INDERSCIENCE PUBLISHERS
Linking academia, business and industry through research

International Journal of Automation and Control

ISSN online: 1740-7524 - ISSN print: 1740-7516
https://www.inderscience.com/ijaac

Few-shot reasoning-based safe reinforcement learning framework for autonomous robot navigation

Weiqiang Wang, Xu Zhou, Benlian Xu, Siwen Chen, Mingli Lu, Jun Li, Yuejiang Gu

DOI: 10.1504/IJAAC.2024.10055043

Article History:

Received:	16 December 2022
Last revised:	26 January 2023
Accepted:	10 February 2023
Published online:	30 November 2023

Few-shot reasoning-based safe reinforcement learning framework for autonomous robot navigation

Weiqiang Wang and Xu Zhou*

School of Mechanical Engineering,
Changshu Institute of Technology,
Suzhou, 215500, China
Email: weiqiangwang2022@hotmail.com
Email: xuzhou@cslg.edu.cn
*Corresponding author

Benlian Xu

School of Electronic and Information Engineering,
Suzhou University of Science and Technology,
Suzhou, 215009, China
Email: xu_benlian@usts.edu.cn

Siwen Chen

Software Institute,
Nanjing University,
Nanjing, 210008, China
Email: 181250013@smail.nju.edu.cn

Mingli Lu

School of Electrical and Automatic Engineering,
Changshu Institute of Technology,
Suzhou, 215500, China
Email: luml@cslg.edu.cn

Jun Li

School of Automation,
Nanjing University of Science and Technology,
Nanjing, 210094, China
Email: lijun1008@163.com

Yuejiang Gu

R&D Center,
General Elevator Co., Ltd.,
Suzhou, 215232, China
Email: guyj@tydt.com

Abstract: Unsafe explorations in the training phase hinder the practical deployment of reinforcement learning (RL) on autonomous robots. Some safe RL methods use safety constraints from prior or external knowledge to reduce or avoid unsafe explorations, but such knowledge is usually unavailable in practice, especially in unknown environments. In this work, we propose a few-shot reasoning-based safe reinforcement learning framework that includes a new few-shot learning method with dynamic support set to reason the safety of unexplored actions and hence guide safer action selection. Additionally, it endows robots with the capability of reverting to previous safe states and reflecting on failures to update the dynamic support set and further improve the accuracy of safety reasoning. Experimental results show that our new few-shot learning method is more accurate, and our proposed framework can significantly reduce the number of failures in the learning phase, especially for long-term autonomy.

Keywords: safe reinforcement learning; few-shot learning; dynamic support set; autonomous robots.

Reference to this paper should be made as follows: Wang, W., Zhou, X., Xu, B., Chen, S., Lu, M., Li, J. and Gu, Y. (2024) 'Few-shot reasoning-based safe reinforcement learning framework for autonomous robot navigation', *Int. J. Automation and Control*, Vol. 18, No. 1, pp.30–52.

Biographical notes: Weiqiang Wang is currently pursuing an MEng in Mechanical Engineering at Yancheng Institute of Technology, Yancheng, China. His research interests include reinforcement learning, robot planning, and computer vision.

Xu Zhou received his BEng in Automation at Nanjing University of Information Science and Technology in Nanjing, China, in 2011, an MEng in Control Theory and Control Engineering at Nanjing University of Science and Technology, in 2014, and a PhD in Mechanical Engineering at Colorado School of Mines in Golden, Colorado, USA. He is currently an Assistant Professor with the Department of Mechanical Engineering at Changshu Institute of Technology in Suzhou, China. His current research interests include intelligent robot control, reinforcement learning, and knowledge-based systems.

Benlian Xu is a Professor with the School of Electronic and Information Engineering at Suzhou University of Science and Technology, Suzhou, China. He received his PhD in Control Science and Engineering from the Nanjing University of Science and Technology, Nanjing, in 2006. As a Visiting Fellow, he was invited to join the research project on Biomedical Image Analysis in the University of Melbourne, the University of Western

Australia and the Australian National University, in 2009, 2012 and 2018, respectively. His current research interests focus on multi-object tracking, swarm intelligence, and simultaneous localisation and mapping.

Siwen Chen received a BEng in the Software Institute (SWI) of Nanjing University in Nanjing, China, in 2022. His current research interests include software engineering, autonomous robots, and computer vision.

Mingli Lu received her PhD in Control Science and Engineering at the Nanjing University of Science and Technology, in 2016. Currently, she is a professor with School of Electrical and Control Engineering at Changshu Institute of Technology. Her present research interests are image processing and object tracking.

Jun Li received a BEng in Automatic Control at the East China Institute of Technology in Nanjing, China, in 1991, an MEng in Control Theory and Control Engineering at the East China Institute of Technology in Nanjing, China, in 1994, and a PhD in Control Science and Engineering at the Nanjing University of Science and Technology in Nanjing, China, in 1998. He is currently a Professor with School of Automation at Nanjing University of Science and Technology in Nanjing, China. His current research interests include intelligent control, robot learning, and computer vision.

Yuejiang Gu is a senior engineer and the General Manager of R&D Center at the General Elevator Co., Ltd. He is also the Director of General Elevator Co., Ltd.

1 Introduction

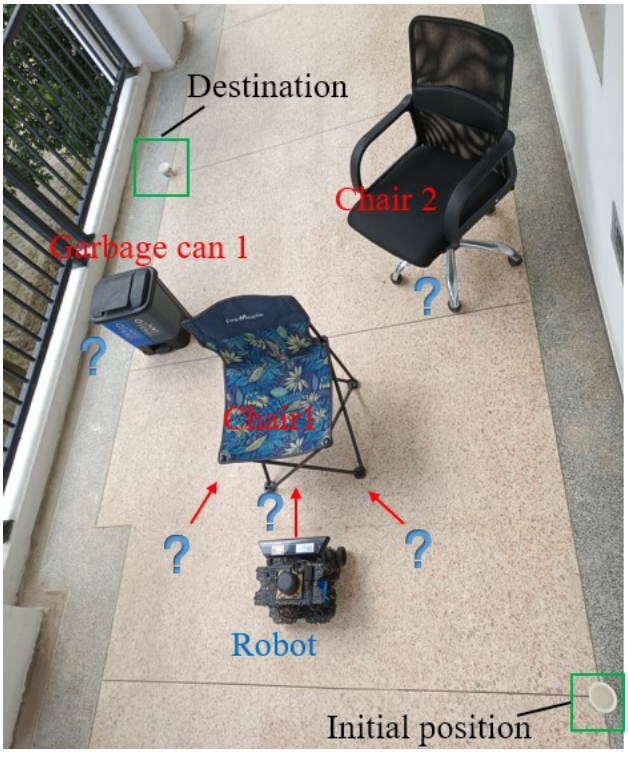
Reinforcement learning (RL) shows great potential in autonomous robot navigation, especially in unknown environments, as it can adapt to the environment by learning from the interactions between robots and environments. In order to find the optimal policy, the robot should have sufficient explorations and learn from both good and bad consequences to maximise the cumulative rewards it obtains. When applying RL to real-world autonomous robot navigation, however, unsafe actions and bad consequences may make the robot suffer from safety issues, such as collisions that cause damage to the robot or the environment. As most robots for navigation are safety-critical, it is crucial to guarantee learning safety in the practical deployment of RL.

For the safety issue during the learning phase of RL, two general categories of safe RL methods have been reported (Garcıa and Fernandez, 2015). One is transforming the traditional optimisation criteria that maximise the expectation of the return to more comprehensive criteria respecting learning safety. This method reduces the probability of risks by changing the final objective the agent needs to optimise, but it does not strictly prevent the risk from happening. The other methods modify the exploration process by choosing safe actions through the incorporation of external knowledge, such as environmental knowledge. However, we do not always have access to external knowledge that can be directly used in practice, especially in unknown environments. Considering that prohibiting the selection of unsafe actions is a fundamental and

effective way, it comes to the question whether the robot can reason useful safety knowledge from limited explorations by itself.

