arXiv:2403.18539v2 [cs.LG] 30 Mar 2024

Safe and Robust Reinforcement Learning: Principles and Practice

Taku Yamagata[1*] and Raúl Santos-Rodríguez[1]

[1*]Intelligent System Laboratory, University of Bristol, Bristol, UK.

*Corresponding author(s). E-mail(s): taku.yamagata@bristol.ac.uk;
Contributing authors: ensr@bristol.ac.uk;

Abstract

Reinforcement Learning (RL) has shown remarkable success in solving relatively complex tasks, yet the deployment of RL systems in real-world scenarios poses significant challenges related to safety and robustness. This paper aims to identify and further understand those challenges thorough the exploration of the main dimensions of the safe and robust RL landscape, encompassing algorithmic, ethical, and practical considerations. We conduct a comprehensive review of methodologies and open problems that summarizes the efforts in recent years to address the inherent risks associated with RL applications.

After discussing and proposing definitions for both safe and robust RL, the paper categorizes existing research works into different algorithmic approaches that enhance the safety and robustness of RL agents. We examine techniques such as uncertainty estimation, optimisation methodologies, exploration-exploitation trade-offs, and adversarial training. Environmental factors, including sim-to-real transfer and domain adaptation, are also scrutinized to understand how RL systems can adapt to diverse and dynamic surroundings. Moreover, human involvement is an integral ingredient of the analysis, acknowledging the broad set of roles that humans can take in this context.

Importantly, to aid practitioners in navigating the complexities of safe and robust RL implementation, this paper introduces a practical checklist derived from the synthesized literature. The checklist encompasses critical aspects of algorithm design, training environment considerations, and ethical guidelines. It will serve as a resource for developers and policymakers alike to ensure the responsible deployment of RL systems in many application domains.

Keywords: reinforcement learning, safe AI, robust Markov decision process, constrained Markov decision process

1 Structure

This paper provides a general overview of the definitions, approaches and practical considerations for safe and robust reinforcement learning (RL). We intend to cover a wide range of topics and approaches related to safe and robust RL, but fully accepting that these are not only ambiguously defined and used, but fast evolving concepts.

Here, we start by introducing a basic formulation of RL framework in Sec. 2. We collect the most common definitions of the terms – *safety* and *robustness* in Sec. 3. Based on those, we propose two working definitions for the remainder of the work.

The following three sections (Sec. 4 to 6) introduce and categorize various safe and robust RL approaches. Figure 1 shows the overview of RL framework and highlights where the relevant categories introduced in these sections fit. We provide a summary of the literature introduced in this paper in Appendix A. It includes two figures (Fig. A1 and A2) that show an overview of both the safe RL and robust RL literature, placing the works in chronological order (from top to bottom).

Section 4 focuses on how to train the agent's policy to achieve safety and robustness. It has three main components – criteria, method and exploration. The `optimisation criteria` related to the study of objective functions that are used to achieve safety. The `optimisation method` provides a overview of approaches to achieve the criteria. The `exploration` part focuses on methods for exploration that count with a safety ingredient. Exploration is a key research topic because exploration and safety are conflicting concepts – in practice, it is a hard trade-off to explore while maintaining the safety.

Section 5 discusses approaches incorporating additional data/knowledge – data, simulators and human knowledge. In principle, collecting more information about the environment, the more likely to be able to improve on the safety front. We explore the different solutions for adding these extra sources of information into the learning process, while assessing its value. Section 6 deals specifically with *human-in-the-Loop*, which usually sits in between the agent and environment, intervening, interfacing and guiding the interactions. For example, we present alternatives for humans to give feedback or *shaping* the reward, or humans changing the agent's action altogether to maintain safety.

In subsequent sections (Sec. 7 and 8), we explore a broad spectrum of topics relevant to different aspects of safety and robustness of RL. For example, until this point, the discussion is centered around the standard RL framework. Section 7 looks into alternative RL paradigms, specifically multi-agent RL and hierarchical RL, as well as other domains that are intrinsically linked to the concept of safe and robust RL. In Sec. 8, we discuss ethical aspects of RL agents for real-world safety critical applications, as we argue that the definition of ethical requirements tailored to the target application domain should be prevalent in any development and deployment.

In Sec. 9, we design and provide practitioners with a checklist for designing a safe and robust RL system. It summarises important design choices and offers a workflow to develop and deploy a safe and robust RL solutions in practice.

Each section of this paper is designed to be stand-alone, requiring no prior knowledge from other sections. However, understanding a basic RL formulation, as outlined in Section 2, is assumed. Readers may directly navigate to sections that align with their interests.

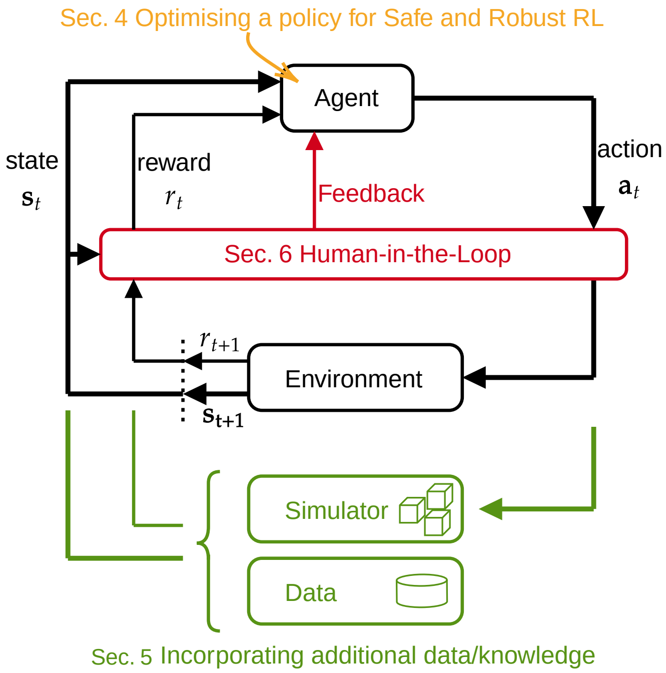


Fig. 1 Overview of the reinforcement learning setting (in black) with relevant components (coloured) to the safe and robust RL introduced in Sec.4 to Sec.6

2 Reinforcement Learning

Here, we introduce a minimal formulation of RL paradigm. RL is an abstract framework for any learning process that involves sequentially interacting with an environment to achieve a certain objective [1]. The learner is called the *agent*. It interacts with the *environment*, observes its consequences, and receives a reward (or a cost) signal – a special numerical assessment the current situation. The agent outputs a sequence of actions to maximise the cummulative reward (or minimise the cost) as shown in Fig. 2.

More formally, the agent and environment interactions are discretised into a sequence of time steps, $t = 0, 1, 2, \ldots$. At each time step t, the agent observes the state of the environment $\mathbf{s}_t$, then decides an action $\mathbf{a}_t$. In the next time step, the environment updates the state based on the agent's action – it becomes $\mathbf{s}_{t+1}$, and also generates the reward $r_{t+1} \in \mathbb{R}$. The agent basically learns a mapping from the state to the action that maximises the total amount of reward it receives over the long run. The mapping is called *policy* denoted as $\pi_t(\mathbf{a}|\mathbf{s})$ that indicates the probability of $\mathbf{a}_t = \mathbf{a}$ when the state is $\mathbf{s}_t = \mathbf{s}$.

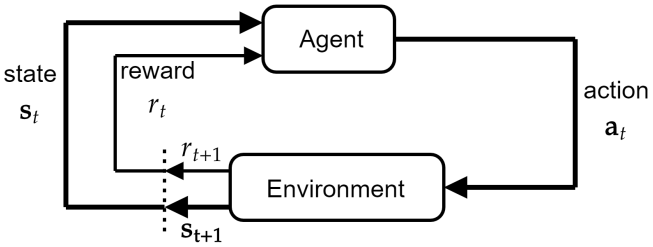


Fig. 2 Overview of the reinforcement learning setting.

Reinforcement learning differs from traditional *supervised learning*. While the latter learns from examples provided by experts, RL learns from its interactions with the environment. RL is suitable when it is obvious what we would like to achieve (the goal), but it is not so obvious how to achieve it. For example, it is clear what we would like to achieve in the chess game (taking the opponent's King and winning), but it is not apparent how to achieve it. In general, RL learns how to achieve the goal autonomously from the interactions (playing games) via a trial-and-error approach.

3 Definitions

In this Section, instead of starting by providing the reader with a formal definition of safe and robust RL, we collect and discuss definitions already present in the literature for each concept. We aim to show the diversity and lack of consesus, while focusing on the similarities among definitions.

3.1 Robust reinforcement learning

According to the Cambridge dictionary[1], the definitions of the word *robust* refer to "(of a person or animal) strong and healthy, or (of an object or system) strong and unlikely to break or fail." Also, according to Wikipedia[2], *robustness* "is the property of being strong and healthy in constitution. When it is transposed into a system, it refers to the ability of tolerating perturbations that might affect the system's functional body. In the same line robustness can be defined as the ability of a system to resist change without adapting its initial stable configuration."

Certain uses of *robust RL* are relatively consistent across the all RL literature. To the best of our knowledge, the term *robust RL* is first introduced by Jun Morimoto and Kenji Doya [2]. They define it as,

[1] https://dictionary.cambridge.org/dictionary/english/robust
[2] https://en.wikipedia.org/wiki/Robustness

> "a new reinforcement learning (RL) paradigm that explicitly takes into account input disturbance as well as modeling errors."

Concurrently, G. Iyengar also proposed robust dynamic programming [3], a method aimed to achieve robust RL. It defines the method as:

> "Systematically mitigate the sensitivity of the dynamic programming optimal policy to ambiguity in the underlying transition probabilities"

More recent survey papers suggest the following definitions, which align with the above Moriomoto and Doya's definition.

