

# SRL-VIC: A Variable Stiffness-Based Safe Reinforcement Learning for Contact-Rich Robotic Tasks

Heng Zhang<sup>1</sup>, Gokhan Solak<sup>1</sup>, Gustavo J. G. Lahr<sup>1</sup>, and Arash Ajoudani<sup>1</sup>, *Member, IEEE*

**Abstract**—Reinforcement learning (RL) has emerged as a promising paradigm in complex and continuous robotic tasks, however, safe exploration has been one of the main challenges, especially in contact-rich manipulation tasks in unstructured environments. Focusing on this issue, we propose SRL-VIC: a model-free safe RL framework combined with a variable impedance controller (VIC). Specifically, *safety critic* and *recovery policy* networks are pre-trained where *safety critic* evaluates the safety of the next action using a risk value before it is executed and the *recovery policy* suggests a corrective action if the risk value is high. Furthermore, the policies are updated online where the task policy not only achieves the task but also modulates the stiffness parameters to keep a safe and compliant profile. A set of experiments in contact-rich maze tasks demonstrate that our framework outperforms the baselines (without the recovery mechanism and without the VIC), yielding a good trade-off between efficient task accomplishment and safety guarantee. We show our policy trained on simulation can be deployed on a physical robot without fine-tuning, achieving successful task completion with robustness and generalization.

**Index Terms**—Compliance and impedance control, reinforcement learning (RL), robotics and automation in construction.

## I. INTRODUCTION

EXPLORING unstructured environments is a key skill for future robotics, where the robots will be employed in unknown and potentially hostile surroundings. The inherent physical interactions in such scenarios pose risks to robots, humans, and the environment. These risks are increased in the context of automated learning and exploration, e.g., in RL. Indeed, ensuring the safety of RL in tasks involving rich contact remains an ongoing challenge, especially in unstructured environments.

Manuscript received 11 January 2024; accepted 24 April 2024. Date of publication 2 May 2024; date of current version 8 May 2024. This letter was recommended for publication by Associate Editor J. Rojas and Editor J. Kober upon evaluation of the reviewers' comments. This work was supported in part by the European Union's Horizon 2020 Research and Innovation Programme SOPHIA under Grant 871237 and in part by CONCERT under Grant 101016007. (Corresponding author: Heng Zhang.)

Heng Zhang is with Human-Robot Interfaces and Interaction Lab, Istituto Italiano di Tecnologia, 16121 Genoa, Italy, and also with the Ph.D. Program of National Interest in Robotics and Intelligent Machines (DRIM), Università di Genova, 16126 Genoa, Italy (e-mail: heng.zhang@iit.it).

Gokhan Solak, Gustavo J. G. Lahr, and Arash Ajoudani are with Human-Robot Interfaces and Interaction Lab, Istituto Italiano di Tecnologia, 16121 Genoa, Italy.

The video is available at <https://youtu.be/ksWXR3vByoQ>.

This letter has supplementary downloadable material available at <https://doi.org/10.1109/LRA.2024.3396368>, provided by the authors.

Digital Object Identifier 10.1109/LRA.2024.3396368

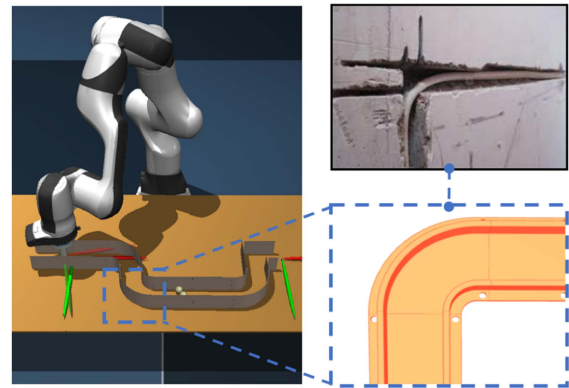


Fig. 1. Maze exploration is an unstructured contact-rich task that has practical applications such as cable-lying in walls. It also has similarities to search missions in dark and narrow environments. The agent does not have access to vision, thus it should navigate using only the contact information.

We study the problem of safe RL on a maze exploration scenario as an illustrative example of contact-rich tasks. This scenario is inspired by the common construction task of installing electricity cables within walls, as depicted in Fig. 1. Traditionally, these tasks are done by workers and are time-consuming and labor-intensive. These characteristics indicate high potential for robotic automation, but raise several challenges: 1) The environment is unknown to the robot; 2) Visual perception may be difficult to deploy, thus physical contact is essential; 3) Frequent contact with the environment causes safety issues. To this end, we abstracted the cable-lying task into a robot exploration task since all the slots in the wall are composed of two basic shapes (turns and straight sections) as shown in Fig. 1. Then, we created an experimental setup that includes a 7-DoF robot arm and a maze with obstacles as shown in Fig. 1.

In tasks involving rich contact, the robot must engage in continuous interaction with the environment, presenting a notable safety risk that could result in damage to either the robot itself or objects in its surroundings. Impedance control is widely used to alleviate such hazards during physical robot interaction [1]. Moreover, variable impedance control (VIC) methods enable adapting impedance parameters to the changing conditions of the task or the surrounding environment [2], [3]. The robot can trade between compliance and accuracy to maximize task performance while staying safe. Nevertheless, determining the adjustment of these parameters in VIC, particularly in unstructured

environments with dynamic obstacles, remains a challenging question.

Learning-based approaches became a powerful way to address this challenge, with a wide variety of methods for tuning impedance parameters [2]. In previous works, such parameters were learned from human (or expert) demonstrations using Dynamic movement primitives (DMPs) [4], inverse reinforcement learning (IRL) [5] and skill priors [6]; however, providing demonstrations in advance is relatively time-consuming and labor-intensive. Also, it can be impractical to provide expert demonstrations in some tasks where the environment is unstructured or potentially hostile.

In summarizing the current state-of-the-art research on contact-rich tasks, two notable approaches emerge: 1) Some studies employ VIC to ensure safe performance while utilizing RL for learning impedance parameters [6], [7]; 2) Other works leverage the predictive capabilities of neural networks to anticipate the safety value of future actions, allowing for informed decisions on whether to pursue a safer course of action in advance [8], [9]. Each of these ideas has its own advantage: the former is to adjust the behaviour of the robot reactively based on the observed contact forces, while the latter is to proactively select a safer robot behaviour by predicting the safety of future actions, therefore avoiding actions that would generate dangerous amounts of contact force. They both improve the safety of robots when performing contact-rich tasks to some extent.