For example, as shown in Figure 1, there are three obstacles in the environment: chair 1, chair 2, and garbage can 1. The robot faces one side of chair 1 and assumes that it has got the safety knowledge in this position. The safe exploration mechanism aims to prevent the robot from choosing actions toward chair 1 at this position and to recognise chair 1 when facing other sides. Moreover, the knowledge obtained around chair 1 can be helpful to the inference of chair 2 or even for other items in the environment like luggage 1. It is significant to emphasise that safety knowledge is not available as prior knowledge but acquired from the environment during the exploration. To this end, a safe exploration method that can online learn safety specifications from limited explorations to ensure the selection of safe actions is expected.

Figure 1 Safety reasoning for actions in unknown environments (see online version for colours)



Notes: The robot is expected to reason and exclude unsafe actions toward to unseen obstacles based on the information that has been already explored by RL.

Furthermore, even if the action is safe when selected for exploration, it may still lead to unsafe consequences during the execution due to many unexpected disturbances, like the sudden changes in the friction between the robot and the ground. Theoretically, once the failure occurs, RL should terminate the current episode and restart from the initial position with another episode. This transition from one episode to the next episode is

easy in simulations, but it can be difficult in practical robot autonomy, where human interventions are not expected or allowed. In fact, as the already explored states are safe, it is easier and less time-consuming for the robot to go back to the previous safe states rather than resetting back to the starting point. Also, the failures usually contain more hidden safety information, so it is better for the RL-based robot to have the capability of self-recovering to previous safe states and reflecting on failures for further safe explorations.

In this work, we propose a few-shot reasoning-based safe RL framework for the practical deployment of RL in autonomous robot navigation. It improves learning safety through two important parts:

1 The safe exploration part: This part uses a new few-shot learning method with dynamic support set to reason safety relations between different actions and different obstacles and hence exclude unsafe actions in the exploration process.

2 The self-recovery part: This part can self-reverse to previous safe states when failures occur and learn from these failures to update the support set used in the few-shot learning.

Our method is demonstrated by a robot navigation example where obstacles are assumed to cause collisions. The main contributions of this work can be summarised as follows. Firstly, we develop a safe RL framework for autonomous robot navigation. The safe exploration part in this framework can reason the safety of unexplored actions and prevent unsafe actions from executing. The self-recovery part can make better use of failures to improve the reasoning performance in the safe exploration part. The self-recovery part can also revert the robot to safe states once a failure happens, which avoids frequent resets and hence is more suitable for practical application. Secondly, we propose a new few-shot learning algorithm with dynamic support set to reason the safety of unexplored actions so to improve the safety in the learning phase. The support set dynamically changed with collected real-world samples can improve the robot's adaptation to unknown environments. Lastly, we provide experiment results to present the effectiveness of our proposed framework.

2 Related work

RL methods have been gradually applied in robot navigation with the development of artificial intelligence (Tai et al., 2017). A popular application scenario is local obstacle avoidance (Duguleana and Mogan, 2016; Kato et al., 2017; Kato and Morioka, 2019), which is the basis of the other scenarios and can be extended to more complex navigation tasks (Zhu and Zhang, 2021), such as indoor navigation (Zhu et al., 2016; Devo et al., 2020a,b). These works focus more on traditional learning performance, but when applying RL in practical applications, it brings up a new question – the learning safety. Due to trial and error, RL needs to try various actions including both safe and unsafe actions to find the optimal policy. In practical applications, unsafe actions will cause unacceptable consequences to robots or environments, which motivates the development of safe RL.

The safety problem in RL is one of the significant problems in artificial intelligence safety (Amodei et al., 2016). One solution to the safety issue is formulating the problem

as a constrained Markov decision process (CMDP) (Altman, 2021). Two common approaches to solve the CMDP problem are primal-dual method (Chen et al., 2021; Xu et al., 2020; Paternain et al., 2019; Tessler et al., 2018) and constrained policy optimisation (CPO) method (Achiam et al., 2017; Yang et al., 2020). Other approaches have been proposed to solve the CMDP problem recently, like the Lyapunov method (Chow et al., 2018; Huh and Yang, 2020) and the safe layer method (Dalal et al., 2018). However, these methods are limited in the type of constraints because they require a parametrisation of the policy (policy gradient methods), which is not applicable to value-based algorithms. Additionally, these methods tend to base on dynamics or an appropriate design of risks, but this knowledge is hard to be obtained in practice. Some works solve the safety issue from other perspectives. Alshiekh et al. (2018) introduce a shielded RL that can synthesise a shield concerning the safety specifications to prevent the agent from choosing unsafe actions. Sui et al. (2019) and Turchetta et al. (2016) assume that the state spaces and the safety function have some form of regularity to keep safe. Zhou (2021) uses a potential function to shape a safe reward that biases safe explorations and gets significantly fewer failures. Saunders et al. (2017) and Turchetta et al. (2021) introduce human supervision to help the agent explore safely. In practical navigation, however, it is important for the robot to acquire safety knowledge that is not specific and obtain safety knowledge by itself. In this work, we reflect on the failures during the exploration and adopt the few-shot learning method to reason the safety of different actions.

Few-shot learning is one of the meta-learning algorithms that aim at recognising categories from very few labeled samples like humans (Biederman, 1987). Recently, most works related to few-shot learning focus on the recognition accuracy by different networks (Vinyals et al., 2016; Finn et al., 2017a; Snell et al., 2017; Sung et al., 2018; Ravi and Larochelle, 2017) in some well-known datasets such as miniImagenet (Vinyals et al., 2016) and Omniglot (Lake et al., 2011), and some works apply few-shot learning in computer vision, natural language processing, audio processing, robotics, and so on (Lu et al., 2020). In robotics fields, few-shot learning is usually applied in visual navigation (Finn et al., 2017a; Li et al., 2017; Jamal and Qi, 2019), imitation learning (Duan et al., 2017; Finn et al., 2017b), robot manipulation (Xie et al., 2018; Xu et al., 2018). So far, using few-shot learning to improve the learning safety in RL-based robot navigation has not been reported. Furthermore, the support set in the traditional few-shot learning is constant, which may have a poor performance on some unseen but similar obstacles in the environment. Thus, our work designs a support set that can be dynamically changed with respect to environments to improve obstacle recognition accuracy and safety reasoning accuracy to further improve safety of explorations.

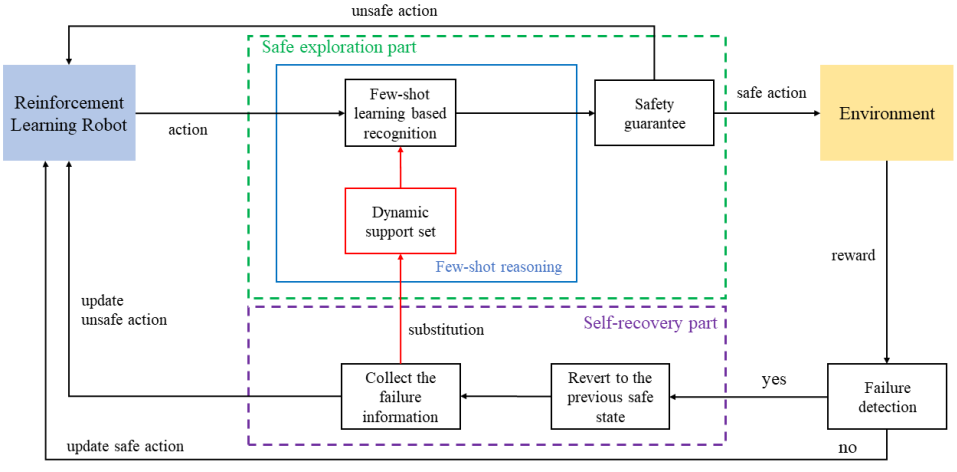
3 Methodology

In this section, we describe our few-shot reasoning-based safe RL framework as shown in Figure 2, through two parts:

1 a safe exploration part that uses few-shot learning with dynamic support set to recognise unseen obstacles and exclude unsafe actions in the future explorations

2 a self-recovery part that can self-reverse to previous states when failures occur and self-reflect on failures to update the support set.

