FHCPL: An Intelligent Fixed-Horizon Constrained Policy Learning System for Risk-Sensitive Industrial Scenario

Ke Lin, *Graduate Student Member, IEEE*, Duantengchuan Li, Yanjie Li, *Member, IEEE*, Shiyu Chen, and Xindong Wu, *Fellow, IEEE*

***Abstract*—In many industrial scenarios, safety is a crucial factor to consider. In this article, we focus on the safe reinforcement learning problem that maximizes total rewards while enabling agents to avoid risks. We propose an intelligent fixed-horizon constrained policy learning (FHCPL) system, which allows agents to obtain high returns while maintaining risk avoidance behaviors. For discrete cases, a two-stage policy iteration algorithm, named fixed-horizon constrained policy iteration, is proposed, in which the safety of the learned policy is guaranteed. In the first stage, a policy that satisfies the safety constraint is obtained. In the second stage, a final learned policy that can get high returns while satisfying the safety constraint is reached. For continuous cases, we present the fixed-horizon constrained policy optimization algorithm. Empirical results demonstrate that, with the advantage of the fixed-horizon risk, the FHCPL achieves superior performance in terms of reward maximization and risk avoidance.**

***Index Terms*—Fixed-horizon constraint, policy learning, reinforcement learning (RL), risk-sensitive industrial scenario, safe RL.**

NOMENCLATURE

Symbols	*Notations*
π, π_θ	Policy and a policy parameterized with $\boldsymbol{\theta}$.
C	Cost threshold in CMDP.
h	Safe threshold.
$\mathcal{M}$	Markov decision process.
$\boldsymbol{s}, \boldsymbol{a}$	State and an action.
$\mathcal{S}, \mathcal{A}$	State space and action space in an MDP.
$P(\cdot \mid \boldsymbol{s}, \boldsymbol{a})$	State transition probability under $\boldsymbol{s}, \boldsymbol{a}$.
$P_\pi(\boldsymbol{s})$	Probability of $\boldsymbol{s}$ under π.
r, c	Reward function and cost function.
$\gamma, \bar{\gamma}$	Discount factors for reward and cost.
N	Value of the fixed-horizon.
$V^\pi, \mathcal{V}_N^\pi$	State and risk state value function under π.
Q^π	State-action value function under π.
$\mathcal{Q}_N^\pi$	Risk state-action value function under π.
$\mathcal{V}_N^{\bar{\gamma}\pi}, \mathcal{Q}_N^{\bar{\gamma}\pi}$	Risk value function with discount factor $\bar{\gamma}$.
$\mathcal{V}_\infty^\pi, \mathcal{Q}_\infty^\pi$	Risk value function with infinite horizon.
$\mathcal{S}_s, \mathcal{S}_u$	Safe state set and unsafe state set.
$\phi(\cdot)$	Distribution about safe state set.
$\mathcal{X}$	Transient state set.
$\mathcal{X}_s, \mathcal{X}_u$	Safe & unsafe transient state set.
$A^\pi, \mathcal{A}_N^\pi$	Reward & risk advantage function under π.
$J(\pi), J_c(\pi)$	Expected total reward and total cost of π.

Manuscript received 23 February 2023; revised 11 September 2023; accepted 15 November 2023. Date of publication 21 December 2023; date of current version 4 April 2024. This work was supported in part by the National Natural Science Foundation of China under Grant 61977019 and Grant U1813206, and in part by the Shenzhen Fundamental Research Program under Grant JCYJ20220818102415033 and Grant JSGG20201103093802006. Paper no. TII-23-0624. *(Corresponding author: Yanjie Li.)*

Ke Lin, Yanjie Li, and Shiyu Chen are with the Department of Control Science and Engineering, Harbin Institute of Technology (Shenzhen), Shenzhen 518055, China (e-mail: schris_lin@stu.hit.edu.cn; autolyj@hit.edu.cn; chenshiyu@stu.hit.edu.cn).

Duantengchuan Li is with the School of Computer Science, Wuhan University, Wuhan 430072, China (e-mail: dtclee1222@gmail.com).