Compared with other previous work, in this letter, we take a step towards combining the advantages of an RL-based VIC approach with the advantages of using a predictive safe RL approach. We introduce a new safe-RL framework incorporating VIC for contact-rich tasks. Following the recovery RL (RRL) method [8], during the pre-training phase, we train a *Safety-Critic* network using procedurally collected offline data, which can predict a risk value based on the current state and the next action. Meanwhile, we pre-train a *recovery policy* network, which can sample a safe action if the risk value exceeds the safety threshold. In the online training phase, a task policy is trained with the goal of maximizing the reward. Unlike previous works, we do not use expert demonstrations to accelerate task policy training, given that the environment is unknown to the robot.

We compare the proposed framework with scenarios where VIC is absent and where a recovery mechanism is not implemented. The results of our simulation experiments indicate that the combination of VIC and recovery mechanism improves the safety and efficiency of the solution.

To summarize, the main contributions of our work are listed as follows:

- 1) We present a safe RL framework with VIC for contact-rich tasks without prior knowledge of the environment. Prior to action execution, the *safety critic* predicts the safety of the action. The *recovery policy* is activated for risky situations. During the action execution, the contact force is regulated with the help of VIC to enhance safety.
- 2) We developed a contact-rich maze exploration task in simulations, and empirically demonstrated the proposed framework outperforms baselines.

- 3) We demonstrate that our policy trained on simulation can be deployed on a physical robot without fine-tuning, successfully achieving the tasks even with different type of obstacles, maze sizes and shapes, and flange sizes.

## II. RELATED WORK

### A. Contact-Rich Robotic Tasks

The solutions for contact-rich tasks have undergone a shift from contact modeling analysis [10] to learning-based approaches [11], [12]. However, safety issues have always been an inescapable aspect in such contact-rich tasks.

### B. Variable Impedance Control for Contact-Rich Tasks

Implementing VIC enables a robot to exhibit a certain degree of compliance, contributing to the achievement of safe behavior in contact-rich tasks. However, it is a challenge to pre-define the impedance parameters according to different tasks or different stages of the task. To solve this problem, Yang et al. [13] utilized muscle surface electromyography (sEMG) signals to extract the demonstrator's variable stiffness which were passed to the robot. In this type of method, however, specialized equipment is required. Lee et al. [14] proposed a stiffness modulation method that achieved safe and robust performance in a door opening task, while it focused on stiffness in joint space. A self-turning impedance was proposed in [3], which ensures compliant contact and low contact forces based on two planning strategies. However, the direction of motion of the robot will not change unless large contact forces are generated, which means that the actual trajectory executed by the robot may not be optimal with respect to energy and time. Furthermore, some scholars [6], [15], [16] learn different impedance parameters from collected expert demonstration trajectories. However, it is difficult to demonstrate trajectories for tasks like ours, i.e., blind maze exploration. Moreover, some works address it by utilizing the impedance properties of the human body. Yang et al. [13], [17] transferred the stiffness profile from the upper limb of the human to a humanoid dual-arm robot for contact-rich tasks.

Therefore, these methods of obtaining variable impedance parameters either need demonstrations or require special devices or are specific to a single task.

### C. Safe RL for Contact-Rich Tasks

RL-based works focus on contact-rich tasks can be divided into two groups depending on when contact safety is increased. On the one hand, a safe controller with variable stiffness can be learned via RL. Specifically, Martin-Martin et al. [7] compared the effect of different action spaces in RL with variable impedance control in end-effector space for contact-rich tasks, such as surface-wiping and door-opening. Some work [4], [5], [6] leveraged demonstrations to learn variable impedance parameters as part of the action space. On the other hand, a safe *action* or *policy* can be learned with RL to significantly enhance safety during the training stage. For instance, MoPA-RL [18] augmented the action space with long-horizon planning to improve safety. *Safety critic* network can be trained that generate a

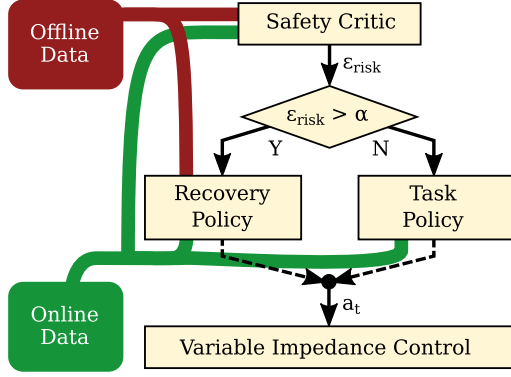


Fig. 2. Proposed framework: we combine the recovery-based safe RL approach [8] with VIC to solve the contact-rich maze exploration task. We first use an automated procedure to collect the offline data and pre-train our safety critic and recovery policy. Then, we train all learning components using online data. The risk value  $\epsilon_{risk}$  is used to activate either the task or recovery policy. The action  $a_t$  is chosen by the activated component, and it is fed to the VIC. Our action includes a relative position change and desired stiffness vector  $\{K_x, K_y\}$ .

risk value to help task policy sample a conservative or relatively safe action, works such as RRL [8] and CSC [19].

However, in contrast to these, our proposed framework combines the strengths of both. We improve safety in terms of both impedance parameters in the controller and safety prediction during training. Furthermore, unlike most of the existing methods, we do not use expert demonstrations for task learning. Our method uses a scripted procedure to demonstrate only the concept of collision constraints which is a more immediate and simpler concept compared to a long-term task, hence easier to program.

### III. APPROACH

We introduce the details of our approach in this section. In Section III-A, the fundamentals of impedance control in task space are presented, which is our downstream controller. Then, we describe the proposed RRL-based VIC in Section III-B. A diagram of the proposed framework is shown in Fig. 2.