Figure 2 Few-shot learning-based safe RL framework for autonomous robot navigation (see online version for colours)



Notes: The safe exploration part excludes unsafe actions by recognising and reasoning similar, unseen obstacles from known ones. When a failure is detected, the self-recovery part recovers the robot to a previous safe state and reflects on the failure for subsequent safer explorations.

3.1 Problem statement

The framework is for practical, safe autonomous robot navigation in unknown environments. There are a few categories of obstacles in the environment, and each with several samples for the auxiliary inference by the robot. The goal of the robot is to find an optimal path in the unknown environment autonomously and avoid as many collisions as possible.

In order to simulate a real-world scenario, we make some assumptions as follows:

1 While we do not know the specifications of obstacles such as the shape, size, colour, and so on, we assume knowing the categories of obstacles in the environment. For example, we could know there are boxes as obstacles in the environment, but the size and colour of these boxes are different, which is not known beforehand.

2 The obstacles in the environment will not cause lethal damage to robots, i.e., the robot is able to recover from the collision. For example, if the robot moves forward and collides with a box, it can move backward to achieve a safe position.

3 The robot can know the specific obstacle category after the collision. This may be achieved by taking a closer and clearer picture or using impact sensors. This is out of our work's scope, so we do not discuss it more here.

3.2 Safe exploration

We introduce the safe exploration part through two modules: few-shot reasoning and safety guarantee. As shown in Figure 2, the main module, the few-shot reasoning module, uses few-shot learning with a dynamic support set to recognise obstacles in the environment and reason for unsafe actions toward these obstacles. Another module is the safety guarantee module, which focuses on the decision-making of the agent. Specifically, the safety guarantee module monitors the actions selected by the learning agent and corrects them if and only if the chosen action is unsafe. The criterion for judging whether the action is safe or not is learned from the few-shot reasoning module.

3.2.1 Few-shot reasoning

In an unknown environment, few-shot reasoning module can be seen as a recognition module with the function of reasoning similar categories of obstacles. We first make a simple introduction to the few-shot learning algorithm. Few-shot learning tends to have three datasets: training set, support set, and query set. The training set usually has thousands of different kinds of data with labels, and it is used to train the model with the capability of telling the difference between two common categories. The support set and the query set share the same labels, but the categories of data in them never occur in the training set. The support set provides some samples for the model to refer while the query set is the real testing set the model needs to classify. We call the target few-shot problem C-way K-shot if the support set contains K labeled examples for each of C unique classes. In the few-shot learning research field, the size of K is typically 1 or 5, that is to say, one-shot or five-shot.

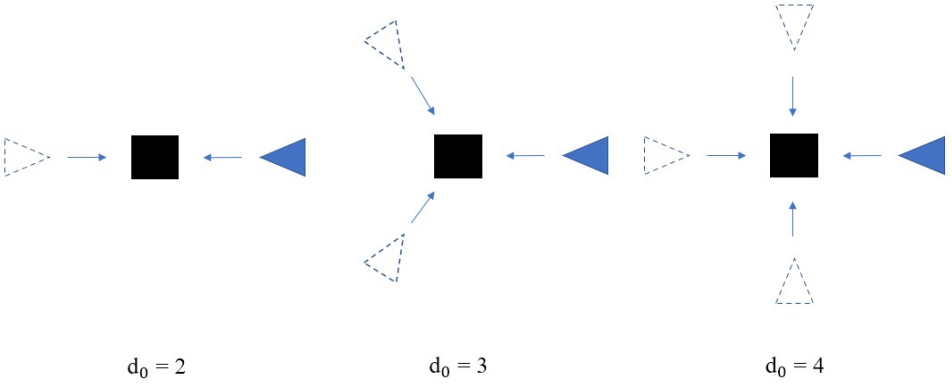
In our work, we use the relation network (Sung et al., 2018) as a recogniser for the recognition of obstacles in the navigation because the relation network can be seen as providing a learnable nonlinear classifier for determining relationships, while other few-shot learning networks like twin network and prototype network use a linear classifier of relationships. We think the models learned by the relation network may have a better performance on our own data due to the similarities between the navigation environments.

In practical navigation, each side of the obstacles can be captured as a picture belonging to the query set. All the pictures are provided for recognition. Our recognition is a 5-way 5-shot problem, and the samples in the support set never occur in the training set. We do not make any changes to the trained model and still use the similar expression in Sung et al. (2018) to describe our problem. Specifically, our training set is based on miniImagenet (Vinyals et al., 2016) and follows the split introduced by Ravi and Larochelle (2017), with 64 classes for training. Our support set and query set are data of obstacles in navigation instead of the miniImagenet dataset. The support set can be denoted as $S = \{(x_i, y_i)\}_{i=1}^{m} (m = K \times C)$, where x_i and y_i represent the sample and its label. Before we define the query set in the form of real navigation, we first divide the area around the obstacle into d_0 parts starting from the east in a clockwise direction. Our work divides the area into d_0 parts because our action space is discrete. For the continuous action space, it can be any position around the obstacle at a certain distance. Then the query set can be denoted as:

$$Q = \left\{\left(x_{js(Area_{di})}\right), y_j\right\}_{j=1}^{n}, di = 1, 2, ..., d_0, \tag{1}$$

where $Area_{di}$ represents the area of di^{th} part, and $s\left(Area_{di}\right)$ is the initial position in $Area_{di}$ towards the obstacle at a constant distance, and then $x_{js(Area_{di})}$ means the picture captured from $s\left(Area_{di}\right)$ of the j^{th} obstacle. Some examples are depicted in Figure 3 for intuitive description. n is the number of obstacles in the environment.

Figure 3 An illustration of sensing an obstacle from different positions (see online version for colours)



Notes: The value of d_0 means how many sides of an obstacle can be observed, which is also the number of this obstacle images in the query set.

Therefore, the recognition problem (few-shot reasoning algorithm) can be defined as:

$$r_{i,j}s(Area_{di}) = g_{\phi}\left(C\left(f_{\varphi}(x_i), f_{\varphi}\left(x_{js(Area_{di})}\right)\right)\right),$$
$$i = 1, 2, 3, 4, 5, di = 1, 2, ..., d_0, \tag{2}$$

the relation network combines the feature map of samples in the support set and query set, computes the relation scores between different categories in the support set, and then chooses the category of the biggest score to complete the classification problem (see Sung et al., 2018 for more details). Different from Sung et al. (2018), the support set is dynamic in our settings, which is further explained in the dynamic support set part.

3.2.2 Safety guarantee

In this part, we discuss how to guarantee safety from two perspectives. First, if recognising an obstacle correctly, how to prohibit the robot from choosing the actions toward that obstacle? Second, if recognising incorrectly, how to prohibit the robot from choosing this action again after collision?