Xindong Wu is with the Research Center for Knowledge Engineering at the Zhejiang Lab, Hangzhou 311121, China (e-mail: xwu@zhejianglab.com).

This article has supplementary material provided by the authors and color versions of one or more figures available at https://doi.org/10.1109/TII.2023.3336225.

Digital Object Identifier 10.1109/TII.2023.3336225

I. INTRODUCTION

REINFORCEMENT learning (RL) has achieved remarkable success in many decision-making problems, such as Go [1], robotic control [2], [3], games [4], [5], recommender systems [6], [7], and resource scheduling [8], [9], in which the goal of an agent is to maximize the cumulative reward. Moreover, in the field of control theory, adaptive dynamic programming [10] is also proposed to handle such related problems. For instance, Yu et al. [11] proposed an adaptive optimal time-varying formation tracking protocol for disturbed high-order multi-agent systems. Wei et al. [12] proposed the finite approximation errors to tackle the optimal control problems with infinite horizon nonlinear systems. However, many industrial scenarios are risk-sensitive, such as the oil and gas industry, coal and metal mining industry, and aviation and aerospace industry, in which agents must avoid all environmental risks while maximizing returns.

Safe RL is usually used to deal with such risk-sensitive decision-making problems, the aim of which is to get a policy π that can satisfy some safety constraints (e.g., collaborative robots do not harm workers) while maximizing the cumulative rewards (e.g., robots successfully assemble parts). With the intelligent policy, an agent can perform a series of actions, and finally safely achieve some decision-making goals in a specific

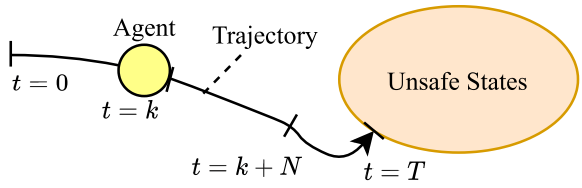

Fig. 1. In typical safe RL, the cumulative cost of the entire trajectory is considered. Since the trajectory in the figure enters the unsafe states in the last few steps, this trajectory will be considered a bad trajectory, and the probability of occurrence of such trajectories will become smaller. However, the fixed-horizon constraint only considers the first N steps from the current time step. Therefore, the proposed constraint is not influenced by unsafe behaviors in the distant future. Thus, the constraint is actually relaxed, and it can make the agent explore with less conservativeness while keeping a sense of risk avoidance. Then, higher returns become easier to achieve.

industrial scenario. A comprehensive overview of safe RL can be found in [13]. For safe RL algorithms, to make the agent avoid risks, one common approach is to incorporate constraints into the existing policy optimization algorithms, e.g., policy gradient [14], trust region policy optimization [15], and proximal policy optimization (PPO) [16]. Then, constrained optimization techniques can be applied to tackle this type of problem, such as the Karush–Kuhn–Tucker (KKT) condition [17] or Lagrangian relaxation [18], [19]. In addition, there are also Lyapunov-based [20] and model-based [21] methods to tackle the safe RL problem.

In the methods described above, the constrained Markov decision process (CMDP) [22], [23] is considered, where the constraint is formulated as the total cost, which must be less than a certain threshold C. However, limited by extra constraints in the CMDP, the agent's exploration becomes conservative, which prevents the agent from getting a satisfactory return [24]. This is because such constraints consider the cumulative cost of the entire horizon. Thus, unsafe behavior in the remote future can affect the agent's current decisions, as illustrated in Fig. 1. In addition, modifying the discount factor in the cost can alleviate this problem to a certain extent. However, the degree of the constraint is sensitive to the adjustment of the discount factor, especially when the horizon is long [25]. Nevertheless, typical RL methods without constraints can achieve high returns in risk-sensitive environments, but the agent is not equipped with risk avoidance behavior. Therefore, how to reduce an agent's conservativeness in exploration and make a tradeoff between obtaining high returns and avoiding risks becomes an urgent problem.

