ANNUAL REVIEWS

Annual Review of Control, Robotics, and Autonomous Systems

Safe Learning in Robotics: From Learning-Based Control to Safe Reinforcement Learning

Lukas Brunke,[1,2,3,*] Melissa Greeff,[1,2,3,*] Adam W. Hall,[1,2,3,*] Zhaocong Yuan,[1,2,3,*] Siqi Zhou,[1,2,3,*] Jacopo Panerati,[1,2,3] and Angela P. Schoellig[1,2,3]

[1]Institute for Aerospace Studies, University of Toronto, Toronto, Ontario, Canada; email: lukas.brunke@robotics.utias.utoronto.ca, melissa.greeff@robotics.utias.utoronto.ca, adam.hall@robotics.utias.utoronto.ca, justin.yuan@robotics.utias.utoronto.ca, siqi.zhou@robotics.utias.utoronto.ca, jacopo.panerati@utoronto.ca, angela.schoellig@utoronto.ca

[2]University of Toronto Robotics Institute, Toronto, Ontario, Canada

[3]Vector Institute for Artificial Intelligence, Toronto, Ontario, Canada

Annu. Rev. Control Robot. Auton. Syst. 2022. 5:411–44

First published as a Review in Advance on January 26, 2022

The *Annual Review of Control, Robotics, and Autonomous Systems* is online at control.annualreviews.org

https://doi.org/10.1146/annurev-control-042920-020211

*These authors contributed equally to this article

Keywords

safe learning, robotics, robot learning, learning-based control, safe reinforcement learning, adaptive control, robust control, model predictive control, machine learning, benchmarks

Abstract

The last half decade has seen a steep rise in the number of contributions on safe learning methods for real-world robotic deployments from both the control and reinforcement learning communities. This article provides a concise but holistic review of the recent advances made in using machine learning to achieve safe decision-making under uncertainties, with a focus on unifying the language and frameworks used in control theory and reinforcement learning research. It includes learning-based control approaches that safely improve performance by learning the uncertain dynamics, reinforcement learning approaches that encourage safety or robustness, and methods that can formally certify the safety of a learned control policy. As data- and learning-based robot control methods continue to gain traction, researchers must understand when and how to best leverage them in real-world scenarios where safety is imperative, such as when operating in close proximity

to humans. We highlight some of the open challenges that will drive the field of robot learning in the coming years, and emphasize the need for realistic physics-based benchmarks to facilitate fair comparisons between control and reinforcement learning approaches.

1. INTRODUCTION

Robotics researchers strive to design systems that can operate autonomously in increasingly complex scenarios, often in close proximity to humans. Examples include self-driving vehicles (1), aerial delivery (2), and the use of mobile manipulators for service tasks (3). However, the dynamics of these complex applications are often uncertain or only partially known—for example, the mass distribution of a carried payload might not be given a priori. Uncertainties arise from various sources. For example, the robot dynamics may not be perfectly modeled, sensor measurements may be noisy, and/or the operating environment may not be well characterized or may include other agents whose dynamics and plans are not known.

In these real-world applications, robots must make decisions despite having only partial knowledge of the world. In recent years, the research community has multiplied its efforts to leverage data-based approaches to address this problem. This was motivated in part by the success of machine learning in other areas, such as computer vision and natural language processing.

A crucial, domain-specific challenge of learning for robot control is the need to implement and formally guarantee the safety of the robot's behavior, not only for the optimized policy (or controller, which is essential for the certification of systems that interact with humans) but also during learning, to avoid costly hardware failures and improve convergence. Ultimately, these safety guarantees can only be derived from the assumptions and structure captured by the problem formalization.

Researchers in both control theory and machine learning—reinforcement learning (RL) in particular—have proposed approaches to tackle this problem. Control theory has traditionally taken a model-driven approach (see **Figure 1**): It leverages a given dynamics model and provides guarantees with respect to known operating conditions. RL has traditionally taken a data-driven approach, which makes it highly adaptable to new contexts at the expense of providing formal guarantees. Combining model-driven and data-driven approaches, and leveraging the advantages of each, is a promising direction for safe learning in robotics. The methods we review encourage robustness (by accounting for the worst-case scenarios and taking conservative actions), enable adaptation (by learning from online observations and adapting to unknown situations), and build and leverage prediction models (based on a combination of domain knowledge, real-world data, and high-fidelity simulators).

While control is still the bedrock of all current robot applications, the body of safe RL literature has ballooned from tens of publications to more than a thousand in just the few years since its most recent review (4). Physics-based simulation (5), which we leverage in our open-source benchmark implementation (6, 7), has played an important role in the recent progress of RL; however, the transfer to real systems remains a research area in itself (8).

Previous review works have focused on specific techniques—for example, learning-based model predictive control (MPC) (9), iterative learning control (10, 11), model-based RL (12), data-efficient policy search (13), imitation learning (14), or the use of RL in robotics (15, 16) and in optimal control (17)—without emphasizing the safety aspect. Recent surveys on safe learning control have focused on either control-theoretic (18) or RL (19) approaches and do not provide a unifying perspective.

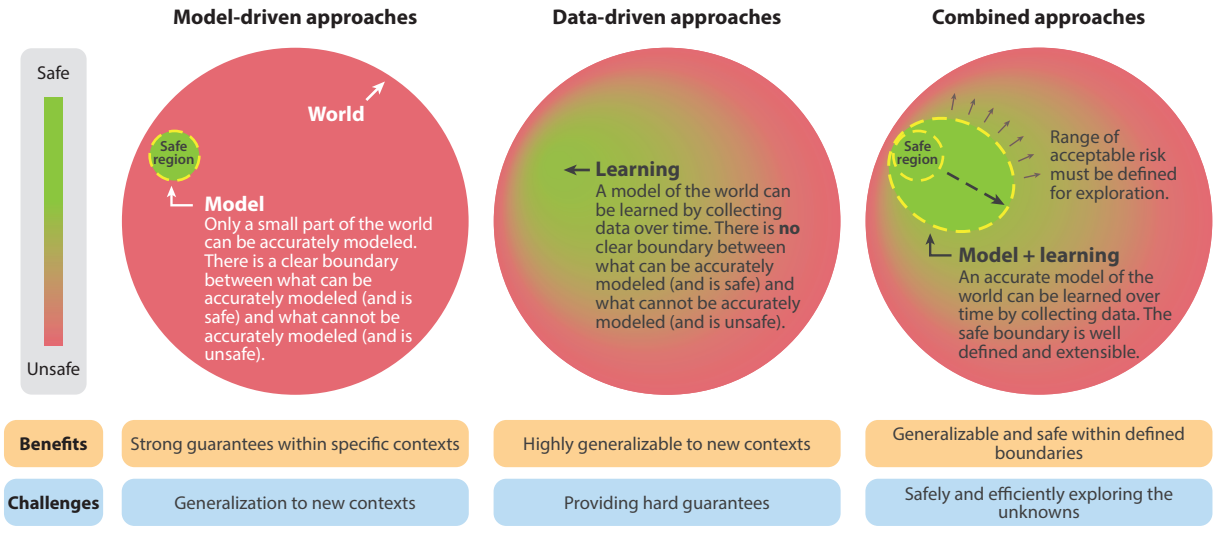

Figure 1

A comparison of model-driven, data-driven, and combined approaches.

In this article, we provide a bird's-eye view of the most recent work in learning-based control and RL that implement safety and provide safety guarantees for robot control. We focus on safe learning control approaches where the data generated by the robot system are used to learn or modify the feedback controller (or policy). We hope to help shrink the gap between the control and RL communities by creating a common vocabulary and introducing benchmarks for algorithm evaluation that can be leveraged by both (20, 21). Our target audience is researchers, with either a control or RL background, who are interested in a concise but holistic perspective on the problem of safe learning control. While we do not cover perception, estimation, planning, or multiagent systems, we do connect our discussion to these additional challenges and opportunities.

2. PRELIMINARIES AND BACKGROUND ON SAFE LEARNING CONTROL

In this review, we are interested in the problem of safe decision-making under uncertainties using machine learning (i.e., safe learning control). Intuitively, in safe learning control, our goal is to allow a robot to fulfill a task while respecting a set of safety constraints despite the uncertainties present in the problem. In this section, we define the safe learning control problem (Section 2.1) and provide an overview of how the problem of decision-making under uncertainties has traditionally been tackled by the control (Section 2.2) and RL (Section 2.3) communities. We highlight the main limitations of these approaches and articulate how novel data-based, safety-focused methods can address these gaps (Section 2.4).

2.1. Problem Statement

We formulate the safe learning control problem as an optimization problem to capture the efforts of both the control and RL communities. The optimization problem has three main components (see **Figure 2**): (*a*) a system model that describes the dynamic behavior of the robot, (*b*) a cost function that defines the control objective or task goal, and (*c*) a set of constraints that specify the safety requirements. The goal is to find a controller or policy that computes commands (also

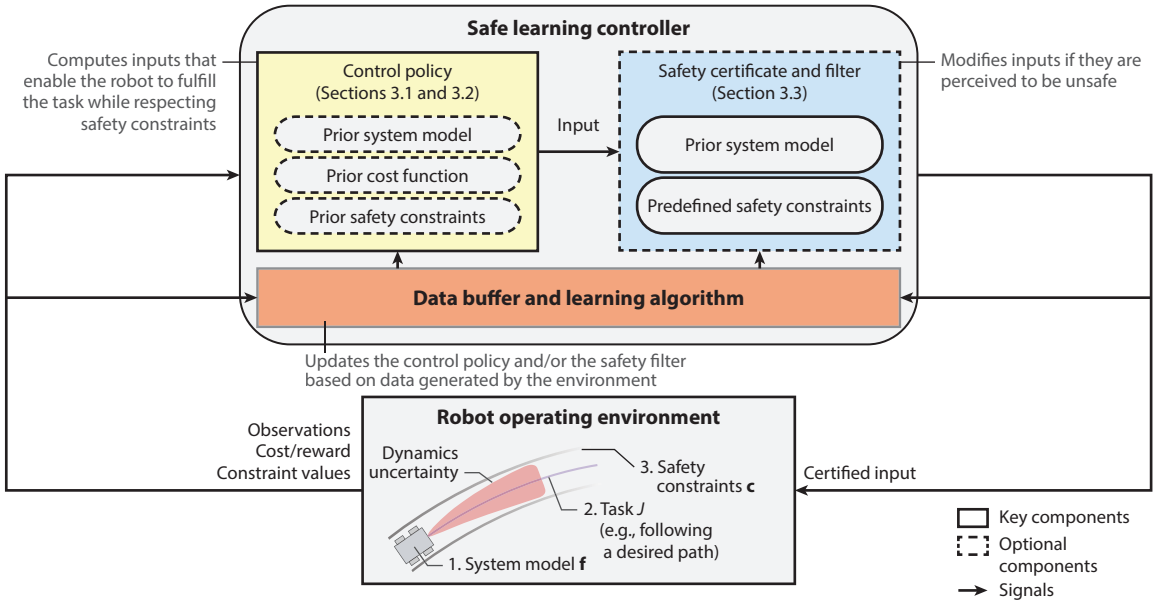

Figure 2

Block diagram representing the safe learning control approaches reviewed in this article. The three main components of the safe learning control problem are the cost function J, the system model $\mathbf{f}$, and the constraints $\mathbf{c}$, all of which may be initially unknown. Data are used to update the control policy (see Sections 3.1 and 3.2) or the safety filter (see Section 3.3).

called inputs) that enable the system to fulfill the task while respecting given safety constraints. In general, any of the components could be initially unknown or only partially known. Below, we first introduce each of the three components and then conclude by stating the overall safe learning control problem.

2.1.1. System model. We consider a robot whose dynamics can be represented by the following discrete-time model:

$$\mathbf{x}_{k+1} = \mathbf{f}_k(\mathbf{x}_k, \mathbf{u}_k, \mathbf{w}_k), \qquad 1.$$

where $k \in \mathbb{Z}_{\geq 0}$ is the discrete-time index; $\mathbf{x}_k \in \mathbb{X}$ is the state, with $\mathbb{X}$ denoting the state space; $\mathbf{u}_k \in \mathbb{U}$ is the input, with $\mathbb{U}$ denoting the input space; $\mathbf{f}_k$ is the dynamics model of the robot; and $\mathbf{w}_k \in \mathbb{W}$ is the process noise distributed according to a distribution $\mathcal{W}$. Equation 1 is analogous to the transition function in RL. Throughout this review, we assume direct access to (possibly noisy) measurements of the robot state $\mathbf{x}_k$ and neglect the problem of state estimation. Equation 1 represents many common robot platforms (e.g., quadrotors, manipulators, and ground vehicles). More complex models (e.g., partial differential equations) may be necessary for other robot designs.

Transition function: a transition probability model $T_k(\mathbf{x}_{k+1} \mid \mathbf{x}_k, \mathbf{u}_k)$ that is commonly used in RL as an alternative representation of the system model

2.1.2. Cost function. The robot's task is defined by a cost function. We consider a finite-horizon optimal control problem with time horizon N. Given an initial state $\mathbf{x}_0$, the cost is computed based on the sequence of states $\mathbf{x}_{0:N} = \{\mathbf{x}_0, \mathbf{x}_1, \ldots, \mathbf{x}_N\}$ and the sequence of inputs $\mathbf{u}_{0:N-1} = \{\mathbf{u}_0, \mathbf{u}_1, \ldots, \mathbf{u}_{N-1}\}$:

$$J(\mathbf{x}_{0:N}, \mathbf{u}_{0:N-1}) = l_N(\mathbf{x}_N) + \sum_{k=0}^{N-1} l_k(\mathbf{x}_k, \mathbf{u}_k), \qquad 2.$$

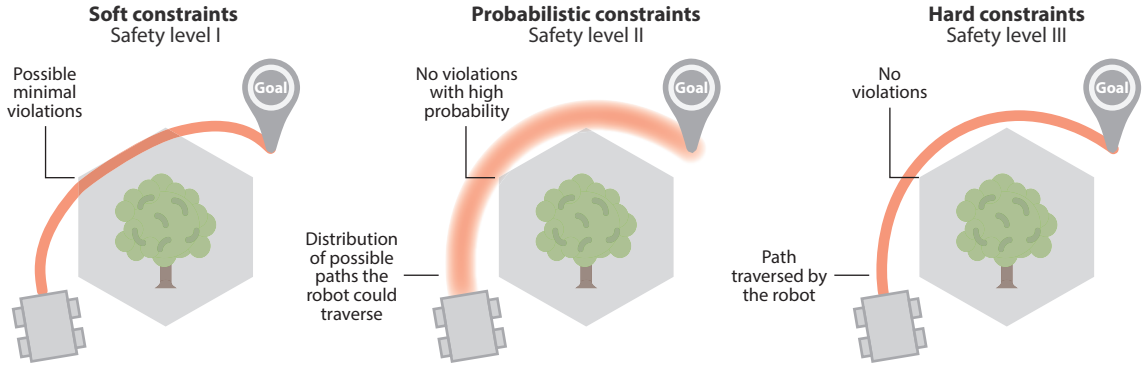

Figure 3

The three safety levels.

where $l_k : \mathbb{X} \times \mathbb{U} \mapsto \mathbb{R}$ is the stage cost incurred at each time step k (analogous to discounted rewards in RL), and $l_N : \mathbb{X} \mapsto \mathbb{R}$ is the terminal cost incurred at the end of the N-step horizon. The stage and terminal cost functions map the state and input sequences, which may be random variables, to a real number, and may, for example, include the expected value or variance operators.

2.1.3. Safety constraints. Safety constraints ensure, or encourage, the safe operation of the robot and include (*a*) state constraints $\mathbb{X}_c \subseteq \mathbb{X}$, which define the set of safe operating states (e.g., the lane in self-driving); (*b*) input constraints $\mathbb{U}_c \subseteq \mathbb{U}$ (e.g., actuation limits); and (*c*) stability guarantees (e.g., the robot's motion converging to a desired path) (22). To encode the safety constraints, we define n_c constraint functions: $\mathbf{c}_k(\mathbf{x}_k, \mathbf{u}_k, \mathbf{w}_k) \in \mathbb{R}^{n_c}$, with each constraint c_k^j being a real-valued, time-varying function. Starting with the strongest guarantee, we introduce three levels of safety: hard, probabilistic, and soft constraints (illustrated in **Figure 3**). In practice, safety levels are often mixed. For example, input constraints are typically hard constraints, but state constraints may be soft constraints (e.g., encouraging small acceleration values). Probabilistic constraints guarantee a constraint with high probability and are often the best we can achieve with a learning approach.

2.1.3.1. Safety level III: constraint satisfaction guaranteed. The system satisfies hard constraints:

$$c_k^j(\mathbf{x}_k, \mathbf{u}_k, \mathbf{w}_k) \leq 0 \qquad 3.$$

for all times $k \in \{0, \ldots, N\}$ and constraint indexes $j \in \{1, \ldots, n_c\}$.

2.1.3.2. Safety level II: constraint satisfaction with high probability. The system satisfies probabilistic constraints:

$$\Pr\left(c_k^j(\mathbf{x}_k, \mathbf{u}_k, \mathbf{w}_k) \leq 0\right) \geq p^j, \qquad 4.$$

where $\Pr(\cdot)$ denotes the probability and $p^j \in (0, 1)$ defines the likelihood of the jth constraint being satisfied, for all times $k \in \{0, \ldots, N\}$ and $j \in \{1, \ldots, n_c\}$. The chance constraint in Equation 4 is identical to the hard constraint in Equation 3 for $p^j = 1$.