#### A. Impedance Control in Task Space

The robotic manipulator rigid body model in joint space may be written as

$$M(\mathbf{q})\ddot{\mathbf{q}} + C(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{g}(\mathbf{q}) = \boldsymbol{\tau} + \boldsymbol{\tau}_{ext}, \quad (1)$$

where  $\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}} \in \mathbb{R}^n$  are the position, velocity, and acceleration vectors in joint space, respectively, and  $n$  is the number of DoF.  $M(\mathbf{q}) \in \mathbb{R}^{n \times n}$  is the joint space inertia matrix,  $C(\mathbf{q}, \dot{\mathbf{q}}) \in \mathbb{R}^n$  the Coriolis term,  $\mathbf{g}(\mathbf{q}) \in \mathbb{R}^n$  the gravity term, and  $\boldsymbol{\tau} \in \mathbb{R}^n$  is the actuation torques. The external torques generated by the interaction forces are given by  $\boldsymbol{\tau}_{ext} = \mathbf{J}^T(\mathbf{q})\mathbf{F}_{ext}$ , being  $\mathbf{J}^T(\mathbf{q}) \in \mathbb{R}^{n \times 6}$  the Jacobian transpose matrix.

The actuation torques are chosen to compensate for the Coriolis and gravitational vectors, plus to deal with the external forces risen by contact with the environment through an impedance model in task space:

$$\mathbf{F}_{ext} = \mathbf{D}^d \dot{\tilde{\mathbf{x}}} + \mathbf{K}^d \tilde{\mathbf{x}}, \quad (2)$$

where  $\tilde{\mathbf{x}}$  is the Cartesian pose error given by the difference between the desired Cartesian pose  $\mathbf{x}_d \in \mathbb{R}^6$  and the actual pose  $\mathbf{x} \in \mathbb{R}^6$ . Accordingly,  $\dot{\tilde{\mathbf{x}}}$  is the velocity error between the desired and actual end-effector's velocity,  $\dot{\mathbf{x}}, \dot{\mathbf{x}}_d \in \mathbb{R}^6$ , respectively. The desired stiffness matrix  $\mathbf{K}^d \in \mathbb{R}^{6 \times 6}$  is a diagonal matrix with variable terms as  $\mathbf{K}^d = \text{diag}\{K_x, K_y, K_z, 0.15K_x, 0.15K_y, 0\}$ , where the rotation elements for  $[K_x, K_y]$  are set as 0.15 times of  $[K_x, K_y]$  inspired by [3] to reduce and simplify the parameters tuning. We set the rotational z-axis stiffness to zero, to keep it indifferent w.r.t. the end-effector roll. The desired Cartesian damping matrix  $\mathbf{D}^d \in \mathbb{R}^{6 \times 6}$  is also diagonal and its terms are defined by the double diagonal rule, i.e.,  $\mathbf{D}_i^d(t) = 2\zeta\sqrt{\mathbf{K}_i^d}$  where  $i$  represents the linear or rotational components in Task Space. We use  $\zeta=0.707$  in our experiments. The multi-dimensional stiffness can be represented with a stiffness ellipsoid having different compliance in different dimensions [3].

#### B. Safe RL for Contact-Rich Tasks

We adopt the RRL framework [8] to actively predict safety hazards and take corrective actions. The original RRL was not specifically designed for contact-rich tasks. In [8] contact was avoided, on the contrary, in our work continuous contact is required but possibly unsafe. Thus, we introduce impedance control in the task space to better handle the contact-rich aspect of the problem. Specifically, we add the Cartesian impedance parameters to the action space in RL.

As shown in Fig. 2, there are three components in our framework: Firstly, similar to RRL, a *safety critic* network and *recovery policy* are trained with the pre-collected offline data. Then, our task policy was trained online with the maze exploration experiments. The output of the task policy is given to the variable impedance controller which controls the robot to achieve the desired actions.

In this framework, the problem is defined as a constrained Markov decision process (CMDP) problem [20]. A tuple  $(\mathcal{S}, \mathcal{A}, R, P, \gamma, \mu, \mathcal{C})$  denotes a CMDP, where  $\mathcal{S}$  is the state space,  $\mathcal{A}$  is the action space,  $R: \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}$  is the reward function,  $P$  is the state transition probability,  $\gamma \in (0, 1)$  is a reward discount factor,  $\mu$  is the starting state distribution and  $\mathcal{C} = \{(c_i: \mathcal{S} \rightarrow \{0, 1\}, \chi_i \in \mathbb{R}) \mid i \in \mathbb{Z}\}$  denote safety constraints that the agent must satisfy. In our task, we set  $\gamma$  as 0.9 and we define the other components as detailed below.

- **State Space  $\mathcal{S}$ .** To make our framework more general to contact-rich exploration tasks instead of this specific maze, we differentiate the state space across different networks: it is 6-dimensional vector for *safety critic* network and *recovery policy* including 6 force/torque  $[F_x, F_y, F_z, T_x, T_y, T_z]$  values measured from a F/T sensor. We do not give the position as an input to the safety network, so that it does not memorise the locations of the walls. While for task policy, we add the position  $[P_x, P_y, P_z]$  of the end-effector, obtaining a 9-dimensional vector as state space so that the agent remembers the position when finding its way to the goal.
- **Action Space  $\mathcal{A}$ .** We combine 2 impedance parameters  $[K_x, K_y]$  with 2 end-effector displacement values

$[\Delta P_x, \Delta P_y]$  which consist of 4-dimensional vector as action space. Note that here the displacement values  $\Delta P$  are not absolute positions in the world coordinate but the desired change w.r.t. the end-effector position. The sample range of  $\Delta P$  is  $[-0.03, 0.03]m$  and the impedance values  $[K_x, K_y]$  are within the sample range of  $[300, 1000]N/m$ . We project the stiffness values onto the  $\Delta P$  vector components as  $[K_x^*, K_y^*]$  in the controller:

$$\begin{aligned} K_x^* &= |K_x \cdot \Delta P_x / \|\Delta P\| + |K_y \cdot -\Delta P_y / \|\Delta P\| \\ K_y^* &= |K_x \cdot \Delta P_y / \|\Delta P\| + |K_y \cdot \Delta P_x / \|\Delta P\| \end{aligned} \quad (3)$$

- **Reward Function  $R$ .** We use a negative Euclidean distance between the current position  $P_{cur}$  of the end-effector and the goal position  $P_{goal}$  multiplied by a constant  $c_{pos}=100$ . We introduce a penalty for high-force collisions  $r_{col}=-250$  and a penalty for leaving from the entrance  $r_{ent}=-500$  to satisfy our task requirements. Also we add a bonus for reaching the goal  $r_{goal}=1000$  to increase the speed of convergence.
- **Safety Constraints  $\mathcal{C}$ .** The force constraint requires the measured force magnitude to stay below a threshold.

