Safe Reinforcement Learning with Contrastive Risk Prediction

Anonymous Author(s)
Affiliation
Address
email

Abstract

As safety violations can lead to severe consequences in real-world applications, the increasing deployment of Reinforcement Learning (RL) in safety-critical domains such as robotics has propelled the study of safe exploration for reinforcement learning (safe RL). In this work, we propose a risk preventive training method for safe RL, which learns a binary classifier based on contrastive sampling to predict the probability of a state-action pair leading to unsafe states. Based on the predicted risk probabilities, risk preventive trajectory exploration and optimality criterion modification can be simultaneously conducted to induce safe RL policies. We conduct experiments in robotic simulation environments. The results show the proposed approach outperforms existing model-free safe RL approaches, and yields comparable performance with the state-of-the-art model-based method.

1 Introduction

Reinforcement Learning (RL) offers a great set of technical tools for many real-world decision making systems, such as robotics, that require an agent to automatically learn behavior policies through interactions with the environments [1]. Conversely, the applications of RL in real-world domains also pose important new challenges for RL research. In particular, many real-world robotic environments and tasks, such as human-related robotic environments [2], helicopter manipulation [3, 4], autonomous vehicle [5], and aerial delivery [6], have very low tolerance for violations of safety constraints, as such violation can cause severe consequences. This raises a substantial demand for safe reinforcement learning techniques.

Safe reinforcement learning investigates RL methodologies with critical safety considerations, and has received increased attention from the RL research community. In safe RL, in addition to the reward function [7], an RL agent often deploys a cost function to maximize the discounted cumulative reward while satisfying the cost constraint [8–10]. A comprehensive survey of safe RL categorizes the safe RL techniques into two classes: modification of the optimality criterion and modification of the exploration process [11]. For modification of the optimality criterion, previous works mostly focus on the modification of the reward. Many works [12–17] pursue such modifications by shaping the reward function with penalizations induced from different forms of cost constraints. For modification of the exploration process, safe RL approaches focus on training RL agents on modified trajectory data. For example, some works deploy backup policies to recover from safety violations to safer trajectory data that satisfy the safety constraint [18–20].

In this paper, we propose a novel risk preventive training (RPT) method to tackle the safe RL problem. The key idea is to learn a contrastively estimated classification model to predict the risk—the probability of a state-action pair leading to unsafe states, which can then be deployed to modify both the exploration process and the optimality criterion. In terms of exploration process modification, we collect trajectory data in a risk preventive manner based on the predicted probability of risk. A

Submitted to 38th Conference on Neural Information Processing Systems (NeurIPS 2024). Do not distribute.

trajectory is terminated if the next state falls into an unsafe region that has above-threshold risk values. Regarding optimality criterion modification, we reshape the reward function by penalizing it with the predicted risk for each state-action pair. Benefiting from the generalizability of risk prediction, the proposed approach can avoid safety constraint violations much early in the training phase and induce safe RL policies, while previous works focus on backup policy and violate more safety constraints by interacting with the environment in the unsafe regions. Moreover, we further deploy a simple unsafe-state augmentation strategy for the proposed method to increase the sample efficiency of the encountered unsafe states and reduce the safety violations of the RL agent in the experiments. We conduct experiments using four robotic simulation environments on MuJoCo [21]. Our model-free approach produces comparable performance with a state-of-the-art model-based safe RL method SMBPO [16] and greatly outperforms other model-free safe RL methods. The main contributions of the proposed work can be summarized as follows:

- This is the first work that introduces a contrastive sampling based classifier to perform risk prediction and conduct safe RL exploration.
- With its proficient risk prediction capabilities, the proposed approach possesses the essential capacity to simultaneously modify the exploration process through risk preventive trajectory collection and adjust the optimality criterion through reward reshaping.
- As a model-free safe RL method, the proposed approach achieves comparable performance to the state-of-the-art model-based safe RL method and outperforms the model-free methods in multiple benchmark robotic simulation environments.

2 Related Works

Many methods have been developed for safe RL. Garcıa and Fernández [11] provided a survey categorizing safe RL methods into categories of modifying the optimality criterion and modifying the exploration process.

