashion, 2) new certification techniques are required since most existing certification methods are only based on *prediction consistency* without explicitly taking specific (safety) goals into account, 3) these constraints are usually correlated with others or even the environments which makes the certification process of high complexity. As a result, how to certify the safety of RL by categorizing the safe constraints into different levels, and characterizing the sufficient conditions for certifying safety is of great importance.

Finally, certifying the *generalization* for RL has a lot of real-world implications. For instance, if an autonomous vehicle is trained in a rural environment, it is important to know its performance certification when driving in an urban area. In this case, how to measure the difference between different environments or observations, how to define the

certification goal (e.g., the lower bound of reward given shifted observations), and how to train a certifiably generalizable RL would lead to interesting work with both theoretical and practical impacts. In addition, the generalization and robustness certifications are naturally connected. For instance, the robustness certification is on the individual input level while generalization certification is on the input distribution level; certification for robustness can be viewed as the "worst-case" shift while that for generalization can be viewed as certifying under "nature distribution shift". In this way, some certification techniques for robustness can be potentially leveraged for certifying generalization under different conditions. How to analyze their fundamental connections or indications is also an interesting direction to be explored. 5.1.2 How to design evaluation platforms for trustworthy RL?

It is vital to design proper platforms and evaluation metrics to assess the trustworthiness of a proposed RL algorithm. For a continuous environment, the most common platform is the MuJoCo environment [184], where one can vary the agent's mass, ground friction, joint damping, and armature to test robustness. Atari Game is another popular platform for observation-based attacks. Safety evaluation environments, such as the AI Safety Gridworlds environment [96], Safety Gym [148], and Bullet Safety Gym [60], design control tasks with safety zones. The evaluation of generalization focus on changing irrelevant factors, e.g., the background of the environment [33]. In addition, compositional generalization is evaluated by combining different pre-defined sub-tasks [119].

These platforms greatly facilitate the development of trustworthy RL. However, there are three critical limitations of these existing benchmarks. The most important one is still a large sim-to-real gap between the benchmark and the real world. Evaluation of robustness, safety, and generalization only in simple tasks may not generalize well in complex real-world applications. The second limitation is that most of the benchmarks use pre-defined tasks and parameters set by the creators, which lacks diversity and may be subject to human biases. Last but not least, although tasks are usually randomly sampled, the distribution rarely triggers critical events with catastrophic consequences, resulting in an underestimation of risks and very slow convergence of the results [40]. Some recent benchmarks [199] use realistic 3D simulators to construct real-world scenarios and use accelerated evaluation methods [10, 204] to emphasize the rare safety-critical cases. However, there is a trade-off between the modeling error and evaluation error [74].

These attempts are pioneers toward the efficient and accurate evaluation of certain aspects. However, we are still far away from attaining satisfying benchmarks for evaluating robustness, safety, and generalization. Comprehensive and standard benchmarks are urgently required to make RL algorithms deploy in the real world.

5.2 What is the relation between the different aspects of trustworthiness?

We have discussed the trustworthiness of RL from three different aspects. As a result, we have at least four dimensions to evaluate a trustworthy RL system: the safety, robustness, and generalization aspects of the system as well as the nominal task performance of an RL system i.e. the original functional goal of the RL. However, how to design a trustworthy RL system that is safe, robust, generalizable, and has high task performance is still an open problem and is rarely studied in the literature. To better understand this problem, we need to answer one question first: what are the relations between the four aforementioned dimensions? While most existing papers only focus on one or two of them, we believe that they are not orthogonal to each other. We provide several hypotheses and thoughts regarding the relations between them, which might be helpful for future trustworthy RL studies.

The trade-off between trustworthiness and task performance. We could observe that improving any aspects of trustworthiness might potentially induce a drop in the best possible task performance. For instance, improving the safety of an RL agent may lead the agent to be conservative in exploring high-rewarding regions, and thus has relatively lower task performance than the unsafe one [108]; increasing the robustness against adversarial perturbations may Manuscript submitted to ACM

over-smooth the policy, and thus decrease the task performance; training a generalizable policy on multiple tasks may decrease the performance on a single task due to the limitation of model capacity. As a result, we can see that improving the trustworthiness may be at the cost of sacrificing the optimal task performance, which is also a reflection of the no-free-lunch theorem. How to determine the sweet point to balance them is an interesting problem, and how to better understand the trade-offs could help us adjust the trustworthiness as needed based on different applications.

The interconnections between the trustworthiness aspects. We believe that the conceptions of safety, robustness, and generalization have non-negligible overlaps and are not orthogonal to each other, though they are mostly discussed independently in the RL literature. For example, we could view a robust RL policy in terms of observation noises as a generalization capability to the states around the training samples. Similarly, a generalizable policy for unseen scenarios can be regarded as the robustness property to dynamics uncertainties. Therefore, they are inseparable in a certain context, and we can also see similar discussions and thoughts in the general machine learning domain [202]. On the other hand, safety is also closely related to generalization and robustness, because an agent cannot be regarded as safe if it is not generalizable to novel environments or robust against adversarial perturbations. A recent work unveils the connections between robustness and safety by showing that a safe policy in a noise-free environment may not be safe under adversarial attacks [108], and we hope to see more interdisciplinary discussions in this direction.

5.3 How to co-design trustworthy RL with the physical agent and environment?

While there are tremendous developments on the computational side of trustworthy RL, the progress of designing and optimizing the physical properties of the agents are generally lagged behind [120]. Similar to organisms, the RL agents in the real world have both the "brain" which provides cognition capability and "body" which serves the sensing and actuating organs after decisions are made. Some prior works have explored this direction in simulated environments. To find a robot's morphology that is nearly optimal for a specific task, Gupta et al. [63] uses genetic algorithms to mutate and evolve a population of learning agents. Similarly, Ha [64] uses sampling-based algorithms to update the physical parameters of the agent. More recently, Yuan et al. [216] proposes to use Graph Neural Networks to model both the control policy and morphing strategy in order to jointly optimize both of them. Most of these methods show advantages compared with non-co-optimized agents.

Some key challenges of the "physical trustworthy RL" still remain. For example, how to model the physical structures, sensing, and actuation systems so that we could use gradient methods to efficiently train the whole agent. How to build an easy-to-use simulation platform to optimize a structural and modular system? How to design step-wise rather than episodic training to accelerate the training? We believe co-design the software and hardware of RL agents would provide a path to the future RL agents could be more adaptive to the changing test environments and more resilient to damages or adversarial environments, resulting in a new generation of trustworthy intelligent agents.

5.4 How to achieve the human-centric design for trustworthy RL?

In addition to the intrinsic perspectives of trustworthy RL discussed in this paper (i.e., robustness, safety, generalization), the extrinsic perspectives which mainly involve human feedback have also raised great attention and need to be further explored for the "societal trustworthy RL". In particular, such extrinsic perspectives broadly include the *explainability*, *privacy* protection for sensitive individual information, *ethics*, and general human trust on trained RL agents or models. Different trustworthy RL algorithms have also been studied in the literature by considering such human-centric design [4, 51, 128, 175] to bridge the *trust* with human.

Human behavior modeling and human-machine interaction. First, it is important and challenging to model human behaviors and interactions between human and RL agents. For instance, behavior analysis of an RL agent would be important to understand if a trained RL model is egoistic or altruistic, which is beneficial to build human trust In particular, human's sub-optimal behaviors cast a great challenge on human behavior modeling, which raises a key open problem in trustworthy RL. Inferring an agent's intent given history observations or opponent policies is widely studied in inverse reinforcement learning (IRL) (see [12] and references therein). However, IRL assumes the demonstrated behaviors are near-optimal, which is frequently violated by human. Human's sub-optimal behavior depends on various factors. For instance, Ziebart et al. [227] assumes human's sub-optimal behavior as an effect of random noise inspired by the Boltzmann noisily-rational decision model in cognitive science [14] and economic behavior [123]. Human's risk sensitivity (risk-seeking or risk-averse) is also widely known to cause sub-optimal behaviors. [147] uses the Prospect theory [84] to account for decision makers' different altitudes towards gains and losses. Reddy et al. [149] and Golub et al. [55] argue that human's sub-optimal is due to model missepecification: humans's sub-optimal behavior w.r.t. true environment dynamics model may be near-optimal w.r.t. the user's internal beliefs of the dynamics model.