We usually give a negative reward to unsafe actions to encourage the agent not to select those actions in RL. However, receiving negative rewards cannot fully prohibit the selection of unsafe actions. Unsafe actions can still be sampled in RL methods, although the probability is low during the later phases of training. A RL problem is usually described as a Markov decision process (Suttong and Barto, 1998), which is defined as a tuple $M = \langle S, A, P, R, \gamma \rangle$, where S is a finite set of states s, A is a finite

set of actions a, $P : S \times A \to S$ is a state transition probability matrix denoting the probability of transferring to the next state, R is a reward function, γ is a discount factor, $\gamma \in [0, 1]$. For our question, we first define a critical-states set:

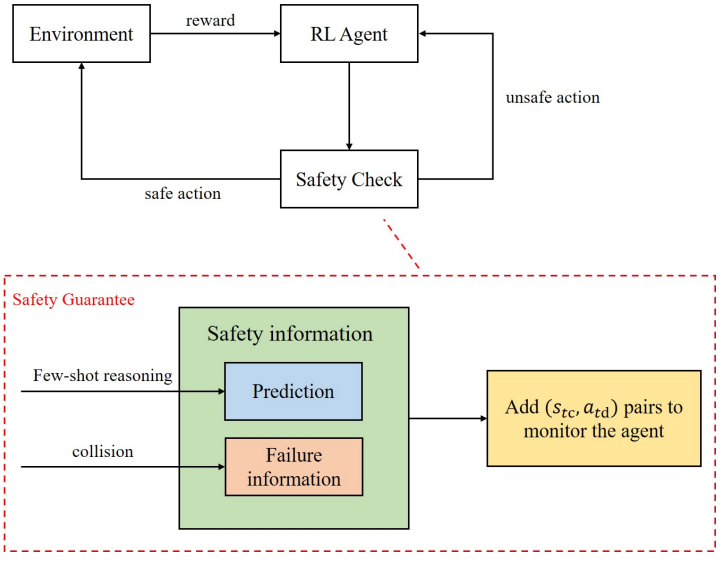
$$S_c = \{s_t \mid \forall s_t \in S, \exists s_{t+1} \in S_{obs}\}, \tag{3}$$

to denote all the critical positions around the obstacles, where S_{obs} is the set of all the obstacles. Afterward, we can define the dangerous actions set:

$$A_d = \{a_t \mid \forall a_t \in A, \exists s_{t+1} \in S_{obs}, \text{ s.t. }, s_t \in S_c, s_t \times a_t \to s_{t+1}\}, \tag{4}$$

to denote all the actions which make the agent transit from critical states to obstacle states. Thus, we use the pair (s_{tc}, a_{td}) to denote one specific critical state and the action towards a specific obstacle in this state. (s_{tc}, a_{td}) pairs are important hidden safety information that should be exploited, and the safety guarantee module's duty is to monitor the agent not to choose a_{td} at state s_{tc} as shown in Figure 4. The safety information is originated from two sources. First, it is from the few-shot reasoning module. If the module recognises the obstacle correctly, which means a collision will not happen and the robot is facing the obstacle at the critical position s_{tc}. At this time, the safety guarantee module will collect the information and prevent the robot from taking action a_{td} as a prediction. Second, if the few-shot reasoning module recognises incorrectly, which means a collision will happen, and the self-recovery part can collect the real obstacle information to safety guarantee after the collision. All the safety information is gained during the training process and the safety guarantee module gradually expands its safety specifications.

Figure 4 Safety guarantee module (see online version for colours)



Notes: The module collects safety information in two ways: the few-shot recogniser and the failure information after collisions. With the safety information, it monitors the actions selected by the learning agent and corrects them if and only if the chosen action is unsafe.

RL methods tend to have the safety issue during the learning phase. Here we use the classic Q-learning algorithm as an example for its easy implementation. In fact, our framework can work with other RL methods because the key safe exploration part and the self-recovery part are generally independent of a specific RL method. The navigation problem can be summarised as:

Maximise:

$$Q^{\pi}(s,a) = E_{\pi}\left\{\sum_{k=0}^{\infty}\gamma^{k} r_{t+k+1} \mid s_t = s, a_t = a\right\}, \tag{5}$$

Subject to:

$$a_t \notin A_d, \forall t \in [0, +\infty], \tag{6}$$

where $Q^{\pi}(s,a)$ is the long-term expectation that the Q-learning algorithm needs to optimise, the objective of the agent is to maximise $Q^{\pi}(s,a)$ as well as not violating the safety specifications. The safety guarantee module guides the action selection by restricting the agent. More specifically, if the action in the safety guarantee module is sampled by the agent in the action selection process, the safety guarantee part will prohibit the unsafe action and reselect a safe action. The action selection policy remains similar to Q-learning. Hence, the improved selection policy (using $\varepsilon - greedy$ policy) can be formed as follows:

For the probability of $1-\varepsilon$:

$$\pi(s_t, a_t) = \begin{cases} argmax_a Q(s_t, a_t), \text{if } a_t \notin A_d \\ argmax_a Q'(s_t, a_t), \text{if } a_t \in A_d \end{cases}. \tag{7}$$

For the probability of ε:

$$\pi(s_t, a_t) = \begin{cases} random_a Q(s_t, a_t), \text{if } a_t \notin A_d \\ random_a Q'(s_t, a_t), \text{if } a_t \in A_d \end{cases}, \tag{8}$$

where $Q(s_t, a_t)$ is the Q-value of all actions, and $Q'(s_t, a_t)$ is the Q-value of all actions except unsafe actions. Finally, either the safe action guided by safety guarantee module or the unsafe action resulting in collisions are evaluated by Bellman equation:

$$\begin{aligned} Q_{t+1}(s_{t+1}, a_{t+1}) = {} & Q_t(s_t, a_t) \\ & + \alpha\left[r_t + \gamma \max_{a_{t+1}} Q_{t+1}(s_{t+1}, a_{t+1}) - Q_t(s_t, a_t)\right]. \end{aligned} \tag{9}$$

3.3 Self-recovery with dynamic support set

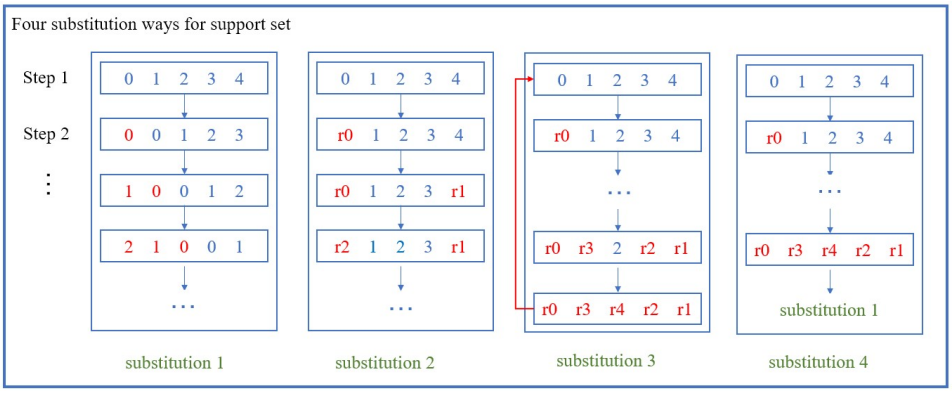
In the real-world environment, when an autonomous robot collides with an obstacle during the training process, the RL method terminates the current episode and starts a new one. In the practical deployment of RL on autonomous robots, it is not always possible to restart training from the initial position. Thus, we develop a self-recovery module to go back to the previous safe state without ending the current episode. Meanwhile, the occurred failures can be used to update the support set for few-shot learning in safe exploration.

We consider the robot's trajectory as a discrete sequence of states and actions like:

$$\tau = \{s_1, a_1, s_2, a_2, ..., s_t, a_t\}. \tag{10}$$

For example, at time t, the robot in state s_t executes action a_t, and then converts to s_{t+1}, which is an obstacle state and results in a collision. The self-recovery mechanism aims to reverse the robot to one of the past recorded t states $s_1, s_2, ..., s_t$ that are safe states without changing the framework of RL. In this work, we only consider reverting to the nearest state in the past, which uses the least memory.

Figure 5 Sample substitution in dynamic support set (see online version for colours)



Notes: Once a collision happens, the picture is substituted in the corresponding category for one of the five samples in different ways. Thus, the support set is composed of five categories, each with five dynamically changed samples.

When a collision happens, the robot first updates this unsafe action by Bellman Equation, and then reverts to the previous safe state by the self-recovery module. Meanwhile, the robot receives the real category of the obstacle due to assumption (3). Then the robot sends this message to the safety guarantee module to prohibit this unsafe action. In addition, the photo captured by the robot during a collision is a piece of useful information for the robot to reflect on itself and improve its capability of reasoning. The happening of collision is caused by the inaccuracy of the few-shot reasoning module, and the reason for low accuracy is that there is a big difference between samples in the support set and instances in the real environment (query set). Therefore, an intuitive idea to optimise this problem is to substitute some samples in the support set with the pictures in the query set, as pictures captured from different sides of an obstacle share similar features. Also, the background of the environment in the query set is closer to the real environment than the original support set.