1) **Safety Critic and Recovery Policy:** Based on [21], our *safety critic* learn a critic function  $Q_{risk}^\pi$ :

$$\begin{aligned} Q_{risk}^\pi(s_t, a_t) &= \mathbb{E}_\pi \left[ \sum_{t'=t}^{\infty} \gamma^{t'-t} c_{t'} \mid s_t, a_t \right] \\ &= c_t + (1 - c_t) \gamma_{risk} \mathbb{E}_\pi [Q_{risk}^\pi(s_{t+1}, a_{t+1}) \mid s_t, a_t] \end{aligned} \quad (4)$$

where  $c_t = 1$  denotes constraint violation, and  $c_t = 0$  indicates safety in state  $s_t$ .  $Q_{risk}^\pi$  is trained by minimizing a MSE loss function:

$$\begin{aligned} J_{risk}(s_t, a_t, s_{t+1}; \phi) &= \frac{1}{2} \left( \hat{Q}_{\phi, risk}^\pi(s_t, a_t) \right. \\ &\quad \left. - \left( c_t + (1 - c_t) \gamma_{risk} \mathbb{E}_{a_{t+1} \sim \pi(\cdot | s_{t+1})} \left[ \hat{Q}_{\phi, risk}^\pi(s_{t+1}, a_{t+1}) \right] \right) \right)^2 \end{aligned} \quad (5)$$

Similar to [8], we use an off-policy DDPG [22] algorithm to learn *recovery* policy  $\pi_{rec}$  by performing gradient descent on the *safety critic* network  $\hat{Q}_{\phi, risk}^\pi(s, \pi_{rec}(s))$ .

2) **Pre-Training:** By providing a set of offline data for the agent, we expect our agent to learn the relationship of safety with contact force and motion. It is possible to avoid catastrophic collisions caused by unexpected actions during online training by learning the implicit relationship between the two in advance. Details of the data collection procedure are described in Section IV-B1. Unlike online training where the agent searches for actions that maximise the reward, in this stage, we run a scripted behavior to collect examples of collisions without any reward. This approach reduces the problem complexity by separating task learning and constraint learning which are otherwise interlaced.

We do not pre-train the task policy, because this requires recording complex expert demonstrations which are costly and difficult to obtain for a blind maze exploration task.

We pre-train the *safety critic* network and *recovery* policy with the offline data which are collected in simulation. Specifically, We use 256 as batch size according to the number of hidden units in the network and update the *safety critic* network and *recovery* policy 10000 times in the pre-training stage.

3) **Task Policy:** The aim of the task policy is to finish the task and obtain maximum rewards within the constraints of safety. We train the task policy by the SAC algorithm [23], which is based on the idea of maximum entropy. The main difference from other RL algorithms is that SAC maximizes the entropy while optimizing the policy for higher cumulative returns. For our task, the benefit of introducing entropy into the learning stage is the increased randomization. Therefore, the agent can sufficiently explore the state space to avoid getting stuck in the local optimum. Our task policy works in complementarity with the recovery network described in III-B1. Action is selected as follows depending on the risk value:

$$a_t = \begin{cases} a_t^{\pi_{task}} & Q(s_t, a_t^{\pi_{task}}) \leq \epsilon_{risk} \\ a_t^{\pi_{rec}} & Q(s_t, a_t^{\pi_{rec}}) > \epsilon_{risk} \end{cases} \quad (6)$$

where  $a_t^{\pi_{task}} \sim \pi_{task}(\cdot | s_t)$  and  $a_t^{\pi_{rec}} \sim \pi_{rec}(\cdot | s_t)$ . Different from task policy, the *recovery* policy is trained to minimize the  $\hat{Q}_{\phi, risk}^\pi(s_t, a_t)$  so that it can provide a safe action  $a_t^{\pi_{rec}}$  with small risk value i.e.  $Q(s_t, a_t^{\pi_{rec}}) < \epsilon_{risk}$ .

4) **Online Training:** All of the policy and *safety critic* networks are updated online, where task policy  $\pi_{task}$  is only trained in the online RL stage while recovery policy  $\pi_{rec}$  and *safety critic* are also pre-trained with offline data.

In every single step of training, we update the replay buffer with online data where each one is denoted as a tuple  $[s_t, a_t, r_t, s_{t+1}, done]$ . We set 500 as horizon for each episode, in each transition, actions are sampled from the policies and are executed continuously until one of the termination conditions is met. The termination conditions are: violating a constraint, leaving the maze from the entrance, achieving success, and reaching the maximum step count. The networks are updated online after collecting each observation.

## IV. EVALUATION

In the following evaluation experiments, we aim to study whether the proposed framework is: 1) safer; 2) more efficient, in comparison to our baselines, which are standard RL with VIC, and RRL with constant stiffness ( $K=300$  and  $1000 N/m$ ). We label the baselines in this order as Std\_RL-VIC, SRL-K300, SRL-K1000. In Std\_RL-VIC, we use the same RL algorithm without the recovery mechanism to highlight the added benefit of our framework that combines VIC with RRL. The other baselines SRL-K300 and SRL-K1000 are used to evaluate the benefit of the VIC. Safety can be judged by the absence of excessively high interaction forces, whereas efficiency can be defined as attaining high task successes and reward with less training.

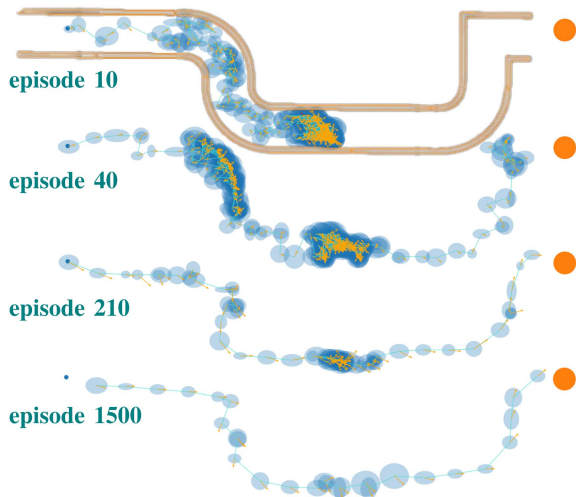


Fig. 3. Trajectories of the end-effector during different stages of training, in top-down view<sup>1</sup>.