2.1.3.3. Safety level I: constraint satisfaction encouraged. The system encourages constraint satisfaction. This can be achieved in different ways. One is to add a penalty term to the objective function that discourages the violation of constraints with a high cost. A nonnegative ϵ_j is added

Discounted reward: the typical RL stage cost expressed as a reward r_k discounted by $\gamma \in [0, 1]$: $l_k = -\gamma^k r_k(\mathbf{x}_k, \mathbf{u}_k)$

Stability: the boundedness of the system output or state (e.g., asymptotic stability in the sense of Lyapunov requires the system to converge to a desired state)

to the right-hand side of the inequality shown in Equation 3, for all times $k \in \{0, \dots, N\}$ and $j \in \{1, \dots, n_c\}$,

$$c_k^j(\mathbf{x}_k, \mathbf{u}_k, \mathbf{w}_k) \leq \epsilon_j, \tag{5.}$$

and an appropriate penalty term $l_\epsilon(\boldsymbol{\epsilon}) \geq 0$, with $l_\epsilon(\boldsymbol{\epsilon}) = 0 \iff \boldsymbol{\epsilon} = \mathbf{0}$, is added to the objective function. The vector $\boldsymbol{\epsilon}$ includes all elements ϵ_j and is an additional variable of the optimization problem. Alternatively, although $c_k^j(\mathbf{x}_k, \mathbf{u}_k, \mathbf{w}_k)$ is a stepwise quantity, some approaches aim to provide guarantees on its expected value $E[\cdot]$ only on a trajectory level:

$$J_{cj} = E\left[\sum_{k=0}^{N-1} c_k^j(\mathbf{x}_k, \mathbf{u}_k, \mathbf{w}_k)\right] \leq d_j, \tag{6.}$$

where J_{cj} represents the expected total constraint cost, and d_j defines the constraint threshold. The constraint function can also be discounted as $\gamma^k c_k^j(\mathbf{x}_k, \mathbf{u}_k, \mathbf{w}_k)$, similar to the stage cost. Note that Equation 6 can represent a probabilistic constraint, for example, if c_k^j takes the form of an indicator function over some state–input set. Then, bounding the expectation translates to bounding the probability of entering this set. Equation 6 can also represent a stepwise hard constraint. For example, nonnegative c_k^j and $d_j = 0$ effectively reduce the bounded expectation to $c_k^j(\mathbf{x}_k, \mathbf{u}_k, \mathbf{w}_k) = 0$, or zero violation across all steps.

2.1.4. Formulation of the safe learning control problem. The functions introduced above—the system model $\mathbf{f}$, the constraints $\mathbf{c}$, and the cost function J—represent the true functions of the robot control problem. In practice, $\mathbf{f}$, $\mathbf{c}$, and J may be unknown or partially known. Without loss of generality, we assume that each of the true functions $\mathbf{f}$, $\mathbf{c}$, and J can be decomposed into a nominal component $\bar{(\cdot)}$, reflecting our prior knowledge, and an unknown component $\hat{(\cdot)}$, to be learned from data. For instance, the dynamics model $\mathbf{f}$ can be decomposed as

$$\mathbf{f}_k(\mathbf{x}_k, \mathbf{u}_k, \mathbf{w}_k) = \bar{\mathbf{f}}_k(\mathbf{x}_k, \mathbf{u}_k) + \hat{\mathbf{f}}_k(\mathbf{x}_k, \mathbf{u}_k, \mathbf{w}_k), \tag{7.}$$

where $\bar{\mathbf{f}}$ is the prior dynamics model and $\hat{\mathbf{f}}$ represents the uncertain dynamics.

Safe learning control leverages our prior knowledge $\mathcal{P} = \{\bar{\mathbf{f}}, \bar{\mathbf{c}}, \bar{J}\}$ and the data collected from the system $\mathcal{D} = \{\mathbf{x}^{(i)}, \mathbf{u}^{(i)}, \mathbf{c}^{(i)}, l^{(i)}\}_{i=0}^{i=D}$ to find a policy (or controller) $\boldsymbol{\pi}_k(\mathbf{x}_k)$ that achieves the given task while respecting all safety constraints:

$$\text{safe learning control} : (\mathcal{P}, \mathcal{D}) \mapsto \boldsymbol{\pi}_k, \tag{8.}$$

where $(\cdot)^{(i)}$ denotes a sample of a quantity $(\cdot)_k$ and D is the data set size. More specifically, we aim to find a policy $\boldsymbol{\pi}_k$ that best approximates the true optimal policy $\boldsymbol{\pi}_k^\star$, which is the solution to the following optimization problem:

$$J^{\boldsymbol{\pi}^\star}(\bar{\mathbf{x}}_0) = \min_{\boldsymbol{\pi}_{0:N-1}, \boldsymbol{\epsilon}} \quad J(\mathbf{x}_{0:N}, \mathbf{u}_{0:N-1}) + l_\epsilon(\boldsymbol{\epsilon}) \tag{9a.}$$

$$\text{subject to} \quad \mathbf{x}_{k+1} = \mathbf{f}_k(\mathbf{x}_k, \mathbf{u}_k, \mathbf{w}_k), \quad \mathbf{w}_k \sim \mathcal{W}, \quad \forall k \in \{0, \dots, N-1\}, \tag{9b.}$$

$$\mathbf{u}_k = \boldsymbol{\pi}_k(\mathbf{x}_k), \tag{9c.}$$

$$\mathbf{x}_0 = \bar{\mathbf{x}}_0, \tag{9d.}$$

$$\text{safety constraints according to either Equations 3–5 or Equation 6, and } \boldsymbol{\epsilon} \geq \mathbf{0}, \tag{9e.}$$

where $\bar{\mathbf{x}}_0 \sim \mathcal{X}_0$ is the initial state, with $\mathcal{X}_0$ being the initial state distribution, and $\boldsymbol{\epsilon}$ and l_ϵ are introduced to account for the soft safety constraint case (safety level I, Equation 5) and are set to

zero, for example, if only hard and probabilistic safety constraints are considered (safety levels II and III).

2.2. A Control Theory Perspective

Safe decision-making under uncertainty has a long history in the field of control theory. Typical assumptions are that a model of the system is available and that it either is parameterized by an unknown parameter or has bounded unknown dynamics and noise. While the control approaches are commonly formulated using continuous-time dynamic models, they are usually implemented in discrete time with sampled inputs and measurements.

Adaptive control typically considers systems modeled as parametric models with uncertain parameters and adapts the controller or model online to optimize performance. Adaptive control requires knowledge of the parametric form of the uncertainty (23) and typically considers a dynamics model that is affine in $\mathbf{u}$ and the uncertain parameters $\theta \in \Theta$:

$$\mathbf{x}_{k+1} = \bar{\mathbf{f}}_{\mathbf{x}}(\mathbf{x}_k) + \bar{\mathbf{f}}_{\mathbf{u}}(\mathbf{x}_k)\mathbf{u}_k + \bar{\mathbf{f}}_{\theta}(\mathbf{x}_k)\theta, \tag{10.}$$

where $\bar{\mathbf{f}}_{\mathbf{x}}$, $\bar{\mathbf{f}}_{\mathbf{u}}$, and $\bar{\mathbf{f}}_{\theta}$ are known functions often derived from first principles and Θ is a possibly bounded parameter set. The control input is $\mathbf{u}_k = \pi(\mathbf{x}_k, \hat{\theta}_k)$, which is parameterized by $\hat{\theta}_k$. The parameter $\hat{\theta}_k$ is adapted by using either a Lyapunov function to guarantee that the closed-loop system is stable or model reference adaptive control (MRAC) to make the system behave as a predefined stable reference model (23). Adaptive control is typically limited to parametric uncertainties and relies on a specific model structure. Moreover, adaptive control approaches tend to overfit to the latest observations, and convergence to the true parameters is generally not guaranteed (23, 24). These limitations motivate the learning-based adaptive control approaches in Section 3.1.1.

Robust control is a control design technique that guarantees stability for prespecified bounded disturbances, which can include unknown dynamics and noise. In contrast to adaptive control, which adapts to the parameters currently present, robust control finds a suitable controller for all possible disturbances and keeps the controller unchanged after the initial design. Robust control is limited largely to linear time-invariant systems with a linear nominal model $\bar{\mathbf{f}}(\mathbf{x}_k, \mathbf{u}_k) = \bar{\mathbf{A}}\mathbf{x}_k + \bar{\mathbf{B}}\mathbf{u}_k$ and unknown dynamics $\hat{\mathbf{f}}_k(\mathbf{x}_k, \mathbf{u}_k, \mathbf{w}_k) = \hat{\mathbf{A}}\mathbf{x}_k + \hat{\mathbf{B}}\mathbf{u}_k + \mathbf{w}_k \in \mathbb{D}$, with $\mathbb{D}$ being known and bounded and $\bar{\mathbf{A}}$, $\bar{\mathbf{B}}$, $\hat{\mathbf{A}}$, and $\hat{\mathbf{B}}$ being static matrices of appropriate size—that is,

$$\mathbf{x}_{k+1} = (\bar{\mathbf{A}} + \hat{\mathbf{A}})\mathbf{x}_k + (\bar{\mathbf{B}} + \hat{\mathbf{B}})\mathbf{u}_k + \mathbf{w}_k. \tag{11.}$$

Robust control design techniques, such as robust H_∞ and H_2 control design (25), yield controllers that are robustly stable for all $\hat{\mathbf{f}}_k \in \mathbb{D}$. Robust control can be extended to nonlinear systems whose dynamics can be decomposed into a linear nominal model $\bar{\mathbf{f}}$ and a nonlinear function $\hat{\mathbf{f}}$ with known bound $\hat{\mathbf{f}} \in \mathbb{D}$ (26).

Robust MPC extends classical adaptive and robust control by additionally guaranteeing state and input constraints, $\mathbf{x}_k \in \mathbb{X}_c$ and $\mathbf{u}_k \in \mathbb{U}_c$, for all possible bounded disturbances $\hat{\mathbf{f}} \in \mathbb{D}$. At every time step k, MPC solves a constrained optimization problem over a control input sequence $\mathbf{u}_{k:k+H-1}$ for a finite horizon H, and applies the first optimal control input to the system and then resolves the optimization problem in the next time step based on the current state (27). A common approach in robust MPC is tube-based MPC (28), which uses a nominal prediction model $\bar{\mathbf{f}}(\mathbf{x}_k, \mathbf{u}_k)$ in the MPC optimization and tightens the constraints to account for unmodeled dynamics. A stabilizing controller keeps the true state inside a bounded set of states around the nominal state, called a tube, for all possible disturbances. Since the nominal states satisfy the tightened constraints and the true states stay inside the tube around the nominal states, constraint satisfaction for the true states is guaranteed. Tube-based MPC typically considers a

Parametric model: a model that depends on a finite number of parameters that typically reflect our prior knowledge about the system structure

Lyapunov function: a positive definite function $L : \mathbb{X} \to \mathbb{R}_{\geq 0}$ used to prove the stability of a dynamical system (with respect to an equilibrium)

linear nominal model $\bar{\mathbf{f}}(\bar{\mathbf{x}}_k, \bar{\mathbf{u}}_k) = \bar{\mathbf{A}}\bar{\mathbf{x}}_k + \bar{\mathbf{B}}\bar{\mathbf{u}}_k$ with nominal state $\bar{\mathbf{x}}_k$ and input $\bar{\mathbf{u}}_k$. In its simplest implementation, prior knowledge of set $\mathbb{D}$ is combined with a known stabilizing linear controller, $\mathbf{u}_{k,\text{stab}} = \mathbf{K}(\mathbf{x}_k - \bar{\mathbf{x}}_k)$ with gain $\mathbf{K}$, to determine the bounded tube Ω_{tube} from the matrices $\bar{\mathbf{A}}$, $\bar{\mathbf{B}}$, and $\mathbf{K}$ and the set $\mathbb{D}$. The stabilizing controller $\mathbf{u}_{k,\text{stab}}$ keeps all potential errors within the tube, $\mathbf{x}_k - \bar{\mathbf{x}}_k \in \Omega_{\text{tube}}$, for all k. For the nominal model, tube-based MPC solves the following constrained optimization problem at every time step k to obtain the optimal sequence $\bar{\mathbf{u}}^*_{0:H-1}$:

$$J^*_{\text{RMPC}}(\bar{\mathbf{x}}_k) = \min_{\bar{\mathbf{u}}_{0:H-1}} \quad l_H(\mathbf{z}_H) + \sum_{i=0}^{H-1} l_i(\mathbf{z}_i, \bar{\mathbf{u}}_i) \tag{12a.}$$

$$\text{subject to} \quad \mathbf{z}_{i+1} = \bar{\mathbf{A}}\mathbf{z}_i + \bar{\mathbf{B}}\bar{\mathbf{u}}_i, \quad \forall i \in \{0, \dots, H-1\}, \tag{12b.}$$

$$\mathbf{z}_i \in \mathbb{X}_c \ominus \Omega_{\text{tube}}, \quad \bar{\mathbf{u}}_i \in \mathbb{U}_c \ominus \mathbf{K}\Omega_{\text{tube}}, \tag{12c.}$$

$$\mathbf{z}_0 = \bar{\mathbf{x}}_k, \quad \mathbf{z}_H \in \mathbb{X}_{\text{term}}, \tag{12d.}$$

where $\mathbf{z}_i$ is the open-loop nominal state at time step $k + i$, and $\mathbb{X}_c \ominus \Omega_{\text{tube}} = \{\mathbf{x} \in \mathbb{X} : \mathbf{x} + \boldsymbol{\omega} \in \mathbb{X}_c, \forall \boldsymbol{\omega} \in \Omega_{\text{tube}}\}$ and $\mathbb{U}_c \ominus \mathbf{K}\Omega_{\text{tube}} = \{\mathbf{u} \in \mathbb{U} : \mathbf{u} + \mathbf{K}\boldsymbol{\omega} \in \mathbb{U}_c, \forall \boldsymbol{\omega} \in \Omega_{\text{tube}}\}$ are, respectively, the state and input constraints tightened using the bounded tube Ω_{tube}, with $\ominus$ denoting the Pontryagin difference. Combined with the stabilizing control input $\mathbf{u}_{k,\text{stab}}$, the control input $\mathbf{u}_k = \mathbf{u}_{k,\text{stab}} + \bar{\mathbf{u}}^*_0$ is applied to the system at every time step. Stability and the satisfaction of the tightened constraints are guaranteed by selecting the terminal cost l_H in Equation 12a such that after the prediction horizon H the nominal state $\mathbf{z}_H$ is within a terminal constraint set $\mathbb{X}_{\text{term}}$ (see Equation 12d), at which point a known linear controller can be safely applied (27).

Both robust control and robust MPC are conservative, as they guarantee stability and—in the case of MPC—state and input constraints for the worst-case scenario. This usually yields poor performance (9). For example, a conservative uncertainty set $\mathbb{D}$ generates a large tube Ω_{tube}, resulting in tight hard constraints that are prioritized over cost optimization. Learning-based robust control and robust MPC improve performance by using data to (*a*) learn a less conservative state- and input-dependent uncertainty set $\mathbb{D}$ and/or (*b*) learn the unknown dynamics $\hat{\mathbf{f}}$ and, as a result, reduce the remaining model uncertainty (see Sections 3.1.2 and 3.1.3, respectively).

2.3. A Reinforcement Learning Perspective

RL is the standard machine learning framework to address the problem of sequential decision-making under uncertainty. Unlike traditional control, RL generally does not rely on an a priori dynamics model $\bar{\mathbf{f}}$ and can be directly applied to uncertain dynamics $\mathbf{f}$. However, the lack of explicit assumptions and constraints in many of the works limits their applicability to safe control. RL algorithms attempt to find $\boldsymbol{\pi}^*$ while gathering data and knowledge of $\mathbf{f}$ from interaction with the system—initially taking random actions and then improving afterward. A long-standing challenge of RL, which hampers safety during the learning stages, is the exploration–exploitation dilemma—that is, whether to (*a*) act greedily with the available data or (*b*) explore, which means taking suboptimal (and possibly unsafe) actions $\mathbf{u}$ to learn a more accurate $\hat{\mathbf{f}}$.

RL typically assumes that the underlying control problem is a Markov decision process (MDP). An MDP comprises a state space $\mathbb{X}$, an input (action) space $\mathbb{U}$, stochastic dynamics (also called a transition model), and a per-step reward function. When all the components of an MDP are known (in particular, $\mathbf{f}$ and J from Section 2.1), then it solves the problem in Equations 9a–9c without the constraints. Dynamic programming algorithms such as value and policy iteration can be used to find an optimal policy $\boldsymbol{\pi}^*$. Many RL approaches, however, make no assumptions on any part of $\mathbf{f}$ being known a priori.

We can distinguish model-based RL approaches, which learn an explicit model $\hat{\mathbf{f}}$ of the system dynamics $\mathbf{f}$ and use it to optimize a policy, from model-free RL algorithms. The latter algorithms (29) can be broadly categorized as (*a*) value function–based methods, learning an action-value function; (*b*) policy-search and policy-gradient methods, directly trying to find an optimal policy $\boldsymbol{\pi}^*$; and (*c*) actor–critic methods, learning both a value function (critic) and the policy (actor). We also note that the convergence of these methods has been shown for simple scenarios but is still a challenge for more complex scenarios or when function approximators are used (30).

There are multiple practical hurdles to the deployment of RL algorithms in real-world robotics problems (8). These challenges include (*a*) the continuous, possibly high-dimensional $\mathbb{X}$ and $\mathbb{U}$ in robotics (often assumed to be finite, discrete sets in RL); (*b*) the stability and convergence of the learning algorithm (31) (necessary, albeit not sufficient, to produce a stable policy); (*c*) learning robust policies from limited samples; (*d*) the interpretability of the learned policy, especially in deep RL when leveraging neural networks for function approximation (29); and, importantly, (*e*) providing provable safety guarantees.

The exploration–exploitation dilemma can be mitigated using Bayesian inference. This is achieved by computing posterior distributions over the states $\mathbb{X}$, possible dynamics $\mathbf{f}$, or total cost J from past observed data (32). These posteriors provide explicit knowledge of the problem's uncertainty and can be used to guide exploration. In practice, however, full posterior inference is almost always prohibitively expensive, and concrete implementations must rely on approximations.