Modification of the optimality criterion. Since optimizing the conventional reward signal does not ensure the avoidance of safety violations, leading to the exploration of modifying the optimality objective based on risk notions [22, 23], probabilities of visiting risky states [24], etc. Achiam et al. [20] proposed Constrained Policy Optimization (CPO) to update safe policies by optimizing the primal-dual problem in trust regions. Recently, reward shaping techniques [25, 26] have been integrated into safe RL. Tessler et al. [14] introduced Reward Constrained Policy Optimization (RCPO) by penalizing the normal training policy. Thomas et al. [16] reshaped reward functions using a model-based predictor, treating unsafe states as absorbing states to train the RL agent with penalized rewards. Xu et al. [27] developed Constrained Penalized Q-learning (CPQ) using a cost critic to learn constraint values during exploration and penalizing the Bellman operator in policy training to stop the updates for potentially unsafe states.

Modification of the exploration process. Previous works have optimized safe RL policies by adjusting exploration processes during interaction with the environment. For instance, [28, 3, 29] guided exploration based on prior environmental knowledge. Similarly, [30, 31] constrained exploration learning using demonstration data. More recent approaches like [18, 19] focused on utilizing backup policies from safe regions to prevent safety violations. If the agent undertakes a potentially risky action, the task policy is replaced with a guaranteed safe backup policy. Yu et al. [32] defined safe regions as feasible sets and used reachability analysis to expand these sets beyond traditional energy-based methods. Jayant and Bhatnagar [33] introduced a model-based deep RL agent that efficiently learns an ensemble of transition dynamics in an online environment and restricts exploration with a performance ratio.

Safe RL is crucial in environments like human-related robotic settings where safety violations can lead to catastrophic failures [2]. Robotic simulation environments such as MuJoCo, developed by Todorov et al. [21], facilitate research in RL applications for robotics. Thomas et al. [16] extended the MuJoCo environment to define safety violations in robotic simulations, making it an ideal test bed for safe RL methods.

3 Preliminary

Reinforcement learning (RL) has been broadly used to train robotic agents by maximizing the discounted cumulative rewards. The representation of a reinforcement learning problem can be formulated as a Markov Decision Process (MDP) $M = (\mathcal{S}, \mathcal{A}, \mathcal{T}, \mathcal{R}, \gamma)$ [7], where $\mathcal{S}$ is the state space for all observations, $\mathcal{A}$ is the action space for available actions, $\mathcal{T} : \mathcal{S} \times \mathcal{A} \rightarrow \mathcal{S}$ is the transition dynamics, $\mathcal{R} : \mathcal{S} \times \mathcal{A} \rightarrow [r_{min}, r_{max}]$ is the reward function, and $\gamma \in (0, 1)$ is the discount factor. An agent can start from a random initial state s_0 to take actions and interact with the MDP environment by receiving rewards for each action and moving to new states. Such interactions can produce a transition (s_t, a_t, r_t, s_{t+1}) at each time-step t with $s_{t+1} = \mathcal{T}(s_t, a_t)$ and $r_t = r(s_t, a_t)$, while a sequence of transitions comprise a trajectory $\tau = (s_0, a_0, r_0, s_1, a_1, r_1, \cdots, s_{|\tau|+1})$, where $|\tau| + 1$ denotes the length of trajectory τ—i.e., the number of transitions. The goal of RL is to learn an optimal policy $\pi^\star : \mathcal{S} \rightarrow \mathcal{A}$ that can maximize the expected discounted cumulative reward (return): $\pi^\star = \arg\max_\pi \; J_r(\pi) = \mathbb{E}_{\tau\sim\mathcal{D}_\pi}[\sum_{t=0}^{|\tau|} \gamma^t r_t]$