We propose four substitution ways for the dynamic support set depicted in Figure 5. The blue numbers represent five samples of one category in the support set. We use the red numbers 0, 1, 2, ... to represent the photo sequence that is recognised incorrectly during the explorations. The first substitution way replaces the samples in order for each step, one in and one out. In the other substitution ways, we use red numbers r1, r2, ... to denote that we replace a random position of the five samples, but the random substitution rules are different. The second substitution way replaces one of the five samples randomly, and any sample is likely to be replaced at each step even if it was

just replaced at the previous step. Different from substitution way 2, the third and fourth substitution way only replace the samples by choosing one position in blue numbers randomly, which means it replaces one of the five blue number positions randomly in step 1, replaces one of the four blue number positions randomly in step 2 and so on. After all the original samples are replaced, the third substitution way restarts from step 1, while the fourth substitution way continues with the first substitution way with the substitution order recorded in the first round of substitution way 4.

3.4 Algorithm summary

Our full algorithm is shown in Algorithm 1, which consists of safe exploration and self-recovery. Safe exploration is always activated to learn from experiences and guarantee safety, while the self-recovery part is executed only when a failure occurs.

Algorithm 1 Our algorithm

Initialise $Q(s, a)$ arbitrarily
for each episode **do**
  Choose a_t from s_t using policy derived from Q $(\varepsilon - greedy)$
  if $a_t \in A_d$ **then**
    Reselect a_t ▷ Safety guarantee
  else
    if detecting a picture **then**
      Compute $r_{i,js}$ and do classification ▷ few-shot reasoning
      if recognising correctly **then**
        $A_d = A_d \cup \{a_t\}$ ▷ append a_t to safety guarantee module
        End this episode
      else
        $r(s, a) = r_{failure}$
      end if
    end if
  end if
  if $r(s, a) = r_{failure}$ **then**
    $A_d = A_d \cup \{a_t\}$
    Update the Q-value of a_t
    Do substitution for dynamic support set
    $s_{t+1} = s_t$ ▷ Revert to s_t by self-recovery
  else
    Take action a_t, observe r_t, s_{t+1}
    Update Q-value
  end if
end for
Return the optimal policy

4 Experiments and discussion

4.1 Experiment setup

In this work, we conduct an experiment to test the accuracy of few-shot learning with dynamic support set in the navigation problem and validate our safe RL framework in a

5×5 grid-world environment. The gird-world environment is about the robot navigation in an unknown maze with several obstacles.

The grid-world environment is shown on the left of Figure 6. The red block in the upper left corner of the grid environment is regarded as a mobile robot, and the yellow oval in the lower right corner represents the target location to be reached by the robot, and the black blocks represent obstacles in the unknown environment. Once the target point is reached, the agent will be given a reward of 1. Once it collides with an obstacle, the mobile robot will get a reward of –1. For other individual steps, the reward is 0. The robot needs to find the optimal path to the target location in several iterations, and it can move in four directions: north, south, east, and west, but it can move only one grid at one time. When the robot moves to the positions around the obstacle and chooses the action towards the obstacle, a picture is received by the robot for recognition. If the few-shot reasoning module fails to recognise the obstacle, the robot reverts to the previous safe state. An episode ends if the agent has taken more than 200 steps. An ε-greedy action selection method is used for Q-learning with $\varepsilon = 0.1$. The discount rate γ is set to 0.9, and the learning rate α of the agent is set to 0.01.

Figure 6 The maze navigation environment and samples in query set and support set (see online version for colours)

As formula (1) says, each side of an obstacle is a test picture of the query set. In our environment, the hyperparameter d_0 is four, which means all the pictures are captured from north, south, east, and west. Our query set consists of five categories which contain chairs, kettles, cartons, basins, and baskets, and each category has eight pictures. The pictures are captured from four sides of two obstacles, and we define the first picture in four pictures in the middle of Figure 6 as the orientation of this obstacle. We rotate the obstacle 45 degrees clockwise to represent another specific obstacle that belongs to the same category as shown in the middle of Figure 6 shows. Our support set is shown on the right of Figure 6. In the first experiment, we test the 40 pictures in the query set by counting how many pictures can be recognised correctly with different substituted numbers in the support set, which aims to validate the effectiveness of the few-shot learning algorithm with dynamic support set. In the second experiment, nine of the ten obstacles in the query set are assigned randomly to the obstacles in the environment and this means all the pictures in the environment are not repetitive. Meanwhile, the

orientation of each obstacle is random, too. If two obstacles are next to each other, the image of the adjacent sides cannot be captured by the robot. We apply our safe RL framework to the autonomous robot navigation in the grid-world environment and compare it with baselines with the index of collision number.

4.2 Experiment results

4.2.1 Few-shot learning accuracy

In our work, the support set is a 5-way 5-shot setting and the training set is the same as the relation network which is based on miniImagenet. Considering the substitution of one category has an influence on all the test data, we first test all 40 pictures in the query set, and then substitute one of the pictures in the support set with one of the pictures in the query set. The substitution is executed in the same category and the pictures selected from the support set and query set are randomly chosen. The result is shown in Table 1.

Table 1 Few-shot learning accuracy using one sample substitution

		Test accuracy						
Category		*0*	*1*	*2*	*3*	*4*	*Overall*	*Mean*
No substitution		*0/8*	*3/8*	*2/8*	*8/8*	*4/8*	*17/40*	*42.5%*
Substituted category	0	186/800	495/800	168/800	748/800	364/800	1,961/4,000	49.03%
	1	0/800	689/800	132/800	771/800	292/800	1,884/4,000	
	2	5/800	581/800	383/800	742/800	388/800	2,099/4,000	
	3	4/800	500/800	136/800	792/800	314/800	1,746/4,000	
	4	12/800	500/800	335/800	692/800	577/800	2,116/4,000	
Substituted category	0	*95/700*	496/800	155/800	747/800	373/800	1,866/3,900	48.04%
(no same picture)	1	0/800	*596/700*	134/800	756/800	303/800	1,789/3,900	
	2	11/800	574/800	*331/700*	748/800	389/800	2,053/3,900	
	3	0/800	500/800	127/800	697/700	316/800	1,640/3,900	
	4	13/800	500/800	334/800	685/800	*487/700*	2,019/3,900	

Category 0 has very low recognition accuracy while category 3 has very high recognition accuracy and other categories' accuracy is balanced among the original recognition results of five types of obstacles without substitution, which shows that our data selection is representative. We first substitute a picture chosen from eight pictures in category 0 of the query set randomly for the picture chosen from five pictures in category 0 of the support set, and do the substitution 100 times, and then each category has 800 pictures to test, 4,000 pictures in total. We count how many pictures can be recognised correctly to measure the effectiveness of substitution in category 0. Afterward, we execute the same operation on category 1, category 2, category 3, and category 4, respectively, and compute the mean accuracy. The substitution category (no same picture) means that when counting the pictures recognised correctly, we do not test the pictures being sampled to replace the support set. Thus, there are only 700 pictures in this category being counted. The result shows that the test accuracy increases from 42.5% to 49.03%, 48.04% (no same picture) when the substituted picture number is 1.

The substituted category tends to have higher accuracy, and the result shows that the effect on the whole test data is positive.

Furthermore, we do more experiments on substituting more pictures of one category in the support set, as shown in Table 2, because the real substitution process during navigation can be continuous. We substitute some of the pictures in the support set with the same number of pictures in the query set, the selection is also random, and we compute the accuracy respectively when the substituted number is 1 to 5. The result shows that the recognition accuracy increases gradually with the increase of substitution number. When it comes to the no same pic condition, the accuracy decreases a little compared to the previous result, which illustrates that when the support set contains the test picture in the query set, the test picture tends to be recognised more correctly. And the mean accuracy only increases when the substitution number changes from 1 to 2, or 3 to 4. There could be two possible reasons. First, the substitution process has the property of randomness, and those pictures which pose bad influences on the accuracy may be sampled more frequently. Second, the whole substitution process is dynamically balanced. Substitutions in one category can bring both good effects and bad effects to the recognition accuracies of other categories and hence cause small fluctuations of the mean accuracy. In summary, the accuracy of few-shot learning with dynamic support is better than the original algorithm in our navigation environment, which is consistent with our theory.