To achieve constraint satisfaction (over states or sequences of states) and robustness to different, possibly noisy dynamics $\mathbf{f}$, constrained MDPs (CMDPs) and robust MDPs are extensions of traditional MDPs that more closely resemble the problem statement in Section 2.1.

CMDPs (33) extend simple MDPs with constraints and optimize the problem in Equation 9 when Equation 9e takes the discounted form of safety level I's Equation 6. We refer to the discounted constraint cost J_{c^j} under policy $\boldsymbol{\pi}$ as $J^{\boldsymbol{\pi}}_{c^j}$. Traditional approaches to solve CMDPs, such as linear programming and Lagrangian methods, often assume discrete state–action spaces and cannot properly scale to complex tasks such as robot control. Deep RL promises to mitigate this problem, yet applying it to the constrained problem still suffers from the computational complexity of the off-policy evaluation (not to be confused with offline RL) of trajectory-level constraints J_{c^j} (34). In Section 3.2.3, we present some recent advances in CMDP-based work that feature (*a*) integration of deep learning techniques in CMDPs for more complex control tasks, (*b*) provable constraint satisfaction throughout the exploration or learning process, and (*c*) constraint transformation for the efficient evaluation of J_{c^j} from data collected off-policy.

Robust MDPs (35), inspired by robust control (see Section 2.2), extend MDPs such that the dynamics can include parametric uncertainties or disturbances, and the cost of the worst-case scenario is optimized. This is captured by the min–max optimization problem:

$$J^{\boldsymbol{\pi}}(\bar{\mathbf{x}}_0) = \min_{\boldsymbol{\pi}_{0:N-1}} \max_{\hat{\mathbf{f}} \in \mathbb{D}} \quad J(\mathbf{x}_{0:N}, \mathbf{u}_{0:N-1}) \tag{13a}$$

$$\text{subject to} \quad \text{Equations 9b–9d}, \tag{13b}$$

where $\mathbb{D}$ is a given uncertainty set of $\hat{\mathbf{f}}$. To keep solutions tractable, practical implementations typically restrict $\mathbb{D}$ to certain classes of models. This can limit the applicability of robust MDPs beyond toy problems. Recent work (36–38) applied deep RL to robust decision-making, targeting key theoretical and practical hurdles such as how to effectively model uncertainty with deep neural networks (DNNs) and how to efficiently solve the min–max optimization (e.g., via sampling or two-player, game-theoretic formulations). These ideas, including adversarial RL and domain randomization, are presented in Section 3.2.4.

Value function: under policy $\boldsymbol{\pi}_k$, a function of state $\mathbf{x}_k$, $V^{\boldsymbol{\pi}_k}(\mathbf{x}_k)$, equal to the expected return J of applying $\boldsymbol{\pi}_k$ from $\mathbf{x}_k$

Action-value function: under $\boldsymbol{\pi}_k$, a function of action $\mathbf{u}_k$ and state $\mathbf{x}_k$, $Q^{\boldsymbol{\pi}_k}(\mathbf{x}_k, \mathbf{u}_k)$, equal to the expected return J when taking the action $\mathbf{u}_k$ at $\mathbf{x}_k$ and then following $\boldsymbol{\pi}_k$

Neural network: a computational model with interconnected layers of neurons (parameterized by weights) that can be used to approximate highly nonlinear functions

Off-policy evaluation: in RL, improving the value function estimates with data collected operating under different policies; on-policy updates, by contrast, use only data generated by the current policy

Offline RL: RL approaches that use data previously collected by a (possibly unknown) policy but do not interact with the environment until deployed

2.4. Bridging Control Theory and Reinforcement Learning for Safe Learning Control

Gaussian process (GP): a probabilistic model specifying a distribution over functions

When designing a learning-based controller, we typically have two sources of information: our prior knowledge and data generated by the robot system. Control approaches rely on prior knowledge and on assumptions such as parametric dynamics models to provide safety guarantees. RL approaches typically make use of expressive learning models to extract patterns from data that facilitate learning complex tasks, but these can impede the provision of any formal guarantees. In recent literature, we see an effort from both the control and RL communities to develop safe learning control algorithms, with the goal of systematically leveraging expressive models for closed-loop control (see **Figure 4**). Questions that arise from these efforts include how control-theoretic tools can be applied to expressive machine learning models and how expressive models can be incorporated into control frameworks.

Expressive learning models can be categorized as deterministic (e.g., standard DNNs) or probabilistic [e.g., Gaussian processes (GPs) and Bayesian linear regression]. Deep learning techniques such as feedforward neural networks, convolutional neural networks, and long short-term memory networks have the advantage of being able to abstract large volumes of data, enabling real-time execution in a control loop. On the other hand, their probabilistic counterparts, such as GPs and Bayesian linear regression, provide model output uncertainty estimates that can be naturally blended into traditional adaptive and robust control frameworks. We note that there are approaches aiming to combine the advantages of the two types of learning (e.g., Bayesian neural networks), and quantifying uncertainty in deep learning is still an active research direction.

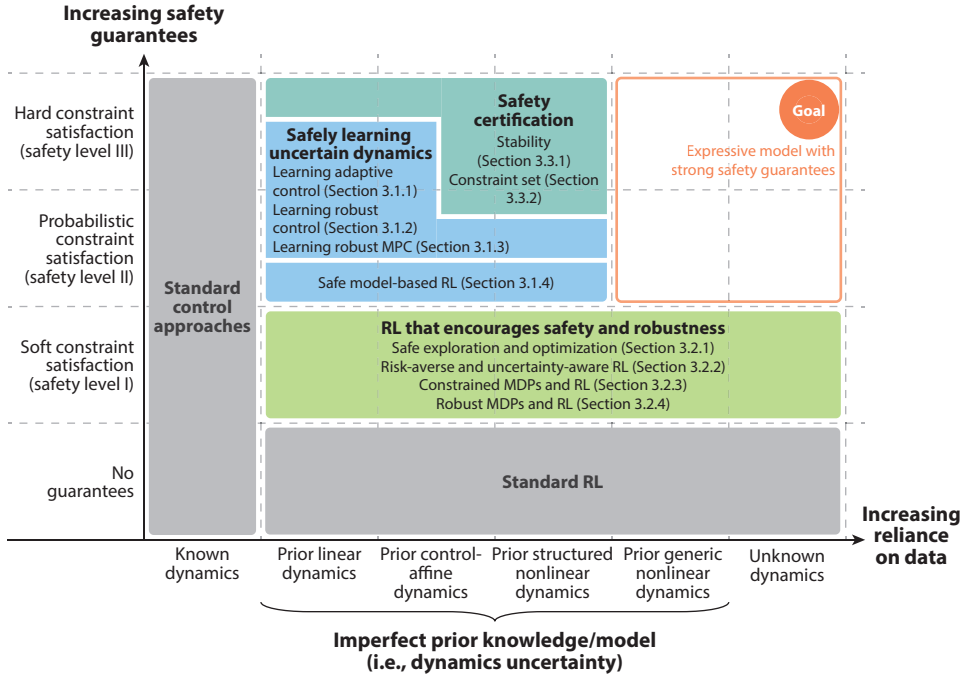

Figure 4

Summary of the safe learning control approaches reviewed in Section 3. Abbreviations: MDP, Markov decision process; MPC, model predictive control; RL, reinforcement learning.

In this review, we focus on approaches that address the problem of safe learning control at two stages: (*a*) online adaptation or learning, where online data are used to adjust the parameters of the controller, the robot dynamics model, the cost function, or the constraint functions during closed-loop operation, and (*b*) offline learning, where data collected from each trial are recorded and used to update a model in a batch manner in between trials of closed-loop operation. In safe learning control, data are generally used to address the issue of uncertainties in the problem formulation and reduce the conservatism in the system design, while the safety aspect boils down to knowing what is unknown and cautiously accounting for the incomplete knowledge via algorithm design.

Lipschitz continuity: a function $\mathbf{h} : \mathbb{A} \mapsto \mathbb{B}$ for which a bounded change in input yields a bounded change in output

3. SAFE LEARNING CONTROL APPROACHES

The ability to guarantee safe robot control is inevitably dependent on the amount of prior knowledge available and the types of uncertainties present in the problem of interest. In this section, we discuss approaches for safe learning control in robotics based on the following three categories (see **Figures 2** and **4**):

- Learning uncertain dynamics to safely improve performance: These works rely on an a priori model of the robot dynamics. The robot's performance is improved by learning the uncertain dynamics from data. Safety is typically guaranteed based on standard control-theoretic frameworks, achieving safety level II or III.
- Encouraging safety and robustness in RL: These works encompass approaches that usually do not have knowledge of an a priori robot model or the safety constraints. Rather than providing hard safety guarantees, these approaches encourage safe robot operation (safety level I), for example, by penalizing dangerous actions.
- Certifying learning-based control under dynamics uncertainty: These works aim to provide safety certificates for learning-based controllers that do not inherently consider safety constraints. These approaches modify the learning controller output by constraining the control policy, leveraging a known safe backup controller, or modifying the controller output directly to achieve stability and/or constraint satisfaction. They typically achieve safety level II or III.

Figure 4 categorizes the approaches reviewed in this section based on their safety level and reliance on data. A more detailed summary of the approaches can be found in **Supplemental Table 1** in the **Supplemental Material**.

Supplemental Material >

3.1. Learning Uncertain Dynamics to Safely Improve Performance

In this section, we consider approaches that improve the robot's performance by learning the uncertain dynamics and provide safety guarantees (safety level II or III) via control frameworks such as adaptive control, robust control, and robust MPC (outlined in Section 2.2). These approaches make assumptions about the unknown parts of the problem (e.g., Lipschitz continuity) and often rely on a known model structure (e.g., a control-affine or linear system with bounded uncertainty) to prove stability and/or constraint satisfaction (see **Figure 4**).

3.1.1. Integrating machine learning and adaptive control. There are three main ideas to incorporating online machine learning into traditional adaptive control (see Section 2.2), each with its own distinct benefits: (*a*) using black-box machine learning models to accommodate nonparametric unknown dynamics; (*b*) using probabilistic learning and explicitly accounting for the learned model uncertainties, in order to achieve cautious adaptation; and (*c*) augmenting adaptive

control with deep learning approaches for experience memorization in order to minimize the need for readaptation.

3.1.1.1. Learning nonparametric unknown dynamics with machine learning models. One goal of integrating adaptive control and machine learning is to improve the performance of a robot subject to nonparametric dynamics uncertainties. Instead of using Equation 10, we consider a system with nominal dynamics $\bar{\mathbf{f}}(\mathbf{x}_k, \mathbf{u}_k) = \bar{\mathbf{A}}\mathbf{x}_k + \bar{\mathbf{B}}\mathbf{u}_k$ and unknown dynamics $\hat{\mathbf{f}}(\mathbf{x}_k, \mathbf{u}_k) = \bar{\mathbf{B}}\,\psi_k(\mathbf{x}_k)$, where $\psi_k(\mathbf{x}_k)$ is an unknown nonlinear function without an obvious parametric structure. Learning-based MRAC approaches (39, 40) make the uncertain system behave as the linear nominal model $\bar{\mathbf{f}}$ by using a combination of $\mathcal{L}_1$ adaptation (41) and online learning to approximate $\psi_k(\mathbf{x}_k)$ by $\hat{\psi}_k(\mathbf{x}_k) = \hat{\psi}_{\mathcal{L}_1,k} + \hat{\psi}_{\text{learn},k}(\mathbf{x}_k)$, where $\hat{\psi}_{\mathcal{L}_1,k}$ is the input computed by the $\mathcal{L}_1$ adaptive controller and $\hat{\psi}_{\text{learn},k}$ is the input computed by the learning module, which can be a neural network (39) or a GP (40). The estimated $\hat{\psi}_k(\mathbf{x}_k)$ is then used in the controller $\boldsymbol{\pi}_k(\mathbf{x}_k)$ to account for the unknown nonlinear dynamics $\psi_k(\mathbf{x}_k)$, improving a linear nominal control policy designed based on $\bar{\mathbf{f}}$. In these approaches, fast adaptation and stability guarantees are provided by the $\mathcal{L}_1$ adaptation framework (safety level III), while the learning module provides additional flexibility to capture the unknown dynamics. The addition of learning improves the performance of the standard $\mathcal{L}_1$ adaptive controller and allows fast adaptation to be achieved at a lower sampling rate (39).

3.1.1.2. Cautious adaptation with probabilistic model learning. Another set of adaptive control approaches leverage probabilistic models to achieve cautious adaptation by weighting the contribution of the learned model based on the model output uncertainty. We consider a system with nominal dynamics $\bar{\mathbf{f}}(\mathbf{x}_k, \mathbf{u}_k) = \bar{\mathbf{f}}_{\mathbf{x}}(\mathbf{x}_k) + \bar{\mathbf{f}}_{\mathbf{u}}(\mathbf{x}_k)\mathbf{u}_k$ and unknown dynamics of the same form, $\hat{\mathbf{f}}(\mathbf{x}_k, \mathbf{u}_k) = \hat{\mathbf{f}}_{\mathbf{x}}(\mathbf{x}_k) + \hat{\mathbf{f}}_{\mathbf{u}}(\mathbf{x}_k)\mathbf{u}_k$, where $\hat{\mathbf{f}}_{\mathbf{x}}(\mathbf{x}_k)$ and $\hat{\mathbf{f}}_{\mathbf{u}}(\mathbf{x}_k)$ are unknown nonparametric nonlinear functions. In a model inversion–based MRAC framework, an approximate feedback linearization is achieved via the nominal model to facilitate the design of MRAC, and a GP-based adaptation approach is used to compensate for feedback linearization errors due to the unknown dynamics (42). To account for the uncertainty in the GP model learning, the controller relies on the GP model only if the confidence in the latter is high: $\boldsymbol{\pi}_k(\mathbf{x}_k) = \boldsymbol{\pi}_{\text{nom}}(\mathbf{x}_k) - \gamma(\mathbf{x}_k, \mathbf{u}_k)\,\boldsymbol{\pi}_{\text{learn},k}(\mathbf{x}_k)$, where $\boldsymbol{\pi}_{\text{nom}}(\mathbf{x}_k)$ is the control policy designed based the nominal model, $\boldsymbol{\pi}_{\text{learn},k}(\mathbf{x}_k)$ is the adaptive component designed based on the GP model, and $\gamma(\mathbf{x}_k, \mathbf{u}_k) \in [0, 1]$ is a scaling factor, with 0 indicating low confidence in the GP. The stability of the overall system (safety level III) is guaranteed via a stochastic stability analysis, and the efficacy of the approach has been demonstrated in quadrotor experiments (42, 43).

3.1.1.3. Memorizing experience with deep architectures. Apart from compensating for nonlinear and nonparametric dynamics uncertainties, deep learning approaches have also been applied to adaptive control for memorizing generalizable feature functions as the system adapts. In particular, References 44 and 45 proposed an asynchronous DNN adaptation approach. Similarly to References 39 and 40, these works consider a linear nominal model $\bar{\mathbf{f}}(\mathbf{x}_k, \mathbf{u}_k) = \bar{\mathbf{A}}\mathbf{x}_k + \bar{\mathbf{B}}\mathbf{u}_k$ and unknown nonlinear dynamics $\hat{\mathbf{f}}(\mathbf{x}_k, \mathbf{u}_k) = \bar{\mathbf{B}}\,\psi_k(\mathbf{x}_k)$, with $\psi_k(\mathbf{x}_k)$ being an unknown nonlinear function. In the proposed approach, the last layer of the DNN is updated at a higher frequency for fast adaptation, while the inner layers are updated at a lower frequency to memorize pertinent features for the particular operation regimes. To provide safety guarantees, References 44 and 45 derived an upper bound on the sampling complexity of the DNN to achieve a prescribed level of modeling error and leveraged this result to show that Lyapunov stability of the adapted system can be guaranteed (safety level III) by ensuring that the modeling error of the DNN is lower than a

given bound. In contrast to other MRAC approaches, which usually do not retain a memory of the past experience, the inner layers of the asynchronous DNN store relevant features that facilitate adaptation when similar scenarios arise in the future.

3.1.2. Learning-based robust control. Learning-based robust control improves the performance of classical robust control (described in Section 2.2) by using data to improve the linear dynamics model and reduce the uncertainty in Equation 11.

3.1.2.1. Using a Gaussian process dynamics model for linear robust control. The conservative performance of robust control (described in Section 2.2) is improved by updating the linear dynamics model and uncertainty in Equation 11 with a GP (46). The unknown nonlinear dynamics $\hat{\mathbf{f}}(\mathbf{x}_k, \mathbf{u}_k)$ are learned as a GP, which is then linearized about an operating point. Linearizing the GP (as opposed to directly fitting a linear model) allows data close to the operating point to be prioritized. The uncertain linear dynamics in Equation 11 are assumed to be modeled as $\hat{\mathbf{A}} = \mathbf{A}_0 + \tilde{\mathbf{A}}\boldsymbol{\Delta}$ and $\hat{\mathbf{B}} = \mathbf{B}_0 + \tilde{\mathbf{B}}\boldsymbol{\Delta}$, where $\mathbf{A}_0$ and $\mathbf{B}_0$ are obtained from the linearized GP mean, $\tilde{\mathbf{A}}$ and $\tilde{\mathbf{B}}$ are obtained from the linearized GP variance (often two standard deviations), and $\boldsymbol{\Delta}$ represents a matrix with elements taking any value in the range of $[-1, +1]$. Further performance improvement is achieved by modeling $\tilde{\mathbf{A}}$ and $\tilde{\mathbf{B}}$ as state dependent (47). Additionally, Reference 48 achieved better performance than Reference 46 by leveraging the GP's distribution, while maintaining the same level of safety. The main advantage of these approaches is that they can robustly guarantee safety level II stability while improving performance, which is achieved by shrinking the GP uncertainty as more data are added, thus improving the linear model $\mathbf{A}_0$ and $\mathbf{B}_0$ and reducing the uncertain component $\tilde{\mathbf{A}}$ and $\tilde{\mathbf{B}}$. This approach has been shown on a quadrotor (46). However, these methods are limited to stabilization tasks and do not account for state and input constraints.