3.1 Safe Exploration for Reinforcement Learning

Safe exploration for Reinforcement Learning (safe RL) studies RL with critical safety considerations. For a safe RL environment, in addition to the reward function, a cost function can also exist to reflect the risky status of each exploration step. The process of safe RL can be formulated as a Constrained Markov Decision Process (CMDP) [34], $\hat{M} = (\mathcal{S}, \mathcal{A}, \mathcal{T}, \mathcal{R}, \gamma, c, d)$, which introduces an extra cost function c and a cost threshold d into MDP. An exploration trajectory under CMDP can be written as $\tau = (s_0, a_0, r_0, c_0, s_1, \cdots, s_{|\tau|+1})$, where the transition at time-step t is $(s_t, a_t, s_{t+1}, r_t, c_t)$, with a cost value c_t induced from the cost function $c_t = c(s_t, a_t)$. CMDP monitors the safe exploration process by requiring the cumulative cost $J_c(\pi)$ does not exceed the cost threshold d, where $J_c(\pi)$ can be defined as the expected total cost of the exploration, $J_c(\pi) = \mathbb{E}_{\tau\sim\mathcal{D}_\pi}[\sum_{t=0}^{|\tau|} c_t]$ [12]. Safe RL hence aims to learn an optimal policy $\pi^\star$ that can maximize the expected discounted cumulative reward subjecting to a cost constraint, as follows:

$$\pi^\star = \arg\max_\pi \; J_r(\pi) = \mathbb{E}_{\tau\sim\mathcal{D}_\pi}\left[\sum\nolimits_{t=0}^{|\tau|} \gamma^t r_t\right] \qquad (1)$$
$$\text{s.t. } J_c(\pi) = \mathbb{E}_{\tau\sim\mathcal{D}_\pi}\left[\sum\nolimits_{t=0}^{|\tau|} c_t\right] \leq d.$$

4 Method

Robot operations typically have low tolerance for risky/unsafe states and actions, since a robot could be severely damaged in real-world environments when the safety constraint being violated. Similar to the work in [9], in this work we adopt a strict setting for the safety constraint such that any “unsafe" state can cause violation of the safety constraint and the RL agent will terminate an exploration trajectory when encountering an “unsafe" state. We have the following definition:

Definition 1. *For a state s and an action a, the value of the cost function $c(s, a)$ can either be 0 or 1. When $c(s, a) = 0$, the induced state $\mathcal{T}(s, a)$ is defined as a safe state; when $c(s, a) = 1$, the induced state $\mathcal{T}(s, a)$ is defined as an unsafe state, which triggers the violation of safety constraints and hence causes the termination of the trajectory.*

Based on this definition, the cost threshold d in Eq. (1) should be set strictly to 0. The agent is expected to learn a safe policy π that can operate with successful trajectories containing only safe states. Towards this goal, we propose a novel risk prediction method for safe RL. The proposed method deploys a contrastive classifier to predict the probability of a state-action pair leading to unsafe states, which can be trained during the exploration process of RL and generalized to previously unseen states.

With risk prediction probabilities, a more informative cumulative cost $J_c(\pi)$ can be formed to prevent unsafe trajectories and reshape the reward in each transition of a trajectory to induce safe RL policies. Previous safe RL methods in the literature can typically be categories into two classes: modification of the optimality criterion and modification of the exploration process [11]. With safety constraints and risk predictions, the proposed approach (to be elaborated below) has the capacity and is expected to incorporate the strengths of both categories of safe RL techniques.

4.1 Risk Prediction with Contrastive Classification

Although an RL agent would inevitably encounter unsafe states during the initial stage of the exploration process in an unknown environment, we aim to quickly learn from the unsafe experience through statistical learning and generalize the recognition of unsafe trajectories to prevent risk for future exploration. Specifically, we aim to compute the probability of a state-action pair leading to unsafe states, i.e., $p(y=1|s_t,a_t)$, where $y \in \{0,1\}$ denotes a random variable that indicates whether (s_t,a_t) leads to an unsafe state $s_u \in S_U$. The set of unsafe states, S_U, can be either pre-given or collected during initial exploration. However, directly training a binary classifier to make such predictions is impractical as it is difficult to judge whether a state-action pair is *safe*—i.e., never leading to unsafe states.

For this purpose, we propose to train a contrastive classifier $F_\theta(s_t,a_t)$ with model parameter θ to discriminate a positive state-action pair (s_t,a_t) in a trajectory that leads to unsafe states (unsafe trajectory) against random state-action pairs from the overall distribution of any trajectory. Such a contrastive form of learning can conveniently avoid the impractical identification problem of absolute negative (i.e., safe) state-action pairs.