Table 2 Few-shot learning accuracy using substitutions with different number of samples

Substitution number	*Mean accuracy*	
0	42.5%	
Substitution number	*Mean accuracy*	*Mean accuracy (no same picture)*
1	49.03%	48.04%
2	50.83%	48.46%
3	52.47%	48.41%
4	53.64%	48.68%
5	54.91%	48.60%

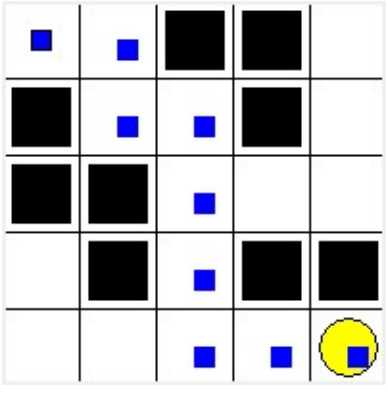
4.2.2 Comparisons of different RL algorithms

The robot ought to navigate in the grid-world environment in Figure 6 and find an optimal path to the goal position. Our framework is to ensure exploration safety during the training phase. Figure 7 shows the final optimal path the robot has learned.

Figure 8 shows the number of collisions by using different algorithms in the grid-world environment. We do 100 times independent experiments and each with 500 episodes. Then we record the collision number of each independent experiment. Owing to that all the RL algorithms can suffer from unsafe actions, the Q-learning algorithm is compared as a baseline to represent the RL algorithms without safety guarantee property. The other algorithm, Q-learning with the shield, is used as another baseline using shield RL algorithm (Alshiekh et al., 2018). We modify it by adding safety specifications during exploration rather than being computed before exploration for fair comparisons. The safe and self-recoverable RL framework (SSRL) (Wang et al., 2022)

can be seen as a rule-based algorithm based on Q-learning with the shield. It can predict the other three unsafe actions when a failure towards an obstacle occurs. The result in Figure 8 shows that algorithms with the safety guarantee can significantly reduce the number of collisions than the algorithm without a safety guarantee. SSRL and few-shot reasoning (without a dynamic support set) are comparable, and both of them have a better performance in safety exploration than Q-learning with shield. Although the performance of SSRL is slightly better than few-shot reasoning, the rule of SSRL is not always applicable in real navigations while our method does not have this restriction.

Figure 7 The optimal path (see online version for colours)



Notes: The blue dots represent the optimal path and all the RL methods for comparison are able to find it within 500 episodes.

Figure 8 Comparisons of collisions between different algorithms (see online version for colours)

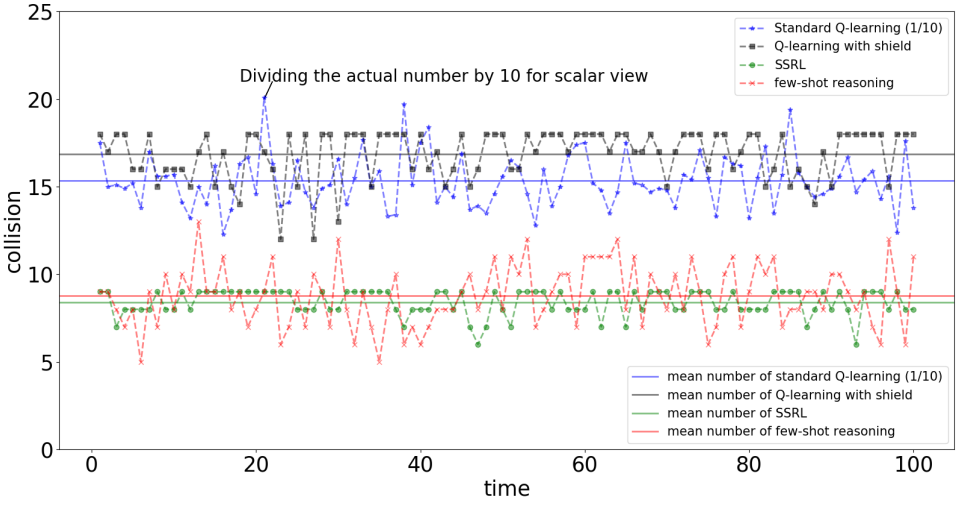
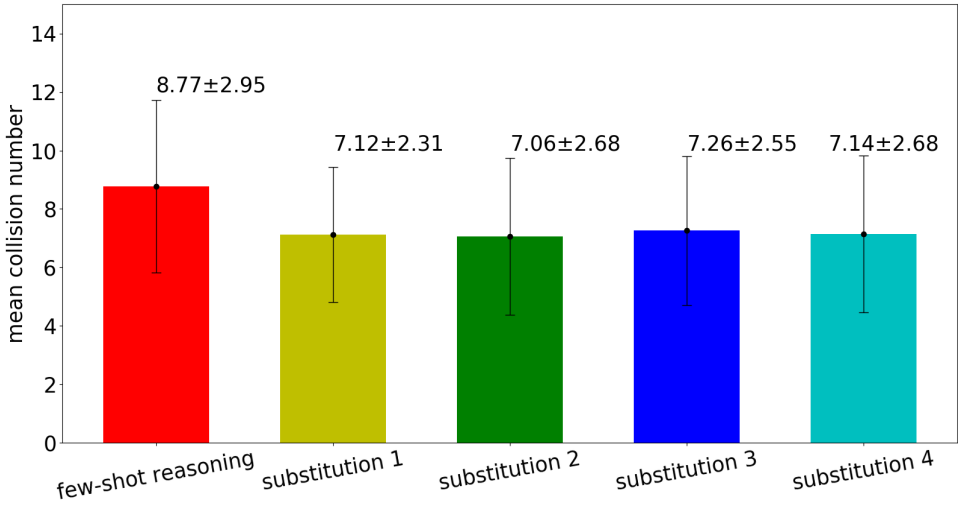


Figure 9 shows the comparison of collisions between the few-shot reasoning method with different sample substitution ways. The result shows that all the different

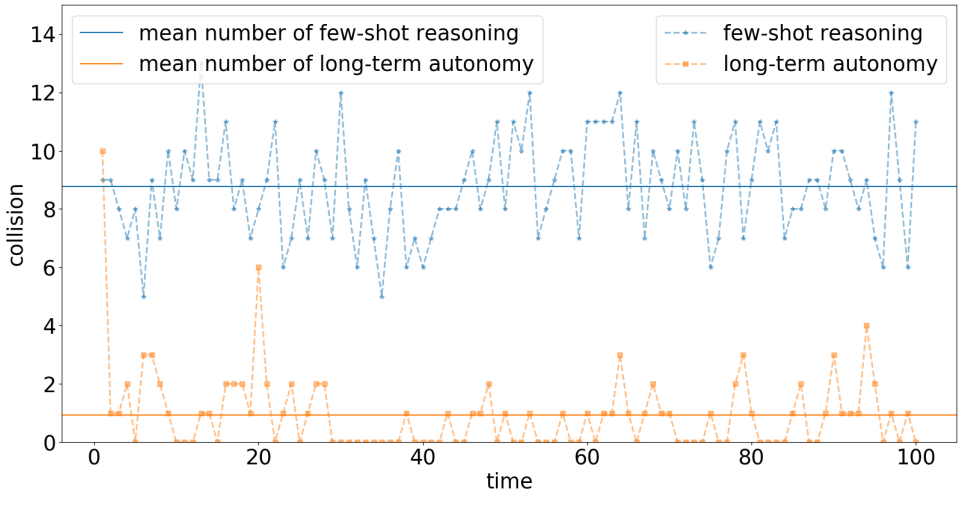
substitution ways are comparable and can further reduce the collision number than few-shot reasoning with no substitution.