In this article, we address the safe RL issue from the perspective of improving constraints. We propose a fixed-horizon constraint on risk occurrence probability, which is defined as the probability of entering unsafe states within a fixed-horizon being less than a threshold h. Note that different from the finite-horizon in RL, fixed-horizon RL is a recently proposed concept, which only considers the cumulative reward over the next N steps from the current time step [26]. Our constraint is featured by greater leniency and the ability to allow agents to explore in environments more aggressively while keeping a sense of risk avoidance, as shown in Fig. 1.

The main contributions of this work are summarized as follows.

1) We propose a fixed-horizon constraint that can alleviate the problem of conservativeness in general cumulative cost constraints in typical safe RL. Thus, the agent becomes more likely to receive higher returns.
2) For discrete state space and action space problems, we give a two-stage fixed-horizon constrained policy iteration (FHCPI) algorithm. Starting from an arbitrary policy, our algorithm can eventually converge to a local optimal policy whose safety is guaranteed.
3) For a special case when the safe threshold is 0, being global optimal under the safety constraint is guaranteed by the proposed two-stage FHCPI algorithm. The excellent performance of FHCPI has also been demonstrated in a discrete grid world experiment.
4) For continuous situations, we give an approximate performance difference lemma for our constraint and propose the fixed-horizon constrained policy optimization (FHCPO) algorithm. Extensive experiments show that our algorithm achieves superior performance to the existing typical safe RL methods in terms of total reward and collision rate (unsafe state ratio).

II. Background and Related Work

A. Constrained Markov Decision Process

Safe RL problems are usually modeled as a CMDP [23]. A CMDP can be represented by a tuple $\mathcal{M}=(\mathcal{S},\mathcal{A},P,r,c,\gamma)$, where $\mathcal{S}$ and $\mathcal{A}$ denote the state and action space, respectively. $P(\boldsymbol{s}'|\boldsymbol{s},\boldsymbol{a})$ represents the probability of the next state $\boldsymbol{s}'\in\mathcal{S}$ given the current state $\boldsymbol{s}\in\mathcal{S}$ and action $\boldsymbol{a}\in\mathcal{A}$. In addition, $r:\mathcal{S}\times\mathcal{A}\times\mathcal{S}\to\mathbb{R}$ and $c:\mathcal{S}\times\mathcal{A}\times\mathcal{S}\to\mathbb{R}$ are the reward and the cost given by the environment on each transition, respectively. $\gamma\in[0,1)$ is a discount factor. A standard CMDP aims to find a policy π that can maximize the expected sum of rewards while satisfying constraints to make agents have risk-sensitive behavior. A CMDP problem is formally defined as

$$\begin{aligned}\max_{\pi}\quad & \mathbb{E}_{\pi}\left[\sum_{t=0}^{\infty}\gamma^t r(\boldsymbol{s}_t,\boldsymbol{a}_t,\boldsymbol{s}_{t+1})\right]\\ \text{s.t.}\quad & \mathbb{E}_{\pi}\left[\sum_{t=0}^{\infty}\gamma^t c(\boldsymbol{s}_t,\boldsymbol{a}_t,\boldsymbol{s}_{t+1})\right]\le C\end{aligned}\tag{1}$$

where C is the cost threshold. Moreover, the state value function $V^{\pi}(\boldsymbol{s})$ and the state-action value function $Q^{\pi}(\boldsymbol{s},\boldsymbol{a})$ are introduced to evaluate the performance of the policy. Specifically, they are defined as the discounted total reward under the policy π given the initial state and action. From the definition, the following equations hold [27]:

$$V^{\pi}(\boldsymbol{s})=\sum_{\boldsymbol{a}}\pi(\boldsymbol{a}|\boldsymbol{s})\sum_{\boldsymbol{s}'}P(\boldsymbol{s}'|\boldsymbol{s},\boldsymbol{a})\left[r+\gamma V^{\pi}(\boldsymbol{s}')\right]\tag{2}$$

$$Q^{\pi}(\boldsymbol{s},\boldsymbol{a})=\sum_{\boldsymbol{s}'}P(\boldsymbol{s}'|\boldsymbol{s},\boldsymbol{a})\left[r+\gamma V^{\pi}(\boldsymbol{s}')\right].\tag{3}$$