> "Robust RL aims to learn a robust optimal policy that accounts for model uncertainty of the transition probability to systematically mitigate the sensitivity of the optimal policy in perturbed environments" [4]

> "robustness—in the scope considered in this survey—refers to the ability to cope with variations or uncertainty of one's environment. In the context of reinforcement learning and control, robustness is pursued w.r.t. specific uncertainties in system dynamics, e.g., varying physical parameters" [5]

In this paper, we slightly broaden the definition and consider uncertainty in all information the agent receives (not just uncertainty in the environment). We propose then to consider *robust RL* as follows.

Definition 1 (Robust RL)**.** *Robust RL methodologies are those than can cope with (or systematically mitigate the sensitivity of) all the relevant sources of uncertainty of the environment, taking into account any other information that the agent receives.*

3.2 Safe reinforcement learning

Following a similar procedure, according to the Cambridge dictionary[3], the general definition of the word *safe* is "not in danger or likely to be harmed". Also according to Wikipedia[4], the word *safety* is defined as "the state of being safe, the condition of being protected from harm or other danger. Safety can also refer to the control of recognized hazards in order to achieve an acceptable level of risk."

While the concept of *safety* in RL is relatively clear, the terminology *safe RL* is heavily overloaded. Roughly the following four aspects of safe RL are considered in the literature.

I. Consistent performance.
II. Maintaining safety constraints.
III. Aligned with the true objective.
IV. Accepting human intervention.

I) Consistent performance. requires the agent to always perform well in various conditions. This requirement is similar to the robust RL.

II) Maintaining safety constraints. requires the agent to maintain certain constraints defined by the system.

[3] https://dictionary.cambridge.org/dictionary/english/safe
[4] https://en.wikipedia.org/wiki/Safety

III) Aligned with the true objective. means that the agent's objective must be aligned with the task's true objective (or human intention). The agent cannot be safe if the given objective is not aligned with the true objective, no matter how good the agent algorithm is.

IV) Accepting human intervention. indicates that the agent (or system) must have a mechanism for human intervention (or "emergency stop"). We do not think this feature alone makes the agent safe; rather, this feature is mandatory for all real-world RL applications.

Below, we introduce several definitions of safe RL in literature with the Roman numbers (I to IV) that indicate which category the definition belongs.

The most generic (and high level) definition of *safe RL* would be by Javier Garcıa et al. [6]

> "the process of learning policies that maximize the expectation of the return in problems in which it is important to ensure reasonable system performance and/or respect safety constraints during the learning and/or deployment processes" I),II)

The definition by Shangding Gu et al. [7] explicitly includes adversary attacks while accommodating a broader sense of *safety*, which includes mitigating undesirable situations and reducing risk.

> "about optimizing cost objectives, avoiding adversary attacks, improving undesirable situations, reducing risk, and controlling agents to be safe" II)

Yongshuai Liu et.al. [8] and Lukas Brunke et al. [9] employ the concept of cost and define *safety* as controlling the cost. This definition is originated from a constrained Markov decision process (CMDP) [10] framework.

> "The safe RL agent's objective is to maximise long-term reward while keeping certain costs under their respective constraints." II)

Dylan Hadfield-Menell et al. [11] argue from slightly different prospective and define the *safe RL* as:

> "Safe RL has a mechanism for a human to interfere the agent effectively." IV)

This definition relates to the following note by Norbert Wiener [12] in one of the earliest explanations of the problems that arise when a powerful autonomous system operates with an incorrect objective.

> "If we use, to achieve our purposes, a mechanical agency with whose operation we cannot interfere effectively . . . we had better be quite sure that the purpose put into the machine is the purpose which we really desire." III) or IV)

Wiener's notion could lead to two lines of approaches to the safety of the autonomous system. One introduces an effective intervention mechanism, and the other has the right reward function, and an algorithm can achieve the goal – indeed, most *safe RL* approaches are belong to either of these lines.

In this paper, we define *safe RL* in the similar way as Javier Garcıa et al. [6], but also added aspects of III) and IV) as Dylan Hadfield-Menell et al. [11].

Definition 2 (Safe RL). *Safe RL is the process of learning policies that maximize the expectation of the return in problems in which it is important to ensure reasonable system performance and/or respect safety constraints during the learning and/or deployment processes. Also, the system must have the right objectives (the reward function aligned with the objective of the task) and a mechanism for humans to intervene.*

This definition also covers the robust RL aspect as well, so we use it as our working definition of safe and robust RL for the remainder of the paper.

4 Optimising a policy for Safe and Robust RL

This section outlines various training methodologies essential for developing a safe and robust RL agent. Initially, we present helpful concepts such as Markov decision process (MDP), robust MDP, and constrained MDP, which are foundational for understanding different optimisation strategies. Subsequently, we explore diverse optimisation criteria and techniques to fulfil these criteria to ensure a safe and robust RL.

4.1 Robust and Constrained Markov Decision Process

RL framework can be seen as a Markov decision process when the environment and the agent hold the Markovian property. It can be extended to a robust Markov process and a constrained Markov process. They are closely related to robust and safe RL, and they are very useful concepts to define some of safe and robust RL algorithms, so this section describes the definitions of these Markov processes.

A Markov decision process (MDP) is defined as a tuple $(\mathcal{S}, \mathcal{A}, r, \mathcal{P}, \mu, \gamma)$ [1], where $\mathcal{S}$ is the set of states, $\mathcal{A}$ is the set of actions, $r : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \mapsto \mathbb{R}$ is the reward function, $\mathcal{P} : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \mapsto [0, 1]$ is the state transition probability function, $\mu : \mathcal{S} \mapsto [0, 1]$ is the initial state distribution and $\gamma \in [0, 1]$ is the discount factor for the future reward. A policy $\pi : \mathcal{S} \mapsto P(\mathcal{A})$ is a mapping from states to a probability distribution over actions. A standard MDP aims to learn a policy π that maximises the discounted cumulative reward:

$$\arg\max_{\pi} J_r^{\pi} = \mathbb{E}_{\tau\sim\pi}\left[\sum_{t=0}^{\infty} \gamma^t r\left(s_t, a_t, s_{t+1}\right)\right], \tag{1}$$

where $\tau = (s_0, a_0, s_1, a_1, \cdots)$ denotes a trajectory, and $\tau \sim \pi$ denotes trajectories sampled from the policy π.

An robust Markov decision process (RMDP) extends the definition of the standard MDP by introducing $\mathscr{P}$ an uncertainty set for the state transition probabilities. An RMDP is defined as a tuple $(\mathcal{S}, \mathcal{A}, r, \mathscr{P}, \mu, \gamma)$. It guarantee the highest discounted accumulative reward with the given uncertainty set:

$$\arg\max_{\pi} J_{r,\mathscr{P}}^{\pi} = \inf_{\mathcal{P}\in\mathscr{P}} \mathbb{E}_{\tau\sim\pi,\mathcal{P}}\left[\sum_{t=0}^{\infty} \gamma^t r\left(s_t, a_t, s_{t+1}\right)\right], \tag{2}$$

where $\tau \sim \pi, \mathcal{P}$ denotes sampled trajectories from the policy π and the state transition probability $\mathcal{P}$. The uncertainty set $\mathscr{P}$ is typically for the state transition probabilities. However, in general, it can be for any parameters in the target environment.

A constrained Markov decision process (CMDP) is also a concept that extends the standard MDP by introducing cost functions C in addition to the reward function. It is defined as a tuple $(\mathcal{S}, \mathcal{A}, r, C, \mathcal{P}, \mu, \gamma)$. The cost functions $c_i \in C, c_i : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \mapsto \mathbb{R}$ are constrained suitably for the target application. The types of constraints are discussed in the following section. A CMDP aims to learn a policy π that maximises the discounted cumulative reward while it satisfies all of its necessary constraints. Formally, the CMDP becomes the following conditional optimisation problem:

$$\begin{gathered}\arg\max_{\pi} J_r^\pi = \mathbb{E}_{\tau\sim\pi}\left[\sum_{t=0}^{\infty}\gamma^t r\left(s_t, a_t, s_{t+1}\right)\right], \\ s.t.\ J_{c_i}^\pi \leq \epsilon_i \forall i,\end{gathered} \tag{3}$$

where $J_{c_i}^\pi$ denotes a statistical measure over the i-th cost function values from the trajectory sampled with the policy π and ϵ_i is the allowed upper bound for the measure. Various types of constraints are realised by defining appropriate $J_{c_i}^\pi$.

4.2 Optimisation criteria

In this subsection, we will review two optimisation criteria commonly used for safe and robust RL. For a normal RL setting, the optimisation criterion is to maximise the total (discounted) reward. A safe and robust RL scenario requires slightly different criteria. The first type is robust RL criteria, which aims to maximise the expected total reward under some worst-case scenarios or distributional assumptions. The second type is constrained RL criteria, which imposes additional constraints on the agent's actions or outcomes. We will describe these criteria in the rest of this section, and discuss some approaches to achieve them in the following section.

Robust RL criterion

The former optimisation criterion indicates that the agent consistently achieves a certain level of accumulated reward within a given uncertainty in the RL framework. Often, it assumes a certain level of uncertainty in the environment and maximises the expected total reward in the worst case. This category is equivalent to the robust RL [2]. This task setup is also often referred to RMDP [3].

Constrained RL criterion

The latter definition is more common in the recent safe RL literature. Typically it introduces a cost function in addition to the reward function, and the agent tries to maximise the expected accumulated reward while maintaining particular constraints regarding the cost. The constraints can be categorised as Fig.3 [8]. The cumulative constraints require the sum or mean of a given cost to be within a specific limit. e.g. for an electric car navigation task, the sum of energy consumption (cost) must be less than the available battery (limit) before reaching a charging station (or home). The

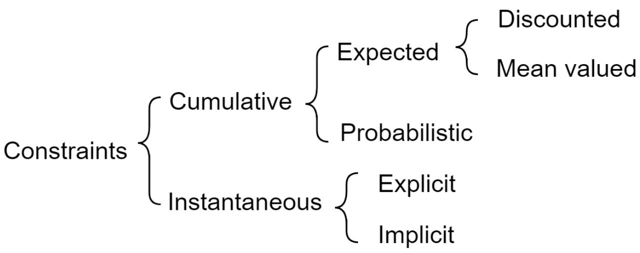


Fig. 3 Categories of constraints.

cumulative constraints are split into two categories – expectations and probability of the costs. The former includes discounted cumulative constraints and mean-valued constraints [10].