Explainable RL for human trust. In addition to the formal modeling of human-machine interaction, making machine decision-making interpretable to human is a key step to building human trust. Existing efforts have been made to construct structured RL to explore the semantics of the decision-making process [80, 143], provide visualization for RL algorithms [152, 192], and explore low-dimensional interpretations [53]. However, the dynamic nature of RL and uncertainties in the environments make it challenging to fully provide a quantitative and verifiable explanation. Thus, it is important and interesting to explore different structures and architectures of RL, such as compositional structures and integration with domain knowledge to provide a semantic and logical explanation for RL.

Privacy-preserving RL to protect sensitive individual information. As the training of RL algorithms requires a large amount of data, such as collected traffic scenarios/driving behaviors for autonomous driving and human medical statistics for intelligent diagnosis, the privacy issues of trained RL models have raised great concerns [106, 157]. Clearly, such data usually contains lots of sensitive information about individuals such as personal behaviors and health status; differentially private RL algorithms [111], game theoretic approaches [34], and locally private RL [131] have been proposed to protect the private information in RL. However, from the human perspective, it is not intuitive to understand the privacy protection levels. For instance, a patient may not know what level of ϵ is suitable for his/her personal data under the (ϵ, δ) -differentially private RL training. As a result, new privacy measurement metrics, privacy protection criteria, and data valuation approaches need to be further studied to gain human trust.

Ethical and fair RL with human trust. Given the fact that RL agents are usually adopted in human daily life, such as treatment assistant robots and cleaning robots, it is of great importance to ensure the ethics of RL agents. For instance, a recent large-scale benchmark has been provided to study the ethical issues in RL video games and try to remove toxic or violent dialogues in games [70]. Similarly, different studies have been conducted to identify the fairness problems in RL as well as mitigate them [78, 104]. On the other hand, the large training observation space and non-linear decision space of RL make it more challenging to train ethical and fair RL agents with human feedback. Thus, not only from the technical perspectives such as different ethics and fairness issues identification and mitigation methods, but also from the policy-making perspectives such as ethics and fairness regulations for RL need to be better understood and studied.

Overall, the extrinsic *human trust* is widely studied in various contexts leading to different definitions, such as an attitude, an intention, or behavior [98]. Trustworthiness in automation [95] is defined as human's attitude that an automated agent will help achieve the individual's goal in uncertain situations. Researchers already show that trust affects human's reliance on intelligent agents, which may lead to under- or over-reliance and thus influence the overall Manuscript submitted to ACM

system performance [134]. It is still an open problem about enhancing human trust in RL agents by endowing RL agents the capability to generate explainable behaviors, understand human preferences and even affect human's behaviors, respect human privacy, and facilitate fairness in teamwork.

6 Conclusion

In this survey, we aim to clarify the terminology of robustness, safety, and generalization of RL, analyze their intrinsic vulnerabilities, introduce work tackling these problems, and summarize popular benchmarks, thus bringing together disparate threads of studies together in a unified framework. We hope that this survey will serve as a touch-point and reference for scientists, engineers, and policymakers in the trustworthy RL domain and spur further research.

We summarise the key takeaways of this survey below.

- Distributionally Robust RL algorithms provide a natural formulation to encode prior knowledge by choosing a
 proper ambiguity set. However, it is still unclear how to choose an appropriate ambiguity set. Hence principles
 of choosing ambiguity set should be focused on in the future.
- Robust training with adversaries could result in over-conservative policies and training instabilities. Better
 adversary attack formulation along with robust training should be proposed, such as attacks constrained by
 known rules or regulated by the RL agents' performance.
- Safe RL has multiple definitions in ML, control, and robotic communities, each with its corresponding practical applications. More studies are needed to understand the pros and cons of each direction for different use cases.
- Ensuring safety for a complex system is still a challenging problem, especially when the risk of constraint violations is hard to predict or the domain knowledge is limited. Ensuring safety during the RL training phase or in a nonstationary environment are still wide open areas.
- Although there exist many environments designed for evaluating RL, few are designed to evaluate their trustworthiness. The research community will largely benefit from a standard evaluation environment and pipeline that can provide diverse options for different evaluation protocols and variations.
- Compared to robust RL and safe RL, generalizable RL is a relatively new field. A unified paradigm with rigorous
 and verifiable theoretical analysis and standard benchmarks is urgently needed to advance this area.
- Sim2Real transfer still poses great challenges to the ML community, which is critical to the real-world deployment of RL agents. On the one hand, we need more publicly available training simulators with high fidelity in simulating complex domains such as soft objects, contacting forces, and realistic terrains, as well as effective evaluation and certification paradigms for rare but catastrophic scenarios. On the other hand, we need to develop more practical algorithms that are able to adapt online to unseen real-world environments.
- More research is needed to understand the interconnection between the different aspects of trustworthiness
 as well as the nominal functional performance of an RL system. For example, in settings with environment
 discrepancies, robust RL and generalizable RL are found to be closely related. The connections between safety
 and robustness or generalization are also observed yet have not been thoroughly established and discussed.
- While we mainly focus on the intrinsic vulnerabilities of RL for "digital trustworthy intelligence", more work is
 needed for "physical trustworthy intelligence" that considers both the RL algorithm design also the physical
 agent and environment design in the whole life cycle for trustworthy autonomy. This is still a wide-open area.
- Finally, intelligent autonomy requires the human-centric consideration for "societal trustworthy" RL, which
 includes human modeling and interaction, explainability, privacy, trust, ethics, and fairness.

Acknowledgement

We thank (in alphabetical order) Mansur Arief, Hanjiang Hu, Haohong Lin, Linyi Li, Jielin Qiu, Fan Wu, Chulin Xie, and Jiacheng Zhu for discussion and comments on drafts of this work.

Author Contributions

Mengdi Xu, Zuxin Liu, and Peide Huang wrote most parts of Sections 2, 3, and 4 respectively. Wenhao Ding and Zhepeng Cen wrote parts of section 4 and 3, respectively. Bo Li and Ding Zhao led the formalism of the survey, guided the students, wrote Sections 1 and 6 and parts of Section 5, and revised the whole manuscript.