Specifically, inspired by the noise contrastive estimation based classifier design in the literature [35, 36], we propose to learn $F_\theta(s_t,a_t)$ as a binary classifier via weighted contrastive sampling by sampling unsafe state-action pairs as positive samples and sampling general state-action pairs as contrastive negative samples. Let $p(s_t,a_t|y=1)$ denote the presence probability of a state-action pair (s_t,a_t) in a trajectory that leads to unsafe states, and $p(y=1)$ denote the distribution probability of unsafe trajectory in the environment. The contrastive classifier $F_\theta(s_t,a_t)$ is then defined as follows:

$$F_\theta(s_t,a_t) = \frac{p(s_t,a_t|y=1)p(y=1)}{p(s_t,a_t|y=1)p(y=1)+p(s_t,a_t)}, \tag{2}$$

where $p(y=1)$ is used as weight for the positive samples which are only from the unsafe trajectories, and weight 1 is given to the *contrastively-negative* samples which are from the overall distribution. This binary classifier identifies the state-action pairs in unsafe trajectories contrastively from general pairs in the overall distribution.

From the definition of $F_\theta(s_t,a_t)$ in Eq.(2), one can derive the probability of interest, $p(y=1|s_t,a_t)$, using the Bayes' theorem, as follows:

$$p(y=1|s_t,a_t) = \frac{p(s_t,a_t|y=1)p(y=1)}{p(s_t,a_t)} \quad = \frac{F_\theta(s_t,a_t)}{1-F_\theta(s_t,a_t)}, \tag{3}$$

where the derivation from the fraction in the top row to the term expressed in F_θ in the second row can be done easily by dividing both the numerator and denominator of the top row fraction with the same term $[p(s_t,a_t|y=1)p(y=1)+p(s_t,a_t)]$. As the normal output range, $[0,1]$, of the probabilistic classifier $F_\theta(s_t,a_t)$ could lead to unbounded values $p(y=1|s_t,a_t) \in [0,\infty]$ through Eq.(3), we propose to first rescale the output of classifier $F_\theta(s_t,a_t)$ to the range of $[0,0.5]$ when calculating $p(y=1|s_t,a_t)$ via Eq.(3).

Based on the contrastive sampling principle of F_θ, we optimize the contrastive classifier's parameter θ using maximum likelihood estimation (MLE) with the following log-likelihood objective function:

$$L(\theta) \;=\; \mathbb{E}_{p(s_t,a_t|y=1)p(y=1)}\big[\log F_\theta(s_t,a_t)\big] + \mathbb{E}_{p(s_t,a_t)}\big[\log(1-F_\theta(s_t,a_t))\big]. \tag{4}$$

By setting the derivative of $L(\theta)$ w.r.t. F_θ to zero, it is easy to verify that the definition of F_θ in Eq.(2) can achieve the maximum of this MLE objective w.r.t. F_θ.

4.2 Risk Preventive Trajectory

Based on Definition 1, a trajectory terminates when the RL agent encounters an unsafe state and triggers safety constraint violation. It is however desirable to minimize the number of such safety violations during the policy training process and learn a good policy in safe regions. The risk prediction classifier we proposed above provides a convenient tool for this purpose by predicting the probability of a state-action pair leading to unsafe states, $p(y=1|s_t,a_t)$. Based on this risk prediction, we have the following definition for unsafe regions:

Definition 2. *A state-action pair (s_t,a_t) falls into an* **unsafe region** *if the probability of (s_t,a_t) leading to unsafe states is greater than a threshold η: $p(y=1|s_t,a_t) > \eta$, where $\eta \in (0,1)$.*

With this definition, an RL agent can pursue risk preventive trajectories to avoid safety violations by staying away from unsafe regions. Specifically, we can terminate a trajectory before violating the safety constraint by judging the potential risk—*i.e.*, the probability of $p(y = 1|s_t, a_t)$.