### A. Experiment Setup

We evaluate the proposed framework on a maze exploration task using the 7-DoF Franka Emika Panda arm in Mujoco simulation environment as shown in Fig. 1. We use a peg-shaped flange mounted on the end-effector of the robot with the diameter of 30 mm and the length of 55 mm. The maze channel is 50 mm in width 70.35 cm in length. Because there is a small space for the robot to explore width-wise, we can treat it also as an extended peg-in-hole task. The goal for the agent is to reach from entrance to exit without constraint violations. We adhere to safety level 1 [24], in which the agent will receive a penalty when it violates the constraints, and the current training episode will be terminated.

This problem combines two challenges of finding the way to exit without vision, and staying safe by avoiding high contact forces. It creates a dilemma as the contacts are absolutely necessary to navigate while they are also risky. Furthermore, we add movable obstacles in the maze that requires the robot to push harder. Thus, resorting to a conservative policy cannot solve the problem. The agent needs to learn both the safe behaviour and the task achievement simultaneously.

We implement the obstacles as three heavy balls in simulation, and a pile of bolts in real-world. We position the obstacles in the middle of the maze.

We choose MuJoCo simulator (version 2.3.3) for its favorable multi-contact physics computation, suitable for contact-rich tasks. However, this required us to split the maze mesh into smaller convex components because non-convex objects are not allowed in MuJoCo. We used V-HACD<sup>2</sup> library to decompose our maze into 128 convex primitive shapes and assembled them into the same maze.

<sup>1</sup>The orange circle denotes goal point of the task, the cyan circles and yellow arrows along the trajectory represent the stiffness ellipsoids and the motion directions respectively.

<sup>2</sup>[Online]. Available: <https://github.com/kmammou/v-hacd>

The safe recovery RL part of our work is implemented based on the RRL codebase<sup>3</sup>. We follow the same hyperparameter procedure, except for: 10 training updates per simulation step, 0.65 as  $\gamma_{risk}$ , 0.7 as  $\epsilon_{risk}$  and 500 as the maximum number of actions for each episode. Another difference from RRL is that we normalize the actions of the RL model and denormalize them before applying them in the robot controller, for faster convergence.

### B. Simulation Experiments

We first collect offline data to pre-train the safety critic and the recovery policy. Then, we train all approaches online and compare their performances.

1) *Offline Data Collection*: In order to teach the notion of risk, we generate random states and actions that are likely to cause constraint violation. The robot moves to one of the six predefined points inside the maze, then moves in a random direction, which is likely to collide with the maze walls. The number of transitions are split equally between each point. In order to make the collected data more diverse, we add noise to the coordinates of each point, so that we have many different starting positions. Then, we sample a *move direction* as an angle between  $0^\circ$  and  $360^\circ$  and a *move size* in the action range. Lastly, the robot is commanded to move in that direction and the transition is stored as a tuple  $(s_t, a_t, c_t, s_{t+1}, done)$ . Offline data for pretraining are given 40018 transitions of data, 185 of these transitions contain constraint-violating states.

2) *Online Training*: We train each setup online for 3 runs with 1500 episodes in each run. In each episode, the robot needs to move to the start point and then begin exploring the maze, until it meets termination conditions. We train the policy using 1 chunk of 20 cores and one GPU (NVIDIA Tesla V100 16 Gb) from the cluster Franklin High Performance Computing (HPC)<sup>4</sup>. The average training time for 3 runs of each experiment SRL-VIC, Std\_RL-VIC, SRL-K1000, SRL-K300 is 14.47, 7.32, 9.08, 68.28 hours respectively. The time gets shorter when episodes terminate early by constraint violation or task success.

Fig. 3 shows the evolution of the robot's behaviour with increasing episodes of online learning. In the beginning, the agent fails because it violates the constraints while interacting with the obstacles. The agent learns to complete the task in later episodes. The stiffness ellipsoids indicate that the agent learns to adapt the stiffness to different conditions such as obstacle or wall contact. We observe that the trajectory becomes smoother and the robot finishes the task faster with further training.

3) *Results*: We compare the proposed framework with baselines in four different perspectives: ratio of successes/violations, reward, cumulative task successes, cumulative violations. Specifically, cumulative task successes indicate the effectiveness of task completion; while cumulative violations show the number of constraint violations that indicates the safety; ratio of successes/violations show the trade-off between task completion and safety guarantee. The results of the simulation training are

<sup>3</sup>[Online]. Available: <https://github.com/abalakrishna123/recovery-rl>

<sup>4</sup>We gratefully acknowledge the Data Science and Computation Facility and its Support Team at Fondazione Istituto Italiano di Tecnologia.

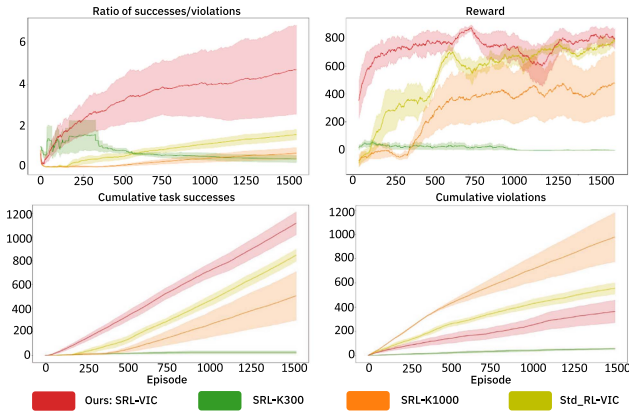


Fig. 4. Learning curves for maze exploration. Our framework (SRL-VIC) outperforms others, particularly in terms of ratio of successes/violation and cumulative violations, i.e., our framework is safer than others. Furthermore, it learns faster, achieving success at an earlier stage. The large deviation area in the ratio of successes/violations comes from the division operation; the cumulative successes and violations do not exhibit large variation.