B. Policy Optimization With Constraint

For deep RL with constraints, a series of algorithms are proposed by employing constrained optimization methods. Primal-dual policy optimization (PDO) [28] employs the Lagrangian relaxation technique to derive the hybrid gradient with respect to the objective function and the cost to optimize the policy. Constrained policy optimization (CPO) [17] approximates the original problem by Taylor expansion and utilizes the trust region optimization method to ensure that the next new policy satisfies constraints. Interior-point policy optimization (IPO) [29] adopts a logarithmic barrier function as a penalty to constrain the update of the policy. In projection-based constrained policy optimization (PCPO) [30], a projection step is performed to project π_k onto the feasible domain to get the next policy π_{k+1}, which meets the constraints. Other policy optimization methods for safe RL include [18], [31]. However, similar to the CMDP, the constraint form in these methods is discounted cumulative cost of the entire horizon, shown in (1), which may make the agent conservative in exploration, as illustrated in Fig. 1. In this article, we propose a fixed-horizon constraint to reduce the agent's conservativeness while making the agent satisfy safety constraints, which can be used for both discrete and continuous cases.

III. Problem Statement and Definition

A. Risk Definition and Evaluation

In this article, we denote sets associated with the state as: $\mathcal{X}$ indicates a transient state set and $\mathcal{X}_s$ denotes a safe transient state set. $\mathcal{S}_s$ and $\mathcal{S}_u$ indicate a safe and an unsafe state set, respectively.

Definition 1: The risk state value function $\mathcal{V}_N^\pi(\boldsymbol{s})$ is defined as the probability of entering unsafe states when starting from $\boldsymbol{s}$ and following π within N transitions. In addition, $\mathcal{Q}_N^\pi(\boldsymbol{s},\boldsymbol{a})$ is defined as the probability of entering unsafe states when starting from $\boldsymbol{s}$, taking action $\boldsymbol{a}$, and following π within N transitions. They can be written as

$$\begin{aligned}\mathcal{V}_N^\pi(\boldsymbol{s}) &= P_\pi(\boldsymbol{s}_{t+1} \text{ or } \cdots \text{ or } \boldsymbol{s}_{t+N} \in \mathcal{S}_u \mid \boldsymbol{s}_t = \boldsymbol{s}) \\ \mathcal{Q}_N^\pi(\boldsymbol{s},\boldsymbol{a}) &= P_\pi\left(\boldsymbol{s}_{t+1} \text{ or } \cdots \text{ or } \boldsymbol{s}_{t+N} \in \mathcal{S}_u \middle| \begin{matrix}\boldsymbol{s}_t = \boldsymbol{s} \\ \boldsymbol{a}_t = \boldsymbol{a}\end{matrix}\right).\end{aligned} \tag{4}$$

By definition of $\mathcal{V}_N^\pi(\boldsymbol{s})$ and $\mathcal{Q}_N^\pi(\boldsymbol{s},\boldsymbol{a})$, if we set $c(\boldsymbol{s},\boldsymbol{a},\boldsymbol{s}') = 1, \boldsymbol{s}' \in \mathcal{S}_u$, $c(\boldsymbol{s},\boldsymbol{a},\boldsymbol{s}') = 0, \boldsymbol{s}' \notin \mathcal{S}_u$, then we have $\mathcal{V}_N^\pi(\boldsymbol{s}) = \mathbb{E}_\pi[\sum_{k=0}^{\min(N-1,\tau_\pi(\boldsymbol{s}_t))} c(\boldsymbol{s}_{t+k},\boldsymbol{a}_{t+k})|\boldsymbol{s}_t = \boldsymbol{s}]$, where $\tau_\pi(\boldsymbol{s}_t)$ is the first time step (starting from $\boldsymbol{s}_t$) that the agent meets an unsafe state under the policy π, and we can get the following recursive formulas:

$$\mathcal{V}_N^\pi(\boldsymbol{s}) = \mathbb{E}_\pi[c(\boldsymbol{s},\boldsymbol{a},\boldsymbol{s}') + \mathcal{V}_{N-1}^\pi(\boldsymbol{s}')] \quad \forall \boldsymbol{s} \in \mathcal{S}_s \tag{5}$$

$$\mathcal{Q}_N^\pi(\boldsymbol{s},\boldsymbol{a}) = \mathbb{E}_{\boldsymbol{s}'\sim P(\cdot|\boldsymbol{s},\boldsymbol{a})}[c(\boldsymbol{s},\boldsymbol{a},\boldsymbol{s}') + \mathcal{V}_{N-1}^\pi(\boldsymbol{s}')] \quad \forall \boldsymbol{s} \in \mathcal{S}_s. \tag{6}$$