Without a doubt, the threshold η is a key for determining the length $T = |\hat{\tau}|$ of an early stopped risk preventive trajectory $\hat{\tau}$. We make the following assumption for deriving a lemma:

Assumption 1. *For a trajectory $\tau = \{s_0, a_0, r_0, c_0, s_1, \cdots, s_H\}$ that leads to an unsafe state $s_H \in S_U$, the risk prediction probability $p(y = 1|s_t, a_t)$ increases linearly along transition steps within a base neighborhood of the unsafe region that can be defined through $p(y = 1|s_t, a_t) \geq \eta_b$ with a threshold $\eta_b \in (0, \eta)$.*

Lemma 1. *Assume Assumption 1 holds. Let H denote the length of an unsafe trajectory $\tau = \{s_0, a_0, r_0, c_0, s_1, \cdots, s_H\}$ that terminates at an unsafe state $s_H \in S_U$. The numbers of transition steps, T and T_b, along this trajectory to the unsafe region determined by η in Definition 2 and its neighborhood determined by η_b, respectively, satisfy $T \approx \lfloor \frac{\eta - \eta_b}{1-\eta_b} H + \frac{1-\eta}{1-\eta_b} T_b \rfloor$.*

This lemma demonstrates the influence of the risk control threshold η on the length of collected trajectories. Given η_b (and hence T_b), a larger η value will allow more effective explorations with longer trajectories to facilitate policy learning, but also tighten the unsafe region and increase the possibility of violating safety constraints.

4.3 Risk Preventive Reward Shaping

With Definition 1, the safe RL formulation in Eq. (1) can hardly induce a safe policy since there are no intermediate costs before encountering an unsafe state. With the risk prediction classifier proposed above, we can rectify this drawback by defining the cumulative cost function $J_c(\pi)$ using the risk prediction probabilities, $p(y = 1|s_t, a_t)$, over all encountered state-action pairs. Specifically, we adopt a reward-like discounted cumulative cost as follows:

$$J_c(\pi) = \mathbb{E}_{\tau \sim \mathcal{D}_\pi} \left[\sum\nolimits_{t=0}^{|\tau|} \gamma^t p(y = 1|s_t, a_t) \right], \tag{5}$$

which uses the predicted risk as the estimated cost. Moreover, instead of solving safe RL as a constrained discounted cumulative reward maximization problem, we propose to use Lagrangian relaxation [37] to convert the constrained maximization problem, CMDP, in Eq. (1) into an unconstrained optimization problem, which is equivalent to shaping the reward function $\mathcal{R}$ with risk penalties:

$$\min_{\lambda \geq 0} \max_{\pi} \; [J_r(\pi) - \lambda(J_c(\pi) - d)] \tag{6}$$

$$\Longleftrightarrow \min_{\lambda \geq 0} \max_{\pi} \; [J_r(\pi) - \lambda J_c(\pi)] \tag{7}$$

$$\Longleftrightarrow \min_{\lambda \geq 0} \max_{\pi} \; \mathbb{E}_{\tau \sim \mathcal{D}_\pi} \left[\sum\nolimits_{t=0}^{|\tau|} \gamma^t (r_t - \lambda p(y = 1|s_t, a_t)) \right] \tag{8}$$

where $r_t - \lambda p(y = 1|s_t, a_t)$ can be treated as the risk penalty reshaped reward. The Lagrangian dual variable λ controls the degree of reward shaping with the predicted risk value.

4.4 Risk Preventive Training Algorithm

The overall risk preventive RL training procedure for the proposed safe RL method is presented in Algorithm 1, which trains a contrastive classifier F_θ (line 21) for risk prediction, and performs safe reinforcement learning by simultaneously enforcing risk preventive trajectory exploration (line 15-17) and risk preventive reward shaping (line 13).

4.5 Data Augmentation for Contrastive Learning

As the goal of safe RL is to minimize the encountering of unsafe states, it is desirable to produce an effective risk predictor with very limited risky state-action pairs. To this end, we propose to extend RPT by designing a simple *data augmentation* procedure, producing a data augmented method, RPT+DA, for comparison. The proposed data augmentation solely *enhances the training*

Algorithm 1 Risk Preventive Training

Input: Initial policy π_ϕ, classifier F_θ, trajectory set $D = \emptyset$,
set of unsafe state-action pairs S_U, threshold η, η_b;
penalty factor λ, set of unsafe trajectory length $\mathcal{H} = \emptyset$
Output: Trained policy π_ϕ