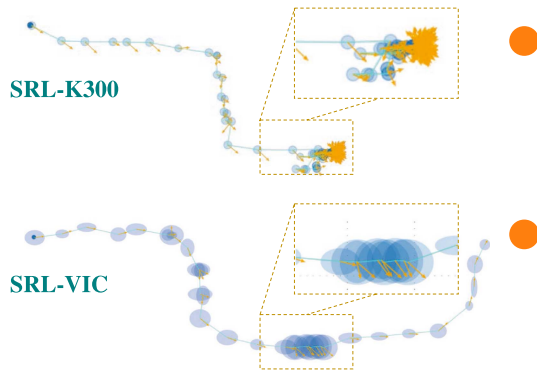


Fig. 5. Different behaviours when encountering obstacles at training episode 1000. Top (SRL-K300): The robot cannot move forward and fails due to low stiffness. Bottom (SRL-VIC): VIC approach switches to high stiffness when there are some obstacles on the way. The method learns to safely push the obstacles without exceeding the force threshold.

presented in Fig. 4. In terms of safety, our method achieves minimal cumulative violations apart from SRL-K300. Although SRL-K300 has the minimum cumulative violations among all setups, it gets stuck in the obstacle area and cannot move forward. Thus, our method has an overwhelming advantage in the ratio of successes/violations (Fig. 4). We also tested the final learned model of SRL-VIC 100 times in simulation and observed that it was successful in all cases without any violation.

Our method achieves the best results also in means of task successes (Fig. 4). Although Std\_RL-VIC performed fairly well in means of reward and cumulative task successes, our framework improves it further. The difference is even stronger in the success/violation ratio results. Meanwhile, our method significantly outperforms the other *recovery*-based methods (SRL-K300 and SRL-K1000). SRL-K300 cannot obtain high performance because it is not stiff enough to deal with obstacles. Fig. 5 shows the advantage of our method over SRL-K300 when encountering obstacles. Please note the increased size of the stiffness ellipsoids when the robot makes contact with the obstacles.

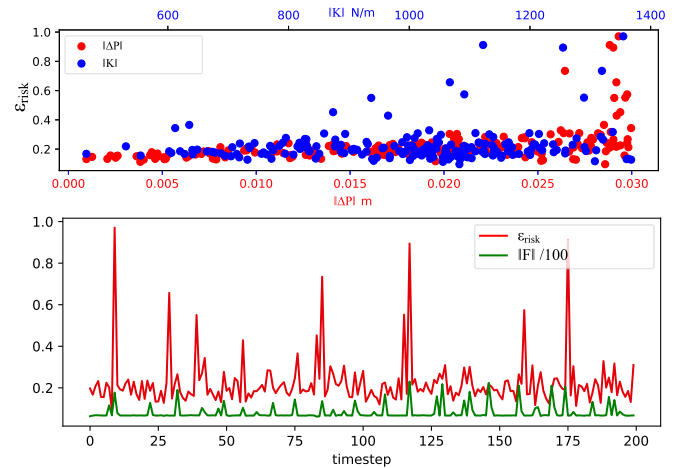


Fig. 6. Relation between the risk value and state/action values in the offline test. Top: The relationship of the risk  $\epsilon_{risk}$  to stiffness  $\|K\|$  and desired move  $\|\Delta P\|$  values. Bottom: Plot of  $\epsilon_{risk}$  and contact force  $\|F\|$ . The force magnitude is scaled down by  $1/100$ .

Regarding speed, SRL-VIC has to take relatively conservative actions in unsafe situations. Therefore, it achieved the task in slightly longer times compared to SRL-K1000 and Std\_RL-VIC, but it was still faster than K300.

The results strongly demonstrate that our method provides a good trade-off between safety and task performance by combining the benefits of VIC and SRL. The fact that Std\_RL-VIC is the second-best method, performing better than the other baselines of SRL with constant stiffness values, confirms the importance of VIC in dealing with contact-rich tasks. The VIC allows switching between the low-stiffness behaviour and high-stiffness behaviour under different conditions to use the advantage of both. We also observe that it stays stiffer along the motion direction while it stays more compliant laterally. However, the proposed method improves the performance even further as the safety critic provides a predictive mechanism to avoid dangerous situations before getting too close. It decreases the violations, thus allows the task policy to explore more states without failing. Additionally, the recovery policy takes the robot away from the risky state, making it harder to get stuck.

4) *Safety Critic Behaviour*: In order to understand the behaviour of our *safety critic* network, we visualise the relationship between the predicted risk value and the state/action values. The robot was commanded to repetitively apply a random action 200 times; starting on a random point (with 0.5 cm deviation), and moving in random direction (in  $[0^\circ, 360^\circ]$ ) with random move size and stiffness. We recorded the risk value produced for each of these cases and plotted them in Fig. 6.

As seen in the figure, higher  $\Delta P$  and  $K$  values are correlated with higher  $\epsilon_{risk}$ . This matches the human intuition that stronger actions are riskier. We also see that the model is sensitive to the contact force, as the spikes in  $F$  are aligned with the spikes in  $\epsilon_{risk}$ . However, it does not merely follow the force, as there is a more complex relationship between the state/action values and risk. These outcomes confirm that the system acts reasonably.

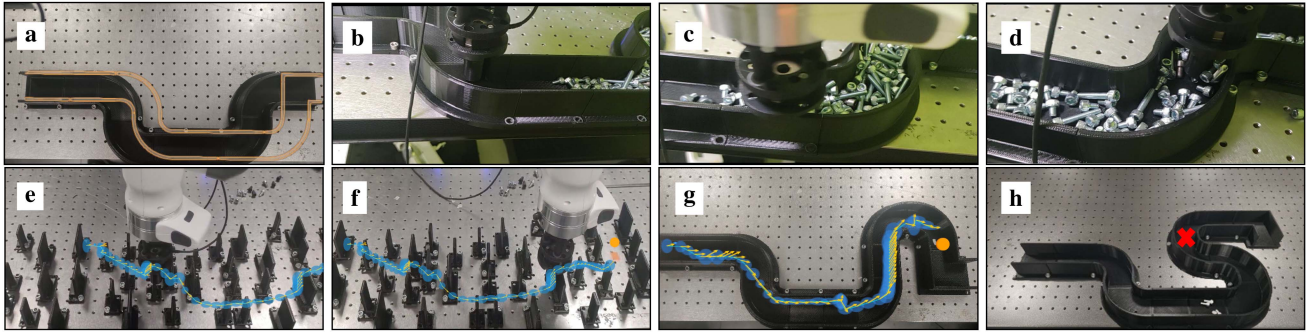


Fig. 7. (a) *Maze shape-1*, without obstacles. The yellow overlay shows the original maze from the simulation. (b) The robot meets the pile of bolts halfway. (c) The obstacles create congestion at the curve. (d) The SRL-VIC policy overcomes the obstacles by increasing the stiffness. (e, f) The *complex task scenario* with small wall segments at new positions. (g) *Maze shape-2*. (h) *Maze shape-3*, failed at red X marker.