Figure 9 Comparisons of collisions between different sample substitution ways (see online version for colours)



Notes: The collision number is described by the mean number and the variance of the 100 times independent experiments.

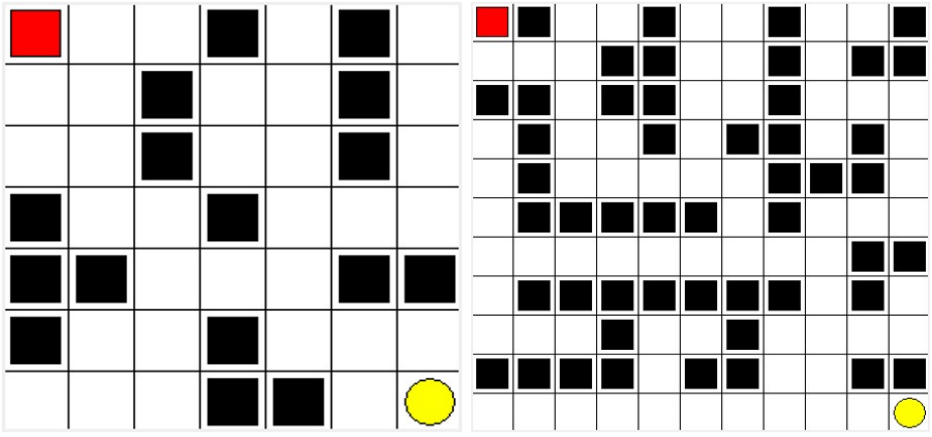
Figure 10 Comparisons of collisions between restoring the support set and not restoring the support set for subsequent experiments after completing the first one (see online version for colours)



Notes: Not restoring the support set can be considered a long-term autonomy scenario.

After completing the first independent experiment, we find that the collision number can be reduced to a very low level if we do not restore the support set to the original set in the following experiments, which is shown in Figure 10. As the recognition accuracy has increased a lot at the later stage of the first independent experiment, it is not that difficult to recognise the same picture like the first time, which validates the effectiveness of the dynamic support set, too. This result also corresponds with the few-shot accuracy experiment: the accuracy increases when the substituted picture is in both support set and query set, because most pictures in the query set are also in the support set in the second independent experiment. The practical meaning of this result is that the autonomous robot can still ensure safety to a large extent when the obstacles' positions change with time. Therefore, it is good for the long-term autonomy of mobile robots.

Figure 11 The maze navigation environment, (a) is a 7 × 7 grid world with 49 states and 16 of them are obstacles, and (b) is a 11 × 11 grid world with 121 states and 49 of them are obstacles (see online version for colours)



(a) (b)

Table 3 Mean collision numbers within 500 episodes in different maze sizes

Method / *Maze size*	*5 × 5*	*7 × 7*	*11 × 11*
Sarsa (Rummery and Niranjan, 1994)	192.3	Not converged	Not converged
Constrained RL (Achiam et al., 2017)	29.51	55.74	144.62
Our method without dynamic support set	8.77	18.39	48.44
Our method with dynamic support set	7.145	9.75	15.9

For a more comprehensive comparison, we have done more experiments with larger maze sizes as shown in Figure 11. Two more RL methods, Sarsa and constrained RL, are also added for comparison and the results can be seen in Table 3. Compared to Q-learning, the conservative exploration mechanism of Sarsa generally makes it explore in a safer way. However, in our 5 ∗ 5 maze experiment, Sarsa has more collisions than

Q-learning. The possible reason is that we have set only one optimal path surrounded by many obstacles which is hard for Sarsa to avoid. The conservative exploration also takes more time for Sarsa to converge, which can be found as not converged within 500 episodes in $7 * 7$ and $11 * 11$ environments. Since constrained RL methods are policy-based methods, it is difficult to have a direct comparison between our method and constrained RL methods. Thus, we make some modifications to apply the idea of the CPO method to the value-based RL method. To be specific, we regard one collision as constraint 1, and the threshold is set to end the current episode when the cumulative constraint on the trajectory is larger than 0.01. Meanwhile, the value of ε decreases by 0.03 for each constraint violation until it decreases to 0.01. Other experiment settings remain unchanged. The results in the last row of Table 3 are the average of the sum of the collisions using four substitution ways. From Table 3, it can be seen that our method, no matter with or without the dynamic support set, has fewer collisions than constrained RL methods.

4.3 Discussion

The results above bring up two interesting thoughts:

1 How to manage the support set dynamically by substituting those pictures which have bad influences on the classification and hence hinder the inference?

2 How can our framework be further modified to solve the problems of a dynamic environment?

These questions deserve further investigations in the future. In addition, we also find that our framework can accelerate the convergence of the RL algorithm because the safe exploration part reduces the unnecessary explorations, and the self-recovery part reduces the steps to restart a new episode from the initial position when a failure occurs.

Although our framework can work with different RL algorithms, it may need some necessary modifications or prerequisites for any specific RL algorithm to work in practice. For example, Q-learning has limitations such as the dimension disaster problem and the incapability to deal with the continuous state space or action space. Thus, when the practical environment is continuous and large, we may need to discretise the continuous environment to small square areas before directly using our method. We can also use a parameterised neural network to approximate the estimation of $Q(s, a)$ function to deal with the large dimension problem.

Moreover, traditional navigation algorithms such as A star and Dijkstra tends to rely on simultaneous localisation and mapping as a prerequisite and are supplemented by dynamic local obstacle avoidance algorithms such as the dynamic window approach may have a better performance in structured environments. However, in some unknown complex environments, where it is difficult to build a map, the traditional navigation algorithms may have poor performance or even not work, while the RL-based robot can still explore autonomously with the reasoning part and self-recovery part.

5 Conclusions

In this paper, we propose a few-shot reasoning-based safe RL framework for the practical deployment of RL on autonomous robots. With the safe exploration part to recognise obstacles and reason and prohibit unsafe actions, the robot can explore safer than traditional RL methods. Moreover, the self-recovery part makes RL more applicable to the real world by enabling the robot to revert to previous safe states for continuous learning. Experiments show that our method can significantly improve learning safety. The future work includes designing a more sophisticated reasoning method for unsafe actions. We will also validate our method with real-world experiments that use the turtlebot3 robot to navigate in a manually created maze environment.

Acknowledgements

This work was supported by National Natural Science Foundation of China (Nos. 61876024, 62241102), and partly by the higher education colleges in Jiangsu Province (No. 21KJA510003), Suzhou Municipal Science and Technology Plan Project (No. SYG202129). and Zhangjiagang Municipal Science and Technology Plan Project (No. ZKYY2222).

References

Achiam, J., Held, D., Tamar, A. and Abbeel, P. (2017) 'Constrained policy optimization', in *International Conference on Machine Learning*, pp.22–31.

Alshiekh, M., Bloem, R., Ehlers, R., Könighofer, B., Niekum, S. and Topcu, U. (2018) 'Safe reinforcement learning via shielding', in *Proceedings of the AAAI Conference on Artificial Intelligence*, Louisiana, USA, pp.2669–2678.

Altman, E. (2021) *Constrained Markov Decision Processes: Stochastic Modeling, Routledge*, Boca Raton.

Amodei, D., Olah, C., Steinhardt, J., Christiano, P., Schulman, J. and Mané, D. (2016) *Concrete Problems in AI Safety* [online] https://arxiv.org/abs/1606.06565.

Biederman, I. (1987) 'Recognition-by-components: a theory of human image understanding', *Psychological Review*, Vol. 94, No. 2, pp.115–147.

Chen, Y., Dong, J. and Wang, Z. (2021) *A Primal-Dual Approach to Constrained Markov Decision Processes* [online] http://arxiv.org/abs/2101.10895.