for $k = 1, 2, ..., K$ **do**
    $T_b = 0$
    for $t = 0, 1, ..., T_{\max}$ **do**
        Sample transition $(s_t, a_t, r_t, c_t, s_{t+1})$ from the environment with policy π_ϕ.
        if $c_t > 0$ **then**
            Add the risky state-action (s_t, a_t) into S_U; add length t to $\mathcal{H}$.
            Increase λ if necessary
            Stop trajectory and break.
        end if
        Sample next action a_{t+1} as $a_{t+1} = \pi_\phi(\cdot|s_{t+1})$.
        Compute p_t and p_{t+1} via Eq. (3)
        If $p_t \geq \eta_b$, then set $T_b = t$
        Penalize reward r_t with p_t: $\hat{r}_t = r_t - \lambda p_t$
        Add transition to the trajectory set, such that: $D = D \cup (s_t, a_t, \hat{r}_t, s_{t+1})$
        if $p_{t+1} > \eta$ **then**
            Stop trajectory and break.
        end if
    end for
    Sample risky state-action pairs from S_U
    Sample transitions from D: $(s_t, a_t, \hat{r}_t, s_{t+1}) \sim D$
    Update classifier F_θ by maximizing $L(\theta)$ in Eq (4)
    Update policy π_ϕ with shaped reward $J_{\hat{r}}(\pi)$ in Eq (8)
end for

of contrastive classifier for risk prediction, with no additional interaction with the environment or trajectory generation. Specifically, we perform data augmentation only for the data sampled from the set of risky states S_U. For each sampled risky state-action pair (s_t, a_t), we propose to produce an augmented state $\hat{s}_t$ by adding a random Gaussian noise sampled from the standard normal distribution $\mathcal{N}(0, 1)$ to each entry of the observed data s_t. We can repeat this process to generate multiple (e.g., n) augmented states for each s_t. In our experiments, we used $n = 3$. Together with a_t, each $\hat{s}_t$ can be used to form an additional risky state-action pair $(\hat{s}_t, a_t)$ for training the contrastive classifier. The hypothesis is that without any prior information about the environment, the training of the proposed contrastive classifier highly depends on the data collected during the agent's interactions with the environment, especially on the limited number of observed unsafe states. By using the proposed data augmentation technique above, we expect to improve the unsafe states' sample efficiency and the generalizability of the approach on discriminating unsafe states and hence reduce the possible safety violations during the exploration process.

5 Experiment

5.1 Experimental Settings

Experimental Environments Following the experimental setting in [16], we adopted four robotics simulation environments, *Ant, Cheetah, Hopper,* and *Humanoid*, based on the MuJoCo simulator [21]. For *Ant* and *Hopper*, a robot violates the safety constraint when it falls over. For *Cheetah*, a robot violates the safety constraint when its head flips on the ground, which is modified from the HalfCheetah environment with extra safety constraint [16]. For *Humanoid*, the human-like robot violates the safety constraint when the head of the robot falls to the ground. The RL agent cumulates returns by operating in the environment. As shown in Algorithm 1, the RL trajectory terminates when either the RL agent encounters safety violation, the maximum length is reached, or the preventive trajectory break takes place.

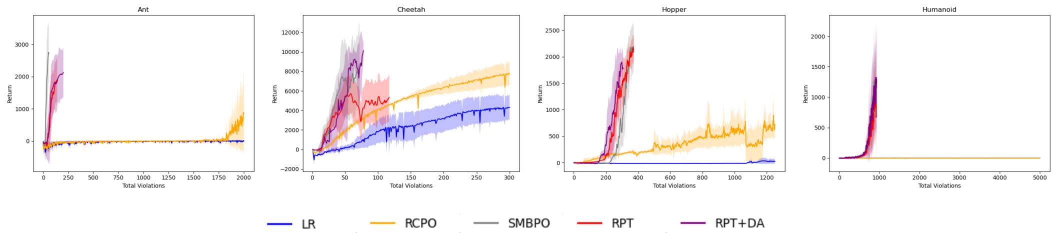

Figure 1: For each method, each plot presents the undiscounted return vs. the total number of violations. The curve shows the mean of the return over five runs, while the shadow shows the standard deviation.

Comparison Methods We compare the proposed Risk Preventive Training (RPT) approach with three state-of-the-art safe RL methods: SMBPO [16], RCPO [14], and LR [12].