References

- [1] Mohammed Amin Abdullah, Hang Ren, Haitham Bou Ammar, Vladimir Milenkovic, Rui Luo, Mingtian Zhang, and Jun Wang. 2019. Wasserstein robust reinforcement learning. arXiv preprint arXiv:1907.13196 (2019).
- [2] Joshua Achiam, David Held, Aviv Tamar, and Pieter Abbeel. 2017. Constrained policy optimization. In International Conference on Machine Learning. PMLR. 22–31.
- [3] HLEG AI. 2019. High-level expert group on artificial intelligence., 6 pages.
- [4] Ronald L Akers and Wesley G Jennings. 2015. Social learning theory. The handbook of criminological theory 4 (2015), 230-240.
- [5] Ilge Akkaya, Marcin Andrychowicz, Maciek Chociej, Mateusz Litwin, Bob McGrew, Arthur Petron, Alex Paino, Matthias Plappert, Glenn Powell, Raphael Ribas, et al. 2019. Solving rubik's cube with a robot hand. arXiv preprint arXiv:1910.07113 (2019).
- [6] Eitan Altman. 1998. Constrained Markov decision processes with total cost criteria: Lagrangian approach and dual linear program. Mathematical methods of operations research 48, 3 (1998), 387–417.
- [7] Haitham Bou Ammar, Eric Eaton, Paul Ruvolo, and Matthew Taylor. 2014. Online multi-task learning for policy gradient methods. In International conference on machine learning. PMLR, 1206–1214.
- [8] Marcin Andrychowicz, Filip Wolski, Alex Ray, Jonas Schneider, Rachel Fong, Peter Welinder, Bob McGrew, Josh Tobin, OpenAI Pieter Abbeel, and Wojciech Zaremba. 2017. Hindsight experience replay. Advances in neural information processing systems 30 (2017).
- [9] Rika Antonova, Silvia Cruciani, Christian Smith, and Danica Kragic. 2017. Reinforcement learning for pivoting task. arXiv preprint arXiv:1703.00472
- [10] Mansur Arief, Zhiyuan Huang, Guru Koushik Senthil Kumar, Yuanlu Bai, Shengyi He, Wenhao Ding, Henry Lam, and Ding Zhao. 2021. Deep probabilistic accelerated evaluation: A robust certifiable rare-event simulation methodology for black-box safety-critical systems. In *International Conference on Artificial Intelligence and Statistics*. PMLR, 595–603.
- [11] Karol Arndt, Murtaza Hazara, Ali Ghadirzadeh, and Ville Kyrki. 2020. Meta reinforcement learning for sim-to-real domain adaptation. In 2020 IEEE International Conference on Robotics and Automation (ICRA). IEEE, 2725–2731.
- [12] Saurabh Arora and Prashant Doshi. 2021. A survey of inverse reinforcement learning: Challenges, methods and progress. Artificial Intelligence (2021), 103500.
- [13] Yarden As, Ilnura Usmanova, Sebastian Curi, and Andreas Krause. 2022. Constrained Policy Optimization via Bayesian World Models. arXiv preprint arXiv:2201.09802 (2022).
- [14] Chris L Baker, Joshua B Tenenbaum, and Rebecca R Saxe. 2007. Goal inference as inverse planning. In *Proceedings of the Annual Meeting of the Cognitive Science Society*, Vol. 29.
- [15] Shumeet Baluja and Ian Fischer. 2018. Learning to attack: Adversarial transformation networks. In Thirty-second aaai conference on artificial intelligence.
- [16] Vahid Behzadan and Arslan Munir. 2017. Vulnerability of deep reinforcement learning to policy induction attacks. In International Conference on Machine Learning and Data Mining in Pattern Recognition. Springer, 262–275.
- [17] Vahid Behzadan and Arslan Munir. 2017. Whatever does not kill deep reinforcement learning, makes it stronger. arXiv preprint arXiv:1712.09344 (2017).
- [18] Marc G Bellemare, Yavar Naddaf, Joel Veness, and Michael Bowling. 2013. The arcade learning environment: An evaluation platform for general agents. Journal of Artificial Intelligence Research 47 (2013), 253–279.
- [19] Felix Berkenkamp, Matteo Turchetta, Angela P Schoellig, and Andreas Krause. 2017. Safe model-based reinforcement learning with stability guarantees. arXiv preprint arXiv:1705.08551 (2017).
- [20] Shalabh Bhatnagar and K Lakshmanan. 2012. An online actor-critic algorithm with function approximation for constrained markov decision processes. Journal of Optimization Theory and Applications 153, 3 (2012), 688–708.
- [21] Ioana Bica, Daniel Jarrett, and Mihaela van der Schaar. 2021. Invariant Causal Imitation Learning for Generalizable Policies. Advances in Neural Information Processing Systems 34 (2021), 3952–3964.
- [22] Vivek S Borkar. 2005. An actor-critic algorithm for constrained Markov decision processes. Systems & control letters 54, 3 (2005), 207-213.
- [23] Kianté Brantley, Miroslav Dudik, Thodoris Lykouris, Sobhan Miryoosefi, Max Simchowitz, Aleksandrs Slivkins, and Wen Sun. 2020. Constrained episodic reinforcement learning in concave-convex and knapsack settings. arXiv preprint arXiv:2006.05051 (2020).

- [24] Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. 2016. Openai gym. arXiv preprint arXiv:1606.01540 (2016).
- [25] Lukas Brunke, Melissa Greeff, Adam W Hall, Zhaocong Yuan, Siqi Zhou, Jacopo Panerati, and Angela P Schoellig. 2022. Safe learning in robotics: From learning-based control to safe reinforcement learning. Annual Review of Control, Robotics, and Autonomous Systems 5 (2022), 411–444.
- [26] Nicholas Carlini and David Wagner. 2017. Towards evaluating the robustness of neural networks. In 2017 ieee symposium on security and privacy (sp). IEEE, 39-57.
- [27] Baiming Chen, Zuxin Liu, Jiacheng Zhu, Mengdi Xu, Wenhao Ding, and Ding Zhao. 2021. Context-Aware Safe Reinforcement Learning for Non-Stationary Environments. arXiv preprint arXiv:2101.00531 (2021).
- [28] Tong Chen, Jiqiang Liu, Yingxiao Xiang, Wenjia Niu, Endong Tong, and Zhen Han. 2019. Adversarial attack and defense in reinforcement learning-from AI security view. *Cybersecurity* 2, 1 (2019), 1–22.
- [29] Zhi Chen, Pengqian Yu, and William B Haskell. 2019. Distributionally robust optimization for sequential decision-making. *Optimization* 68, 12 (2019), 2397–2426.
- [30] Richard Cheng, Gábor Orosz, Richard M Murray, and Joel W Burdick. 2019. End-to-end safe reinforcement learning through barrier functions for safety-critical continuous control tasks. In Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 33. 3387–3395.
- [31] Yinlam Chow, Mohammad Ghavamzadeh, Lucas Janson, and Marco Pavone. 2017. Risk-constrained reinforcement learning with percentile risk criteria. The Journal of Machine Learning Research 18, 1 (2017), 6070–6120.
- [32] Yinlam Chow, Ofir Nachum, Aleksandra Faust, Edgar Duenez-Guzman, and Mohammad Ghavamzadeh. 2019. Lyapunov-based safe policy optimization for continuous control. arXiv preprint arXiv:1901.10031 (2019).
- [33] Karl Cobbe, Chris Hesse, Jacob Hilton, and John Schulman. 2020. Leveraging procedural generation to benchmark reinforcement learning. In International conference on machine learning. PMLR, 2048–2056.
- [34] Lei Cui, Youyang Qu, Mohammad Reza Nosouhi, Shui Yu, Jian-Wei Niu, and Gang Xie. 2019. Improving data utility through game theory in personalized differential privacy. *Journal of Computer Science and Technology* 34, 2 (2019), 272–286.
- [35] Gal Dalal, Krishnamurthy Dvijotham, Matej Vecerik, Todd Hester, Cosmin Paduraru, and Yuval Tassa. 2018. Safe exploration in continuous action spaces. arXiv preprint arXiv:1801.08757 (2018).
- [36] Michael Dennis, Natasha Jaques, Eugene Vinitsky, Alexandre Bayen, Stuart Russell, Andrew Critch, and Sergey Levine. 2020. Emergent complexity and zero-shot transfer via unsupervised environment design. Advances in neural information processing systems 33 (2020), 13049–13061.
- [37] Esther Derman and Shie Mannor. 2020. Distributional robustness and regularization in reinforcement learning. arXiv preprint arXiv:2003.02894
- [38] Dongsheng Ding, Xiaohan Wei, Zhuoran Yang, Zhaoran Wang, and Mihailo Jovanovic. 2021. Provably efficient safe exploration via primal-dual policy optimization. In International Conference on Artificial Intelligence and Statistics. PMLR, 3304–3312.
- [39] Wenhao Ding, Haohong Lin, Bo Li, and Ding Zhao. 2022. Generalizing Goal-Conditioned Reinforcement Learning with Variational Causal Reasoning. arXiv preprint arXiv:2207.09081 (2022).
- [40] Wenhao Ding, Chejian Xu, Haohong Lin, Bo Li, and Ding Zhao. 2022. A Survey on Safety-critical Scenario Generation from Methodological Perspective. arXiv preprint arXiv:2202.02215 (2022).
- [41] Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio Lopez, and Vladlen Koltun. 2017. CARLA: An open urban driving simulator. In Conference on robot learning. PMLR, 1–16.
- [42] Gabriel Dulac-Arnold, Nir Levine, Daniel J Mankowitz, Jerry Li, Cosmin Paduraru, Sven Gowal, and Todd Hester. 2021. Challenges of real-world reinforcement learning: definitions, benchmarks and analysis. *Machine Learning* 110, 9 (2021), 2419–2468.
- [43] Yonathan Efroni, Shie Mannor, and Matteo Pirotta. 2020. Exploration-exploitation in constrained mdps. arXiv preprint arXiv:2003.02189 (2020).
- [44] Dhivya Elavarasan and P. M. Durairaj Vincent. 2020. Crop Yield Prediction Using Deep Reinforcement Learning Model for Sustainable Agrarian Applications. IEEE Access 8 (2020), 86886–86901. https://doi.org/10.1109/ACCESS.2020.2992480
- [45] Tom Erez, Yuval Tassa, and Emanuel Todorov. 2012. Infinite-horizon model predictive control for periodic tasks with contacts. Robotics: Science and systems VII 73 (2012).
- [46] Michael Everett, Björn Lütjens, and Jonathan P How. 2021. Certifiable Robustness to Adversarial State Uncertainty in Deep Reinforcement Learning. IEEE Transactions on Neural Networks and Learning Systems (2021).
- [47] Tom Everitt, Victoria Krakovna, Laurent Orseau, Marcus Hutter, and Shane Legg. 2017. Reinforcement learning with a corrupted reward channel. arXiv preprint arXiv:1705.08417 (2017).
- [48] Chelsea Finn, Pieter Abbeel, and Sergey Levine. 2017. Model-agnostic meta-learning for fast adaptation of deep networks. In *International Conference on Machine Learning*. PMLR, 1126–1135.
- [49] Justin Fu, Katie Luo, and Sergey Levine. 2017. Learning robust rewards with adversarial inverse reinforcement learning. arXiv preprint arXiv:1710.11248 (2017).
- [50] Victor Gallego, Roi Naveiro, and David Rios Insua. 2019. Reinforcement learning under threats. In Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 33. 9939–9940.
- [51] Yuan Gao, Elena Sibirtseva, Ginevra Castellano, and Danica Kragic. 2019. Fast adaptation with meta-reinforcement learning for trust modelling in human-robot interaction. In 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, 305–312.