### C. Real-World Experiments

We trained the policy in Mujoco for the contact-rich maze-exploration task and deployed it on a physical 7-DOF Franka robot arm without any fine-tuning. In the setup, we use Robot Operating System (ROS) where RL policy is a separate ROS node communicating with the physical robot.

Our policy succeeded in 4 out of 5 trials at first. In the failed case, the robot got stuck at a turning point. It was not highly robust to the real-world environment because our policy was trained with the observations in simulation. This is because of the physical dynamics differences and the imprecision of the F/T sensing in simulation. For this reason, we retrained the policy by adding Ornstein-Uhlenbeck (OU) noise in state observations. OU noise is a type of correlated noise often used in RL for continuous control tasks which helps the agent explore more systematically than purely random noise. The OU process is defined by a differential equation and has a mean-reverting property, which can be useful for stabilizing exploration. The success rate increased to 6/6 after retraining. Fig. 7 shows the deployment on a physical robot. We put bolts of different size as dynamic obstacles (50 M6  $\times$  30 for *small obstacles*, 100 M6  $\times$  8 and 50 M6  $\times$  30 for *mixed obstacles*). A video of the experiments is available at <https://youtu.be/ksWXR3vByoQ>.

**Results:** The SRL-VIC policy transfers successfully from simulation to real world. The transfer success was increased by the addition of state randomization in the simulation training. Apart from the physical dynamics and sensing differences between the simulation and the real world, there were also task-level differences, namely, the shape of the maze and the character of the obstacles.

In the real-world experiments, we tested the generalization capability of our safety model by changing the position of the maze curves and placing wall segments at new positions as shown in Fig. 7. The maze length varied between simulation (70.35 cm) and the real-world (84.62 cm). Additionally, the nature of the obstacles differed from those present in the simulation. The learned policy successfully completed most of these cases, suggesting that the safety mechanism generalizes to different wall positions. The agent can also complete the *Maze shape-1* when tested by changing the flange diameter (1, 2, 3 and 4 cm). Our task policy is trained on a single maze shape;

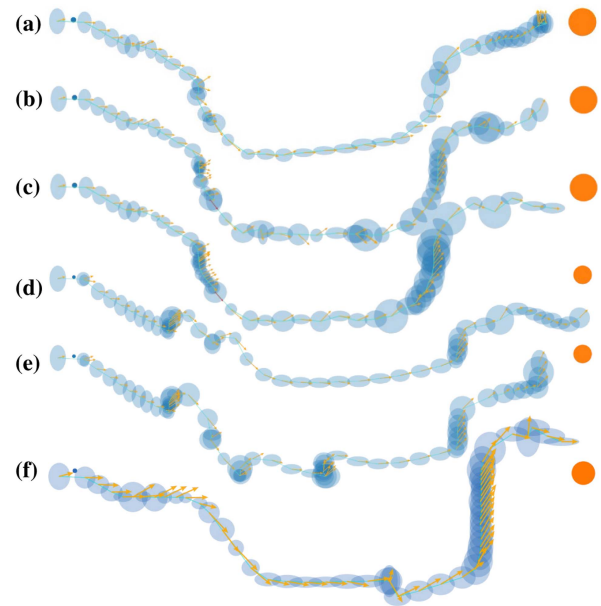


Fig. 8. Trajectories of the end-effector during executing task in the real world. (a-c) *Maze shape-1* (Fig. 7(a)): (a) *no obstacle*, (b) *small obstacles*, (c) *mixed obstacles*. (d, e) *Complex task scenario* with discrete walls, (f) *Maze shape-2* (Fig. 7(g)).

thus, it could not generalize to the case shown in Fig. 7(h). It still did not violate a constraint, but it could not find the exit. Dealing with more diverse shapes of mazes requires training with more diverse shapes as well, however, we do not focus on task generalization in this work.

The end-effector trajectories in real-robot experiments are shown in Fig. 8. The policy adapts the stiffness values effectively by becoming stiff in the goal direction and staying compliant laterally. It also employs distinct strategies for encounters with walls at curves and collisions with movable obstacles. For the latter, it adopts a larger, rounded stiffness ellipse to facilitate pushing through in all directions.

We also plot the measured force ( $F_y$ ), stiffness  $K_y$  and move size ( $\Delta P_x$ ) values w.r.t.  $y$ -position in Fig. 9, as  $y$  is the main axis towards the goal. The stiffness adaptation policy becomes more evident in these plots, comparing the *no obstacle* case to the others. The 2<sup>nd</sup> and 3<sup>rd</sup> cases have larger stiffness in the

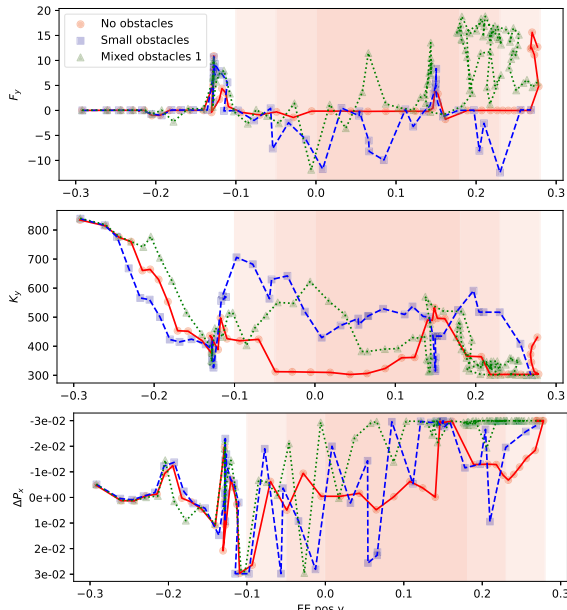


Fig. 9. Force ( $F_y$ ), move size ( $\Delta P_x$ ), and stiffness ( $K_y$ ) values during the real-robot experiments plotted against the  $y$ -position. The pink highlight shows the approximate placement of obstacles.

obstacle area as they observe higher force, while the 1st case increases stiffness only at the curves, while touching the wall. Another difference is seen in the  $\Delta P_x$  action, as the 2nd and 3rd cases move laterally to open the way through the obstacles, while the 1st maintains its  $x$ -position.