5.2 Experimental Results

We compared all the five methods (LR, RCPO, SMBPO, RPT, and RPT+DA) by running each method five times with random seeds in each of the four MuJoCo environments. The performance of each method is evaluated by presenting the corresponding return vs. the total number of violations obtained in the training process. The results for all the methods are presented on the left side of Figure 1, one plot for each robotic simulation environment. The curve for each method shows the learning ability of the RL agent with limited safety violations. From the plots, we can see RPT, RPT+DA and SMBPO achieve large returns with a small number of violations on all the four robotic tasks, and largely outperform the other two methods, RCPO and LR, which have much smaller returns even with large numbers of safety violations. The proposed model-free RPT produces slightly inferior performance than the model-based SMBPO on *Ant* and *Cheetah*, where RPT requires more examples of unsafe states to yield good performance at the initial training stage. Nevertheless, RPT outperforms SMBPO on both *Hopper* and *Humanoid* with smaller number of safety violations. As a model-free safe RL method, RPT produces an overall comparable performance with the model-based method SMBPO. With data augmentation, RPT+DA further improves the performance of RPT on all the four environments, which demonstrates the efficacy of our simple unsafe-state augmentation strategy.

6 Conclusion

Inspired by the increasing demands for safe exploration of Reinforcement Learning, we proposed a novel mode-free risk preventive training method, RPT, to perform safe RL by learning a contrastive-sampling based binary classifier to predict the probability of a state-action pair leading to unsafe states. Based on risk prediction, we produce a systematic scheme to collect risk preventive trajectories that terminate early without triggering safety constraint violations. Moreover, the predicted risk probabilities are also used as penalties to perform reward shaping for learning safe RL policies. A simple data augmentation strategy has also been deployed to improve the efficiency of the observed unsafe-states for RPT. We compared the proposed approach with a few state-of-the-art safe RL methods using four robotic simulation environments. The proposed approach demonstrates comparable performance with the state-of-the-art model-based method and outperforms the model-free safe RL methods.

References

[1] J. Kober, J. A. Bagnell, and J. Peters, "Reinforcement learning in robotics: A survey," *The International Journal of Robotics Research*, 2013.

[2] L. Brunke, M. Greeff, A. W. Hall, Z. Yuan, S. Zhou, J. Panerati, and A. P. Schoellig, "Safe learning in robotics: From learning-based control to safe reinforcement learning," *Annual Review of Control, Robotics, and Autonomous Systems*, 2021.

[3] J. A. Martín H and J. d. Lope, "Learning autonomous helicopter flight with evolutionary reinforcement learning," in *International Conference on Computer Aided Systems Theory*. Springer, 2009.

[4] R. Koppejan and S. Whiteson, "Neuroevolutionary reinforcement learning for generalized control of simulated helicopters," *Evolutionary intelligence*, 2011.

[5] L. Wen, J. Duan, S. E. Li, S. Xu, and H. Peng, "Safe reinforcement learning for autonomous vehicles through parallel constrained policy optimization," in *IEEE International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 2020.

[6] A. Faust, I. Palunko, P. Cruz, R. Fierro, and L. Tapia, "Automated aerial suspended cargo delivery through reinforcement learning," *Artificial Intelligence*, 2017.

[7] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. MIT press, 2018.

[8] O. Mihatsch and R. Neuneier, "Risk-sensitive reinforcement learning," *Machine learning*, 2002.

[9] A. Hans, D. Schneegaß, A. M. Schäfer, and S. Udluft, "Safe exploration for reinforcement learning." in *ESANN*, 2008.

[10] Y. J. Ma, A. Shen, O. Bastani, and J. Dinesh, "Conservative and adaptive penalty for model-based safe reinforcement learning," in *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, 2022.

[11] J. Garcıa and F. Fernández, "A comprehensive survey on safe reinforcement learning," *Journal of Machine Learning Research (JMLR)*, pp. 1437–1480, 2015.

[12] A. Ray, J. Achiam, and D. Amodei, "Benchmarking safe exploration in deep reinforcement learning," *arXiv preprint arXiv:1910.01708*, 2019.

[13] L. Shen, L. Yang, S. Chen, B. Yuan, X. Wang, D. Tao *et al.*, "Penalized proximal policy optimization for safe reinforcement learning," *arXiv preprint arXiv:2205.11814*, 2022.

[14] C. Tessler, D. J. Mankowitz, and S. Mannor, "Reward constrained policy optimization," in *International Conference on Learning Representations (ICLR)*, 2019.