An expected discounted cumulative constraint is of the form

$$J_{c_i}^{\pi} = \mathbb{E}_{\tau \sim \pi} \left[\sum_{t=0}^{\infty} \gamma^t c_i \left(s_t, a_t, s_{t+1} \right) \right] \leq \epsilon_i. \tag{4}$$

An expected mean valued cumulative constraint is of the form

$$J_{c_i}^{\pi} = \mathbb{E}_{\tau \sim \pi} \left[\frac{1}{T} \sum_{t=0}^{T-1} c_i \left(s_t, a_t, s_{t+1} \right) \right] \leq \epsilon_i. \tag{5}$$

While the probabilistic constraints [13] require the probability of the cumulative costs exceeding a threshold (η) is less than ϵ_i.

$$J_{c_i}^{\pi} = P \left(\sum_{t} c_i \left(s_t, a_t, s_{t+1} \right) > \eta \right) \leq \epsilon_i. \tag{6}$$

Instantaneous constraints are constraints on the actions, states or cost functions that must satisfy for each time step. They can be further split into explicit and implicit cases. An explicit constraint has a closed-form expression that can be numerically checked. On the other hand, an implicit constraint does not have an accurate closed-form formulation due to the complexity of the system. Hence it requires learning a function that checks if a given state action pair satisfy requirements. The instantaneous constraints form as

$$J_{c_i}^{\pi} = c_i \left(s_t, a_t, s_{t+1} \right) \leq \epsilon_i. \tag{7}$$

It is hard to show a generic form of implicit constraints. However, instantaneous probabilistic constraints are one of the implicit constraints and can be form of

$$J_{c_i}^{\pi} = P \left(c_i \left(s_t, a_t, s_{t+1} \right) > \eta \right) \leq \epsilon_i. \tag{8}$$

4.3 Optimisation methods

Here, we explore strategies to achieve the criteria introduced in the previous section. We begin by considering optimisation methods for exploitation policy, which solely aim to achieve the optimisation criteria introduced in the above subsection. Subsequently, we introduce methods for safe exploration, which aim to gather diverse training data for training the exploitation policy while maintaining safety.

Optimising exploitation policy for robust RL criterion

For the robust RL criterion, the most straightforward approach would be maximising the reward in the worst case. There are three lines of approaches to achieving empirically robust performance – i) robust adversarial approaches, ii) domain randomisation approaches and iii) approaches employing a statistical measure.

i) The robust adversarial approaches combine RL with adversarial learning that learns a policy to control the environment's parameters to minimise the reward that RL agent obtains while the RL learns a policy maximises it [14, 15]. These two policies are updated alternately (each is updated while fixing the other), attempting to progressively improve the robustness of the RL agent's policy and the strength of its adversary. The difficulty of the approach is that it requires a balance between the ability of RL agent and its adversarial agent. If the adversarial agent possesses too much control ability, RL agent fails to learn any helpful policy, while too little control results in not robust RL policy.

ii) The domain randomisation approaches learn a policy that empirically generalises to a broader range of tasks or environments. Instead of considering the worst-case, it learns from an environment with randomly perturbated parameters [16, 17]. These parameters often have pre-specified ranges.

iii) The approaches employing a statistical measure. Some of the approaches for maximising the worst case scenario may end up with a too-pessimistic solution and performs poorly in the typical condition. An alternative approach would consider a soft worse case, which employs a statistical metric instead of the worst reward such as CVaR [18] or certain percentile point. These statistical measures can be obtained by employing distributional reinforcement learning approaches [19–21] (such as implicit quantile networks (IQN) [21]) that learn the distribution of the value function. However, IQN alone does not capture the Epistemic uncertainties, which is the uncertainty due to lack of knowledge. Epistemic uncertainty can be captured by employing an ensemble approach. This approach is proposed as ensemble quantile networks (EQN) [22] that combines IQN and the ensemble approach to capture Epistemic and Aleatoric uncertainties and learn an appropriate value function distribution for computing the statistical measures like CVaR.

Optimising exploitation policy for constrained RL criterion

For the constrained RL criterion, Lagrangian relaxation is the most straightforward approach to address the constraints [10, 23, 24]. The general form of Lagrangian relaxation is to reduce the problem to an unconstrained problem with Lagrange multipliers.

These adaptive Lagrange multipliers are then used to penalise constraint violations as

$$\min_{\lambda_i \geq 0} \max_{\pi} L(\pi, \lambda) = J_r^{\pi} - \sum_i \lambda_i \left(J_{c_i}^{\pi} - \epsilon_i \right) . \tag{9}$$

However, this approach is sensitive to the initialisation of the Lagrange multipliers and the learning rate. Moreover, the Lagrangian multipliers are solved on a slower time scale, making it difficult to optimise in practice [25, 26]. Another alternative approach employs a particular function to incorporate the constraints and merge them into a single objective without Lagrange multipliers. One such example is interior point optimisation (IPO) [27], a first-order constrained method inspired by the interior-point method [28]. IPO employs logarithmic barrier functions as penalty functions to accommodate the constraints as Eq. 10. It shows the log term go to minus infinity as $J_{c_i}^{\pi}$ getting closer to ϵ_i.

$$\max_{\pi} J_r^{\pi} + \sum_i \frac{1}{t_i} \log \left(-J_{c_i}^{\pi} + \epsilon_i \right) , \tag{10}$$

where t_i is a hyperparamter.

For methods based on strong theoretical justifications, a line of works [29, 30] guarantees performance improvement for each policy update. Trust region optimization (TRPO) [31] adopted it to deep neural network (DNN) parameterised polices, and its successor proximal policy optimization (PPO) [32] established better empirical performances with a much simpler algorithm. Constrained policy optimization (CPO) [25] is inspired by TRPO, and it guarantees keeping the constraints and performance improvement for each iteration. However, CPO is computationally expensive as it uses conjugate gradients for approximating the Fisher Information Matrix, whose approximation error affects the overall performance. Furthermore, CPO only supports constraints that satisfy the Recursive Bellman Equation (i.e. discounted sum constraints), and it is difficult to accommodate multiple constraints.

Exploration policy

In the standard RL setting, the training dataset, or trajectory, is generated through interaction with the environment. This means that the quality of the training data is determined by the policy guiding these interactions. In essence, the exploration policy has a direct impact on the quality of data available for training the agent. It's critical that the policy not only seeks out new states and actions to enrich the training data but also ensures safety. As further elaborated in this paragraph, striking a delicate balance between safety and exploration is essential.

Common RL exploration strategies often rely on some stochasticity in the action choices, such as the ϵ-greedy algorithm. Although they are simple and yet effective in many RL tasks, it does not consider any risk. Hence it might cause many catastrophic failures during the learning process. Moreover, it may result in constant failures even after the learning process due to the randomness in the action choices. Hence, some of the safe RL approaches pay special attention to the exploration process, considering some form of risk. Unfortunately, it is impossible to avoid undesirable consequences completely without accessing a certain amount of external knowledge (or prior knowledge) of the environment. One possible safe exploration strategy is employing Bayesian approaches (e.g. GP [33]) for the environment modeling. It can take prior knowledge and predict future trajectories with uncertainties. It allows the agent to explore a region that might result in high reward while maintain constraints by considering the predictive uncertainties [34, 35]. The limitations of these GP based approaches are difficult to extend to high-dimension state space.

The approaches that do not rely on external knowledge, can not avoid catastrophic failures completely, but they attempt to minimise the failures by considering the risk of the exploration. For example, Moldovan & Abbeel [36] consider safety using ergodicity, where an action is safe if it is still possible to reach every other state after having taken that action. These methods are limited to small, discrete MDPs where exact planning is straightforward. For the algorithms can be applied to complex and high-dimensional tasks, Wachi et al. propose MASE [37] – algorithm employs an uncertainty quantifier for a high-probability guarantee that the safety constraints are not violated and penalises the agent before safety violation, assuming that the agent has access to an "emergency stop" authority – namely a human intervention. Han et al. [38], and Eysenbach et al. [39] propose approaches that learn two policies – one is for maximising the reward (standard RL policy), and the other is for bringing the state back to the initial state (reset policy). Then, they use the reset policy to recover from a potentially unsafe state to maintain their safety. These approaches could fail before they learn when it should use the reset policy and the reset policy itself. Other simpler approach, Gehring and Precup [40] define the risk as a magnitude of temporal difference error for the value function. It discounts an expected future reward with the risk to discourage taking risky actions.

As we discussed in the previous section, approaches like TRPO, PPO and CPO are based on a theoretical guarantee that each policy update improves the performance. Hence, they can be seen as safe exploration approaches. However, they do not have built-in mechanisms to ensure avoiding catastrophic failures. Furthermore, PPO has been found to suffer from a lack of exploration [41]. So, they might need additional mechanisms to ensure safe exploration.

5 Incorporating additional data and knowledge

In reinforcement learning, an agent learns from its own interactions with the environment. However, the agent may sometimes have access to additional knowledge sources or data that can help it improve its performance. For example, the agent may use expert demonstrations, human feedback, or prior knowledge about the task or the environment. This section will explore how to incorporate such knowledge or data into reinforcement learning algorithms. We contrast this approach with the one we discussed in the prior sections, where the agent only uses data (trajectory) obtained by itself. However, some approaches in the prior sections can also be used here. There are various forms of external knowledge – trajectory dataset, a computer model (simulator) of the environment and a person with knowledge of the environment. Each form of knowledge holds different characteristics and hence requires a different approach to extract and incorporate the information. We discuss each of them in the following subsections.