3.1.2.2. Exploiting feedback linearization for robust learning-based tracking. Trajectory tracking convergence, as opposed to the simpler stabilization task performed in Reference 46, is guaranteed by exploiting the special structure of exactly feedback linearizable systems (49). This structure assumes that the nonlinear system dynamics in Equation 1 can be described by a linear nominal model, where $\bar{\mathbf{A}}$ and $\bar{\mathbf{B}}$ have an integrator chain structure. It also assumes that the unknown dynamics are $\hat{\mathbf{f}}(\mathbf{x}_k, \mathbf{u}_k) = \bar{\mathbf{B}}\psi(\mathbf{x}_k, \mathbf{u}_k)$, where $\psi(\mathbf{x}_k, \mathbf{u}_k)$ is an unknown invertible function. A probabilistic upper bound is obtained for $\psi(\mathbf{x}_k, \mathbf{u}_k)$ by learning this function as a GP. A robust linear controller is designed for the uncertain system based on this learned probabilistic bound. The performance is further improved by also updating the feedback linearization (50) through improvements in the estimate of the inverse $\psi^{-1}(\cdot)$. These approaches have been applied to trajectory tracking, with safety level II, on Lagrangian mobile manipulators (49) and quadrotor models (50). However, they hinge on this special structure and cannot account for state and input constraints.

3.1.3. Reducing conservatism in robust model predictive control with learning and adaptation. The conservative nature of robust MPC (Section 2.2) is improved—while still satisfying input and state constraints—through (*a*) robust adaptive MPC, which adapts to parametric uncertainty, and (*b*) learning-based robust MPC, which learns the unknown dynamics $\hat{\mathbf{f}}$ or, in one case, cost $\hat{J}$.

3.1.3.1. Robust adaptive model predictive control. Robust adaptive MPC assumes parametric uncertainties, and either uses data to reduce the set of possible parameters over time or uses an inner-loop adaptive controller and an outer-loop robust model predictive controller. This leads

to improved performance compared with standard robust MPC (see Section 2.2) while satisfying hard constraints (safety level III).

The first set of approaches consider stabilization tasks, where the full system dynamics (Equation 10) are assumed to be linear and stable (51) or linear and unstable (52) with uncertain parameter $\theta \in \Theta_0$:

$$\mathbf{x}_{k+1} = (\bar{\mathbf{A}} + \hat{\mathbf{A}}(\theta))\mathbf{x}_k + (\bar{\mathbf{B}} + \hat{\mathbf{B}}(\theta))\mathbf{u}_k + \mathbf{w}_k. \quad 14.$$

The process noise and the parameters are assumed to be bounded by known sets $\mathbb{W}$ and the initially conservative Θ_0, respectively. Given $\mathbb{W}$ and Θ_0, we can derive a conservative upper bound on the uncertain dynamics $\hat{\mathbf{f}}(\mathbf{x}_k, \mathbf{u}_k, \mathbf{w}_k, \theta) = \hat{\mathbf{A}}(\theta)\mathbf{x}_k + \hat{\mathbf{B}}(\theta)\mathbf{u}_k + \mathbf{w}_k \in \mathbb{D}_0$, where $\mathbb{D}_0$ is a compact set determined from $\mathbb{W}$ and Θ_0. To guarantee constraint satisfaction, tube-based MPC (see Section 2.2) is applied, where the initial tube $\Omega_{\text{tube},0}$ is based on $\mathbb{D}_0$. To reduce the conservatism of the approach, an adaptive control method is introduced to improve the estimate of the parameter set Θ_k and reduce the size of the tube $\Omega_{\text{tube},k}$ at each time step k. This idea has been extended to stochastic process noise for probabilistic constraint satisfaction (safety level II) (53), time-varying parameters (54), and linearly parameterized uncertain nonlinear systems (55, 56). Further performance improvement is achieved by combining robust adaptive MPC with iterative learning (57), which updates the terminal constraint set $\mathbb{X}_{\text{term}}$ in Equation 12d and the terminal cost l_H in Equation 12a after every full iteration using the closed-loop state trajectory and cost (58). In the second approach, an underlying MRAC (see Section 2.2) is used to make the closed-loop system dynamics resemble a linear reference model with bounded disturbance set $\mathbb{D}$ (59). This linear model and its bounds are then used in an outer-loop robust model predictive controller to achieve fast stabilization in the presence of model errors.

3.1.3.2. Learning-based robust model predictive control. Learning-based robust MPC uses data to improve the unknown dynamics estimate, reduce the uncertainty set, or update the cost to avoid states with high uncertainty. Unlike robust adaptive control, learning-based robust MPC considers nonparameterized systems.

Under the assumption of a linear nominal model $\bar{\mathbf{f}}(\mathbf{x}, \mathbf{u}) = \bar{\mathbf{A}}\mathbf{x} + \bar{\mathbf{B}}\mathbf{u}$ and bounded unknown dynamics $\hat{\mathbf{f}}(\mathbf{x}_k, \mathbf{u}_k) \in \mathbb{D}$, the unknown dynamics can be safely learned from data, for example, using a neural network (60). Robust constraint satisfaction (Equation 12c) is guaranteed by using tube-based MPC for the linear nominal model, and performance improvement is achieved by optimizing over the control inputs for the combined nominal and learned dynamics. If the bounded unknown dynamics are assumed to be state dependent, $\hat{\mathbf{f}}(\mathbf{x}_k) \in \mathbb{D}(\mathbf{x}_k)$, instead of using a constant tube Ω_{tube}, then the state constraints in Equation 12c can be tightened based on the state-dependent uncertainty set $\mathbb{D}(\mathbf{x})$ (61). In a numerical stabilization task, a GP is used to model $\hat{\mathbf{f}}(\mathbf{x}_k)$, and its covariance determines $\mathbb{D}(\mathbf{x})$. This yields less conservative, probabilistic state constraints (safety level II).

Learning-based robust MPC can be extended to nonlinear nominal models. Typically, a GP is used to learn the unknown dynamics $\hat{\mathbf{f}}(\mathbf{x}, \mathbf{u})$: The mean updates the dynamics model in Equation 12b, and the state- and input-dependent uncertainty set $\mathbb{D}(\mathbf{x}, \mathbf{u})$ is derived from the GP's covariance and contains the true uncertainty with high probability. Similarly, the state- and input-dependent tube in Equation 12c is determined from the uncertainty set $\mathbb{D}(\mathbf{x}, \mathbf{u})$. The main challenge, compared with using a linear nominal model, is the uncertainty propagation over the prediction horizon in the MPC, because Gaussian uncertainty (obtained from the GP) is no longer Gaussian when propagated through the nominal nonlinear dynamics. Approximation schemes are required, such as using a sigma-point transform (62), linearization (63), exact moment matching (64), or ellipsoidal uncertainty set propagation (65). Additionally, further approximations [e.g.,

fixing the GP's covariances over the prediction horizon (63)] are usually required to achieve real-time implementation. These approximations can lead to violations of the probabilistic constraints (safety level II) in Equation 9e. An alternative approach to address the challenges of GPs uses a neural network regression model that predicts the quantile bounds of the tail of a state trajectory distribution for tube-based MPC (66). However, this approach is hindered by typically nonexhaustive training data sets, which can lead to the underestimation of the tube size.

For repetitive tasks, another approach is to adjust the cost function based on data instead of the system model. The predicted cost error is learned from data using the difference between the predicted cost at each time step and the actual closed-loop cost at execution. By adding this additional term to the cost function, the MPC penalizes states that had previously resulted in a higher closed-loop cost than expected (Equation 12a) (67), resulting in reliable performance despite model errors.

3.1.4. Safe model-based reinforcement learning with a priori dynamics. Safe model-based RL augments model-based RL (see Section 2.3) with safety guarantees (see **Figure 4**). Stability can be probabilistically guaranteed (safety level II) under the assumption that the known nominal model $\bar{\mathbf{f}}(\mathbf{x}_k, \mathbf{u}_k)$ and the unknown part $\hat{\mathbf{f}}(\mathbf{x}_k, \mathbf{u}_k)$ are Lipschitz continuous with known Lipschitz constants (68). Given a Lyapunov function, an initial safe policy $\boldsymbol{\pi}_0$, and a GP to learn the unknown dynamics $\hat{\mathbf{f}}$, a control policy $\boldsymbol{\pi}_k$ is chosen so that it maximizes a conservative estimate of the region of attraction (ROA). The most uncertain states (based on the GP's covariance) inside the ROA are explored, which reduces the uncertainty over time and allows the ROA to be extended. The practical implementation resorts to discrete states for tractability and retains the stability guarantees while being suboptimal in (exploration) performance.

3.2. Encouraging Safety and Robustness in Reinforcement Learning

The approaches in this section are safety-augmented variations of the traditional MDP and RL frameworks. In general, these methods do not assume knowledge of an a priori nominal model $\bar{\mathbf{f}}$, and some also learn the reward or step cost l (69) or the safety constraints $\mathbf{c}$ (70). Rather than providing strict safety guarantees, these approaches encourage constraint satisfaction during and after learning, or the robustness of the learned control policy $\boldsymbol{\pi}$ to uncertain dynamics (safety level I; see **Figure 4**). In plain MDP formulations, (*a*) states and inputs (or actions) are assumed to have known, often discrete and finite, domains but are not further constrained while searching for an optimal policy $\boldsymbol{\pi}^*$, and (*b*) only loose assumptions are made on the dynamics $\mathbf{f}$, such as the system satisfying the Markov property (71).

A previous taxonomy of safe RL (4)—covering research published up until 2015—distinguished methods that either modify the exploration process with external knowledge or modify the optimality criterion J with a safety factor. However, the number and breadth of publications in RL, including safe RL, have since greatly increased (72). Because recent works in safe RL are numerous and diverse, we provide a high-level review of the significant trends with an emphasis on robotics, including (*a*) safe exploration of MDPs, (*b*) risk-aware RL, (*c*) RL with CMDPs, and (*d*) robust RL.

3.2.1. Safe exploration and optimization. Exploration in RL poses a challenge to its safety, as it must select inputs with unpredictable consequences in order to learn about them.

3.2.1.1. Safe exploration. Reference 73 used the notion of ergodicity to tackle the problem of safely exploring an MDP. Policy updates that preserve the ergodicity of the MDP enable the system to return to an arbitrary state from any state. Thus, the core idea is to restrict the

Lipschitz constant: a positive scalar ρ that bounds the change in the output of a Lipschitz continuous function to a maximum of ρ times the change in the input

Region of attraction (ROA): the set of states from which a closed-loop system converges to an equilibrium $\mathbf{x}^*$

Markov property: the property that the probability of being in state $\mathbf{x}_k$ at time k depends only on the state at time $k-1$, $\mathbf{x}_{k-1}$, and, in MDPs, on $\mathbf{u}_{k-1}$

Ergodicity: for an MDP, the property that, by following a given policy, any state is reachable from any other state

space of eligible policies to those that make the MDP ergodic (with at least a given probability). Exactly solving this problem, however, is NP-hard. Reference 73 solved a simplified problem using a heuristic exploration algorithm (74), which leads to suboptimal but safe exploration that only considers a subset of the ergodic policies. In practice, this method was demonstrated in two simulated scenarios with discrete, finite $\mathbb{X}$ and $\mathbb{U}$. In a different approach from recoverable exploration via ergodicity, References 70 and 75 developed safe exploration strategies with constraint satisfaction. They used a safety layer to convert an optimal but potentially unsafe action $\mathbf{u}_{\text{learn},k}$ from a neural network policy into the closest safe action $\mathbf{u}_{\text{safe},k}$ with respect to some safety state constraints. Both works involved solving a constrained least-squares problem, $\mathbf{u}_{\text{safe},k} = \arg\min_{\mathbf{u}_k} \|\mathbf{u}_k - \mathbf{u}_{\text{learn},k}\|_2^2$, which is akin to the safety filter approaches covered in Section 3.3.2. Concretely, Reference 75 assumed full knowledge of the (linear) constraint functions and solved $\mathbf{u}_{\text{safe},k}$ using a differentiable quadratic programming solver. By contrast, Reference 70 assumed that constraints are unknown a priori but can be evaluated. Thus, the approach learns the linear approximations of these constraints and then uses them in the solver. Notably, References 70 and 75 considered single-time-step state constraints. In Section 3.2.3, we discuss methods that deal with more general trajectory-level constraints (Equation 6).

3.2.1.2. Safe optimization. Several works have addressed the problem of safely optimizing an unknown function (typically the cost function), often exploiting GP models (76). Safety refers to sampling inputs that do not violate a given safety threshold (Equations 3 and 4). These approaches fall under the category of Bayesian optimization (77) and include SafeOpt (78); SafeOpt-MC (79), an extension to multiple constraints; StageOpt (80), a more efficient two-stage implementation; and GoSafe (81), used for exploration beyond the initial safe region. In particular, SafeOpt infers two subsets of the safe set from the GP model—one with candidate inputs to extend the safe set, and one with candidate inputs to optimize the unknown function—from which it greedily picks the most uncertain. In SafeMDP, Reference 69 applied the ideas pioneered by SafeOpt to MDPs, resulting in the safe exploration of MDPs with an unknown cost function $l(\mathbf{x}, \mathbf{u})$, which the paper modeled as a GP. In SafeMDP, the single-step reward represents the safety feature that should not violate a given threshold. Another extension of SafeOpt (78), SafeExpOpt-MDP (82), treats the safety feature c^j and the MDP's cost l as two separate, unknown functions, allowing for the constraint of the former and the optimization of the latter. Reference 76 provided a recent survey of these techniques that highlighted the distinction between safe learning in regression (i.e., minimizing the selection and evaluation of nonsafe training inputs) and safe exploration in MDPs and stochastic dynamical systems such as Equation 9b (i.e., selecting action inputs that also preserve ergodicity).

3.2.1.3. Learning a safety critic. A safety critic is a learnable action-value function $Q^{\boldsymbol{\pi}}_{\text{safe}}$ that can detect whether a proposed action can lead to unsafe conditions. References 83–85 used this critic with various fallback schemes to determine a safer alternative input. These works differ from those in Section 3.3.2 in that the filtering criterion depends on a model-free, learned value function, which can only grant the satisfaction of safety level I.

Reference 83 used safety Q-functions for reinforcement learning (SQRL) to (*a*) learn a safety critic from only abstract, sparse safety labels (e.g., a binary indicator) and (*b*) transfer knowledge of safe action inputs to new but similar tasks. SQRL trains $Q^{\boldsymbol{\pi}}_{\text{safe}}$ to predict the future probability of failure in a trajectory and uses it to filter out unsafe actions from the policy $\boldsymbol{\pi}$. Knowledge transfer is achieved by pretraining $Q^{\boldsymbol{\pi}}_{\text{safe}}$ and $\boldsymbol{\pi}$ in simulations and then fine-tuning $\boldsymbol{\pi}$ on the new task (with similar dynamics $\mathbf{f}$ and safety constraints), still in simulation, while reusing $Q^{\boldsymbol{\pi}}_{\text{safe}}$ to discriminate unsafe inputs. However, the success of the final safe policy still depends on the

task- and environment-specific hyperparameters (which must be found via parameter search prior to the actual experiment). Building on this work, recovery RL (84) additionally learns a recovery policy π_{rec} to produce fallback actions for Q^{π}_{safe} as an alternative to filtering out unsafe inputs and resorting to potentially suboptimal ones in π. Reference 85 extended conservative Q-learning (86), an approach that mitigates the value function overestimation of Q-learning, and proposed the conservative safety critic algorithm. Similarly to SQRL, this algorithm assumes sparse safety labels and uses Q^{π}_{safe} for action filtering, but it trains Q^{π}_{safe} to upper bound the probability of failure and ensures provably safe policy improvement at each iteration.

Conditional value at risk: a risk measure that is equal to the average of the samples below the α-percentile of the total cost J

Distributional RL: RL that aims at modeling and learning the distribution of the cost J rather than its expected value

3.2.2. Risk-averse reinforcement learning and uncertainty-aware reinforcement learning. Safety in RL can also be encouraged by deriving and using risk (or uncertainty) estimates during learning. These estimates are typically computed for the system dynamics or the overall cost function and leveraged to produce more conservative (and safer) policies.

Risk can be defined as the probability of collision for a robot performing a navigation task (87, 88). A collision model, captured by a neural network ensemble trained with Monte Carlo dropout, predicts the probability distribution of a collision, given the current state and a sequence of future actions. The collision-averse behavior is then achieved by incorporating the collision model in an MPC planner.

Reference 89 extended model-based probabilistic ensembles with trajectory sampling (PETS) (90) to propose cautious adaptation for safe RL (CARL). Ensembles are collections of learned models used to mitigate noise or capture the stochastic dynamics. CARL has two training phases: (*a*) a pretraining phase that is not risk aware, where a PETS agent is trained on different system dynamics, and (*b*) an adaptation phase, where the agent is fine-tuned on the target system by taking risk-averse actions. CARL also defines two notions of risk, one to avoid low-reward trajectories and another to avoid catastrophes (e.g., irrecoverable states or constraint violations). In safety-augmented value estimation from demonstrations (SAVED) (91), a PETS agent is used to predict the probability of a robot's collision and to evaluate a chance constraint for safe exploration. Similar to the safety critic methods described in Section 3.2.1, SAVED learns a value function from sparse costs, which it uses as a terminal cost estimate.

To learn risk-averse policies using only offline data, Reference 92 optimized for a risk measure of the cost, such as conditional value at risk. Instead of using model ensembles, as in References 87–89, Reference 92 used distributional RL to explicitly model the distribution of the total cost of the task (control of a simulated one-dimensional car) and offline learning to improve scalability.