By letting $\mathcal{V}_0^\pi(\boldsymbol{s}) \equiv \mathcal{Q}_0^\pi(\boldsymbol{s},\boldsymbol{a}) \equiv 0$, $\forall \boldsymbol{s} \in \mathcal{S}$, $\forall \boldsymbol{a} \in \mathcal{A}$ and initializing $\mathcal{V}_k^\pi(\boldsymbol{s}) = 0$, $\forall \boldsymbol{s} \in \mathcal{S}, k = 1,\ldots,N$, we can apply (5) iteratively to evaluate the proposed risk for *safe states* to get $\mathcal{V}_k^\pi(\boldsymbol{s}), \boldsymbol{s} \in \mathcal{S}_s$. Then, for *unsafe states*, we set $\mathcal{V}_k^\pi(\boldsymbol{s}) \equiv 1$ after iteration. Moreover, we define the discounted version of $\mathcal{V}_N^\pi(\boldsymbol{s})$ and $\mathcal{Q}_N^\pi(\boldsymbol{s},\boldsymbol{a})$, which can be written as, $\mathcal{V}_N^{\bar{\gamma}\pi}(\boldsymbol{s}) = \mathbb{E}_\pi[c + \bar{\gamma}\mathcal{V}_{N-1}^{\bar{\gamma}\pi}(\boldsymbol{s}')|\boldsymbol{s}]$, $\mathcal{Q}_N^{\bar{\gamma}\pi}(\boldsymbol{s},\boldsymbol{a}) = \mathbb{E}_{\boldsymbol{s}'\sim P(\cdot|\boldsymbol{s},\boldsymbol{a})}[c + \bar{\gamma}\mathcal{V}_{N-1}^{\bar{\gamma}\pi}(\boldsymbol{s}')|\boldsymbol{s},\boldsymbol{a}]$, where $\bar{\gamma} \in [0,1)$. In fact, when we evaluate $\mathcal{V}_N^\pi(\boldsymbol{s})$ by sampling, we can just regard unsafe states as terminal states, then we can rewrite the risk state value function as

$$\mathcal{V}_N^\pi(\boldsymbol{s}) = \mathbb{E}_\pi\left[\sum_{k=0}^{N-1} c(\boldsymbol{s}_{t+k},\boldsymbol{a}_{t+k},\boldsymbol{s}_{t+k+1})\middle|\boldsymbol{s}_t = \boldsymbol{s}\right]. \tag{7}$$

Remark 1: Intuitively, the risk of a policy π is defined as the probability that the policy π will enter unsafe states within the next N time steps in the future. A higher probability value means a greater risk. Moreover, unsafe states should be predefined in the unsafe state set $\mathcal{S}_u$. For example, in mobile robot scenarios, unsafe states can be defined as collisions between robots and obstacles. In electrical grid scenarios, unsafe states can be defined as excessive current on electric wires or the power transmitted on wires that cannot meet the demand of consumers.

B. Objective Function

To maximize the cumulative reward and make the agent avoid risks, we consider the following CMDP problem:

$$\min_\pi -\mathbb{E}_\pi\left[\sum_{t=0}^{\infty}\gamma^t r(\boldsymbol{s}_t,\boldsymbol{a}_t)\right], \text{s.t.}\, \mathbb{E}_{\boldsymbol{s}\sim\phi(\cdot)}\left[\mathcal{V}_N^\pi(\boldsymbol{s})\right] \le h \tag{8}$$