5.1 Trajectory dataset

If the trajectory dataset is obtained from the interactions between a target environment and an expert human (or an algorithm), then we could rely on a behaviour cloning (BC) type of approach that simply learns a mapping from a state to an action from the dataset. If the dataset is obtained from interactions between a target environment and non-expert, we could employ one of offline RL approaches [42], which learns an optimal policy from the dataset while avoiding actions out of the dataset distribution. In case of the dataset is sampled from a similar but not target environment, it requires robust RL approach, which takes into account the unknown difference between these environments. It learns a policy that performs well in the worst case of given uncertainty in the environment [2, 3]. Alternatively, we can utilise meta-learning to acquire a prior for the target environment. An important note in meta-learning is that the learned prior should cover the actual environment [43, 44]. Achieving this requires a dataset that encompasses diverse realisations of the environment. If such diversity is lacking, we may need to relax the prior distribution to ensure it adequately covers the true environment.

The quality of a trajectory dataset is a crucial factor. Ideally, an expert demonstration dataset should encompass all states an agent will likely encounter. If the dataset is obtained from non-experts, ideally, it includes all state-action pairs [45]. Even when a single trajectory does not encompass the entire optimal path, RL algorithms are capable of learning optimal behaviour from these suboptimal trajectories. This capability, known as *stitching*, is a vital attribute of offline RL algorithms [46]. Without the *stitching* ability, an RL algorithm would require a dataset that contains the complete optimal trajectory [47].

5.2 Simulators

Computer models can simulate the environment for many applications and can be used to train an RL agent or create a training dataset. Computer models are often more accessible and less risky than the actual environment, as they do not incur any costs or consequences for failing a task. Therefore, they are useful for pre-training the agent before deploying it to the real environment [48]. However, a naive approach may not work well due to the discrepancy between the computer model and the real environment. A possible solution to the issue would be a robust RL approach. This problem is known as ”sim-to-real” and has been studied extensively [49, 50].

5.3 Human knowledge

Human knowledge can be a valuable source of guidance for safe reinforcement learning, but it also poses some challenges. How can we obtain human knowledge in a way that is efficient, reliable and scalable [51]? We could use different methods of human input, such as demonstration (where humans provide examples of desired behaviour) [52], feedback (where humans evaluate the agent's actions) [53] or intervention (where humans correct the agent's actions) [54, 55]. Owing to the diverse strategies available for incorporating human knowledge, a dedicated section (Sec. 6 human-in-the-loop) is available for further details.

6 Human-in-the-loop

A human-in-the-loop approach is another possible approach for achieving safe RL. The most robust approach of this category would have a mechanism for a human to intervene in the agent action [54, 56]. When the human finds that the agent's action violates constraints, the human takes over the system and applies alternative action instead of the agent. This method guarantees zero violation of the constraints assuming the human can always provide the right actions. However, it is hard to scale to complex environments because the human cost would be prohibitively high. Alternatively, some works propose having a machine-learning model for intervening and replacing unsafe actions with safe ones [55, 57], yet they still require the model to be reliable. Other approaches are that humans advise which actions to take or give feedback regarding the actions the agent just took (i.e. right or wrong) [53, 58, 59]. The advice or feedback will guide the agent's learning process and help achieve a good policy quickly.

Reinforcement learning with human feedback is a research area that has gained renewed attention in recent years, especially with the application of ChatGPT [60, 61], a conversational agent trained with human preferences. This approach addresses the challenge of defining a suitable reward function for complex tasks by learning from the feedback of human evaluators. This line of research is primarily based on preference-based reinforcement learning (PbRL) [62–67], which learns human preference through relative feedback, such as pair-wise comparisons and rankings. They model human feedback with the Bradley-Terry model [68] for the pair-wise comparisons and Plackett-Luce model [69, 70] for the rankings. The Bradley-Terry model is a special case of the Plackett-Luce model, and it was first introduced by Zermero [71] and heavily studied in the years since, particularly following its rediscovery by Bradley and Terry [68]. It learns a reward function from the human feedback and then train an agent with the learned rewards, or the agent learn a policy directly from the human feedback.

Furthermore, human feedback can help RL agents overcome some of their challenges, such as sparse rewards, misaligned objectives or unsafe exploration [58, 72]. It provides additional guidance, correction, or evaluation of their behaviour through feedback. Therefore, RL with human feedback is an important research area for developing safe and robust RL systems that align with human values and preferences.

7 Related problems and formulations

7.1 Complex reinforcement learning paradigms

This paper primarily focuses on the standard RL setting. However, in this section, we touch on safety and robustness issues on other RL settings – namely, multi-agent RL and hierarchical RL settings.

7.1.1 Multi-agent RL

Multi-agent reinforcement learning (MARL) is a sub-field of RL that focuses on investigating the behaviour of multiple learning agents that coexist in a shared environment. Each agent is motivated by either the global rewards or its own rewards, developing interesting behaviours that can be characterized as collaborative [73]. In some environments, these individual rewards may be opposed to other agents' rewards, resulting in complex group dynamics.

Safe MARL works often define a multi-agent version of CMDP. We found three types of the definitions. Some works define the multi-agent version of CMDP as a tuple $(\mathcal{S}, \{\mathcal{A}_j\}_{j\in\mathcal{N}}, r, \{C_j\}_{j\in\mathcal{N}}, \mathcal{P}, \mu, \gamma)$. Where $\mathcal{N} = [1, \dots, n]$ is a set of agents, $\mathcal{A}_j$ is the action space for the agent j and C_j is the set of cost functions for the agent j. With this formulation, the rewards are common to all agents, while the cost functions and constraints can be different between agents. Hence, the agents try to maximise the common rewards while maintaining each of their constraints. Each agent's policy π_j is optimised as the following equations [74, 75].

$$\begin{gathered}\arg\max_{\pi_j} \mathbb{E}_{\tau\sim\pi_j}\left[\sum_{t=0}^{\infty}\gamma^t r\left(s_t, a_t, s_{t+1}\right)\right], \\ s.t.\ J_{c_i}^{\pi} \leq \epsilon_i \forall i,\ \text{where } c_i \in C_j.\end{gathered} \tag{11}$$

The second formulation define it as $(\mathcal{S}, \{\mathcal{A}_j\}_{j\in\mathcal{N}}, \{r_j\}_{j\in\mathcal{N}}, \{C_j\}_{j\in\mathcal{N}}, \mathcal{P}, \mu, \gamma)$, where r_j is the reward function for the agent j. This formulation assumes different reward functions amongst agents. The agents try to maximise their own rewards and maintain their constraints. So, each agent's policy π_j is optimised as the following equations [76].

$$\begin{gathered}\arg\max_{\pi_j} \mathbb{E}_{\tau\sim\pi_j}\left[\sum_{t=0}^{\infty}\gamma^t r_j\left(s_t, a_t, s_{t+1}\right)\right], \\ s.t.\ J_{c_i}^{\pi} \leq \epsilon_i \forall i,\ \text{where } c_i \in C_j.\end{gathered} \tag{12}$$

The last definition is for a distributed scenario, and it defines their CMDP as $(\mathcal{S}, \{\mathcal{A}_j\}_{j\in\mathcal{N}}, \{r_j\}_{j\in\mathcal{N}}, \mathcal{G}, \{C_j\}_{j\in\mathcal{N}}, \mathcal{P}, \mu, \gamma)$, where $\mathcal{G} = (\mathcal{N}, \mathcal{E})$ indicates available communication link between agents. Each agent receives its own rewards and it tries to maximise the average rewards across all agents [77].

$$\begin{gathered}\arg\max_{\pi} \mathbb{E}_{\tau\sim\pi}\left[\sum_{t=0}^{\infty}\sum_{j=1}^{n}\frac{1}{n}\gamma^t r_j\left(s_t, a_t, s_{t+1}\right)\right], \\ s.t.\ J_{c_i}^{\pi} \leq \epsilon_i \forall i,\ \text{where } c_i \in C_j,\end{gathered} \tag{13}$$

where $\pi = \{\pi_j\}_{j=1:n}$ is a set of all agents' policy.

Multi-agent constrained policy optimisation (MACPO) [74] is the multi-agent version of CPO algorithm [25]. It is the first model-free safe MARL algorithm and guarantee monotonic improvement in reward, while theoretically satisfying safety constraints. However, it is computationally expensive and the algorithm has some approximations which causes some errors and troubles in practice [7]. Multi-agent proximal policy optimisation (MAPPO) [74] is also model free safe MARL algorithm which simplify the MACPO algorithm. However, MACPPO does not provide a guarantee for the constraints. Safe decentralized policy gradient (Safe Dec-PC) [77] is decentralised algorithm for safe MARL. It uses the third multi-agentCMDP model in above and assumes each agent has communication link to other (neighbour) agents.

7.1.2 Hierarchical RL

Hierarchical reinforcement learning (HRL) is a method that decomposes a reinforcement learning problem into a hierarchy of subproblems or tasks. The most common HRL agent structure consists of two layers of hierarchy. The higher layer invokes the lower-level agents by giving them a sub-goal to achieve or selecting a policy from multiple lower-level policies. The higher layer is trained to maximise the rewards from the target environment, while the lower level policies are trained to maximise intrinsic rewards generated by the high-level agent.

There are some works have been done for safe HRL, however they are quite limited. Hierarchical safe reinforcement learning (HiSaRL) [78] is a two-level hierarchy method; the high-level agent generates a safe and efficient path, and the low-level agent ensures runtime safety with a Lyapunov function-based approach. Hierarchical program triggered reinforcement learning (HPRL) [79] employs a structured program for the high-level agent, and it triggers one of the low-level policies that are trained for a specific movement (car manoeuvres, i.e. turn left/right). The structured program in the high-level agent defines rule-based safety specifications, and formal verification is used to ensure safety. The authors of [80] propose a two-level hierarchical RL algorithm with the high-level agent providing a sub-goal to the low-level agent. The low-level agent is trained to maximise the intrinsic rewards that the high-level agent generates. It also has a safety layer that replaces a potentially risky action generated by the low-level agent with a safe action. The high-level agent is trained to produce effective sub-goals that maximise the rewards and minimise the safety layer intervention. Limitations of this approach are: i) The safety layer only looks one time-step ahead; hence, it cannot effectively intervene for long time horizon constraints. ii) Its safety guarantee is heavily dependent upon the accuracy of the one-step-ahead cost prediction model. It might be hard to assess the reliability of the DNN model.