3.2.3. Constrained Markov decision processes and reinforcement learning. The CMDP framework is frequently used in safe RL, as it introduces constraints that can express arbitrary safety notions in the form of Equation 6. RL for CMDPs, however, faces two important challenges (see Section 2.3): how to incorporate and enforce constraints in the RL algorithm and how to efficiently solve the constrained RL problem—especially when using deep learning models, which are the de facto standard in nonsafe RL. In this section, we cover three approaches that aim to address these challenges: (*a*) Lagrangian methods for RL, (*b*) generalized Lyapunov functions for constraints, and (*c*) backward value functions. However, most of the work in this area remains confined to naive simulated tasks, motivating further research on their applicability in real-world control.

3.2.3.1. Lagrangian methods in reinforcement learning optimization. In References 93 and 94, the CMDP constrained optimization problem (see Section 2.3) is first transformed into an equivalent unconstrained optimization problem over the primal variable π and dual variable λ

using the Lagrangian function $\mathcal{L}(\boldsymbol{\pi}, \lambda)$, and RL is used as a subroutine in the primal–dual updates for Equation 15b:

$$\mathcal{L}(\boldsymbol{\pi}, \lambda) = J^{\boldsymbol{\pi}} + \sum_j \lambda_j \left(J_{c^j}^{\boldsymbol{\pi}} - d_j\right), \tag{15a.}$$

$$(\boldsymbol{\pi}^*, \lambda^*) = \arg\max_{\lambda \geq 0} \min_{\boldsymbol{\pi}} \mathcal{L}(\boldsymbol{\pi}, \lambda) \quad \text{subject to} \quad \text{Equations 9b–9d.} \tag{15b.}$$

In particular, Reference 93 defined a constraint on the conditional value at risk of cost $J^{\boldsymbol{\pi}}$ and used policy-gradient or actor–critic methods to update the policy in Equation 15b. Reference 94 subsequently improved on this work by incorporating off-policy updates of the dual variable (with the on-policy primal–dual updates), and showed empirically that this method achieves better sample efficiency and faster convergence. Reference 34 extended a standard trust-region RL algorithm (95) to CMDPs using a novel bound that relates the expected cost of two policies to their state-averaged divergence (where a divergence is a measure of the similarity between probability distributions). The key idea is performing primal–dual updates with surrogates or approximations of the cost $J^{\boldsymbol{\pi}}$ and constraint cost $J_{c^j}^{\boldsymbol{\pi}}$ derived from the bound. The benefits are twofold: The surrogates can be estimated with only state–action data, bypassing the challenge of trajectory evaluation from off-policy data, and the updates guarantee monotonic policy improvement and near constraint satisfaction at each iteration (safety level I). However, unlike in References 93 and 94, each update involves solving the dual variables from scratch, which can be computationally expensive.

3.2.3.2. A Lyapunov approach to safe reinforcement learning. Lyapunov functions are used extensively in control to analyze system stability and are a powerful tool to translate a system's global properties into local conditions. References 96 and 97 used Lyapunov functions to transform the trajectory-level constraints $J_{c^j}^{\boldsymbol{\pi}}$ in Equation 6 into stepwise, state-based constraints. This approach allows a more efficient computation of $J_{c^j}^{\boldsymbol{\pi}}$ and mitigates the cost of off-policy evaluation; however, it also requires the system to start from a baseline policy $\boldsymbol{\pi}_0$ that already satisfies the constraints. In Reference 96, the authors proposed four different algorithms to solve CMDPs by combining traditional RL methods and the Lyapunov constraints, but they are applicable only to discrete input spaces (with continuous state spaces). In subsequent work (97), the authors extended the approach to continuous input spaces and standard policy-gradient methods, addressing its computational tractability.

3.2.3.3. Learning backward value functions. Reference 98 proposed backward value functions to overcome the excessive computational cost in the previous approaches (34, 97). Similar to a (forward) value function $V^{\boldsymbol{\pi}}$ that estimates the total future cost from each state, a backward value function $V^{b,\boldsymbol{\pi}}$ estimates the accumulated cost up to the current state. We can decompose a trajectory-level constraint at any time step k as the sum of $V_{c^j}^{\boldsymbol{\pi}}(\mathbf{x}_k)$ and $V_{c^j}^{b,\boldsymbol{\pi}}(\mathbf{x}_k)$ for the constraint cost $J_{c^j}^{\boldsymbol{\pi}}$. This decomposition also alleviates the problem of off-policy evaluation, as these value functions can be learned concurrently and efficiently via temporal difference methods (71). In practice, $V_{c^j}^{\boldsymbol{\pi}}$, $V_{c^j}^{b,\boldsymbol{\pi}}$, and $V^{\boldsymbol{\pi}}$ are jointly learned (98) and used for policy improvement at each time step, allowing the implementation of safety level I. The approach is intended for discrete action spaces but can be adapted to continuous ones (70).

3.2.4. Robust Markov decision processes and reinforcement learning. Works in this section aim to implement robustness in RL—specifically, learning policies that can operate under disturbances and generalize across similar tasks or robotic systems. This is typically done by framing the learning problem as a robust MDP (Equation 13). Reference 99 developed robust RL,

which implements an actor–disturber–critic architecture. The authors observed that the learned policy and value function coincide with those derived analytically from H_∞ control theory for linear systems (see Section 2.2). However, more recent robust RL literature often abstains from assumptions about dynamics or disturbances and applies model-free RL (100) to seek empirically robust performance at the expense of theoretical guarantees. Below, we introduce two lines of work: (*a*) robust adversarial RL, which explicitly models the min–max problem in Equation 13 in a game-theoretic fashion, and (*b*) domain randomization, which approximates the same problem in Equation 13 by learning on a set of randomized perturbed dynamics.

3.2.4.1. Robustness through adversarial training. Combining RL with adversarial learning (101) results in robust adversarial RL (36), where the robust optimization problem (Equation 13) is set up as a two-player, discounted zero-sum Markov game in which an agent (protagonist) learns policy $\boldsymbol{\pi}$ to control the system and another agent (adversary) learns a separate policy to destabilize the system. The two agents learn in an alternating fashion (each is updated while fixing the other), attempting to progressively improve both the robustness of the protagonist's policy and the strength of its adversary. Reference 37 extended this work with risk-aware agents (see Section 3.2.2), with the protagonist being risk averse and the adversary being risk seeking. This method learns an ensemble of deep Q-networks (102) and defines the risk of an action based on the variance of its value predictions. In another extension of the same work, Reference 38 trained a population of adversaries (rather than a single one), making the resulting protagonist less exploitable by new adversaries. Finally, Reference 103 proposed certified lower bounds for the value predictions from a deep Q-network (102), given bounded observation perturbations. The action selection is based on these value lower bounds, assuming adversarial perturbation.

3.2.4.2. Robustness through domain randomization. Domain randomization methods aim to learn policies that generalize to a wide range of tasks or systems. Instead of concerning worst-case scenarios, learning happens on systems with randomly perturbed parameters (e.g., inertial properties and friction coefficients), which often have prespecified ranges. This effectively induces a robust set that approximates the uncertainty set $\mathbb{D}$ in Equation 13 and allows one to use any standard RL method. In Reference 104, a quadrotor learns vision-based flight in simulation with randomized scenes. Using this model-free policy in the real world results in improved collision avoidance performance. Instead of learning a policy directly, a system described in Reference 105 uses learned visual predictions with an MPC controller to enable efficient and scalable real-world performance. Besides the uniform randomization in References 104 and 105, adaptive randomization strategies such as Bayesian search (106) are also a promising direction. Reference 107 adversarially trained and used a discriminator to guide the randomization process that generates systems that are less explored or exploited by the current policy.

3.3. Certifying Learning-Based Control Under Dynamics Uncertainty

In this section, we review methods providing certification to learning-based control approaches that do not inherently account for safety constraints (see **Figure 2**). We divide the discussion into two parts: (*a*) stability certification and (*b*) constraint set certification. The works in this section leverage an a priori dynamics model of the system and provide hard or probabilistic safety guarantees (safety level II or III) under dynamics uncertainties (see **Figure 4**).

3.3.1. Stability certification. This section introduces certification approaches that guarantee closed-loop stability under a learning-based control policy.

Robust positive control invariant safe set: a set $\Omega_{\text{safe}} \subseteq \mathbb{X}_c$ for which there exists a feedback policy $\pi(\mathbf{x})$ that, when starting within the set, the state never leaves the set, for all possible model errors $\mathbb{D}$

3.3.1.1. Lipschitz-based safety certification for deep neural network–based learning controllers. These approaches exploit the expressive power of DNNs for policy parameterization and guarantee closed-loop stability through a Lipschitz constraint on the DNN. Let ρ be a Lipschitz constant of a DNN policy; an upper bound on ρ that guarantees closed-loop stability (safety level III) can be established by using a small-gain stability analysis (108), solving a semidefinite program (109), or applying a sliding mode control framework (110). This bound on ρ can be used in either (*a*) a passive, iterative enforcement approach, where the Lipschitz constant ρ is first estimated [e.g., via semidefinite programming–based estimation (111)] and then used to guide retraining until the Lipschitz constraint is satisfied, or (*b*) an active enforcement approach, where the Lipschitz constraint is directly enforced by the training algorithm of the DNN [e.g., via spectral normalization (110)]. While guaranteeing stability, the Lipschitz-based certification approaches often rely on the particular structure of the system dynamics (e.g., a control-affine structure or linear structure with additive nonlinear uncertainty) to find the certifying Lipschitz constant. It remains to be explored how this idea can be extended to more generic robot systems.

3.3.1.2. Learning regions of attraction for safety certification. The ROA of a closed-loop system is used in the learning-based control literature as a means to guarantee safety. For a nonlinear system with a given state-feedback controller, the ROA is the set of states that is guaranteed to converge to the equilibrium, which is treated as a safe state. This notion of safety provides a way to certify a learning-based controller. It guarantees that there is a region in state space from which the controller can drive the system back to the safe state (safety level III) (112). ROAs can, for example, be used to guide data acquisition for model or controller learning (68, 113).

We consider deterministic closed-loop systems $\mathbf{x}_{k+1} = \mathbf{f}_\pi(\mathbf{x}_k) = \mathbf{f}(\mathbf{x}_k, \boldsymbol{\pi}(\mathbf{x}_k))$, with $\mathbf{f}_\pi(\mathbf{x}_k)$ being Lipschitz continuous. A Lyapunov neural network can be used to iteratively learn the ROA of a controlled nonlinear system from the system's input–output data (112). As compared with the typical Lyapunov functions in control (e.g., quadratic Lyapunov functions), the proposed method uses the Lyapunov neural network as a more flexible Lyapunov function representation to provide a less conservative estimate of the system's ROA. The necessary properties of a Lyapunov function are preserved via the network's architectural design. Reference 113 presented an ROA estimation approach for high-dimensional systems that combines a sum-of-squares programming method for the ROA computation (114) and a dynamics model order reduction technique to curtail computational complexity (115).

3.3.2. Constraint set certification. This section summarizes approaches that provide constraint set certification to a learning-based controller based on the notion of robust positive control invariant safe sets $\Omega_{\text{safe}} \subseteq \mathbb{X}_c$. Certified learning, which can be achieved through a safety filter (9) or shielding (116), finds the minimal modification of a learning-based control input $\mathbf{u}_{\text{learn}}$ (see **Figure 2**) such that the system's state stays inside the set Ω_{safe}:

$$\mathbf{u}_{\text{safe},k} = \underset{\mathbf{u}_k \in \mathbb{U}_c}{\arg\min} \quad \|\mathbf{u}_k - \mathbf{u}_{\text{learn},k}\|_2^2 \tag{16a}$$

$$\text{subject to} \quad \mathbf{x}_{k+1} = \bar{\mathbf{f}}_k(\mathbf{x}_k, \mathbf{u}_k) + \hat{\mathbf{f}}_k(\mathbf{x}_k, \mathbf{u}_k, \mathbf{w}_k) \in \Omega_{\text{safe}}, \tag{16b}$$

$$\forall\, \hat{\mathbf{f}}_k(\mathbf{x}_k, \mathbf{u}_k, \mathbf{w}_k) \in \mathbb{D}(\mathbf{x}_k, \mathbf{u}_k),$$

where the range of possible disturbances $\mathbb{D}(\mathbf{x}_k, \mathbf{u}_k)$ is given. Since the safety filter and controller are usually decoupled, suboptimal behavior can emerge, as the learning-based controller may try to violate the constraints (65).

3.3.2.1. Control barrier functions. Control barrier functions (CBFs) are used to define safe sets. More specifically, the safe set Ω_{safe} is defined as the superlevel set of a continuously differentiable CBF B_c, $B_c : \mathbb{R}^{n_x} \to \mathbb{R}$, as $\Omega_{\text{safe}} = \{\mathbf{x} \in \mathbb{R}^{n_x} : B_c(\mathbf{x}) \geq 0\}$. The function B_c is generally considered for continuous-time, control-affine systems of the form

$$\dot{\mathbf{x}} = \mathbf{f}_{\mathbf{x}}(\mathbf{x}) + \mathbf{f}_{\mathbf{u}}(\mathbf{x})\mathbf{u}, \tag{17.}$$

where $\mathbf{f}_{\mathbf{x}}$ and $\mathbf{f}_{\mathbf{u}}$ are locally Lipschitz, and $\mathbf{x}$ and $\mathbf{u}$ are functions of time (117). In the simplest form, the function B_c is a CBF if there exists a state-feedback control input $\mathbf{u}$ such that the time derivative $\dot{B}_c(\mathbf{x}) = \frac{\partial B_c}{\partial \mathbf{x}}\dot{\mathbf{x}}$ satisfies (117)

$$\sup_{\mathbf{u}\in\mathbb{U}} \frac{\partial B_c}{\partial \mathbf{x}} (\mathbf{f}_{\mathbf{x}}(\mathbf{x}) + \mathbf{f}_{\mathbf{u}}(\mathbf{x})\mathbf{u}) \geq -B_c(\mathbf{x}). \tag{18.}$$

The CBF condition in Equation 18 is a continuous-time version of the robust positive control invariance constraint in Equation 16b. In addition to the robust positive control invariance constraint, a similar constraint for asymptotic stability can be added to Equation 16 in the form of a constraint on the time derivative of a control Lyapunov function (CLF) L_c. However, uncertain dynamics also yield uncertain time derivatives of B_c and L_c.

Control Lyapunov function (CLF): a function whose existence guarantees that there also exists a state-feedback controller $\pi(\mathbf{x})$ that asymptotically stabilizes the system

Learning-based approaches extend CBF and CLF analyses to control-affine systems in Equation 17 with known nominal $\bar{\mathbf{f}}_{\mathbf{x}}$ and $\bar{\mathbf{f}}_{\mathbf{u}}$ and unknown $\hat{\mathbf{f}}_{\mathbf{x}}$ and $\hat{\mathbf{f}}_{\mathbf{u}}$. The time derivative of the unknown dynamics for CBFs and/or CLFs [e.g., $\frac{\partial B_c}{\partial \mathbf{x}}(\hat{\mathbf{f}}_{\mathbf{x}}(\mathbf{x}) + \hat{\mathbf{f}}_{\mathbf{u}}(\mathbf{x})\mathbf{u})$] can be learned from iterative trials (118, 119) or data collected by an RL agent (120, 121). Given a CBF and/or CLF for the true dynamics $\mathbf{f}$, improving the estimate of the CBF's or CLF's time derivative for the unknown dynamics $\hat{\mathbf{f}}$ using data, collected either offline or online, yields a more precise estimate of the constraint in Equation 18. However, any learning error in the CBF's or CLF's time derivative of the unknown dynamics can still lead to applying falsely certified control inputs. To this end, ideas from robust control can be used to guarantee set invariance (safety level III) during the learning process by mapping a bounded uncertainty in the dynamics to a bounded uncertainty in the time derivatives of the CBF or CLF (122, 123), or by accounting for all model errors consistent with the collected data (124). In addition, adaptive control approaches have been proposed to allow safe adaptation of parametric uncertainties in the time derivatives of the CBF or CLF (125, 126). Probabilistic learning techniques for CBFs and CLFs have been used to achieve set invariance probabilistically (safety level II) with varying assumptions about the system dynamics: that the function $\mathbf{f}_{\mathbf{u}}(\mathbf{x})$ is fully known (127, 128), that a nominal model is known (129), or that no nominal model is available (130). A recent extension (131) introduced measurement-robust CBFs that account for errors in the state estimation and allow for safe learning-based updates of the measurement model.

3.3.2.2. Hamilton–Jacobi reachability analysis. Another approach for state constraint set certification of a learning-based controller is via Hamilton–Jacobi reachability analysis. This analysis provides a means to estimate a robust positive control invariant safe set Ω_{safe} under dynamics uncertainties and Equation 16. Consider a nonlinear system subject to unknown but bounded disturbances $\hat{\mathbf{f}}(\mathbf{x}) \in \mathbb{D}(\mathbf{x})$, where $\mathbb{D}(\mathbf{x})$ is assumed to be known but possibly conservative. To compute Ω_{safe}, a two-player, zero-sum differential game is formulated:

$$V(\mathbf{x}) = \max_{\mathbf{u}_{\text{sig}}\in\mathbb{U}_{\text{sig}}} \min_{\hat{\mathbf{f}}_{\text{sig}}\in\mathbb{D}_{\text{sig}}} \left(\inf_{k\geq 0} l_c\left(\phi(\mathbf{x}, k; \mathbf{u}_{\text{sig}}, \hat{\mathbf{f}}_{\text{sig}})\right)\right), \tag{19.}$$

where V is the value function associated with a point $\mathbf{x} \in \mathbb{X}$, $l_c : \mathbb{X} \mapsto \mathbb{R}$ is a cost function that is nonnegative for $\mathbf{x} \in \mathbb{X}_c$ and negative otherwise, $\phi(\mathbf{x}, k; \mathbf{u}_{\text{sig}}, \hat{\mathbf{f}}_{\text{sig}})$ denotes the state at k along a trajectory initialized at $\mathbf{x}$ following input signal $\mathbf{u}_{\text{sig}}$ and disturbance signal $\hat{\mathbf{f}}_{\text{sig}}$, and $\mathbb{U}_{\text{sig}}$ and $\mathbb{D}_{\text{sig}}$ are

collections of input and disturbance signals such that each time instance is in $\mathbb{U}$ and $\mathbb{D}$, respectively. The value function V can be found as the unique viscosity solution of the Hamilton–Jacobi–Isaacs variational inequality (132). The safe set is then $\Omega_{\text{safe}} = \{\mathbf{x} \in \mathbb{X} \mid V(\mathbf{x}) \geq 0\}$. Based on this formulation, we can also obtain an optimally safe policy $\boldsymbol{\pi}^*_{\text{safe}}$ that maximally steers the system toward the safe set Ω_{safe} (i.e., in the greatest ascent direction of V). The Hamilton–Jacobi reachability analysis allows us to define a safety filter for learning-based control approaches to guarantee constraint set satisfaction (safety level II or III). In particular, given Ω_{safe} and $\boldsymbol{\pi}^*_{\text{safe}}$, one can safely learn in the interior of Ω_{safe} and apply the optimally safe policy $\boldsymbol{\pi}^*_{\text{safe}}$ if the system reaches the boundary of Ω_{safe}. To reduce the conservativeness of the approach, Reference 133 proposed a GP-based learning scheme to adapt (and shrink) the unknown dynamics set $\mathbb{D}(\mathbf{x})$ based on observed data.