## V. CONCLUSION AND FUTURE WORK

We presented a framework that integrates the strengths of a safe RL method with VIC. This combination addresses the dual challenges of ensuring safety and maintaining continuous physical contact. Experiments in contact-rich maze tasks demonstrate that our framework achieved a good trade-off between task accomplishment and collision-based safety, and outperformed `Std_RL-VIC`. The inclusion of the VIC adds increased adaptability, creating a better trade-off between safety and success in comparison to the constant impedance approaches. The trained model can transfer to real-world, and also the safety model can generalize to different obstacles and wall positions. In the future, we will evaluate our framework in more generalized task scenarios and introduce model-based RL into our framework to improve the safety and robustness of our proposed framework in unknown environments.

## REFERENCES

- [1] A. Ajoudani, A. M. Zanchettin, S. Ivaldi, A. Albu-Schäffer, K. Kosuge, and O. Khatib, "Progress and prospects of the human-robot collaboration," *Auton. Robots*, vol. 42, no. 5, pp. 957–975, 2018.
- [2] F. J. Abu-Dakka and M. Saveriano, "Variable impedance control and learning—A review," *Front. Robot. AI*, vol. 7, 2020, Art. no. 590681.
- [3] Y. Kato, P. Balatti, J. M. Gandarias, M. Leonori, T. Tsuji, and A. Ajoudani, "A self-tuning impedance-based interaction planner for robotic haptic exploration," *IEEE Robot. Automat. Lett.*, vol. 7, no. 4, pp. 9461–9468, Oct. 2022.
- [4] C. Chang, K. Haninger, Y. Shi, C. Yuan, Z. Chen, and J. Zhang, "Impedance adaptation by reinforcement learning with contact dynamic movement primitives," in *Proc. IEEE/ASME Int. Conf. Adv. Intell. Mechatron.*, 2022, pp. 1185–1191.
- [5] X. Zhang, L. Sun, Z. Kuang, and M. Tomizuka, "Learning variable impedance control via inverse reinforcement learning for force-related tasks," *IEEE Robot. Automat. Lett.*, vol. 6, no. 2, pp. 2225–2232, Apr. 2021.
- [6] Q. Yang, A. Dürr, E. A. Topp, J. A. Stork, and T. Stoyanov, "Variable impedance skill learning for contact-rich manipulation," *IEEE Robot. Automat. Lett.*, vol. 7, no. 3, pp. 8391–8398, Jul. 2022.
- [7] R. Martin-Martin, M. A. Lee, R. Gardner, S. Savarese, J. Bohg, and A. Garg, "Variable impedance control in end-effector space: An action space for reinforcement learning in contact-rich tasks," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2019, pp. 1010–1017.
- [8] B. Thananjeyan et al., "Recovery RL: Safe reinforcement learning with learned recovery zones," *IEEE Robot. Automat. Lett.*, vol. 6, no. 3, pp. 4915–4922, Jul. 2021.
- [9] S. Kim, J. Kwon, T. Lee, Y. Park, and J. Perez, "Safety-aware unsupervised skill discovery," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2023, pp. 894–900.
- [10] H. Qiao, B. Dalay, and R. Parkin, "Robotic peg-hole insertion operations using a six-component force sensor," *Proc. Inst. Mech. Eng. Part C: J. Mech. Eng. Sci.*, vol. 207, no. 5, pp. 289–306, 1993.
- [11] M. A. Lee et al., "Making sense of vision and touch: Learning multimodal representations for contact-rich tasks," *IEEE Trans. Robot.*, vol. 36, no. 3, pp. 582–596, Jun. 2020.
- [12] G. Schoettler, A. Nair, J. A. Ojea, S. Levine, and E. Solowjow, "Meta-reinforcement learning for robotic industrial insertion tasks," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2020, pp. 9728–9735.
- [13] C. Yang, P. Liang, Z. Li, A. Ajoudani, C.-Y. Su, and A. Bicchi, "Teaching by demonstration on dual-arm robot using variable stiffness transferring," in *Proc. IEEE Int. Conf. Robot. Biomimetics*, 2015, pp. 1202–1208.
- [14] J. Lee et al., "Upper-body impedance control with variable stiffness for a door opening task," in *Proc. IEEE-RAS Int. Conf. Humanoid Robots*, 2014, pp. 713–719.
- [15] D. Mitrovic, S. Klanke, and S. Vijayakumar, "Learning impedance control of antagonistic systems based on stochastic optimization principles," *Int. J. Robot. Res.*, vol. 30, no. 5, pp. 556–573, 2011.
- [16] Y. Li, G. Ganesh, N. Jarrassé, S. Haddadin, A. Albu-Schäffer, and E. Burdet, "Force, impedance, and trajectory learning for contact tooling and haptic identification," *IEEE Trans. Robot.*, vol. 34, no. 5, pp. 1170–1182, Oct. 2018.
- [17] C. Yang, C. Zeng, C. Fang, W. He, and Z. Li, "A DMPs-based framework for robot learning and generalization of humanlike variable impedance skills," *IEEE/ASME Trans. Mechatron.*, vol. 23, no. 3, pp. 1193–1203, Jun. 2018.
- [18] J. Yamada et al., "Motion planner augmented reinforcement learning for robot manipulation in obstructed environments," in *Proc. Conf. Robot Learn.*, 2021, pp. 589–603.
- [19] H. Bharadhwaj, A. Kumar, N. Rhinehart, S. Levine, F. Shkurti, and A. Garg, "Conservative safety critics for exploration," in *Proc. Int. Conf. Learn. Representations*, 2021. [Online]. Available: <https://openreview.net/forum?id=iaO86DUuKi>
- [20] E. Altman, *Constrained Markov Decision Processes*. Evanston, IL, USA: Routledge, 1995.
- [21] K. Srinivasan, B. Eysenbach, S. Ha, J. Tan, and C. Finn, "Learning to be safe: Deep RL with a safety critic," 2020, *arXiv:2010.14603*.
- [22] T. P. Lillicrap et al., "Continuous control with deep reinforcement learning," 2015, *arXiv:1509.02971*.
- [23] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine, "Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor," in *Proc. Int. Conf. Mach. Learn.*, 2018, pp. 1861–1870.
- [24] L. Brunke et al., "Safe learning in robotics: From learning-based control to safe reinforcement learning," *Annu. Rev. Control Robot. Auton. Syst.*, vol. 5, pp. 411–444, 2022.