7.2 Beyond reinforcement learning

In what follows below, we discuss problems in fields that are closely related to the robust and safe RL space. Some of these areas are too large to be properly described in here, so we just touch the each of them and highlight the similarities and differences to the robust and safe RL approaches mentioned above.

7.2.1 Control theory

The relationship between control theory and RL is that both fields share similar goals, i.e. coming up with a good sequence of actions to achieve desired outcomes. Control theory starts with a known model dynamics (environment's state transitions), while

RL assume they are unknown and learn them from interacting with the environment. Control theory can provide insights, methods and guarantees for safe and robust RL. Meanwhile, RL can extend the applicability and scalability of control theory to more complex and data-driven scenarios. Although the difference of the assumption about the model dynamics, they are closely related and idea developed in one field often can be applied to the other field e.g. model predictive control (MPC) was originally proposed and developed in the control theory society, is commonly used in RL field especially for the model-based RL planning approaches. Likewise, Lyapunov functions are widely used in control theory to prove the stability of a system, and they can also be applied to reinforcement learning to provide its stability [81] and safety [82, 83].

7.2.2 Transfer learning

Transfer learning is a technique that aims to improve the learning efficiency of a machine learning model on a target task by transferring the knowledge contained in different but related tasks [84–86]. Much work has been done to apply transfer learning for RL [87–90]. The main benefit of transfer learning for RL is that it can reduce the dependence on many interactions with the target environment, which may be expensive, scarce, or unsafe to obtain. By exploiting the similarities between tasks, transfer learning can improve the learning efficiency and the quality of the learned policies.

7.2.3 Meta-learning

Meta-learning, or learning to learn, is the science of systematically observing how different machine learning approaches perform on a wide range of learning tasks and then learning from this experience, or meta-data, to learn new tasks much faster than otherwise possible [91–93]. Meta-learning also benefits safe and robust RL by exploiting knowledge from a wide range of tasks and provides a good starting point [94] for the agent or priors to the model parameters [43, 95]. Both can significantly reduce the risk of unsafe explorations.

7.2.4 Sim-to-real

Sim-to-real is a research area that investigates how to transfer reinforcement learning (RL) agents from simulated environments to real-world settings [49, 50, 96]. This is especially relevant for robotics applications, where RL can enable agents to learn complex and adaptive behaviours but also possess difficulties due to the large amount of data needed to learn. Employing the target environment simulator can reduce the cost and risk of training RL agents, but it also introduces a discrepancy between the simulation and the reality, known as the sim-to-real gap. This gap can cause the agent to fail or unpredictable behaviour when deployed in the real world. Sim-to-real techniques aim to develop learning algorithms to produce robust models that can handle the gap and ensure safe and reliable performance.

8 Ethical considerations of safe and robust RL

RL poses significant ethical challenges, especially when applied to real-world tasks that involve human or environmental impacts [97]. For an in-depth review of the ethical implications of artificial intelligence in general, we refer the reader to [98, 99] and the references therein. Additionally, [100] and similar efforts, provide guidelines to address such ethical considerations for practical applications. In this section, we will focus specifically in some of the main ethical risks that arise from safe and robust RL applications, such as safe rewards, accountability and transparency.

Reward misspecification

RL agents learn a policy that maximises the expected sum of the future rewards. So, if the reward function does not match the true objective of the task (reward misspecification), then the learned policy can be useless or even harmful for people around the agent or the environment (reward hacking [101]). For example, an RL agent that controls a self-driving car may learn to drive recklessly if the reward function only considers speed and not safety. Therefore, designing reward functions that align with the desired outcomes and values of the stakeholders is a crucial ethical challenge in RL. To mitigate the risk, we can consider CMDP framework to properly model various constraints of the task – if the task is complex and needs to consider various conditions to maintain its safety and fairness. The challenges for CMDP are that it often requires high computational power to train the agent, and also, it is hard to guarantee to satisfy all the constraints all the time. CMDP requires humans to specify all the constraints and rewards correctly, and it is hard to specify all the constraints for some applications. To mitigate the risk, we could rely on human feedback to estimate the reward function. However, human feedback can be sparse and inaccurate. So, it is required to make the algorithm robust against the sparsity and inaccuracy of the feedback.

Transparency and accountability

Because RL tasks involve a sequence of multiple decisions for the agent's action, it is more difficult to explain the reason behind the decisions than in standard machine learning settings. Also, RL agents learn from their own interactions with the environment. It means that the agent's performance could vary depending on its previous policy. That makes it difficult to guarantee its performance and provide accountability to the agent [102, 103]. One possible workaround for the issues is to use the RL agent as a decision support system. In this case, the RL agent provides suggestions for the human action, and then the human makes a final decision about which action to take [104]. The human is accountable for the decision. To make such a system work, the RL agent must provide a reason behind its suggestions so that the human makes a good final decision. The limitation of this approach is that, for many applications, it is not feasible to accommodate human intervention.

Overall, the ethical issues of RL has two aspects – a risk of misspecified reward and lack of transparency (explainability) of the agent. Safe and robust RL can mitigate the first issue, but not so much for the latter one directly. However, the uncertainty estimation algorithms, that is a important component of safe RL, helps the transparency of the agent. So, safe and robust RL approaches should be able to help making ethical algorithms. However, it is still requires attention to these ethical issues.

9 A checklist for safe and robust RL

This section provides practitioners with a list of items and actions to check in order to design safe and robust RL agents. Some of these elements are general principles that apply to any RL problem with particular implications around safety and robustness, while others are specific to certain domains or scenarios. These are organised in four parts, namely, `specification`, `additional sources of information`, `optimisation` and `safety`. We expect practitioners to navigate through the list on a sequential manner.

Specification

First of all, we need to clarify the requirements of the task at hand. In addition to the normal RL setting specifications, we should consider the following items:

S1) **Ethical requirements.** For real-world applications, it is important to comply ethical requirements which is often specific to a target application. As the first step in a specification stage, we must list up all ethical requirements. They might be related to its transparency (explainability), its fairness (no discriminatory behaviour) or its privacy (protecting user privacy).

S2) **Reward function.** If the reward (or cost) function is defined, ensure it is well aligned with the true objective of the task, and there is no space of *reward hacking*. If the reward function is hard to define, consider relying on human feedback approaches discussed in Sec. 6.

S3) **Variations.** If the environment can potentially be non-stationary over the foreseen deployment period, the agent must be robust against such variations. We need to specify how much variation could happen (specify uncertainty set) and make the agent robust against the variations with approaches discussed in Sec. 4.3 for robust RL criteria.

S4) **Constraints.** If there are any constraints the task needs to maintain, we need to specify them explicitly and persist them rigorously. Several forms of constraints are explained in Sec. 4.2 Constrained RL criterion. Then, the agent must have a method in place to satisfy the constraints.

Additional sources of information:

Next, we consider what kind of additional knowledge we can exploit. As it is always challenging to maintain safety at the early stage of training, we must exploit any knowledge available prior to the agent interacting with the environment.

A1) **Data.** If there are any trajectory data available, we could exploit them with offline RL, meta-learning or simple BC algorithm to learn an initial policy, and then we train it online further. If the trajectory data is obtained from interactions between an expert and the target environment, BC would be fine. However, it is not from an expert, we use offline RL algorithm to recover the best policy within the data distribution. If it is not from the actual target environment but from many environments containing the target environment, then we could rely on a meta-learning approach to learn the distribution of environment parameters.

A2) **Simulator.** If there is a computer simulator for the target environment, we can rely on it to train the agent. However, it is important to note that there are always some differences between the simulator and the real world. We can employ sim-to-real algorithms (touched in Sec. 7.2) or robust RL approaches (Sec. 4.2) to overcome the gap.

A3) **Expert.** If there is anyone who knows the task well, obtaining demonstration data or annotations on existing trajectory data is worth considering. Although extracting helpful information from a human is difficult, such data from experts is valuable, especially when the reward is sparse. The expert data directly gives information about the best action at every time step. With the demonstration data, we can rely on offline RL or BC algorithm to obtain the initial policy. With the expert annotations on the existing dataset, we can rely on a human feedback algorithm to extract the expert's policy.

A4) **Non-experts.** Obtaining demonstration data or annotations on existing trajectory data from many non-experts is very useful. It is not as data efficient as the expert's data. However, it is possible to work out the best policy from it. Again, we can use the offline RL approach with the demonstration data and one of the human feedback approaches with the annotations. They could work out good policy from the data with mixed quality.

Optimisation criteria/method

We now consider algorithms for optimising/training the agent (policy). We consider three aspects below. They are not mutually exclusive; you likely need to guarantee these three aspects.

O1) **Robustness.** To maintain a certain level of robustness, we need to employ approaches introduced in Sec. 4.3 For robust RL criterion. Under given uncertainties of the environment, these approaches maximise the minimum (the worst case) rewards.

O2) **Safety.** For keeping the specified constraints, several approaches are discussed in Sec. 4.3. For constrained RL criterion. Some of them are strong theoretical justifications but computationally expensive. Others are relatively simple algorithms but only empirically proven.

O3) **Exploration.** Exploring while maintaining safety (constraints) is probably the most challenging objective in RL tasks. Exploring tries something unknown (uncertain); hence, it always has a risk of failure. However, there are some approaches for safe exploration introduced in Sec. 4.3.

Safety layer

Finally, we consider applying an extra safety mechanism (safety layer) to guarantee/improve the safety and robustness. The possible approaches are:

L1) **Human intervention.** If the task is feasible to have a human intervention to prevent a failure, this would be the best mechanism to guarantee safety (Sec. 6). Still, it is important to reduce the number of human interventions by employing some of the abovementioned approaches. Also, it is crucial to show the current status of the environment in an easy-to-understand way so that the person can decide when to intervene.