- [52] Javier Garcia and Fernando Fernández. 2015. A comprehensive survey on safe reinforcement learning. Journal of Machine Learning Research 16, 1 (2015), 1437–1480.
- [53] Vilde B Gjærum, Inga Strümke, Ole Andreas Alsos, and Anastasios M Lekkas. 2021. Explaining a Deep Reinforcement Learning Docking Agent Using Linear Model Trees with User Adapted Visualization. Journal of Marine Science and Engineering 9, 11 (2021), 1178.
- [54] Adam Gleave, Michael Dennis, Cody Wild, Neel Kant, Sergey Levine, and Stuart Russell. 2019. Adversarial policies: Attacking deep reinforcement learning. arXiv preprint arXiv:1905.10615 (2019).
- [55] Matthew Golub, Steven Chase, and Byron Yu. 2013. Learning an internal dynamics model from control demonstration. In International Conference on Machine Learning. PMLR, 606–614.
- [56] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. 2014. Explaining and harnessing adversarial examples. arXiv preprint arXiv:1412.6572 (2014).
- [57] Sven Gowal, Krishnamurthy Dvijotham, Robert Stanforth, Rudy Bunel, Chongli Qin, Jonathan Uesato, Relja Arandjelovic, Timothy Mann, and Pushmeet Kohli. 2018. On the effectiveness of interval bound propagation for training verifiably robust models. arXiv preprint arXiv:1810.12715 (2018).
- [58] Vineet Goyal and Julien Grand-Clement. 2022. Robust Markov Decision Processes: Beyond Rectangularity. Mathematics of Operations Research (2022).
- [59] Jake Grigsby and Yanjun Qi. 2020. Measuring visual generalization in continuous control from pixels. arXiv preprint arXiv:2010.06740 (2020).
- [60] Sven Gronauer. 2022. BULLET-SAFETY-GYM: AFRAMEWORK FOR CONSTRAINED REINFORCEMENT LEARNING. (2022).
- [61] Shixiang Gu and Luca Rigazio. 2014. Towards deep neural network architectures robust to adversarial examples. arXiv preprint arXiv:1412.5068 (2014).
- [62] Shangding Gu, Long Yang, Yali Du, Guang Chen, Florian Walter, Jun Wang, Yaodong Yang, and Alois Knoll. 2022. A Review of Safe Reinforcement Learning: Methods, Theory and Applications. arXiv preprint arXiv:2205.10330 (2022).
- [63] Agrim Gupta, Silvio Savarese, Surya Ganguli, and Li Fei-Fei. 2021. Embodied intelligence via learning and evolution. Nature communications 12, 1 (Oct. 2021), 5721.
- [64] David Ha. 2019. Reinforcement Learning for Improving Agent Design. Artificial life 25, 4 (Nov. 2019), 352-365.
- [65] Assaf Hallak, Dotan Di Castro, and Shie Mannor. 2015. Contextual markov decision processes. arXiv preprint arXiv:1502.02259 (2015).
- [66] Nicklas Hansen and Xiaolong Wang. 2021. Generalization in reinforcement learning by soft data augmentation. In 2021 IEEE International Conference on Robotics and Automation (ICRA). IEEE, 13611–13617.
- [67] Aaron Havens, Zhanhong Jiang, and Soumik Sarkar. 2018. Online robust policy learning in the presence of unknown adversaries. Advances in neural information processing systems 31 (2018).
- [68] Ammar Haydari and Yasin Yılmaz. 2022. Deep Reinforcement Learning for Intelligent Transportation Systems: A Survey. IEEE Transactions on Intelligent Transportation Systems 23, 1 (2022), 11–32. https://doi.org/10.1109/TITS.2020.3008612
- [69] Matthias Hein and Maksym Andriushchenko. 2017. Formal Guarantees on the Robustness of a Classifier against Adversarial Manipulation. In NIPS.
- [70] Dan Hendrycks, Mantas Mazeika, Andy Zou, Sahil Patel, Christine Zhu, Jesus Navarro, Dawn Song, Bo Li, and Jacob Steinhardt. 2021. What would jiminy cricket do? towards agents that behave morally. arXiv preprint arXiv:2110.13136 (2021).
- [71] Peide Huang, Mengdi Xu, Fei Fang, and Ding Zhao. 2022. Robust Reinforcement Learning as a Stackelberg Game via Adaptively-Regularized Adversarial Training. arXiv preprint arXiv:2202.09514 (2022).
- [72] Sandy Huang, Nicolas Papernot, Ian Goodfellow, Yan Duan, and Pieter Abbeel. 2017. Adversarial attacks on neural network policies. arXiv preprint arXiv:1702.02284 (2017).
- [73] Yunhan Huang and Quanyan Zhu. 2019. Deceptive reinforcement learning under adversarial manipulations on cost signals. In International Conference on Decision and Game Theory for Security. Springer, 217–237.
- [74] Zhiyuan Huang, Mansur Arief, Henry Lam, and Ding Zhao. 2019. Evaluation uncertainty in data-driven self-driving testing. In 2019 IEEE Intelligent Transportation Systems Conference (ITSC). IEEE, 1902–1907.
- [75] Inaam Ilahi, Muhammad Usama, Junaid Qadir, Muhammad Umar Janjua, Ala Al-Fuqaha, Dinh Thai Hoang, and Dusit Niyato. 2020. Challenges and countermeasures for adversarial attacks on deep reinforcement learning. arXiv preprint arXiv:2001.09684 (2020).
- [76] Inaam Ilahi, Muhammad Usama, Junaid Qadir, Muhammad Umar Janjua, Ala Al-Fuqaha, Dinh Thai Hoang, and Dusit Niyato. 2021. Challenges and countermeasures for adversarial attacks on deep reinforcement learning. IEEE Transactions on Artificial Intelligence 3, 2 (2021), 90–109.
- [77] Yesmina Jaafra, Jean Luc Laurent, Aline Deruyver, and Mohamed Saber Naceur. 2019. Robust reinforcement learning for autonomous driving. (2019).
- [78] Shahin Jabbari, Matthew Joseph, Michael Kearns, Jamie Morgenstern, and Aaron Roth. 2017. Fairness in reinforcement learning. In International conference on machine learning. PMLR, 1617–1626.
- [79] Nick Jakobi. 1997. Evolutionary robotics and the radical envelope-of-noise hypothesis. Adaptive behavior 6, 2 (1997), 325-368.
- [80] Natasha Jaques, Judy Hanwen Shen, Asma Ghandeharioun, Craig Ferguson, Agata Lapedriza, Noah Jones, Shixiang Shane Gu, and Rosalind Picard. 2020. Human-centric dialog training via offline reinforcement learning. arXiv preprint arXiv:2010.05848 (2020).
- [81] Jiamg Ji, Long Yang, Shangding Gu, Yuanpei Chen, Zhouchen Lin, and Yaodong Yang. [n.d.]. Benchmarking Safe Policy Optimization for Constrained Reinforcement Learning. ([n.d.]).