Chow, Y., Nachum, O., Duenez-Guzman, E. and Ghavamzadeh, M. (2018) 'A Lyapunov-based approach to safe reinforcement learning', in *Advances in Neural Information Processing Systems*, NY, USA, pp.8103–8112.

Dalal, G., Dvijotham, K., Vecerik, M., Hester, T., Paduraru, C. and Tassa, Y. (2018) *Safe Exploration in Continuous Action Spaces* [online] http://arxiv.org/abs/1801.08757.

Devo, A., Mezzetti, G., Costante, G., Fravolini, M.L. and Valigi, P. (2020) 'Towards generalization in target-driven visual navigation by using deep reinforcement learning', *IEEE Transactions on Robotics*, Vol. 36, No. 5, pp.1546–1561.

Devo, A., Costante, G. and Valigi, P. (2020) 'Deep reinforcement learning for instruction following visual navigation in 3D maze-like environments', *IEEE Robotics and Automation Letters*, Vol. 5, No. 2, pp.1175–1182.

Duan, Y., Andrychowicz, M., Stadie, B., Ho, O.J., Schneider, J., Sutskever, I., Abbeel, P. and Zaremba, W. (2017) 'One-shot imitation learning', in *Advances in Neural Information Processing Systems*, Long Beach, CA, USA, pp.1087–1098.

Duguleana, M. and Mogan, G. (2016) 'Neural networks based reinforcement learning for mobile robots obstacle avoidance', *Expert Systems with Applications*, Vol. 62, pp.104–115.

Finn, C., Abbeel, P. and Levine, S. (2017a) 'Model-agnostic meta-learning for fast adaptation of deep networks', in *International Conference on Machine Learning*, Sydney, Australia, pp.1126–1135.

Finn, C., Yu, T., Zhang, T., Abbeel, P. and Levine, S. (2017b) 'One-shot visual imitation learning via meta-learning', in *Conference on Robot Learning*, Mountain View, USA, pp.357–368.

Garcıa, J. and Fernandez, F. (2015) 'A comprehensive survey on safe reinforcement learning', *Journal of Machine Learning Research*, Vol. 16, No. 1, pp.1437–1480.

Huh, S. and Yang, I. (2020) *Safe Reinforcement Learning for Probabilistic Reachability and Safety Specifications: A Lyapunov-based Approach* [online] http://arxiv.org/abs/2002.10126.

Jamal, M.A. and Qi, G-J. (2019) 'Task agnostic meta-learning for few-shot learning', in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Long Beach, CA, USA, pp.11711–11719.

Kato, Y. and Morioka, K. (2019) 'Autonomous robot navigation system without grid maps based on double deep Q-network and RTK-GNSS localization in outdoor environments', in *2019 IEEE/SICE International Symposium on System Integration (SII)*, Paris, France, pp.346–351.

Kato, Y., Kamiyama, K. and Morioka, K. (2017) 'Autonomous robot navigation system with learning based on deep Q-network and topological maps', in *2017 IEEE/SICE International Symposium on System Integration (SII)*, Taipei, Taiwan, pp.1040–1046.

Lake, B.M., Salakhutdinov, R., Gross, J. and Tenenbaum, J.B. (2011) 'One shot learning of simple visual concepts', in *Proceedings of the Annual Meeting of the Cognitive Science Society*, Vol. 33, No. 33.

Li, Z., Zhou, F., Chen, F. and Li, H. (2017) *Meta-SGD: Learning to Learn Quickly for Few-Shot Learning* [online] http://arxiv.org/abs/1707.09835.

Lu, J., Gong, P., Ye, J. and Zhang, C. (2020) *Learning from Very Few Samples: A Survey* [online] http://arxiv.org/abs/2009.02653.

Paternain, S., Chamon, L., Calvo-Fullana, M. and Ribeiro, A. (2019) 'Constrained reinforcement learning has zero duality gap', *Advances in Neural Information Processing Systems*, Vancouver, Canada, pp.7555–7565.

Ravi, S. and Larochelle, H. (2017) 'Optimization as a model for few-shot learning', in *International Conference on Learning Representations*, Toulon, France, pp.1–11.

Rummery, G.A. and Niranjan M. (1994) *On-line Q-Learning using Connectionist Systems*, University of Cambridge, Cambridge, UK.

Saunders, W., Sastry, G., Stuhlmueller, A. and Evans, O. (2017) *Trial without Error: Towards Safe Reinforcement Learning via Human Intervention* [online] http://arxiv.org/abs/1707.05173.

Snell, J., Swersky, K. and Zemel, R. (2017) 'Prototypical networks for few-shot learning', *Advances in Neural Information Processing Systems*, pp.4080–4090.

Sui, Y., Gotovos, A., Burdick, J.W. and Krause, A. (2019) 'Safe exploration for optimization with Gaussian processes', in *International Conference on Machine Learning*, pp.997–1005.

Sung, F., Yang, Y., Zhang, L., Xiang, T., Torr, P.H.S. and Hospedales, T.M. (2018) 'Learning to compare: relation network for few-shot learning', in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Salt Lake City, UT, pp.1199–1208.

Sutton, R.S. and Barto, A.G. (1998) *Reinforcement Learning: An Introduction*, MIT Press, Cambridge, Mass.

Tai, L., Paolo, G. and Liu, M. (2017) 'Virtual-to-real deep reinforcement learning: continuous control of mobile robots for mapless navigation', in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Vancouver, Canada, pp.31–36.

Tessler, C., Mankowitz, D.J. and Mannor, S. (2018) *Reward Constrained Policy Optimization* [online] http://arxiv.org/abs/1805.11074.

Turchetta, M., Berkenkamp, F. and Krause, A. (2016) 'Safe exploration in finite Markov decision processes with Gaussian processes', in *Advances in Neural Information Processing Systems*, Barcelona, Spain, pp.4312–4320.

Turchetta, M., Kolobov, A., Shah, S., Krause, A. and Agarwal, A. (2021) *Safe Reinforcement Learning via Curriculum Induction* [online] http://arxiv.org/abs/2006.12136.

Vinyals, O., Blundell, C. and Lillicrap, T. (2016) 'Matching networks for one shot learning', in *Advances in Neural Information Processing Systems*, Barcelona, Spain, pp.3637–3645.

Wang, W., Zhou, X., Xu, B., Lu, M., Zhang, Y. and Gu, Y. (2022) 'A safe and self-recoverable reinforcement learning framework for autonomous robots', in *2022 41st Chinese Control Conference (CCC)*, Hefei, China, pp.3878–3883.

Xie, A., Singh, A., Levine, S. and Finn, C. (2018) 'Few-shot goal inference for visuomotor learning and planning', in *Conference on Robot Learning*, Zurich, Switzerland, pp.40–52.

Xu, D., Nair, S., Zhu, Y., Gao, J., Garg, A., Fei-Fei, L. and Savarese, S. (2018) 'Neural task programming: learning to generalize across hierarchical tasks', in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, Madrid, Spain, pp.3795–3802.

Xu, T., Liang, Y. and Lan, G. (2020) *CRPO: A New Approach for Safe Reinforcement Learning with Convergence Guarantee* [online] http://arxiv.org/abs/2011.05869.

Yang, T-Y., Rosca, J., Narasimhan, K. and Ramadge, P.J. (2020) *Projection-Based Constrained Policy Optimization* [online] http://arxiv.org/abs/2010.03152.

Zhou, X. (2021) 'Operational safe control for reinforcement-learning-based robot autonomy', in *2021 40th Chinese Control Conference*, Shanghai, China, pp.4091–4095.

Zhu, K. and Zhang, T. (2021) 'Deep reinforcement learning based mobile robot navigation: a review', *Tsinghua Science and Technology*, Vol. 26, No. 5, pp.674–691.

Zhu, Y., Mottaghi, R., Kolve, E., Lim, J.J., Gupta, A., Fei-Fei, L. and Farhadi, A. (2016) 'Target-driven visual navigation in indoor scenes using deep reinforcement learning', in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, Marina Bay Sands, Singapore, pp.3357–3364.