L2) **Shielding (formal verification).** Suppose the constraint violation can be detected from the current state of the environment and the agent's action, and there is an action (or sequence of actions) to recover from such a possibly dangerous state. In that case, we can implement a safety layer that detects the constraint violation and replace the action with recovery action.

L3) **Shielding (adaptive).** If it is difficult to have the safety layers mentioned above, it is still possible to have a safety layer that learns when it should intervene and what action to take to recover from the potentially risky states. These approaches cannot guarantee safety, especially during learning. However, it could improve safety if it can learn from a trajectory dataset offline.

L4) **Traceability and explainability.** When the agent fails a task, it is essential to understand why it fails. Such *post-mortem examination* will help understand the failure mechanism and improve the agent algorithm to prevent similar failures in the future. Therefore, traceability (how the agent failed the task) and explainability (why the agent took actions that led to the failure) are important. They do not prevent failures in the current task, but they are essential for preventing similar failures in the future.

The basic level of traceability can be achieved simply by recording all trajectories (state, action and reward for every time step). However, further information on the agent's internal states might be required to understand the reason for the agent's action choices. These internal states are also required for explainability. Explainability can be achieved by tracing internal states combined with a mechanism to provide a human-understandable explanation. The mechanism to generate a human-understandable explanation can be challenging, especially when the agent utilises DNN, and itself is still a significant research area.

10 Conclusion

This paper explored the various aspects of safe and robust reinforcement learning (RL), delving into algorithmic frameworks, ethical implications, and practical considerations. The domain of safe and robust reinforcement learning is extensive and multifaceted, covering all relevant literature would be far beyond the scope of any single review. Our aim was to illustrate various dimensions of this vast field, providing a foundational understanding upon which readers can build. By categorizing existing safe RL algorithms, we have provided a structured overview that summarises the current state of this field. Our aspiration is that this work serves as a resource for a diverse audience, ranging from researchers new to safe and robust RL seeking to understand the overall structure of the field to practitioners aiming to implement safe and robust RL systems in real-world scenarios.

Appendix A Summary of literature and the timeline

Figure A1 and A2 show a summary of the safe RL and robust RL literature, respectively. They categorise the literature into groups and place it in chronological order (from top to bottom). Major categories are highlighted in colour-coded boxes with the relevant sections of this paper, while sub-categories are denoted by black boxes. The black arrows indicate that the paper inspires the other paper in a different category. It is important to note that this summary is not comprehensive; it focuses primarily on significant, recent contributions to the field.

Acknowledgments

This work is supported by the UKRI Turing AI Fellowship EP/V024817/1.

References

[1] Sutton, R.S., Barto, A.G.: Reinforcement Learning. The MIT Press, Cambridge, MA (1998)

[2] Morimoto, J., Doya, K.: Robust reinforcement learning. Neural computation **17**(2), 335–359 (2005)

[3] Iyengar, G.N.: Robust dynamic programming. Mathematics of Operations Research **30**(2), 257–280 (2005)

[4] Chen, S., Li, Y.: An overview of robust reinforcement learning. In: 2020 IEEE International Conference on Networking, Sensing and Control (ICNSC), pp. 1–6 (2020). https://doi.org/10.1109/ICNSC48988.2020.9238129

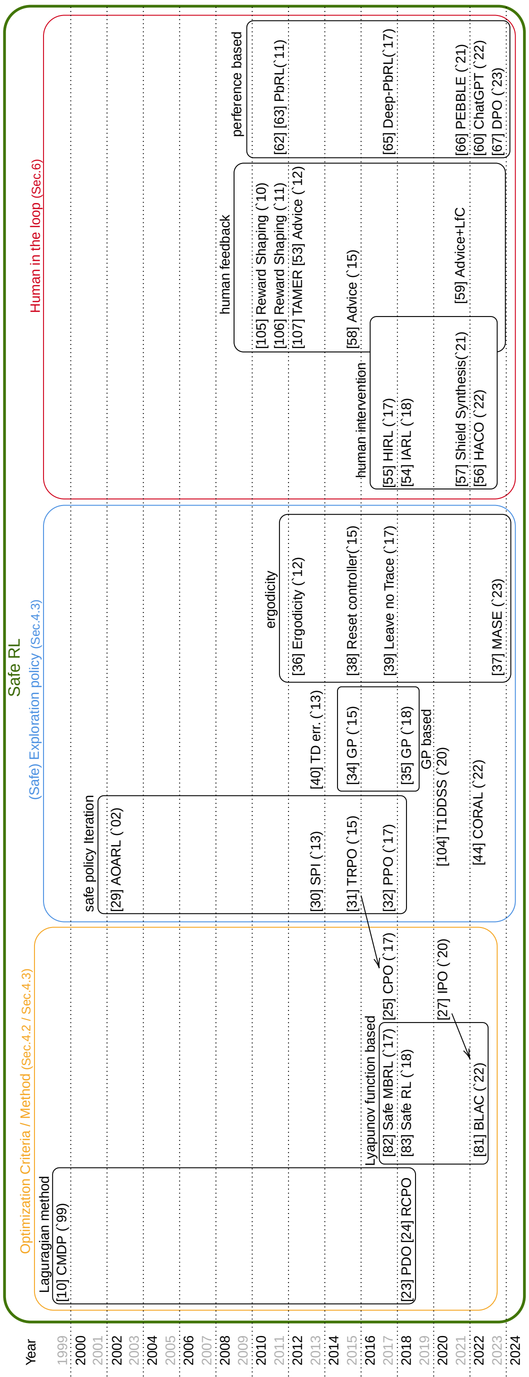
[5] Moos, J., Hansel, K., Abdulsamad, H., Stark, S., Clever, D., Peters, J.: Robust reinforcement learning: A review of foundations and recent advances. Machine Learning and Knowledge Extraction **4**(1), 276–315 (2022) https://doi.org/10.3390/make4010013

[6] Garcıa, J., Fernández, F.: A comprehensive survey on safe reinforcement learning. Journal of Machine Learning Research **16**(1), 1437–1480 (2015)

[7] Gu, S., Yang, L., Du, Y., Chen, G., Walter, F., Wang, J., Yang, Y., Knoll, A.: A review of safe reinforcement learning: Methods, theory and applications. arXiv preprint arXiv:2205.10330 (2022)

[8] Liu, Y., Halev, A., Liu, X.: Policy learning with constraints in model-free reinforcement learning: A survey. In: The 30th International Joint Conference on Artificial Intelligence (IJCAI) (2021)

[9] Brunke, L., Greeff, M., Hall, A.W., Yuan, Z., Zhou, S., Panerati, J., Schoellig,



Abbreviations

Advice+LfC Advice + Learning from a Crowd
AOARL Appromimately Optimal Approximate RL
BLAC Barrier Lyapunov Actor-Critic
CORAL Concept-drift Online Recursive Adaptation Learning
CMDP Constrained Markov Decision Process
CPO Constrained Policy Optimization
DPO Direct Policy Optimization
GP Gaussian Process
HACO Human AI Copilot Optimization
HIRL Human Intervention RL
IARL Intervention Aided RL
IPO Interior-point Policy Optimization
MASE Meta-Algorithm for Safe Exploration
PbRL Preference-based RL
PEBBLE PrE-training and preference-Based learning via relaBeLing Experience
PDO Primal-Dual Optimization
PPO Proximal Policy Optimization
RCPO Reward Constrained Policy Optimization
SPI Safe Policy Iteration
T1DDSS Type 1 Diabetes Decision Support System
TAMER Training Agent Manually via Evaluative Reinforcement
TD err. Temporal Difference Error
TRPO Trust Region Policy Optimization

Fig. A1 The figure organises safe reinforcement learning literature in chronological order, highlighted in colour-coded boxes for major categories and black boxes for sub-categories. Black arrows show inter-paper inspiration across categories.

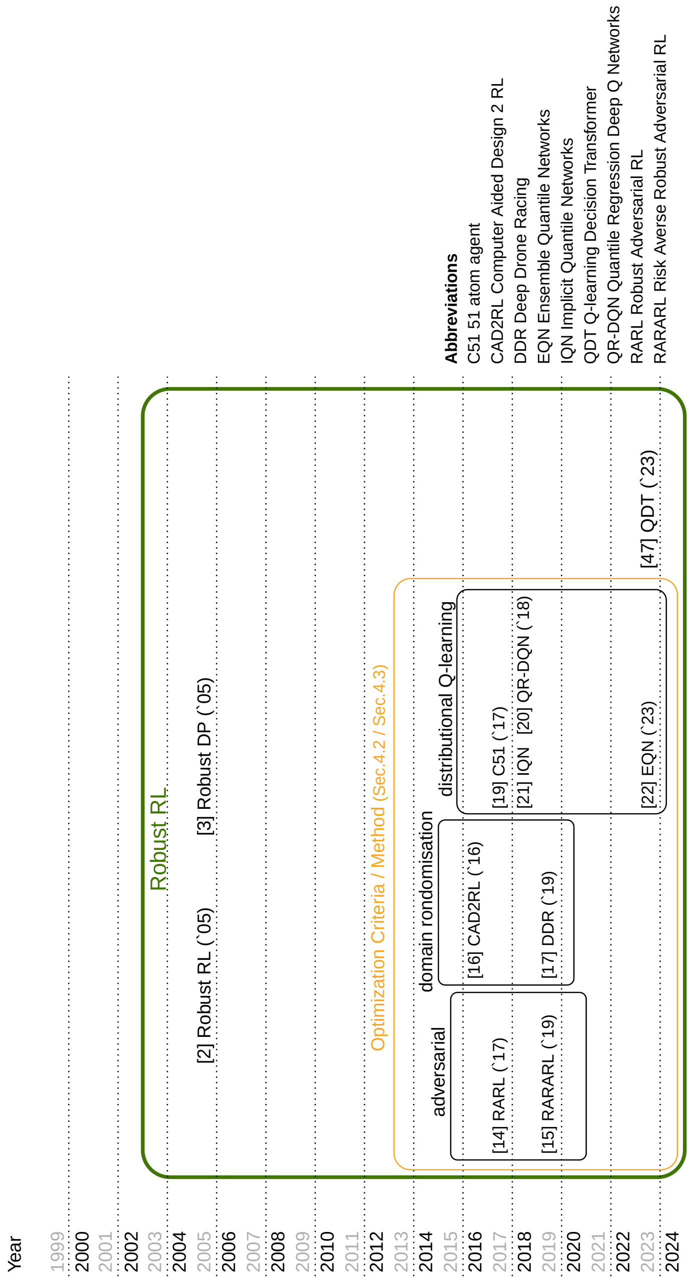


Fig. A2 The figure organises robust reinforcement learning literature in chronological order, highlighted in colour-coded boxes for major categories and black boxes for sub-categories.