- [82] Minqi Jiang, Edward Grefenstette, and Tim Rocktäschel. 2021. Prioritized level replay. In International Conference on Machine Learning. PMLR, 4940–4950.
- [83] Leslie Pack Kaelbling. 1993. Learning to Achieve Goals. In IN PROC. OF IJCAI-93. Morgan Kaufmann, 1094-1098.
- [84] Daniel Kahneman and Amos Tversky. 2013. Prospect theory: An analysis of decision under risk. In Handbook of the fundamentals of financial decision making: Part I. World Scientific, 99–127.
- [85] Christos Kaplanis, Murray Shanahan, and Claudia Clopath. 2018. Continual reinforcement learning with complex synapses. In International Conference on Machine Learning. PMLR, 2497–2506.
- [86] Manuel Kaspar, Juan D Muñoz Osorio, and Jürgen Bock. 2020. Sim2real transfer for reinforcement learning without dynamics randomization. In 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, 4383–4388.
- [87] Nan Rosemary Ke, Olexa Bilaniuk, Anirudh Goyal, Stefan Bauer, Hugo Larochelle, Bernhard Schölkopf, Michael C Mozer, Chris Pal, and Yoshua Bengio. 2019. Learning neural causal models from unknown interventions. arXiv preprint arXiv:1910.01075 (2019).
- [88] Robert Kirk, Amy Zhang, Edward Grefenstette, and Tim Rocktäschel. 2021. A survey of generalisation in deep reinforcement learning. arXiv preprint arXiv:2111.09794 (2021).
- [89] Richard Klima, Daan Bloembergen, Michael Kaisers, and Karl Tuyls. 2019. Robust temporal difference learning for critical domains. arXiv preprint arXiv:1901.08021 (2019).
- [90] Torsten Koller, Felix Berkenkamp, Matteo Turchetta, and Andreas Krause. 2018. Learning-based model predictive control for safe exploration. In 2018 IEEE Conference on Decision and Control (CDC). IEEE, 6059–6066.
- [91] Jernej Kos and Dawn Song. 2017. Delving into adversarial attacks on deep policies. arXiv preprint arXiv:1705.06452 (2017).
- [92] Aounon Kumar, Alexander Levine, and Soheil Feizi. 2021. Policy smoothing for provably robust reinforcement learning. arXiv preprint arXiv:2106.11420 (2021).
- [93] Saurabh Kumar, Aviral Kumar, Sergey Levine, and Chelsea Finn. 2020. One solution is not all you need: Few-shot extrapolation via structured maxent rl. Advances in Neural Information Processing Systems 33 (2020), 8198–8210.
- [94] Erwan Lecarpentier and Emmanuel Rachelson. 2019. Non-Stationary Markov Decision Processes, a Worst-Case Approach using Model-Based Reinforcement Learning, Extended version. arXiv preprint arXiv:1904.10090 (2019).
- [95] John D Lee and Katrina A See. 2004. Trust in automation: Designing for appropriate reliance. Human factors 46, 1 (2004), 50-80.
- [96] Jan Leike, Miljan Martic, Victoria Krakovna, Pedro A Ortega, Tom Everitt, Andrew Lefrancq, Laurent Orseau, and Shane Legg. 2017. AI safety gridworlds. arXiv preprint arXiv:1711.09883 (2017).
- [97] Edouard Leurent. 2018. An Environment for Autonomous Driving Decision-Making. https://github.com/eleurent/highway-env.
- [98] Michael Lewis, Katia Sycara, and Phillip Walker. 2018. The role of trust in human-robot interaction. In Foundations of trusted autonomy. Springer, Cham, 135–159.
- [99] Xiao Li and Calin Belta. 2019. Temporal logic guided safe reinforcement learning using control barrier functions. arXiv preprint arXiv:1903.09885
- [100] Qingkai Liang, Fanyu Que, and Eytan Modiano. 2018. Accelerated primal-dual policy optimization for safe reinforcement learning. arXiv preprint arXiv:1802.06480 (2018).
- [101] Yen-Chen Lin, Zhang-Wei Hong, Yuan-Hong Liao, Meng-Li Shih, Ming-Yu Liu, and Min Sun. 2017. Tactics of adversarial attack on deep reinforcement learning agents. arXiv preprint arXiv:1703.06748 (2017).
- [102] Yen-Chen Lin, Ming-Yu Liu, Min Sun, and Jia-Bin Huang. 2017. Detecting adversarial attacks on neural network policies with visual foresight. arXiv preprint arXiv:1710.00814 (2017).
- [103] Erik Lindgren, Murat Kocaoglu, Alexandros G Dimakis, and Sriram Vishwanath. 2018. Experimental design for cost-aware learning of causal graphs. Advances in Neural Information Processing Systems 31 (2018).
- [104] Weiwen Liu, Feng Liu, Ruiming Tang, Ben Liao, Guangyong Chen, and Pheng Ann Heng. 2020. Balancing between accuracy and fairness for interactive recommendation with reinforcement learning. In Pacific-asia conference on knowledge discovery and data mining. Springer, 155–167.
- [105] Wenliang Liu, Noushin Mehdipour, and Calin Belta. 2021. Recurrent neural network controllers for signal temporal logic specifications subject to safety constraints. IEEE Control Systems Letters (2021).
- [106] Ximeng Liu, Robert H Deng, Kim-Kwang Raymond Choo, and Yang Yang. 2019. Privacy-preserving reinforcement learning design for patient-centric dynamic treatment regimes. IEEE Transactions on Emerging Topics in Computing 9, 1 (2019), 456–470.
- [107] Zuxin Liu, Zhepeng Cen, Vladislav Isenbaev, Wei Liu, Steven Wu, Bo Li, and Ding Zhao. 2022. Constrained variational policy optimization for safe reinforcement learning. In *International Conference on Machine Learning*. PMLR, 13644–13668.
- [108] Zuxin Liu, Zijian Guo, Zhepeng Cen, Huan Zhang, Jie Tan, Bo Li, and Ding Zhao. 2022. On the Robustness of Safe Reinforcement Learning under Observational Perturbations. arXiv preprint arXiv:2205.14691 (2022).
- [109] Zuxin Liu, Hongyi Zhou, Baiming Chen, Sicheng Zhong, Martial Hebert, and Ding Zhao. 2020. Constrained Model-based Reinforcement Learning with Robust Cross-Entropy Method. arXiv preprint arXiv:2010.07968 (2020).
- [110] Björn Lütjens, Michael Everett, and Jonathan P How. 2020. Certified adversarial robustness for deep reinforcement learning. In Conference on Robot Learning. PMLR. 1328–1337.
- [111] Pingchuan Ma, Zhiqiang Wang, Le Zhang, Ruming Wang, Xiaoxiang Zou, and Tao Yang. 2019. Differentially private reinforcement learning. In International Conference on Information and Communications Security. Springer, 668–683.