where $h \in [0,1)$ is a fixed threshold, and ϕ is a distribution w.r.t. safe state set $\mathcal{S}_s$. This constraint means that the probability of entering unsafe states within N steps is less than h. In those methods that build problems based on the CMDP, such as PDO, CPO, IPO, and PCPO, the constraint is defined as $\mathbb{E}[\sum_t^\infty \gamma^t c(\boldsymbol{s}_t,\boldsymbol{a}_t)] \le C$, which considers infinite horizon of risks. This way of defining constraints may make the agent conservative in the training process, as illustrated in Fig. 1. However, the proposed risk only considers the fixed-horizon from the current step, which makes the agent more aggressive in exploration. Therefore, N is a parameter that allows the agent to trade off between risks and rewards. If the value of N becomes larger, the agent tends to be more conservative and cares more about safety. On the contrary, it is more aggressive and more inclined to get higher returns.

IV. Proposed FHCPL System

The overall illustration of the proposed FHCPL system for risk-sensitive industrial scenarios is shown in Fig. 2. Note that many risk-sensitive decision-making problems in industrial scenarios can be transformed into constrained policy learning problems. As a consequence, we proposed the FHCPI to tackle the problem with discrete state and action space, and we also proposed the FHCPO to deal with problems with continuous situations. Note that problems with continuous situations can be solved using iteration-based methods by discretizing states and actions. Also, problems with discrete situations can be solved using gradient-based approaches by interpolating discrete states and actions. However, such transformations will result in a

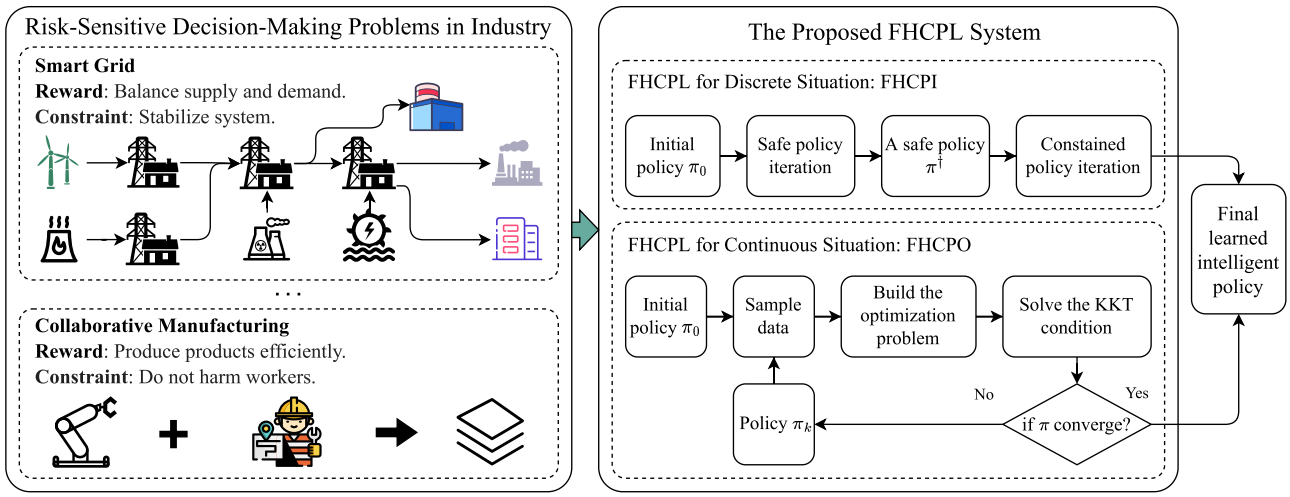

Fig. 2. Overall illustration of the proposed FHCPL system for risk-sensitive industrial scenarios.

Algorithm 1: Safe Policy Iteration (PI).

Input: A risk horizon N. A cost discount factor $\bar{\gamma}$. A threshold h. A set $\mathcal{K} = \{0, \ldots, N-1\}$. $\pi(\boldsymbol{s})$, $\mathcal{V}_\infty^{\bar{\gamma}\pi}(\boldsymbol{s})$, $\mathcal{V}_{N-t}^{\pi}(\boldsymbol{s}), t \in \mathcal{K}, \forall \boldsymbol{s} \in \mathcal{S}$.