A.P.: Safe learning in robotics: From learning-based control to safe reinforcement learning. Annual Review of Control, Robotics, and Autonomous Systems **5**, 411–444 (2022)

[10] Altman, E.: Constrained Markov Decision Processes vol. 7. CRC press, Boca Raton, FL (1999)

[11] Hadfield-Menell, D., Russell, S.J., Abbeel, P., Dragan, A.: Cooperative inverse reinforcement learning. Advances in neural information processing systems **29** (2016)

[12] Wiener, N.: Some moral and technical consequences of automation: As machines learn they may develop unforeseen strategies at rates that baffle their programmers. Science **131**(3410), 1355–1358 (1960)

[13] Geibel, P.: Reinforcement learning for mdps with constraints. In: Machine Learning: ECML 2006: 17th European Conference on Machine Learning Berlin, Germany, September 18-22, 2006 Proceedings 17, pp. 646–653 (2006). Springer

[14] Pinto, L., Davidson, J., Sukthankar, R., Gupta, A.: Robust adversarial reinforcement learning. In: International Conference on Machine Learning, pp. 2817–2826 (2017). PMLR

[15] Pan, X., Seita, D., Gao, Y., Canny, J.: Risk averse robust adversarial reinforcement learning. In: 2019 International Conference on Robotics and Automation (ICRA), pp. 8522–8528 (2019). IEEE

[16] Sadeghi, F., Levine, S.: Cad2rl: Real single-image flight without a single real image. arXiv preprint arXiv:1611.04201 (2016)

[17] Loquercio, A., Kaufmann, E., Ranftl, R., Dosovitskiy, A., Koltun, V., Scaramuzza, D.: Deep drone racing: From simulation to reality with domain randomization. IEEE Transactions on Robotics **36**(1), 1–14 (2019)

[18] Rockafellar, R.T., Uryasev, S., *et al.*: Optimization of conditional value-at-risk. Journal of risk **2**, 21–42 (2000)

[19] Bellemare, M.G., Dabney, W., Munos, R.: A distributional perspective on reinforcement learning. In: International Conference on Machine Learning, pp. 449–458 (2017). PMLR

[20] Dabney, W., Rowland, M., Bellemare, M., Munos, R.: Distributional reinforcement learning with quantile regression. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 32 (2018)

[21] Dabney, W., Ostrovski, G., Silver, D., Munos, R.: Implicit quantile networks for distributional reinforcement learning. In: International Conference on Machine

Learning, pp. 1096–1105 (2018). PMLR

[22] Hoel, C.-J., Wolff, K., Laine, L.: Ensemble quantile networks: Uncertainty-aware reinforcement learning with applications in autonomous driving. IEEE Transactions on Intelligent Transportation Systems (2023)

[23] Chow, Y., Ghavamzadeh, M., Janson, L., Pavone, M.: Risk-constrained reinforcement learning with percentile risk criteria. CoRR **abs/1512.01629** (2015) 1512.01629

[24] Tessler, C., Mankowitz, D.J., Mannor, S.: Reward constrained policy optimization. arXiv preprint arXiv:1805.11074 (2018)

[25] Achiam, J., Held, D., Tamar, A., Abbeel, P.: Constrained policy optimization. In: International Conference on Machine Learning, pp. 22–31 (2017). PMLR

[26] Chow, Y., Ghavamzadeh, M., Janson, L., Pavone, M.: Risk-constrained reinforcement learning with percentile risk criteria. The Journal of Machine Learning Research **18**(1), 6070–6120 (2017)

[27] Liu, Y., Ding, J., Liu, X.: Ipo: Interior-point policy optimization under constraints. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 34, pp. 4940–4947 (2020)

[28] Boyd, S.P., Vandenberghe, L.: Convex Optimization. Cambridge university press, Cambridge, UK (2004)

[29] Kakade, S., Langford, J.: Approximately optimal approximate reinforcement learning. In: Proceedings of the Nineteenth International Conference on Machine Learning, pp. 267–274 (2002)

[30] Pirotta, M., Restelli, M., Pecorino, A., Calandriello, D.: Safe policy iteration. In: International Conference on Machine Learning, pp. 307–315 (2013). PMLR

[31] Schulman, J., Levine, S., Abbeel, P., Jordan, M., Moritz, P.: Trust region policy optimization. In: International Conference on Machine Learning, pp. 1889–1897 (2015). PMLR

[32] Schulman, J., Wolski, F., Dhariwal, P., Radford, A., Klimov, O.: Proximal policy optimization algorithms. arXiv preprint arXiv:1707.06347 (2017)

[33] Rasmussen, C.E.: Gaussian processes in machine learning. In: Summer School on Machine Learning, pp. 63–71. Springer, New York, NY (2003)

[34] Sui, Y., Gotovos, A., Burdick, J., Krause, A.: Safe exploration for optimization with gaussian processes. In: International Conference on Machine Learning, pp. 997–1005 (2015). PMLR

[35] Wachi, A., Sui, Y., Yue, Y., Ono, M.: Safe exploration and optimization of constrained mdps using gaussian processes. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 32 (2018)

[36] Moldovan, T.M., Abbeel, P.: Safe exploration in markov decision processes. arXiv preprint arXiv:1205.4810 (2012)

[37] Wachi, A., Hashimoto, W., Shen, X., Hashimoto, K.: Safe exploration in reinforcement learning: A generalized formulation and algorithms. arXiv preprint arXiv:2310.03225 (2023)

[38] Han, W., Levine, S., Abbeel, P.: Learning compound multi-step controllers under unknown dynamics. In: 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 6435–6442 (2015). IEEE

[39] Eysenbach, B., Gu, S., Ibarz, J., Levine, S.: Leave no trace: Learning to reset for safe and autonomous reinforcement learning. arXiv preprint arXiv:1711.06782 (2017)

[40] Gehring, C., Precup, D.: Smart exploration in reinforcement learning using absolute temporal difference errors. In: Proceedings of the 2013 International Conference on Autonomous Agents and Multi-agent Systems, pp. 1037–1044 (2013)

[41] Wang, Y., He, H., Tan, X., Gan, Y.: Trust region-guided proximal policy optimization. Advances in Neural Information Processing Systems **32** (2019)

[42] Agarwal, R., Schuurmans, D., Norouzi, M.: An optimistic perspective on offline reinforcement learning, pp. 104–114 (2020)

[43] Harrison, J., Sharma, A., Pavone, M.: Meta-learning priors for efficient online bayesian regression, pp. 318–337 (2018)

[44] Yamagata, T., Santos-Rodríguez, R., Flach, P.: Continuous adaptation with online meta-learning for non-stationary target regression tasks. Signals **3**(1), 66–85 (2022)

[45] Liu, H., Laskin, M., Abbeel, P., Lazaric, A., Pinto, L., Yarats, D., Brandfonbrener, D.: Don't change the algorithm, change the data: Exploratory data for offline reinforcement learning. arXiv preprint arXiv:2201.13425 (2022)

[46] Fu, J., Kumar, A., Nachum, O., Tucker, G., Levine, S.: D4RL: Datasets for Deep Data-Driven Reinforcement Learning

[47] Yamagata, T., Khalil, A., Santos-Rodriguez, R.: Q-learning decision transformer: Leveraging dynamic programming for conditional sequence modelling in offline rl. In: International Conference on Machine Learning, pp. 38989–39007 (2023). PMLR

[48] Breyer, M., Furrer, F., Novkovic, T., Siegwart, R., Nieto, J.: Flexible robotic grasping with sim-to-real transfer based reinforcement learning. arXiv preprint arXiv:1803.04996 (2018)

[49] Zhao, W., Queralta, J.P., Westerlund, T.: Sim-to-real transfer in deep reinforcement learning for robotics: a survey. In: 2020 IEEE Symposium Series on Computational Intelligence (SSCI), pp. 737–744 (2020). IEEE

[50] Salvato, E., Fenu, G., Medvet, E., Pellegrino, F.A.: Crossing the reality gap: A survey on sim-to-real transferability of robot controllers in reinforcement learning. IEEE Access **9**, 153171–153187 (2021)

[51] Goecks, V.G.: Human-in-the-loop methods for data-driven and reinforcement learning systems. arXiv preprint arXiv:2008.13221 (2020)

[52] Argall, B.D., Chernova, S., Veloso, M., Browning, B.: A survey of robot learning from demonstration. Robotics and autonomous systems **57**(5), 469–483 (2009)

[53] Griffith, S., Subramanian, K., Scholz, J., Isbell, C.L., Thomaz, A.L.: Policy shaping: Integrating human feedback with reinforcement learning, pp. 2625–2633 (2013)

[54] Wang, F., Zhou, B., Chen, K., Fan, T., Zhang, X., Li, J., Tian, H., Pan, J.: Intervention aided reinforcement learning for safe and practical policy optimization in navigation. In: Conference on Robot Learning, pp. 410–421 (2018). PMLR

[55] Saunders, W., Sastry, G., Stuhlmueller, A., Evans, O.: Trial without error: Towards safe reinforcement learning via human intervention. arXiv preprint arXiv:1707.05173 (2017)