[15] Y. Hu, W. Wang, H. Jia, Y. Wang, Y. Chen, J. Hao, F. Wu, and C. Fan, "Learning to utilize shaping rewards: A new approach of reward shaping," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

[16] G. Thomas, Y. Luo, and T. Ma, "Safe reinforcement learning by imagining the near future," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

[17] J. Zhang, B. Cheung, C. Finn, S. Levine, and D. Jayaraman, "Cautious adaptation for reinforcement learning in safety-critical settings," in *International Conference on Machine Learning (ICML)*, 2020.

[18] B. Thananjeyan, A. Balakrishna, S. Nair, M. Luo, K. Srinivasan, M. Hwang, J. E. Gonzalez, J. Ibarz, C. Finn, and K. Goldberg, "Recovery rl: Safe reinforcement learning with learned recovery zones," *IEEE Robotics and Automation Letters*, 2021.

[19] O. Bastani, S. Li, and A. Xu, "Safe reinforcement learning via statistical model predictive shielding." in *Robotics: Science and Systems*, 2021.

[20] J. Achiam, D. Held, A. Tamar, and P. Abbeel, "Constrained policy optimization," in *International conference on machine learning (ICML)*, 2017.

[21] E. Todorov, T. Erez, and Y. Tassa, "Mujoco: A physics engine for model-based control," in *IEEE/RSJ international conference on intelligent robots and systems*. IEEE, 2012.

[22] R. A. Howard and J. E. Matheson, "Risk-sensitive markov decision processes," *Management science*, 1972.

[23] M. Sato, H. Kimura, and S. Kobayashi, “Td algorithm for the variance of return and mean-variance reinforcement learning,” *Transactions of the Japanese Society for Artificial Intelligence*, 2001.

[24] P. Geibel and F. Wysotzki, “Risk-sensitive reinforcement learning applied to control under constraints,” *Journal of Artificial Intelligence Research (JAIR)*, 2005.

[25] M. Dorigo and M. Colombetti, “Robot shaping: Developing autonomous agents through learning,” *Artificial intelligence*, 1994.

[26] J. Randløv and P. Alstrøm, “Learning to drive a bicycle using reinforcement learning and shaping.” in *International Conference on Machine Learning (ICML)*, 1998.

[27] H. Xu, X. Zhan, and X. Zhu, “Constraints penalized q-learning for safe offline reinforcement learning,” in *AAAI Conference on Artificial Intelligence (AAAI)*, 2022.

[28] K. Driessens and S. Džeroski, “Integrating guidance into relational reinforcement learning,” *Machine Learning*, 2004.

[29] Y. Song, Y.-b. Li, C.-h. Li, and G.-f. Zhang, “An efficient initialization approach of q-learning for mobile robots,” *International Journal of Control, Automation and Systems*, 2012.

[30] P. Abbeel, A. Coates, and A. Y. Ng, “Autonomous helicopter aerobatics through apprenticeship learning,” *The International Journal of Robotics Research*, 2010.

[31] J. Tang, A. Singh, N. Goehausen, and P. Abbeel, “Parameterized maneuver learning for autonomous helicopter flight,” in *IEEE international conference on robotics and automation (ICRA)*, 2010.

[32] D. Yu, H. Ma, S. Li, and J. Chen, “Reachability constrained reinforcement learning,” in *International Conference on Machine Learning (ICML)*. PMLR, 2022.

[33] A. K. Jayant and S. Bhatnagar, “Model-based safe deep reinforcement learning via a constrained proximal policy optimization algorithm,” *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 24 432–24 445, 2022.

[34] E. Altman, *Constrained Markov decision processes: stochastic modeling*. Routledge, 1999.

[35] M. Gutmann and A. Hyvärinen, “Noise-contrastive estimation: A new estimation principle for unnormalized statistical models,” in *Proceedings of the international conference on artificial intelligence and statistics (AISTATS)*, 2010.

[36] A. Mnih and Y. W. Teh, “A fast and simple algorithm for training neural probabilistic language models,” in *International conference on machine learning (ICML)*, 2012.

[37] D. P. Bertsekas, “Nonlinear programming,” *Journal of the Operational Research Society*, 1997.