The general Hamilton–Jacobi reachability analysis framework (132) has also been combined with online dynamics model learning for a target-tracking task (134), with online planning for safe exploration (135) and with temporal difference algorithms for safe RL (136). Reference 137 integrated Hamilton–Jacobi reachability analysis and CBFs to compute smoother control policies while circumventing the need to hand-design appropriate CBFs. In another recent extension, Reference 138 proposed modifications that improve the scalability of the Hamilton–Jacobi safety analysis approach for higher-dimensional systems and demonstrated its use on a 10-dimensional quadrotor trajectory-tracking problem.

3.3.2.3. Predictive safety filters. Predictive safety filters can augment any learning-based controller to enforce state constraints $\mathbf{x} \in \mathbb{X}_{\text{c}}$ and input constraints $\mathbf{u} \in \mathbb{U}_{\text{c}}$. They do this by defining the safe invariant set Ω_{safe} from Equation 16b as the set of states (at the next time step) where a sequence of safe control inputs (e.g., from a backup controller) exists that allows the return to a terminal safe set $\mathbb{X}_{\text{term}}$ or to previously visited safe states.

Model predictive safety certification (MPSC) uses the theory of robust MPC in Section 2.2 and learning-based robust MPC in Section 3.1.3 to filter the output of any learning-based controller, such as the controller of an RL method, to ensure robust constraint satisfaction. The simplest implementation of MPSC (139) uses tube-based MPC and considers the constraints in Equations 12b–12d but replaces the cost in Equation 12a with the cost in Equation 16a to find the closest input $\mathbf{u}_k$ to the learned input $\mathbf{u}_{\text{learn},k}$ at the current time step that guarantees that we will continue to satisfy state and input constraints in the future. The main difference between MPSC and learning-based robust MPC described in Section 3.1.3 is that the terminal safe set $\mathbb{X}_{\text{term}}$ in Equation 12d is not coupled with the selection of the cost function in Equation 12a. Instead, the terminal safe set is conservatively initialized with $\mathbb{X}_{\text{term}} = \Omega_{\text{tube}}$ and can grow to include state trajectories from previous iterations. This approach has been extended to probabilistic constraints by considering a probabilistic tube Ω_{tube} (140) and to nonlinear nominal models (141) (safety level II).

Backup control for safe exploration ensures hard state constraint satisfaction (safety level III) by finding a safe backup controller for any given RL policy $\boldsymbol{\pi}$ (142). Under the assumptions of a known bound $\mathbb{D}(\mathbf{x}, \mathbf{u})$ on the dynamics $\mathbf{f}$ and a distance measure to the state constraints $\mathbb{X}_{\text{c}}$, the backup controller is used to obtain a future state in the neighborhood of a previously visited safe state in some prediction horizon. Before a control input $\mathbf{u}_k$ from $\boldsymbol{\pi}$ is applied to the system, all possible predicted states $\mathbf{x}_{k+1}$ must satisfy (*a*) $\mathbf{x}_{k+1} \in \mathbb{X}_{\text{c}}$ and (*b*) the existence of a safe backup action $\mathbf{u}_{\text{certified}}(\mathbf{x}_{k+1})$. Otherwise, the previous backup control input $\mathbf{u}_{\text{certified}}(\mathbf{x}_k)$ is applied. This procedure guarantees that the system state stays inside a robust positive control invariant set $\Omega_{\text{safe}} \subseteq \mathbb{X}_{\text{c}}$.

4. BENCHMARKS

The approaches discussed in Section 3 have been evaluated in vastly different ways (**Figure 5**). The trends we observe are that (*a*) works that learn uncertain dynamics (Section 3.1) include a

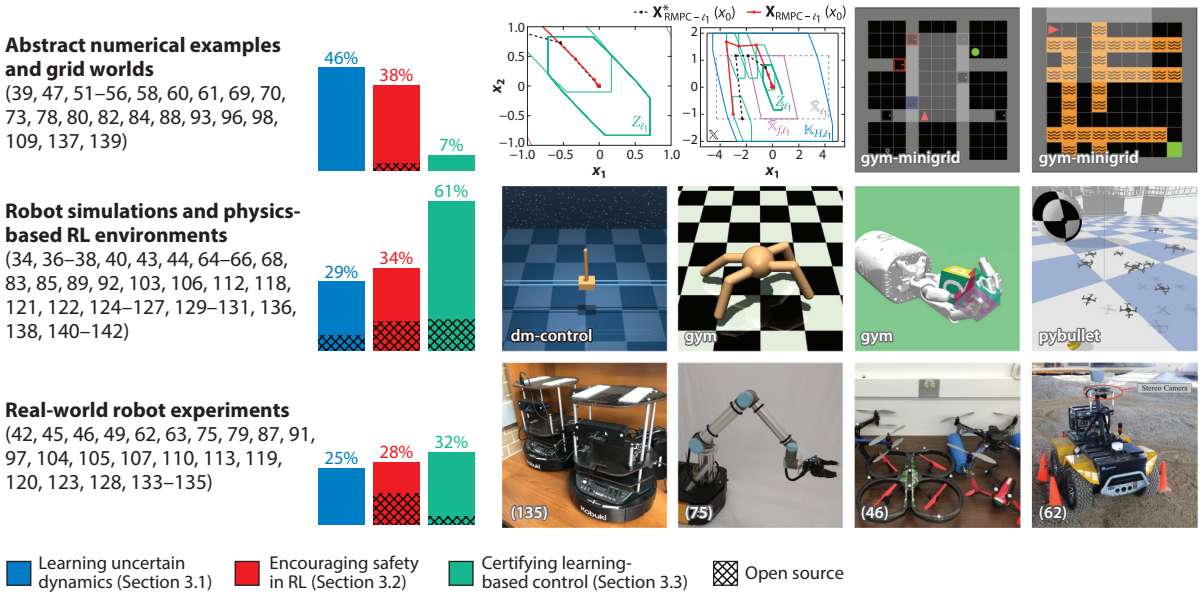

Figure 5

Summary of the environments used for evaluation. With increasing complexity, they can be classified as abstract numerical examples and grid worlds, robot simulations and physics-based RL environments, and real-world robot experiments. The histograms show the prevalence of each category in Sections 3.1–3.3, as well as the fraction whose code is open source. Abbreviation: RL, reinforcement learning.

preponderance of abstract examples; (*b*) works that encourage safety in RL (Section 3.2) still mostly use numerical examples and grid worlds, but robot simulations [often based on physics engines (143) such as MuJoCo] are equally common; and (*c*) works that certify learning-based control (Section 3.3) are still mostly simulated but also account for the largest fraction of real-world experiments. Although numerical examples make it difficult to gauge the practical applicability of a method, we note that even many physics-based RL environments are not representative of existing robotic platforms. In an ideal world, all research would be demonstrated in simulations that closely resemble the target system—and brought to real robots whenever possible.

Furthermore, only a minority of software implementations from published research have been open-sourced. Even in RL, where this is more common (see the red bars in **Figure 5**)—and standardized tools such as `gym` (144) exist—the reproducibility of results (which often rely on careful hyperparameter tuning) remains limited (72). With regard to safety, simple RL environments augmented with constraint evaluation (20) and disturbances (5) have been proposed but lack a unified simulation interface for both safe RL and learning-based control approaches—that is, one that also exposes the available a priori knowledge of a system.

We believe that a necessary stepping-stone for the advancement of safe learning control is to create physics-based environments that (*a*) are simple enough to promote adoption, (*b*) are realistic enough to represent meaningful robotic platforms, (*c*) are equipped with intuitive interfaces for both control and RL researchers, and (*d*) provide useful comparison metrics (e.g., the amount of data required by different approaches).

4.1. Cart–Pole and Quadrotor Benchmark Environments

For these reasons, we created an open-source benchmark suite (6, 7) that simulates two platforms highly popular with both control and RL research: (*a*) a cart–pole system (64, 103, 136) and

(*b*) a quadrotor (44, 66, 129, 142, 145). What sets our implementation apart from previous safe RL environments (5, 20) is the extension of the traditional API (144) with features to facilitate (*a*) the integration of approaches developed by the control theory community (i.e., symbolic nominal models) and (*b*) the evaluation of safety and robustness (i.e., state and input constraints, randomized inertial properties, initial positions and velocities, and external disturbances).

4.2. Safe Learning Control Results

We focus on a constrained stabilization task in both the cart–pole and quadrotor environments. Our results are meant not to establish the superiority of one approach over another but to show how the methods in References 63, 70, and 139, taken from Sections 3.1 (learning uncertain dynamics), 3.2 (encouraging safety in RL), and 3.3 (certifying learning-based control), respectively, can improve control performance while pursuing constraint satisfaction. In doing so, we also show that our benchmark supports algorithms developed by both the control and RL research communities. This also allows us to better compare the data hungriness of the different safe learning control approaches.

For Section 3.1 (learning uncertain dynamics), we implemented a learning-based robust MPC with a GP estimate of $\hat{\mathbf{f}}$ (GP-MPC) to stabilize a quadrotor subject to a state constraint and input constraints, as in Reference 63. A linearization about hover, with a mass and moment of inertia at 150% of the true values, was used as the prior model. Hyperparameter optimization was performed offline, using 800 randomly selected state–action pairs (equivalent to 80 s of training data). **Figure 6** compares the performance of linear MPC, using the (incorrect, heavier) prior model, with the GP-MPC approach. We see that linear MPC predicts the trajectory of the quadrotor to be relatively shallow when maximum thrust is applied, which results in the quadrotor quickly violating the position constraint. By contrast, GP-MPC is able to account for the inaccurate model and satisfies the constraint by a margin proportional to the 95% confidence interval on its predictions, stabilizing the quadrotor.

For Section 3.2 (encouraging safety in RL), we combined the safe exploration approach in Reference 70 with the popular deep RL algorithm proximal policy optimization (PPO) (146) and applied it to cart–pole stabilization with constraints on the cart position. Notably, the task terminates upon any constraint violation. We compared this approach with two baselines: standard PPO and PPO with naive cost shaping (i.e., a penalty when close to constraint violation). Each

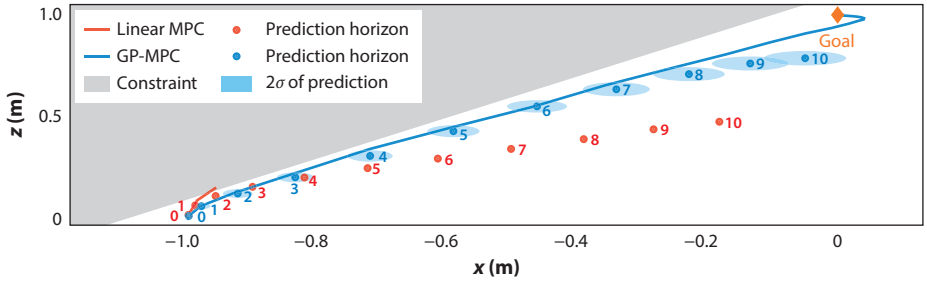

Figure 6

Position comparison of a two-dimensional quadrotor stabilization using linear MPC (*red*) and GP-MPC (*blue*), along with the prediction horizons at the second time step, subject to a diagonal state constraint (*gray*) and input constraints. Abbreviations: GP-MPC, learning-based robust model predictive control with a Gaussian process estimate of $\hat{\mathbf{f}}$; MPC, model predictive control.

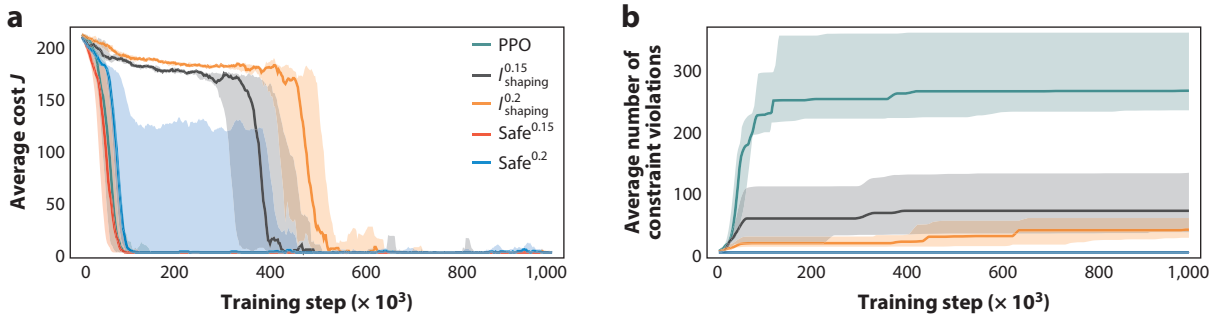

Figure 7

Total (*a*) cost and (*b*) constraint violations during learning for PPO, PPO with cost shaping (for two parameterizations), and PPO with safe exploration (for two slack variable values). Plotted are medians with upper and lower quantiles over 10 seeds. Abbreviation: PPO, proximal policy optimization.

approach used more than 9 h of simulation time to collect training data. **Figure 7** shows that, with sufficient training, the two constraint-aware approaches (cost shaping and safe exploration) achieve their best performance and have substantially fewer constraint violations than standard PPO. In terms of constraint satisfaction, safe exploration outperforms cost shaping without compromising convergence speed. Safe exploration, however, requires careful parameter tuning of the slack variable dictating the responsiveness to near constraint violation.

Finally, for Section 3.3 (certifying learning-based control), we implemented an MPSC algorithm based on Reference 139. This particular formulation uses an MPC framework to modify an unsafe learning controller's actions. Here, a suboptimal PPO controller provides the uncertified inputs trying to stabilize the cart–pole system. The advantages of using MPSC are highlighted in **Figure 8**. In **Figure 8*a***, the inputs are modified by the MPSC early in the stabilization to keep the cart–pole system within the constraint boundaries. **Figure 8*b*** shows that without MPSC, PPO would violate the constraints, but with MPSC, it manages to stay within the boundaries. The plot also shows that MPSC is most active when the system is close to the constraint boundaries. This provides a proof of concept of how safety filters can be combined with RL control to improve safety.

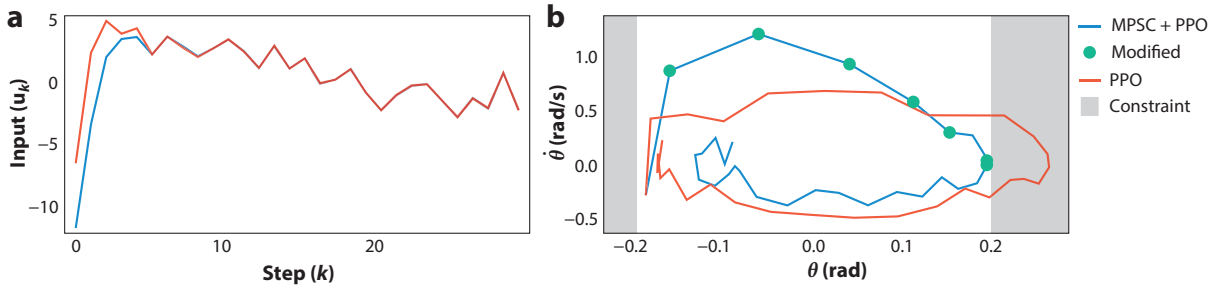

Figure 8

(*a*) Plot of uncertified PPO input (*red*) against certified MPSC + PPO input (*blue*). (*b*) Cart–pole state diagram (θ and $\dot{\theta}$) comparing the MPSC + PPO certified trajectory (*blue*) and the uncertified PPO trajectory (*red*). Green dots show when the MPSC modified the learning controller's input; the MPSC is most active when the system is about to leave the constraint boundary (*gray*) or the set of states from which the MPSC can correct the system. Abbreviations: MPSC, model predictive safety certification; PPO, proximal policy optimization.

5. DISCUSSION AND PERSPECTIVES ON FUTURE DIRECTIONS

The problem of safe learning control is emerging as a crucial topic for next-generation robotics. In this review, we have summarized approaches from the control and the machine learning communities that allow data to be safely used to improve the closed-loop performance of robot control systems. We showed that machine learning techniques, particularly RL, can help generalization toward larger classes of systems (i.e., fewer prior model assumptions), while control theory offers the insights and frameworks necessary to provide constraint satisfaction guarantees and closed-loop stability guarantees during the learning. Despite the many advances to date, there remain many opportunities for future research.