[56] Li, Q., Peng, Z., Zhou, B.: Efficient learning of safe driving policy via human-ai copilot optimization. arXiv preprint arXiv:2202.10341 (2022)

[57] Marta, D., Pek, C., Melsión, G.I., Tumova, J., Leite, I.: Human-feedback shield synthesis for perceived safety in deep reinforcement learning. IEEE Robotics and Automation Letters **7**(1), 406–413 (2021)

[58] Cederborg, T., Grover, I., Isbell, C.L., Thomaz, A.L.: Policy shaping with human teachers, pp. 3366–3372 (2015)

[59] Yamagata, T., McConville, R., Santos-Rodríguez, R.: Reinforcement Learning with Feedback from Multiple Humans with Diverse Skills. (2021). NeurIPS 2021 Workshop on Safe and Robust Control of Uncertain Systems, SafeRL 2021. https://sites.google.com/view/safe-robust-control/home

[60] Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C., Mishkin, P., Zhang, C., Agarwal, S., Slama, K., Ray, A., *et al.*: Training language models to follow

instructions with human feedback. Advances in Neural Information Processing Systems **35**, 27730–27744 (2022)

[61] Casper, S., Davies, X., Shi, C., Gilbert, T.K., Scheurer, J., Rando, J., Freedman, R., Korbak, T., Lindner, D., Freire, P., et al.: Open problems and fundamental limitations of reinforcement learning from human feedback. arXiv preprint arXiv:2307.15217 (2023)

[62] Akrour, R., Schoenauer, M., Sebag, M.: Preference-based policy learning. In: Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2011, Athens, Greece, September 5-9, 2011. Proceedings, Part I 11, pp. 12–27 (2011). Springer

[63] Cheng, W., Fürnkranz, J., Hüllermeier, E., Park, S.-H.: Preference-based policy iteration: Leveraging preference learning for reinforcement learning. In: Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2011, Athens, Greece, September 5-9, 2011. Proceedings, Part I 11, pp. 312–327 (2011). Springer

[64] Wirth, C., Akrour, R., Neumann, G., Fürnkranz, J., *et al.*: A survey of preference-based reinforcement learning methods. Journal of Machine Learning Research **18**(136), 1–46 (2017)

[65] Christiano, P.F., Leike, J., Brown, T., Martic, M., Legg, S., Amodei, D.: Deep reinforcement learning from human preferences. Advances in neural information processing systems **30** (2017)

[66] Lee, K., Smith, L., Abbeel, P.: Pebble: Feedback-efficient interactive reinforcement learning via relabeling experience and unsupervised pre-training. arXiv preprint arXiv:2106.05091 (2021)

[67] Rafailov, R., Sharma, A., Mitchell, E., Ermon, S., Manning, C.D., Finn, C.: Direct preference optimization: Your language model is secretly a reward model. arXiv preprint arXiv:2305.18290 (2023)

[68] Bradley, R.A., Terry, M.E.: Rank analysis of incomplete block designs: I. the method of paired comparisons. Biometrika **39**(3/4), 324–345 (1952)

[69] Duncan, L.R.: Individual choice behavior: A theoretical analysis. Courier Corporation. Google Scholar Google Scholar Reference (1959)

[70] Plackett, R.L.: The analysis of permutations. Journal of the Royal Statistical Society Series C: Applied Statistics **24**(2), 193–202 (1975)

[71] Zermelo, E.: The calculation of tournament results as a maximum-likelihood problem. Mathematische Zeitschrift **29**, 436–460 (1929)

[72] Knox, W.B., Stone, P.: Interactively shaping agents via human reinforcement: The tamer framework. In: Proceedings of the Fifth International Conference on Knowledge Capture, pp. 9–16 (2009)

[73] Tan, M.: Multi-agent reinforcement learning: Independent versus cooperative agents. In: International Conference on Machine Learning (1997). https://api.semanticscholar.org/CorpusID:260435822

[74] Gu, S., Kuba, J.G., Wen, M., Chen, R., Wang, Z., Tian, Z., Wang, J., Knoll, A., Yang, Y.: Multi-agent constrained policy optimisation. arXiv preprint arXiv:2110.02793 (2021)

[75] Liu, C., Geng, N., Aggarwal, V., Lan, T., Yang, Y., Xu, M.: Cmix: Deep multi-agent reinforcement learning with peak and average constraints. In: Machine Learning and Knowledge Discovery in Databases. Research Track: European Conference, ECML PKDD 2021, Bilbao, Spain, September 13–17, 2021, Proceedings, Part I 21, pp. 157–173 (2021). Springer

[76] ElSayed-Aly, I., Bharadwaj, S., Amato, C., Ehlers, R., Topcu, U., Feng, L.: Safe multi-agent reinforcement learning via shielding. arXiv preprint arXiv:2101.11196 (2021)

[77] Lu, S., Zhang, K., Chen, T., Başar, T., Horesh, L.: Decentralized policy gradient descent ascent for safe multi-agent reinforcement learning. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 35, pp. 8767–8775 (2021)

[78] Xiong, Z., Agarwal, I., Jagannathan, S.: Hisarl: A hierarchical framework for safe reinforcement learning. In: SafeAI@ AAAI (2022)

[79] Gangopadhyay, B., Soora, H., Dasgupta, P.: Hierarchical program-triggered reinforcement learning agents for automated driving. IEEE Transactions on Intelligent Transportation Systems **23**(8), 10902–10911 (2021)

[80] Roza, F.S., Roscher, K., Günnemann, S.: Safe and efficient operation with constrained hierarchical reinforcement learning. In: Sixteenth European Workshop on Reinforcement Learning (2023)

[81] Clempner, J.B.: A lyapunov approach for stable reinforcement learning. Computational and Applied Mathematics **41**(6), 279 (2022)

[82] Berkenkamp, F., Turchetta, M., Schoellig, A., Krause, A.: Safe model-based reinforcement learning with stability guarantees. Advances in neural information processing systems **30** (2017)

[83] Chow, Y., Nachum, O., Duenez-Guzman, E., Ghavamzadeh, M.: A lyapunov-based approach to safe reinforcement learning. Advances in neural information processing systems **31** (2018)

[84] Pan, S.J., Yang, Q.: A survey on transfer learning. IEEE Transactions on knowledge and data engineering **22**(10), 1345–1359 (2009)

[85] Weiss, K., Khoshgoftaar, T.M., Wang, D.: A survey of transfer learning. Journal of Big data **3**(1), 1–40 (2016)

[86] Zhuang, F., Qi, Z., Duan, K., Xi, D., Zhu, Y., Zhu, H., Xiong, H., He, Q.: A comprehensive survey on transfer learning. Proceedings of the IEEE **109**(1), 43–76 (2020)

[87] Taylor, M.E., Stone, P.: Transfer learning for reinforcement learning domains: A survey. Journal of Machine Learning Research **10**(7) (2009)

[88] Barreto, A., Dabney, W., Munos, R., Hunt, J.J., Schaul, T., Hasselt, H.V., Silver, D.: Successor features for transfer in reinforcement learning. Advances in Neural Information Processing Systems, 4056–4066 (2017)

[89] Kulkarni, T.D., Saeedi, A., Gautam, S., Gershman, S.J.: Deep successor reinforcement learning. arXiv preprint arXiv:1606.02396 (2016)

[90] Schaul, T., Horgan, D., Gregor, K., Silver, D.: Universal value function approximators, pp. 1312–1320

[91] Vilalta, R., Drissi, Y.: A perspective view and survey of meta-learning. Artificial intelligence review **18**, 77–95 (2002)

[92] Vanschoren, J.: Meta-learning: A survey. arXiv preprint arXiv:1810.03548 (2018)

[93] Hospedales, T., Antoniou, A., Micaelli, P., Storkey, A.: Meta-learning in neural networks: A survey. IEEE transactions on pattern analysis and machine intelligence **44**(9), 5149–5169 (2021)

[94] Finn, C., Abbeel, P., Levine, S.: Model-agnostic meta-learning for fast adaptation of deep networks. 34th International Conference on Machine Learning, ICML 2017 **3**, 1856–1868 (2017)

[95] Grant, E., Finn, C., Levine, S., Darrell, T., Griffiths, T.: Recasting gradient-based meta-learning as hierarchical bayes. arXiv preprint arXiv:1801.08930 (2018)

[96] Matas, J., James, S., Davison, A.J.: Sim-to-real reinforcement learning for deformable object manipulation. In: Conference on Robot Learning, pp. 734–743 (2018). PMLR

[97] Neufeld, E.A., Bartocci, E., Ciabattoni, A., Governatori, G.: Enforcing ethical goals over reinforcement-learning policies. Ethics and Information Technology **24**(4), 43 (2022)

[98] Coeckelbergh, M.: Ai Ethics. The MIT Press, Cambridge, Massachusetts, USA (2020)

[99] Hagendorff, T.: The ethics of ai ethics: An evaluation of guidelines. Minds and Machines **30**(1), 99–120 (2020) https://doi.org/10.1007/s11023-020-09517-8

[100] High-Level Expert Group on AI: Ethics guidelines for trustworthy ai. Report, European Commission, Brussels (April 2019). https://ec.europa.eu/digital-single-market/en/news/ethics-guidelines-trustworthy-ai

[101] Pan, A., Bhatia, K., Steinhardt, J.: The effects of reward misspecification: Mapping and mitigating misaligned models. arXiv preprint arXiv:2201.03544 (2022)

[102] Milani, S., Topin, N., Veloso, M., Fang, F.: Explainable reinforcement learning: A survey and comparative review. ACM Computing Surveys (2023)

[103] Krajna, A., Brcic, M., Lipic, T., Doncevic, J.: Explainability in reinforcement learning: perspective and position. arXiv preprint arXiv:2203.11547 (2022)

[104] Yamagata, T., O'Kane, A., Ayobi, A., Katz, D., Stawarz, K., Marshall, P., Flach, P., Santos-Rodriguez, R.: Model-based reinforcement learning for type 1 diabetes blood glucose control. CEUR Workshop Proceedings **2820**, 72–77 (2020)