- [112] Yuzhe Ma, Xuezhou Zhang, Wen Sun, and Jerry Zhu. 2019. Policy Poisoning in Batch Reinforcement Learning and Control. In Advances in Neural Information Processing Systems, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (Eds.), Vol. 32. Curran Associates, Inc. https://proceedings.neurips.cc/paper/2019/file/315f006f691ef2e689125614ea22cc61-Paper.pdf
- [113] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. 2018. Towards Deep Learning Models Resistant to Adversarial Attacks. In International Conference on Learning Representations. https://openreview.net/forum?id=rJzIBfZAb
- [114] Ajay Mandlekar, Yuke Zhu, Animesh Garg, Li Fei-Fei, and Silvio Savarese. 2017. Adversarially robust policy learning: Active construction of physically-plausible perturbations. In 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, 3932–3939.
- [115] Daniel J Mankowitz, Nir Levine, Rae Jeong, Yuanyuan Shi, Jackie Kay, Abbas Abdolmaleki, Jost Tobias Springenberg, Timothy Mann, Todd Hester, and Martin Riedmiller. 2019. Robust reinforcement learning for continuous control with model misspecification. arXiv preprint arXiv:1906.07516 (2019).
- [116] Shie Mannor, Ofir Mebel, and Huan Xu. 2016. Robust MDPs with k-rectangular uncertainty. Mathematics of Operations Research 41, 4 (2016), 1484–1509.
- [117] Bhairav Mehta, Tristan Deleu, Sharath Chandra Raparthy, Chris J Pal, and Liam Paull. 2020. Curriculum in gradient-based meta-reinforcement learning. arXiv preprint arXiv:2002.07956 (2020).
- [118] Bhairav Mehta, Manfred Diaz, Florian Golemo, Christopher J Pal, and Liam Paull. 2020. Active domain randomization. In Conference on Robot Learning. PMLR, 1162–1176.
- [119] Jorge A Mendez, Marcel Hussing, Meghna Gummadi, and Eric Eaton. 2022. CompoSuite: A Compositional Reinforcement Learning Benchmark. arXiv preprint arXiv:2207.04136 (2022).
- [120] Aslan Miriyev and Mirko Kovač. 2020. Skills for physical artificial intelligence. Nature Machine Intelligence 2, 11 (2020), 658-660.
- [121] Janosch Moos, Kay Hansel, Hany Abdulsamad, Svenja Stark, Debora Clever, and Jan Peters. 2022. Robust Reinforcement Learning: A Review of Foundations and Recent Advances. Machine Learning and Knowledge Extraction 4, 1 (2022), 276–315.
- [122] Igor Mordatch, Kendall Lowrey, and Emanuel Todorov. 2015. Ensemble-cio: Full-body dynamic motion planning that transfers to physical humanoids. In 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, 5307–5314.
- [123] Oskar Morgenstern and John Von Neumann. 1953. Theory of games and economic behavior. Princeton university press.
- [124] Fabio Muratore, Christian Eilers, Michael Gienger, and Jan Peters. 2020. Bayesian domain randomization for sim-to-real transfer. arXiv e-prints (2020), arXiv-2003.
- [125] Anusha Nagabandi, Ignasi Clavera, Simin Liu, Ronald S Fearing, Pieter Abbeel, Sergey Levine, and Chelsea Finn. 2018. Learning to adapt in dynamic, real-world environments through meta-reinforcement learning. arXiv preprint arXiv:1803.11347 (2018).
- [126] Anusha Nagabandi, Chelsea Finn, and Sergey Levine. 2018. Deep online learning via meta-learning: Continual adaptation for model-based rl. arXiv preprint arXiv:1812.07671 (2018).
- [127] Hideaki Nakao, Ruiwei Jiang, and Siqian Shen. 2021. Distributionally Robust Partially Observable Markov Decision Process with Moment-Based Ambiguity. SIAM Journal on Optimization 31, 1 (2021), 461–488.
- [128] Changjoo Nam, Phillip Walker, Michael Lewis, and Katia Sycara. 2017. Predicting trust in human control of swarms via inverse reinforcement learning. In 2017 26th ieee international symposium on robot and human interactive communication (ro-man). IEEE, 528-533.
- [129] Arnab Nilim and Laurent El Ghaoui. 2005. Robust control of Markov decision processes with uncertain transition matrices. Operations Research 53, 5 (2005), 780-798.
- [130] Tuomas Oikarinen, Wang Zhang, Alexandre Megretski, Luca Daniel, and Tsui-Wei Weng. 2021. Robust Deep Reinforcement Learning through Adversarial Loss. In Advances in Neural Information Processing Systems, A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan (Eds.). https://openreview.net/forum?id=eaAM_bdW0O
- [131] Hajime Ono and Tsubasa Takahashi. 2020. Locally private distributed reinforcement learning. arXiv preprint arXiv:2001.11718 (2020).
- [132] Takayuki Osogami. 2015. Robust partially observable Markov decision process. In International Conference on Machine Learning. PMLR, 106-115.
- [133] Marcel Panzer and Benedict Bender. 2022. Deep reinforcement learning in production systems: a systematic literature review. International Journal of Production Research 60, 13 (2022), 4316–4341. https://doi.org/10.1080/00207543.2021.1973138 arXiv:https://doi.org/10.1080/00207543.2021.1973138
- [134] Raja Parasuraman and Victor Riley, 1997. Humans and automation: Use, misuse, disuse, abuse, Human factors 39, 2 (1997), 230–253.
- [135] Santiago Paternain, Luiz FO Chamon, Miguel Calvo-Fullana, and Alejandro Ribeiro. 2019. Constrained reinforcement learning has zero duality gap. arXiv preprint arXiv:1910.13393 (2019).
- [136] Anay Pattanaik, Zhenyi Tang, Shuijing Liu, Gautham Bommannan, and Girish Chowdhary. 2018. Robust Deep Reinforcement Learning with adversarial attacks. In 17th International Conference on Autonomous Agents and Multiagent Systems, AAMAS 2018. International Foundation for Autonomous Agents and Multiagent Systems (IFAAMAS), 2040–2042.
- [137] Xue Bin Peng, Marcin Andrychowicz, Wojciech Zaremba, and Pieter Abbeel. 2018. Sim-to-real transfer of robotic control with dynamics randomization. In 2018 IEEE international conference on robotics and automation (ICRA). IEEE, 3803–3810.
- [138] Zhenghao Peng, Quanyi Li, Chunxiao Liu, and Bolei Zhou. 2022. Safe driving via expert guided policy optimization. In Conference on Robot Learning. PMLR, 1554–1563.
- [139] Tu-Hoa Pham, Giovanni De Magistris, and Ryuki Tachibana. 2018. Optlayer-practical constrained optimization for deep reinforcement learning in the real world. In 2018 IEEE International Conference on Robotics and Automation (ICRA). IEEE, 6236–6243.

- [140] Lerrel Pinto, James Davidson, Rahul Sukthankar, and Abhinav Gupta. 2017. Robust adversarial reinforcement learning. In International Conference on Machine Learning. PMLR, 2817–2826.
- [141] Rémy Portelas, Cédric Colas, Katja Hofmann, and Pierre-Yves Oudeyer. 2020. Teacher algorithms for curriculum learning of deep rl in continuously parameterized environments. In Conference on Robot Learning. PMLR, 835–853.
- [142] Aniruddh Gopinath Puranic, Jyotirmoy Deshmukh, and Stefanos Nikolaidis. 2021. Learning from Demonstrations Using Signal Temporal Logic in Stochastic and Continuous Domains. IEEE Robotics and Automation Letters (2021).
- [143] Angela Radulescu, Yael Niv, and Ian Ballard. 2019. Holistic reinforcement learning: the role of structure and attention. Trends in cognitive sciences 23. 4 (2019), 278–292.
- [144] Hamed Rahimian and Sanjay Mehrotra. 2019. Distributionally robust optimization: A review. arXiv preprint arXiv:1908.05659 (2019).
- [145] Aravind Rajeswaran, Sarvjeet Ghotra, Balaraman Ravindran, and Sergey Levine. 2016. Epopt: Learning robust neural network policies using model ensembles. arXiv preprint arXiv:1610.01283 (2016).
- [146] Mohammad Rasouli and Soroush Saghafian. 2018. Robust partially observable Markov decision processes. (2018).
- [147] Lillian J Ratliff and Eric Mazumdar. 2019. Inverse risk-sensitive reinforcement learning. IEEE Trans. Automat. Control 65, 3 (2019), 1256-1263.
- [148] Alex Ray, Joshua Achiam, and Dario Amodei. 2019. Benchmarking safe exploration in deep reinforcement learning. arXiv preprint arXiv:1910.01708 7 (2019).
- [149] Siddharth Reddy, Anca D Dragan, and Sergey Levine. 2018. Where do you think you're going?: Inferring beliefs about dynamics from behavior. arXiv preprint arXiv:1805.08010 (2018).
- [150] Spencer M Richards, Felix Berkenkamp, and Andreas Krause. 2018. The lyapunov neural network: Adaptive stability certification for safe learning of dynamical systems. In Conference on Robot Learning. PMLR, 466–476.
- [151] Joshua Romoff, Peter Henderson, Alexandre Piché, Vincent Francois-Lavet, and Joelle Pineau. 2018. Reward estimation for variance reduction in deep reinforcement learning. arXiv preprint arXiv:1805.03359 (2018).
- [152] Christian Rupprecht, Cyril Ibrahim, and Christopher J Pal. 2019. Finding and visualizing weaknesses of deep reinforcement learning agents. arXiv preprint arXiv:1904.01318 (2019).
- [153] Fereshteh Sadeghi and Sergey Levine. 2016. Cad2rl: Real single-image flight without a single real image. arXiv preprint arXiv:1611.04201 (2016).
- [154] Fereshteh Sadeghi, Alexander Toshev, Eric Jang, and Sergey Levine. 2018. Sim2real viewpoint invariant visual servoing by recurrent control. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 4691–4699.
- [155] Steindór Sæmundsson, Katja Hofmann, and Marc Peter Deisenroth. 2018. Meta reinforcement learning with latent variable gaussian processes. arXiv preprint arXiv:1803.07551 (2018).
- [156] Soroush Saghafian. 2018. Ambiguous partially observable Markov decision processes: Structural results and applications. Journal of Economic Theory 178 (2018), 1–35.
- [157] Jun Sakuma, Shigenobu Kobayashi, and Rebecca N Wright. 2008. Privacy-preserving reinforcement learning. In Proceedings of the 25th international conference on Machine learning. 864–871.
- [158] Tom Schaul, Daniel Horgan, Karol Gregor, and David Silver. 2015. Universal Value Function Approximators. In Proceedings of the 32nd International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 37), Francis Bach and David Blei (Eds.). PMLR, Lille, France, 1312–1320. https://proceedings.mlr.press/v37/schaul15.html
- [159] Nino Scherrer, Olexa Bilaniuk, Yashas Annadani, Anirudh Goyal, Patrick Schwab, Bernhard Schölkopf, Michael C Mozer, Yoshua Bengio, Stefan Bauer, and Nan Rosemary Ke. 2021. Learning neural causal models with active interventions. arXiv preprint arXiv:2109.02429 (2021).
- [160] John Schulman, Sergey Levine, Pieter Abbeel, Michael Jordan, and Philipp Moritz. 2015. Trust region policy optimization. In International conference on machine learning. PMLR, 1889–1897.
- [161] National Science and Technology Council (US). Select Committee on Artificial Intelligence. 2019. The national artificial intelligence research and development strategic plan: 2019 update. National Science and Technology Council (US), Select Committee on Artificial
- [162] Younggyo Seo, Kimin Lee, Ignasi Clavera Gilaberte, Thanard Kurutach, Jinwoo Shin, and Pieter Abbeel. 2020. Trajectory-wise multiple choice learning for dynamics generalization in reinforcement learning. Advances in Neural Information Processing Systems 33 (2020), 12968–12979.
- [163] Qianli Shen, Yan Li, Haoming Jiang, Zhaoran Wang, and Tuo Zhao. 2020. Deep Reinforcement Learning with Robust and Smooth Policy. In International Conference on Machine Learning. PMLR, 8707–8718.
- [164] Rahul Singh, Abhishek Gupta, and Ness B Shroff. 2020. Learning in Markov decision processes under constraints. arXiv preprint arXiv:2002.12435
- [165] Aman Sinha, Hongseok Namkoong, Riccardo Volpi, and John Duchi. 2017. Certifying some distributional robustness with principled adversarial training. arXiv preprint arXiv:1710.10571 (2017).
- [166] Aman Sinha, Matthew O'Kelly, Hongrui Zheng, Rahul Mangharam, John Duchi, and Russ Tedrake. 2020. Formulazero: Distributionally robust online adaptation via offline population synthesis. In *International Conference on Machine Learning*. PMLR, 8992–9004.
- [167] Elena Smirnova, Elvis Dohmatob, and Jérémie Mary. 2019. Distributionally robust reinforcement learning. arXiv preprint arXiv:1902.08708 (2019).
- [168] Aivar Sootla, Alexander I Cowen-Rivers, Taher Jafferjee, Ziyan Wang, David H Mguni, Jun Wang, and Haitham Ammar. 2022. Sauté RL: Almost surely safe reinforcement learning using state augmentation. In International Conference on Machine Learning. PMLR, 20423–20443.
- [169] Krishnan Srinivasan, Benjamin Eysenbach, Sehoon Ha, Jie Tan, and Chelsea Finn. 2020. Learning to be safe: Deep rl with a safety critic. arXiv preprint arXiv:2010.14603 (2020).