FUTURE ISSUES

1. Capturing a broader class of systems: Work to date has focused on nonlinear systems in the form of Equation 1. While they can model many robotic platforms, robots can also exhibit hybrid dynamics [e.g., legged robots or other contact dynamics with the environment (147)], time-varying dynamics [e.g., operation in changing environments (148, 149)], time delays (e.g., in actuation, sensing, or observing the reward), or partial differential dependencies in the dynamics [e.g., in continuum robotics (150)]. Expanding safe learning control approaches to these scenarios is essential for their broader applicability in robotics.
2. Accounting for imperfect state measurements: The majority of safe learning control approaches assume direct access to (possibly noisy) state measurements and neglect the problem of state estimation. In practice, obtaining accurate state information is challenging due to sensors that do not provide state measurements directly (e.g., images as measurements), inaccurate process and observation models used for state estimation, and/or improper state feature representations. Expanding existing approaches to work with (possibly high-dimensional) sensor data is essential for a broad applicability of these methods in robotics.
3. Considering scalability as well as sampling and computational efficiency: Many of the approaches presented here have been demonstrated only on small toy problems, and applying them to high-dimensional robotics problems is not trivial. Moreover, in practice, we often face issues such as data sparsity, distribution shifts, and the optimality–complexity trade-off for real-time implementations. Efficient robot learning relies on multiple factors, including control architecture design (151), systematic training data collection (152), and appropriate function class selection (153). While current approaches focus on providing theoretical safety guarantees, formal analysis of sampling complexity and computational complexity is indispensable to facilitate the implementation of safe learning control algorithms in real-world robot applications.
4. Verifying system and modeling assumptions: The safety guarantees provided often rely on a set of assumptions (e.g., Lipschitz continuous true dynamics with a known Lipschitz constant or bounded disturbance sets). It is difficult to verify these assumptions prior to a robot's operation. To facilitate algorithm implementation, we also see other approximations being made (e.g., linearization, or data assumed to be independent and identically distributed Gaussian samples). Systematic approaches to verify or quantify the impact of the assumptions and the approximations with minimal (online) data are crucial to allow

the safe learning approaches to be used in real-world applications. This can also include investigations into the interpretability of trained models, especially black-box models such as deep neural networks, for safe closed-loop operation.

DISCLOSURE STATEMENT

The authors are not aware of any affiliations, memberships, funding, or financial holdings that might be perceived as affecting the objectivity of this review.

ACKNOWLEDGMENTS

The authors would like to acknowledge the early contributions to this work by Karime Pereida and Sepehr Samavi, the invaluable suggestions and feedback by Hallie Siegel, and the support from the Natural Sciences and Engineering Research Council of Canada, the Canada Research Chairs program, and the CIFAR AI Chairs program.

LITERATURE CITED

1. Burnett K, Qian J, Du X, Liu L, Yoon DJ, et al. 2021. Zeus: a system description of the two-time winner of the collegiate SAE autodrive competition. *J. Field Robot.* 38:139–66
2. Boutilier JJ, Brooks SC, Janmohamed A, Byers A, Buick JE, et al. 2017. Optimizing a drone network to deliver automated external defibrillators. *Circulation* 135:2454–65
3. Dong K, Pereida K, Shkurti F, Schoellig AP. 2020. Catch the ball: accurate high-speed motions for mobile manipulators via inverse dynamics learning. arXiv:2003.07489 [cs.RO]
4. García J, Fernández F. 2015. A comprehensive survey on safe reinforcement learning. *J. Mach. Learn. Res.* 16:1437–80
5. Dulac-Arnold G, Levine N, Mankowitz DJ, Li J, Paduraru C, et al. 2021. An empirical investigation of the challenges of real-world reinforcement learning. arXiv:2003.11881 [cs.LG]
6. Dyn. Syst. Lab. 2021. safe-control-gym. *GitHub*. **https://github.com/utiasDSL/safe-control-gym**
7. Yuan Z, Hall AW, Zhou S, Brunke L, Greeff M, et al. 2021. safe-control-gym: a unified benchmark suite for safe learning-based control and reinforcement learning. arXiv:2109.06325 [cs.RO]
8. Dulac-Arnold G, Mankowitz D, Hester T. 2019. Challenges of real-world reinforcement learning. arXiv:1904.12901 [cs.LG]
9. Hewing L, Wabersich KP, Menner M, Zeilinger MN. 2020. Learning-based model predictive control: toward safe learning in control. *Annu. Rev. Control Robot. Auton. Syst.* 3:269–96
10. Bristow D, Tharayil M, Alleyne A. 2006. A survey of iterative learning control. *IEEE Control Syst. Mag.* 26(3):96–114
11. Ahn HS, Chen Y, Moore KL. 2007. Iterative learning control: brief survey and categorization. *IEEE Trans. Syst. Man Cybernet. C* 37:1099–121
12. Polydoros AS, Nalpantidis L. 2017. Survey of model-based reinforcement learning: applications on robotics. *J. Intell. Robot. Syst.* 86:153–73
13. Chatzilygeroudis K, Vassiliades V, Stulp F, Calinon S, Mouret JB. 2020. A survey on policy search algorithms for learning robot controllers in a handful of trials. *IEEE Trans. Robot.* 36:328–47
14. Ravichandar H, Polydoros AS, Chernova S, Billard A. 2020. Recent advances in robot learning from demonstration. *Annu. Rev. Control Robot. Auton. Syst.* 3:297–330
15. Kober J, Bagnell JA, Peters J. 2013. Reinforcement learning in robotics: a survey. *Int. J. Robot. Res.* 32:1238–74
16. Recht B. 2019. A tour of reinforcement learning: the view from continuous control. *Annu. Rev. Control Robot. Auton. Syst.* 2:253–79

17. Kiumarsi B, Vamvoudakis KG, Modares H, Lewis FL. 2018. Optimal and autonomous control using reinforcement learning: a survey. *IEEE Trans. Neural Netw. Learn. Syst.* 29:2042–62
18. Osborne M, Shin HS, Tsourdos A. 2021. A review of safe online learning for nonlinear control systems. In *2021 International Conference on Unmanned Aircraft Systems (ICUAS)*, pp. 794–803. Piscataway, NJ: IEEE
19. Tambon F, Laberge G, An L, Nikanjam A, Mindom PSN, et al. 2021. How to certify machine learning based safety-critical systems? A systematic literature review. arXiv:2107.12045 [cs.LG]
20. Ray A, Achiam J, Amodei D. 2019. *Benchmarking safe exploration in deep reinforcement learning*. Preprint, OpenAI, San Francisco, CA. **https://cdn.openai.com/safexp-short.pdf**
21. Leike J, Martic M, Krakovna V, Ortega PA, Everitt T, et al. 2017. AI safety gridworlds. arXiv:1711.09883 [cs.LG]
22. Khalil H. 2002. *Nonlinear Systems*. Upper Saddle River, NJ: Prentice Hall. 3rd ed.
23. Sastry S, Bodson M. 2011. *Adaptive Control: Stability, Convergence and Robustness*. Mineola, NY: Dover
24. Nguyen-Tuong D, Peters J. 2011. Model learning for robot control: a survey. *Cogn. Process.* 12:319–40
25. Zhou K, Doyle J, Glover K. 1996. *Robust and Optimal Control*. Upper Saddle River, NJ: Prentice Hall
26. Dullerud G, Paganini F. 2005. *A Course in Robust Control Theory: A Convex Approach*. New York: Springer
27. Rawlings J, Mayne D, Diehl M. 2017. *Model Predictive Control: Theory, Computation, and Design*. Santa Barbara, CA: Nob Hill
28. Mayne D, Seron M, Raković S. 2005. Robust model predictive control of constrained linear systems with bounded disturbances. *Automatica* 41:219–24
29. Arulkumaran K, Deisenroth MP, Brundage M, Bharath AA. 2017. Deep reinforcement learning: a brief survey. *IEEE Signal Process. Mag.* 34(6):26–38
30. Dai B, Shaw A, Li L, Xiao L, He N, et al. 2018. SBEED: convergent reinforcement learning with nonlinear function approximation. In *Proceedings of the 35th International Conference on Machine Learning*, ed. J Dy, A Krause, pp. 1125–34. Proc. Mach. Learn. Res. 80. N.p.: PMLR
31. Cheng R, Verma A, Orosz G, Chaudhuri S, Yue Y, Burdick J. 2019. Control regularization for reduced variance reinforcement learning. In *Proceedings of the 36th International Conference on Machine Learning*, ed. K Chaudhuri, R Salakhutdinov, pp. 1141–50. Proc. Mach. Learn. Res. 97. N.p.: PMLR
32. Ghavamzadeh M, Mannor S, Pineau J, Tamar A. 2015. Bayesian reinforcement learning: a survey. *Found. Trends Mach. Learn.* 8:359–483
33. Altman E. 1999. *Constrained Markov Decision Processes*. Boca Raton, FL: Chapman & Hall/CRC
34. Achiam J, Held D, Tamar A, Abbeel P. 2017. Constrained policy optimization. In *Proceedings of the 34th International Conference on Machine Learning*, ed. D Precup, YW Teh, pp. 22–31. Proc. Mach. Learn. Res. 70. N.p.: PMLR
35. Nilim A, El Ghaoui L. 2005. Robust control of Markov decision processes with uncertain transition matrices. *Oper. Res.* 53:780–98
36. Pinto L, Davidson J, Sukthankar R, Gupta A. 2017. Robust adversarial reinforcement learning. In *Proceedings of the 34th International Conference on Machine Learning*, ed. D Precup, YW Teh, pp. 2817–26. Proc. Mach. Learn. Res. 70. N.p.: PMLR
37. Pan X, Seita D, Gao Y, Canny J. 2019. Risk averse robust adversarial reinforcement learning. In *2019 International Conference on Robotics and Automation (ICRA)*, pp. 8522–28. Piscataway, NJ: IEEE
38. Vinitsky E, Du Y, Parvate K, Jang K, Abbeel P, Bayen A. 2020. Robust reinforcement learning using adversarial populations. arXiv:2008.01825 [cs.LG]
39. Cooper J, Che J, Cao C. 2014. The use of learning in fast adaptation algorithms. *Int. J. Adapt. Control Signal Process.* 28:325–40
40. Gahlawat A, Zhao P, Patterson A, Hovakimyan N, Theodorou E. 2020. L1-GP: L1 adaptive control with Bayesian learning. In *Proceedings of the 2nd Conference on Learning for Dynamics and Control*, ed. AM Bayen, A Jadbabaie, G Pappas, PA Parrilo, B Recht, et al., pp. 826–37. Proc. Mach. Learn. Res. 120. N.p.: PMLR
41. Hovakimyan N, Cao C. 2010. *L1 Adaptive Control Theory: Guaranteed Robustness with Fast Adaptation*. Philadelphia: Soc. Ind. Appl. Math.
42. Grande RC, Chowdhary G, How JP. 2014. Experimental validation of Bayesian nonparametric adaptive control using Gaussian processes. *J. Aerosp. Inf. Syst.* 11:565–78

43. Chowdhary G, Kingravi HA, How JP, Vela PA. 2015. Bayesian nonparametric adaptive control using Gaussian processes. *IEEE Trans. Neural Netw. Learn. Syst.* 26:537–50
44. Joshi G, Chowdhary G. 2019. Deep model reference adaptive control. In *2019 IEEE 58th Conference on Decision and Control (CDC)*, pp. 4601–8. Piscataway, NJ: IEEE
45. Joshi G, Virdi J, Chowdhary G. 2020. Asynchronous deep model reference adaptive control. arXiv:2011.02920 [cs.RO]
46. Berkenkamp F, Schoellig AP. 2015. Safe and robust learning control with Gaussian processes. In *2015 European Control Conference (ECC)*, pp. 2496–501. Piscataway, NJ: IEEE
47. Holicki T, Scherer CW, Trimpe S. 2021. Controller design via experimental exploration with robustness guarantees. *IEEE Control Syst. Lett.* 5:641–46
48. von Rohr A, Neumann-Brosig M, Trimpe S. 2021. Probabilistic robust linear quadratic regulators with Gaussian processes. In *Proceedings of the 3rd Conference on Learning for Dynamics and Control*, ed. A Jadbabaie, J Lygeros, GJ Pappas, PA Parrilo, B Recht, et al., pp. 324–35. Proc. Mach. Learn. Res. 144. N.p.: PMLR
49. Helwa MK, Heins A, Schoellig AP. 2019. Provably robust learning-based approach for high-accuracy tracking control of Lagrangian systems. *IEEE Robot. Autom. Lett.* 4:1587–94
50. Greeff M, Schoellig AP. 2021. Exploiting differential flatness for robust learning-based tracking control using Gaussian processes. *IEEE Control Syst. Lett.* 5:1121–26
51. Tanaskovic M, Fagiano L, Smith R, Morari M. 2014. Adaptive receding horizon control for constrained MIMO systems. *Automatica* 50:3019–29
52. Lorenzen M, Cannon M, Allgöwer F. 2019. Robust MPC with recursive model update. *Automatica* 103:461–71
53. Bujarbaruah M, Zhang X, Borrelli F. 2018. Adaptive MPC with chance constraints for FIR systems. In *2018 Annual American Control Conference (ACC)*, pp. 2312–17. Piscataway, NJ: IEEE
54. Bujarbaruah M, Zhang X, Tanaskovic M, Borrelli F. 2019. Adaptive MPC under time varying uncertainty: robust and stochastic. arXiv:1909.13473 [eess.SY]
55. Gonçalves GA, Guay M. 2016. Robust discrete-time set-based adaptive predictive control for nonlinear systems. *J. Process Control* 39:111–22
56. Köhler J, Kötting P, Soloperto R, Allgöwer F, Müller MA. 2021. A robust adaptive model predictive control framework for nonlinear uncertain systems. *Int. J. Robust Nonlinear Control* 31:8725–49
57. Rosolia U, Borrelli F. 2018. Learning model predictive control for iterative tasks. A data-driven control framework. *IEEE Trans. Autom. Control* 63:1883–96
58. Bujarbaruah M, Zhang X, Rosolia U, Borrelli F. 2018. Adaptive MPC for iterative tasks. In *2018 IEEE Conference on Decision and Control (CDC)*, pp. 6322–27. Piscataway, NJ: IEEE
59. Pereida K, Brunke L, Schoellig AP. 2021. Robust adaptive model predictive control for guaranteed fast and accurate stabilization in the presence of model errors. *Int. J. Robust Nonlinear Control* 31:8750–84
60. Aswani A, Gonzalez H, Sastry SS, Tomlin C. 2013. Provably safe and robust learning-based model predictive control. *Automatica* 49:1216–26
61. Soloperto R, Müller MA, Trimpe S, Allgöwer F. 2018. Learning-based robust model predictive control with state-dependent uncertainty. *IFAC-PapersOnLine* 51(20):442–47
62. Ostafew CJ, Schoellig AP, Barfoot TD. 2016. Robust constrained learning-based NMPC enabling reliable mobile robot path tracking. *Int. J. Robot. Res.* 35:1547–63
63. Hewing L, Kabzan J, Zeilinger MN. 2020. Cautious model predictive control using Gaussian process regression. *IEEE Trans. Control Syst. Technol.* 28:2736–43
64. Kamthe S, Deisenroth M. 2018. Data-efficient reinforcement learning with probabilistic model predictive control. In *Proceedings of the Twenty-First International Conference on Artificial Intelligence and Statistics*, ed. A Storkey, F Perez-Cruz, pp. 1701–10. Proc. Mach. Learn. Res. 84. N.p.: PMLR
65. Koller T, Berkenkamp F, Turchetta M, Boedecker J, Krause A. 2019. Learning-based model predictive control for safe exploration and reinforcement learning. arXiv:1906.12189 [eess.SY]
66. Fan D, Agha A, Theodorou E. 2020. Deep learning tubes for tube MPC. In *Robotics: Science and Systems XVI*, ed. M Toussaint, A Bicchi, T Hermans, pap. 87. N.p.: Robot. Sci. Syst. Found.