- [170] Dale O Stahl II and Paul W Wilson. 1994. Experimental evidence on players' models of other players. Journal of economic behavior & organization 25, 3 (1994), 309–327.
- [171] Adam Stooke, Joshua Achiam, and Pieter Abbeel. 2020. Responsive safety in reinforcement learning by pid lagrangian methods. In International Conference on Machine Learning. PMLR. 9133–9143.
- [172] Yanchao Sun, Ruijie Zheng, Yongyuan Liang, and Furong Huang. 2022. Who Is the Strongest Enemy? Towards Optimal and Efficient Evasion Attacks in Deep RL. In International Conference on Learning Representations. https://openreview.net/forum?id=JM2kFbJvvI
- [173] Richard S. Sutton and Andrew G. Barto. 1998. Reinforcement learning an introduction. MIT Press. https://www.worldcat.org/oclc/37293240
- [174] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. 2013. Intriguing properties of neural networks. arXiv preprint arXiv:1312.6199 (2013).
- [175] Aaquib Tabrez and Bradley Hayes. 2019. Improving human-robot interaction through explainable reinforcement learning. In 2019 14th ACM/IEEE International Conference on Human-Robot Interaction (HRI). IEEE, 751–753.
- [176] Aviv Tamar, Yonatan Glassner, and Shie Mannor. 2015. Optimizing the CVaR via sampling. In Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 29.
- [177] Jie Tan, Tingnan Zhang, Erwin Coumans, Atil Iscen, Yunfei Bai, Danijar Hafner, Steven Bohez, and Vincent Vanhoucke. 2018. Sim-to-real: Learning agile locomotion for quadruped robots. arXiv preprint arXiv:1804.10332 (2018).
- [178] Chen Tessler, Yonathan Efroni, and Shie Mannor. 2019. Action robust reinforcement learning and applications in continuous control. In International Conference on Machine Learning. PMLR, 6215–6224.
- [179] Chen Tessler, Daniel J Mankowitz, and Shie Mannor. 2018. Reward constrained policy optimization. arXiv preprint arXiv:1805.11074 (2018).
- [180] Brijen Thananjeyan, Ashwin Balakrishna, Suraj Nair, Michael Luo, Krishnan Srinivasan, Minho Hwang, Joseph E Gonzalez, Julian Ibarz, Chelsea Finn, and Ken Goldberg. 2021. Recovery rl: Safe reinforcement learning with learned recovery zones. IEEE Robotics and Automation Letters 6, 3 (2021), 4915–4922.
- [181] Brijen Thananjeyan, Ashwin Balakrishna, Ugo Rosolia, Felix Li, Rowan McAllister, Joseph E Gonzalez, Sergey Levine, Francesco Borrelli, and Ken Goldberg. 2020. Safety augmented value estimation from demonstrations (saved): Safe deep model-based rl for sparse cost robotic tasks. IEEE Robotics and Automation Letters 5, 2 (2020), 3612–3619.
- [182] Sebastian Thrun and Tom M Mitchell. 1995. Lifelong robot learning. Robotics and autonomous systems 15, 1-2 (1995), 25-46.
- [183] Josh Tobin, Rachel Fong, Alex Ray, Jonas Schneider, Wojciech Zaremba, and Pieter Abbeel. 2017. Domain randomization for transferring deep neural networks from simulation to the real world. In 2017 IEEE/RSJ international conference on intelligent robots and systems (IROS). IEEE, 23–30.
- [184] Emanuel Todorov, Tom Erez, and Yuval Tassa. 2012. Mujoco: A physics engine for model-based control. In 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems. IEEE, 5026–5033.
- [185] Edgar Tretschk, Seong Joon Oh, and Mario Fritz. 2018. Sequential attacks on agents for long-term adversarial goals. arXiv preprint arXiv:1805.12487 (2018).
- [186] Matteo Turchetta, Andrey Kolobov, Shital Shah, Andreas Krause, and Alekh Agarwal. 2020. Safe reinforcement learning via curriculum induction. arXiv preprint arXiv:2006.12136 (2020).
- [187] Elise van der Pol, Daniel Worrall, Herke van Hoof, Frans Oliehoek, and Max Welling. 2020. MDP homomorphic networks: Group symmetries in reinforcement learning. Advances in Neural Information Processing Systems 33 (2020), 4199–4210.
- [188] Nelson Vithayathil Varghese and Qusay H Mahmoud. 2020. A survey of multi-task deep reinforcement learning. Electronics 9, 9 (2020), 1363.
- [189] Jingkang Wang, Yang Liu, and Bo Li. 2020. Reinforcement learning with perturbed rewards. In Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 34, 6202–6209.
- [190] Yuh-Shyang Wang, Tsui-Wei Weng, and Luca Daniel. 2019. Verification of neural network control policy under persistent adversarial perturbation. arXiv preprint arXiv:1908.06353 (2019).
- [191] Paweł Wawrzyński. 2009. Real-time reinforcement learning by sequential actor-critics and experience replay. Neural networks 22, 10 (2009), 1484–1497.
- [192] Lindsay Wells and Tomasz Bednarz. 2021. Explainable ai and reinforcement learning—a systematic review of current approaches and trends. Frontiers in artificial intelligence 4 (2021), 550030.
- [193] Lily Weng, Huan Zhang, Hongge Chen, Zhao Song, Cho-Jui Hsieh, Luca Daniel, Duane Boning, and Inderjit Dhillon. 2018. Towards fast computation of certified robustness for relu networks. In International Conference on Machine Learning. PMLR, 5276–5285.
- [194] Wolfram Wiesemann, Daniel Kuhn, and Berç Rustem. 2013. Robust Markov decision processes. Mathematics of Operations Research 38, 1 (2013), 153–183.
- [195] Aaron Wilson, Alan Fern, Soumya Ray, and Prasad Tadepalli. 2007. Multi-task reinforcement learning: a hierarchical bayesian approach. In Proceedings of the 24th international conference on Machine learning. 1015–1022.
- [196] Fan Wu, Linyi Li, Zijian Huang, Yevgeniy Vorobeychik, Ding Zhao, and Bo Li. 2021. arXiv preprint arXiv:2106.09292 (2021).
- [197] Fan Wu, Linyi Li, Zijian Huang, Yevgeniy Vorobeychik, Ding Zhao, and Bo Li. 2022. CROP: Certifying Robust Policies for Reinforcement Learning through Functional Smoothing. In International Conference on Learning Representations.
- [198] Fan Wu, Linyi Li, Chejian Xu, Huan Zhang, Bhavya Kailkhura, Krishnaram Kenthapadi, Ding Zhao, and Bo Li. 2022. COPA: Certifying Robust Policies for Offline Reinforcement Learning against Poisoning Attacks. arXiv preprint arXiv:2203.08398 (2022).

- [199] Chejian Xu, Wenhao Ding, Weijie Lyu, Zuxin Liu, Shuai Wang, Yihan He, Hanjiang Hu, Ding Zhao, and Bo Li. 2022. SafeBench: A Benchmarking Platform for Safety Evaluation of Autonomous Vehicles. arXiv preprint arXiv:2206.09682 (2022).
- [200] Huan Xu and Shie Mannor. 2006. The robustness-performance tradeoff in Markov decision processes. Advances in Neural Information Processing Systems 19 (2006).
- [201] Huan Xu and Shie Mannor. 2010. Distributionally Robust Markov Decision Processes.. In NIPS. 2505-2513.
- [202] Huan Xu and Shie Mannor. 2012. Robustness and generalization. Machine learning 86, 3 (2012), 391-423.
- [203] Mengdi Xu, Wenhao Ding, Jiacheng Zhu, Zuxin Liu, Baiming Chen, and Ding Zhao. 2020. Task-agnostic online reinforcement learning with an infinite mixture of gaussian processes. arXiv preprint arXiv:2006.11441 (2020).
- [204] Mengdi Xu, Peide Huang, Fengpei Li, Jiacheng Zhu, Xuewei Qi, Kentaro Oguchi, Zhiyuan Huang, Henry Lam, and Ding Zhao. 2021. Accelerated Policy Evaluation: Learning Adversarial Environments with Adaptive Importance Sampling. arXiv preprint arXiv:2106.10566 (2021).
- [205] Cheng Xue, Vimukthini Pinto, Chathura Gamage, Ekaterina Nikonova, Peng Zhang, and Jochen Renz. 2021. Phy-q: A benchmark for physical reasoning. arXiv preprint arXiv:2108.13696 (2021).
- [206] Ziang Yan, Yiwen Guo, and Changshui Zhang. 2018. Deep Defense: Training DNNs with Improved Adversarial Robustness. Advances in Neural Information Processing Systems 31 (2018), 419–428.
- [207] Insoon Yang. 2017. A convex optimization approach to distributionally robust Markov decision processes with Wasserstein distance. IEEE control systems letters 1, 1 (2017), 164–169.
- [208] Qisong Yang, Thiago D Simão, Simon H Tindemans, and Matthijs TJ Spaan. 2021. WCSAC: Worst-Case Soft Actor Critic for Safety-Constrained Reinforcement Learning.. In AAAI. 10639–10646.
- [209] Tsung-Yen Yang, Justinian Rosca, Karthik Narasimhan, and Peter J Ramadge. 2020. Projection-based constrained policy optimization. arXiv preprint arXiv:2010.03152 (2020).
- [210] Tsung-Yen Yang, Justinian Rosca, Karthik Narasimhan, and Peter J Ramadge. 2021. Accelerating Safe Reinforcement Learning with Constraint-mismatched Baseline Policies. In International Conference on Machine Learning. PMLR. 11795–11807.
- [211] Chao Yu, Jiming Liu, Shamim Nemati, and Guosheng Yin. 2021. Reinforcement Learning in Healthcare: A Survey. ACM Comput. Surv. 55, 1, Article 5 (nov 2021), 36 pages. https://doi.org/10.1145/3477600
- [212] Haonan Yu, Wei Xu, and Haichao Zhang. 2022. Towards Safe Reinforcement Learning with a Safety Editor Policy. arXiv preprint arXiv:2201.12427
- [213] Ming Yu, Zhuoran Yang, Mladen Kolar, and Zhaoran Wang. 2019. Convergent policy optimization for safe reinforcement learning. arXiv preprint arXiv:1910.12156 (2019).
- [214] Pengqian Yu and Huan Xu. 2015. Distributionally robust counterpart in Markov decision processes. IEEE Trans. Automat. Control 61, 9 (2015), 2538–2543.
- [215] Wenhao Yu, Jie Tan, C Karen Liu, and Greg Turk. 2017. Preparing for the unknown: Learning a universal policy with online system identification. arXiv preprint arXiv:1702.02453 (2017).
- [216] Ye Yuan, Yuda Song, Zhengyi Luo, Wen Sun, and Kris Kitani. 2021. Transform2Act: Learning a Transform-and-Control Policy for Efficient Agent Design. arXiv preprint arXiv:2110.03659 (2021).
- [217] Zhaocong Yuan, Adam W Hall, Siqi Zhou, Lukas Brunke, Melissa Greeff, Jacopo Panerati, and Angela P Schoellig. 2021. safe-control-gym: a Unified Benchmark Suite for Safe Learning-based Control and Reinforcement Learning. arXiv preprint arXiv:2109.06325 (2021).
- [218] Amy Zhang, Clare Lyle, Shagun Sodhani, Angelos Filos, Marta Kwiatkowska, Joelle Pineau, Yarin Gal, and Doina Precup. 2020. Invariant causal prediction for block mdps. In *International Conference on Machine Learning*. PMLR, 11214–11224.
- [219] Amy Zhang, Rowan McAllister, Roberto Calandra, Yarin Gal, and Sergey Levine. 2020. Learning invariant representations for reinforcement learning without reconstruction. arXiv preprint arXiv:2006.10742 (2020).
- [220] Huan Zhang, Hongge Chen, Duane S Boning, and Cho-Jui Hsieh. 2021. Robust Reinforcement Learning on State Observations with Learned Optimal Adversary. In International Conference on Learning Representations. https://openreview.net/forum?id=sCZbhBvqQaU
- [221] Huan Zhang, Hongge Chen, Chaowei Xiao, Bo Li, Mingyan Liu, Duane Boning, and Cho-Jui Hsieh. 2020. Robust Deep Reinforcement Learning against Adversarial Perturbations on State Observations. In Advances in Neural Information Processing Systems, H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin (Eds.), Vol. 33. Curran Associates, Inc., 21024–21037. https://proceedings.neurips.cc/paper/2020/file/f0eb6568ea114ba6e293f903c34d7488-Paper.pdf
- [222] Haoqi Zhang and David C Parkes. 2008. Value-Based Policy Teaching with Active Indirect Elicitation.. In AAAI, Vol. 8. 208–214.
- [223] Xuezhou Zhang, Yuzhe Ma, Adish Singla, and Xiaojin Zhu. 2020. Adaptive reward-poisoning attacks against reinforcement learning. In *International Conference on Machine Learning*. PMLR, 11225–11234.
- [224] Yiming Zhang, Quan Vuong, and Keith Ross. 2020. First order constrained optimization in policy space. Advances in Neural Information Processing Systems (2020).
- [225] Wenshuai Zhao, Jorge Peña Queralta, and Tomi Westerlund. 2020. Sim-to-real transfer in deep reinforcement learning for robotics: a survey. In 2020 IEEE Symposium Series on Computational Intelligence (SSCI). IEEE, 737–744.
- [226] Yuke Zhu, Josiah Wong, Ajay Mandlekar, and Roberto Martín-Martín. 2020. robosuite: A modular simulation framework and benchmark for robot learning. arXiv preprint arXiv:2009.12293 (2020).

Manuscript submitted to ACM

Chicago, IL, USA, 1433-1438.

[227] Brian D Ziebart, Andrew L Maas, J Andrew Bagnell, Anind K Dey, et al. 2008. Maximum entropy inverse reinforcement learning.. In Aaai, Vol. 8.