67. McKinnon CD, Schoellig AP. 2020. Context-aware cost shaping to reduce the impact of model error in receding horizon control. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 2386–92. Piscataway, NJ: IEEE
68. Berkenkamp F, Turchetta M, Schoellig A, Krause A. 2017. Safe model-based reinforcement learning with stability guarantees. In *Advances in Neural Information Processing Systems 30*, ed. I Guyon, UV Luxburg, S Bengio, H Wallach, R Fergus, et al., pp. 908–19. Red Hook, NY: Curran
69. Turchetta M, Berkenkamp F, Krause A. 2016. Safe exploration in finite Markov decision processes with Gaussian processes. In *Advances in Neural Information Processing Systems 29*, ed. DD Lee, M Sugiyama, UV Luxburg, I Guyon, R Garnett, pp. 4312–20. Red Hook, NY: Curran
70. Dalal G, Dvijotham K, Vecerik M, Hester T, Paduraru C, Tassa Y. 2018. Safe exploration in continuous action spaces. arXiv:1801.08757 [cs.AI]
71. Sutton RS, Barto AG. 2018. *Reinforcement Learning: An Introduction*. Cambridge, MA: MIT Press. 2nd ed.
72. Henderson P, Islam R, Bachman P, Pineau J, Precup D, Meger D. 2018. Deep reinforcement learning that matters. In *The Thirty-Second AAAI Conference on Artificial Intelligence*, pp. 3207–14. Palo Alto, CA: AAAI Press
73. Moldovan TM, Abbeel P. 2012. Safe exploration in Markov decision processes. In *Proceedings of the 29th International Conference on Machine Learning (ICML)*, pp. 1451–58. Madison, WI: Omnipress
74. Brafman RI, Tennenholtz M. 2002. R-max – a general polynomial time algorithm for near-optimal reinforcement learning. *J. Mach. Learn. Res.* 3:213–31
75. Pham TH, De Magistris G, Tachibana R. 2018. OptLayer - practical constrained optimization for deep reinforcement learning in the real world. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 6236–43. Piscataway, NJ: IEEE
76. Kim Y, Allmendinger R, López-Ibáñez M. 2021. Safe learning and optimization techniques: towards a survey of the state of the art. arXiv:2101.09505 [cs.LG]
77. Duivenvoorden RR, Berkenkamp F, Carion N, Krause A, Schoellig AP. 2017. Constrained Bayesian optimization with particle swarms for safe adaptive controller tuning. *IFAC-PapersOnLine* 50(1):11800–7
78. Sui Y, Gotovos A, Burdick J, Krause A. 2015. Safe exploration for optimization with Gaussian processes. In *Proceedings of the 32nd International Conference on Machine Learning*, ed. F Bach, D Blei, pp. 997–1005. Proc. Mach. Learn. Res. 37. N.p.: PMLR
79. Berkenkamp F, Krause A, Schoellig AP. 2020. Bayesian optimization with safety constraints: safe and automatic parameter tuning in robotics. arXiv:1602.04450 [cs.RO]
80. Sui Y, Zhuang V, Burdick J, Yue Y. 2018. Stagewise safe Bayesian optimization with Gaussian processes. In *Proceedings of the 35th International Conference on Machine Learning*, ed. J Dy, A Krause, pp. 4781–89. Proc. Mach. Learn. Res. 80. N.p.: PMLR
81. Baumann D, Marco A, Turchetta M, Trimpe S. 2021. GoSafe: globally optimal safe robot learning. arXiv:2105.13281 [cs.RO]
82. Wachi A, Sui Y, Yue Y, Ono M. 2018. Safe exploration and optimization of constrained MDPs using Gaussian processes. In *The Thirty-Second AAAI Conference on Artificial Intelligence*, pp. 6548–55. Palo Alto, CA: AAAI Press
83. Srinivasan K, Eysenbach B, Ha S, Tan J, Finn C. 2020. Learning to be safe: deep RL with a safety critic. arXiv:2010.14603 [cs.LG]
84. Thananjeyan B, Balakrishna A, Nair S, Luo M, Srinivasan K, et al. 2021. Recovery RL: safe reinforcement learning with learned recovery zones. *IEEE Robot. Autom. Lett.* 6:4915–22
85. Bharadhwaj H, Kumar A, Rhinehart N, Levine S, Shkurti F, Garg A. 2021. Conservative safety critics for exploration. arXiv:2010.14497 [cs.LG]
86. Kumar A, Zhou A, Tucker G, Levine S. 2020. Conservative Q-learning for offline reinforcement learning. arXiv:2006.04779 [cs.LG]
87. Kahn G, Villaflor A, Pong V, Abbeel P, Levine S. 2017. Uncertainty-aware reinforcement learning for collision avoidance. arXiv:1702.01182 [cs.LG]
88. Lütjens B, Everett M, How JP. 2019. Safe reinforcement learning with model uncertainty estimates. In *2019 International Conference on Robotics and Automation (ICRA)*, pp. 8662–68. Piscataway, NJ: IEEE

89. Zhang J, Cheung B, Finn C, Levine S, Jayaraman D. 2020. Cautious adaptation for reinforcement learning in safety-critical settings. In *Proceedings of the 37th International Conference on Machine Learning*, ed. HD Daumé III, A Singh, pp. 11055–65. Proc. Mach. Learn. Res. 119. N.p.: PMLR
90. Chua K, Calandra R, McAllister R, Levine S. 2018. Deep reinforcement learning in a handful of trials using probabilistic dynamics models. In *Advances in Neural Information Processing Systems 31*, ed. S Bengio, H Wallach, H Larochelle, K Grauman, N Cesa-Bianchi, R Garnett, pp. 4759–70. Red Hook, NY: Curran
91. Thananjeyan B, Balakrishna A, Rosolia U, Li F, McAllister R, et al. 2020. Safety augmented value estimation from demonstrations (SAVED): safe deep model-based RL for sparse cost robotic tasks. *IEEE Robot. Autom. Lett.* 5:3612–19
92. Urpí NA, Curi S, Krause A. 2021. Risk-averse offline reinforcement learning. arXiv:2102.05371 [cs.LG]
93. Chow Y, Ghavamzadeh M, Janson L, Pavone M. 2017. Risk-constrained reinforcement learning with percentile risk criteria. *J. Mach. Learn. Res.* 18:6070–120
94. Liang Q, Que F, Modiano E. 2018. Accelerated primal-dual policy optimization for safe reinforcement learning. arXiv:1802.06480 [cs.AI]
95. Schulman J, Levine S, Abbeel P, Jordan M, Moritz P. 2015. Trust region policy optimization. In *Proceedings of the 32nd International Conference on Machine Learning*, ed. F Bach, D Blei, pp. 1889–97. Proc. Mach. Learn. Res. 37. N.p.: PMLR
96. Chow Y, Nachum O, Duenez-Guzman E, Ghavamzadeh M. 2018. A Lyapunov-based approach to safe reinforcement learning. In *Advances in Neural Information Processing Systems 31*, ed. S Bengio, H Wallach, H Larochelle, K Grauman, N Cesa-Bianchi, R Garnett, pp. 8103–12. Red Hook, NY: Curran
97. Chow Y, Nachum O, Faust A, Duenez-Guzman E, Ghavamzadeh M. 2019. Lyapunov-based safe policy optimization for continuous control. arXiv:1901.10031 [cs.LG]
98. Satija H, Amortila P, Pineau J. 2020. Constrained Markov decision processes via backward value functions. In *Proceedings of the 37th International Conference on Machine Learning*, ed. HD Daumé III, A Singh, pp. 8502–11. Proc. Mach. Learn. Res. 119. N.p.: PMLR
99. Morimoto J, Doya K. 2005. Robust reinforcement learning. *Neural Comput.* 17:335–59
100. Turchetta M, Krause A, Trimpe S. 2020. Robust model-free reinforcement learning with multi-objective Bayesian optimization. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 10702–708. Piscataway, NJ: IEEE
101. Goodfellow IJ, Pouget-Abadie J, Mirza M, Xu B, Warde-Farley D, et al. 2014. Generative adversarial networks. arXiv:1406.2661 [stat.ML]
102. Mnih V, Kavukcuoglu K, Silver D, Rusu AA, Veness J, et al. 2015. Human-level control through deep reinforcement learning. *Nature* 518:529–33
103. Lütjens B, Everett M, How JP. 2020. Certified adversarial robustness for deep reinforcement learning. In *Proceedings of the Conference on Robot Learning*, ed. LP Kaelbling, D Kragic, K Sugiura, pp. 1328–37. Proc. Mach. Learn. Res. 100. N.p.: PMLR
104. Sadeghi F, Levine S. 2017. CAD2RL: real single-image flight without a single real image. In *Robotics: Science and Systems XIII*, ed. N Amato, S Srinivasa, N Ayanian, S Kuindersma, pap. 34. N.p.: Robot. Sci. Syst. Found.
105. Loquercio A, Kaufmann E, Ranftl R, Dosovitskiy A, Koltun V, Scaramuzza D. 2020. Deep drone racing: from simulation to reality with domain randomization. *IEEE Trans. Robot.* 36:1–14
106. Rajeswaran A, Ghotra S, Ravindran B, Levine S. 2017. EPOpt: learning robust neural network policies using model ensembles. arXiv:1610.01283 [cs.LG]
107. Mehta B, Diaz M, Golemo F, Pal CJ, Paull L. 2020. Active domain randomization. In *Proceedings of the Conference on Robot Learning*, ed. LP Kaelbling, D Kragic, K Sugiura, pp. 1162–76. Proc. Mach. Learn. Res. 100. N.p.: PMLR
108. Zhou S, Helwa MK, Schoellig AP. 2020. Deep neural networks as add-on modules for enhancing robot performance in impromptu trajectory tracking. *Int. J. Robot. Res.* 39:1397–418
109. Jin M, Lavaei J. 2020. Stability-certified reinforcement learning: a control-theoretic perspective. *IEEE Access* 8:229086–100
110. Shi G, Shi X, O'Connell M, Yu R, Azizzadenesheli K, et al. 2019. Neural lander: stable drone landing control using learned dynamics. In *2019 International Conference on Robotics and Automation (ICRA)*, pp. 9784–90. Piscataway, NJ: IEEE

111. Fazlyab M, Robey A, Hassani H, Morari M, Pappas GJ. 2019. Efficient and accurate estimation of Lipschitz constants for deep neural networks. arXiv:1906.04893 [cs.LG]
112. Richards SM, Berkenkamp F, Krause A. 2018. The Lyapunov neural network: adaptive stability certification for safe learning of dynamical systems. In *Proceedings of the 2nd Conference on Robot Learning*, ed. A Billard, A Dragan, J Peters, J Morimoto, pp. 466–76. Proc. Mach. Learn. Res. 87. N.p.: PMLR
113. Zhou Z, Oguz OS, Leibold M, Buss M. 2020. A general framework to increase safety of learning algorithms for dynamical systems based on region of attraction estimation. *IEEE Trans. Robot.* 36:1472–90
114. Jarvis-Wloszek Z, Feeley R, Tan W, Sun K, Packard A. 2003. Some controls applications of sum of squares programming. In *42nd IEEE International Conference on Decision and Control (CDC)*, Vol. 5, pp. 4676–81. Piscataway, NJ: IEEE
115. Schilders WH, Van der Vorst HA, Rommes J. 2008. *Model Order Reduction: Theory, Research Aspects and Applications*. Berlin: Springer
116. Alshiekh M, Bloem R, Udiger Ehlers R, Könighofer B, Niekum S, Topcu U. 2018. Safe reinforcement learning via shielding. In *The Thirty-Second AAAI Conference on Artificial Intelligence*, pp. 2669–78. Palo Alto, CA: AAAI Press
117. Ames AD, Coogan S, Egerstedt M, Notomista G, Sreenath K, Tabuada P. 2019. Control barrier functions: theory and applications. In *2019 18th European Control Conference (ECC)*, pp. 3420–31. Piscataway, NJ: IEEE
118. Taylor AJ, Dorobantu VD, Le HM, Yue Y, Ames AD. 2019. Episodic learning with control Lyapunov functions for uncertain robotic systems. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 6878–84. Piscataway, NJ: IEEE
119. Taylor A, Singletary A, Yue Y, Ames A. 2020. Learning for safety-critical control with control barrier functions. In *Proceedings of the 2nd Conference on Learning for Dynamics and Control*, ed. AM Bayen, A Jadbabaie, G Pappas, PA Parrilo, B Recht, et al., pp. 708–17. Proc. Mach. Learn. Res. 120. N.p.: PMLR
120. Ohnishi M, Wang L, Notomista G, Egerstedt M. 2019. Barrier-certified adaptive reinforcement learning with applications to brushbot navigation. *IEEE Trans. Robot.* 35:1186–205
121. Choi J, Castañeda F, Tomlin C, Sreenath K. 2020. Reinforcement learning for safety-critical control under model uncertainty, using control Lyapunov functions and control barrier functions. In *Robotics: Science and Systems XVI*, ed. M Toussaint, A Bicchi, T Hermans, pap. 88. N.p.: Robot. Sci. Syst. Found.
122. Taylor AJ, Dorobantu VD, Krishnamoorthy M, Le HM, Yue Y, Ames AD. 2019. A control Lyapunov perspective on episodic learning via projection to state stability. In *2019 IEEE 58th Conference on Decision and Control (CDC)*, pp. 1448–55. Piscataway, NJ: IEEE
123. Taylor AJ, Singletary A, Yue Y, Ames AD. 2020. A control barrier perspective on episodic learning via projection-to-state safety. arXiv:2003.08028 [eess.SY]
124. Taylor AJ, Dorobantu VD, Dean S, Recht B, Yue Y, Ames AD. 2020. Towards robust data-driven control synthesis for nonlinear systems with actuation uncertainty. arXiv:2011.10730 [eess.SY]
125. Taylor AJ, Ames AD. 2020. Adaptive safety with control barrier functions. In *2020 American Control Conference (ACC)*, pp. 1399–405. Piscataway, NJ: IEEE
126. Lopez BT, Slotine JJE, How JP. 2021. Robust adaptive control barrier functions: an adaptive and data-driven approach to safety. *IEEE Control Syst. Lett.* 5:1031–36
127. Cheng R, Orosz G, Murray RM, Burdick JW. 2019. End-to-end safe reinforcement learning through barrier functions for safety-critical continuous control tasks. *The Thirty-Third AAAI Conference on Artificial Intelligence*, pp. 3387–95. Palo Alto, CA: AAAI Press
128. Fan DD, Nguyen J, Thakker R, Alatur N, Agha-mohammadi A, Theodorou EA. 2019. Bayesian learning-based adaptive control for safety critical systems. arXiv:1910.02325 [eess.SY]
129. Wang L, Theodorou EA, Egerstedt M. 2018. Safe learning of quadrotor dynamics using barrier certificates. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 2460–65. Piscataway, NJ: IEEE

130. Khojasteh MJ, Dhiman V, Franceschetti M, Atanasov N. 2020. Probabilistic safety constraints for learned high relative degree system dynamics. In *Proceedings of the 2nd Conference on Learning for Dynamics and Control*, ed. AM Bayen, A Jadbabaie, G Pappas, PA Parrilo, B Recht, et al., pp. 781–92. Proc. Mach. Learn. Res. 120. N.p.: PMLR
131. Dean S, Taylor AJ, Cosner RK, Recht B, Ames AD. 2020. Guaranteeing safety of learned perception modules via measurement-robust control barrier functions. arXiv:2010.16001 [eess.SY]
132. Mitchell I, Bayen A, Tomlin C. 2005. A time-dependent Hamilton-Jacobi formulation of reachable sets for continuous dynamic games. *IEEE Trans. Autom. Control* 50:947–57
133. Fisac JF, Akametalu AK, Zeilinger MN, Kaynama S, Gillula J, Tomlin CJ. 2019. A general safety framework for learning-based control in uncertain robotic systems. *IEEE Trans. Autom. Control* 64:2737–52
134. Gillula JH, Tomlin CJ. 2012. Guaranteed safe online learning via reachability: tracking a ground target using a quadrotor. In *2012 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 2723–30. Piscataway, NJ: IEEE
135. Bajcsy A, Bansal S, Bronstein E, Tolani V, Tomlin CJ. 2019. An efficient reachability-based framework for provably safe autonomous navigation in unknown environments. In *2019 IEEE 58th Conference on Decision and Control (CDC)*, pp. 1758–65. Piscataway, NJ: IEEE
136. Fisac JF, Lugovoy NF, Rubies-Royo V, Ghosh S, Tomlin CJ. 2019. Bridging Hamilton-Jacobi safety analysis and reinforcement learning. In *2019 International Conference on Robotics and Automation (ICRA)*, pp. 8550–56. Piscataway, NJ: IEEE
137. Choi JJ, Lee D, Sreenath K, Tomlin CJ, Herbert SL. 2021. Robust control barrier-value functions for safety-critical control. arXiv:2104.02808 [eess.SY]
138. Herbert S, Choi JJ, Sanjeev S, Gibson M, Sreenath K, Tomlin CJ. 2021. Scalable learning of safety guarantees for autonomous systems using Hamilton-Jacobi reachability. arXiv:2101.05916 [cs.RO]
139. Wabersich KP, Zeilinger MN. 2018. Linear model predictive safety certification for learning-based control. In *2018 IEEE Conference on Decision and Control (CDC)*, pp. 7130–35. Piscataway, NJ: IEEE
140. Wabersich KP, Hewing L, Carron A, Zeilinger MN. 2019. Probabilistic model predictive safety certification for learning-based control. arXiv:1906.10417 [eess.SY]
141. Wabersich KP, Zeilinger MN. 2021. A predictive safety filter for learning-based control of constrained nonlinear dynamical systems. *Automatica* 129:109597
142. Mannucci T, van Kampen E, de Visser C, Chu Q. 2018. Safe exploration algorithms for reinforcement learning controllers. *IEEE Trans. Neural Netw. Learn. Syst.* 29:1069–81
143. Liu CK, Negrut D. 2021. The role of physics-based simulators in robotics. *Annu. Rev. Control Robot. Auton. Syst.* 4:35–58
144. Brockman G, Cheung V, Pettersson L, Schneider J, Schulman J, et al. 2016. OpenAI Gym. arXiv:1606.01540 [cs.LG]
145. Panerati J, Zheng H, Zhou S, Xu J, Prorok A, Schoellig AP. 2021. Learning to fly—a Gym environment with PyBullet physics for reinforcement learning of multi-agent quadcopter control. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 7512–19. Piscataway, NJ: IEEE
146. Schulman J, Wolski F, Dhariwal P, Radford A, Klimov O. 2017. Proximal policy optimization algorithms. arXiv:1707.06347 [cs.LG]
147. Wieber PB, Tedrake R, Kuindersma S. 2016. Modeling and control of legged robots. In *Springer Handbook of Robotics*, ed. B Siciliano, O Khatib, pp. 1203–34. Cham, Switz.: Springer
148. McKinnon CD, Schoellig AP. 2018. Experience-based model selection to enable long-term, safe control for repetitive tasks under changing conditions. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 2977–84. Piscataway, NJ: IEEE
149. Chandak Y, Jordan S, Theocharous G, White M, Thomas PS. 2020. Towards safe policy improvement for non-stationary MDPs. In *Advances in Neural Information Processing Systems 33*, ed. H Larochelle, M Ranzato, R Hadsell, MF Balcan, H Lin, pp. 9156–68. Red Hook, NY: Curran
150. Burgner-Kahrs J, Rucker DC, Choset H. 2015. Continuum robots for medical applications: a survey. *IEEE Trans. Robot.* 31:1261–80

151. Mueller FL, Schoellig AP, D'Andrea R. 2012. Iterative learning of feed-forward corrections for high-performance tracking. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 3276–81. Piscataway, NJ: IEEE
152. Dean S, Tu S, Matni N, Recht B. 2018. Safely learning to control the constrained linear quadratic regulator. arXiv:5582–88
153. McKinnon CD, Schoellig AP. 2019. Learning probabilistic models for safe predictive control in unknown environments. In *2019 18th European Control Conference (ECC)*, pp. 2472–79. Piscataway, NJ: IEEE