Procedure:

repeat
  Set $\mathcal{V}_\infty^{\bar{\gamma}\pi}(\boldsymbol{s}) = 0$, $\mathcal{V}_{N-t}^{\pi}(\boldsymbol{s}) = 0, t \in \mathcal{K}, \forall \boldsymbol{s} \in \mathcal{S}$.
  repeat {Policy evaluation}
    for $\boldsymbol{s}$ in $\mathcal{X}_s$ **do**
      $\mathcal{V}_\infty^{\bar{\gamma}\pi}(\boldsymbol{s}) \leftarrow \sum_{\boldsymbol{a}} \pi(\boldsymbol{a}|\boldsymbol{s}) \sum_{\boldsymbol{s}'} P(\boldsymbol{s}'|\boldsymbol{s}, \boldsymbol{a}) [c + \bar{\gamma} \mathcal{V}_\infty^{\bar{\gamma}\pi}(\boldsymbol{s}')]$.
      $\mathcal{V}_{N-t}^{\pi}(\boldsymbol{s}) \leftarrow \mathbb{E}_\pi[c + \mathcal{V}_{N-t-1}^{\pi}(\boldsymbol{s}')], t \in \mathcal{K}$.
    end for
  until $\mathcal{V}_\infty^{\bar{\gamma}\pi}(\boldsymbol{s})$ and $\mathcal{V}_N^{\pi}(\boldsymbol{s}), \forall \boldsymbol{s} \in \mathcal{X}_s$ converge.
  Set $\mathcal{V}_\infty^{\bar{\gamma}\pi}(\boldsymbol{s}) = 1, \boldsymbol{s} \in \mathcal{S}_u$. Get $\mathcal{Q}_\infty^{\bar{\gamma}\pi}(\boldsymbol{s}, \boldsymbol{a})$ by $\mathcal{V}_\infty^{\bar{\gamma}\pi}(\boldsymbol{s})$.
  for $\boldsymbol{s}$ in $\mathcal{X}$ **do** {Policy improvement}
    $\pi(\boldsymbol{s}) \leftarrow \arg\min_{\tilde{\boldsymbol{a}}} \mathcal{Q}_\infty^{\bar{\gamma}\pi}(\boldsymbol{s}, \tilde{\boldsymbol{a}})$.
  end for
until $\mathcal{V}_N^{\pi}(\boldsymbol{s}) \leq h, \forall \boldsymbol{s} \in \mathcal{S}_s$.

Output: π.

loss of optimality. Thus, it is crucial to propose two different algorithms for discrete and continuous cases, respectively.

A. FHCPL for Discrete Situation

Assumption 1: We assume, $\forall \pi$, the agent will eventually transfer to the terminal states in T steps.

Assumption 2 (Exist a safe policy): There exists a policy π that satisfies, $\mathcal{V}_\infty^{\pi}(\boldsymbol{s}) < h$, $\forall \boldsymbol{s} \in \mathcal{S}_s$, where we denote $\mathcal{V}_\infty^{\pi}(\boldsymbol{s}) = \lim_{N \to \infty} \mathcal{V}_N^{\pi}(\boldsymbol{s})$.

For the problem presented in (8) with discrete situations, we present a two-stage FHCPI method. In the first stage, named *safe PI*, given an initial policy, we aim to obtain a safe policy $\pi^\dagger$ that satisfies the constraint. In the second stage, named *constrained PI*, given an initial safe policy $\pi^\dagger$, we aim to obtain a high-return policy π^*, which still meets the constraint.

1) Safe PI: In this stage, we aim to find a policy $\pi^\dagger$ that can satisfy the safety constraint, i.e., $\mathcal{V}_N^{\pi^\dagger}(\boldsymbol{s}) \leq h$, $\forall \boldsymbol{s} \in \mathcal{S}_s$. To theoretically guarantee that such a policy can be obtained by a specific form of policy improvement, we introduce the discounted version of risk state value function and make the risk horizon infinite, i.e., $\mathcal{V}_\infty^{\bar{\gamma}\pi}$ and $\mathcal{Q}_\infty^{\bar{\gamma}\pi}$. Then, we update the policy by

$$\pi'(\boldsymbol{s}) = \arg\min_{\tilde{\boldsymbol{a}}} \mathcal{Q}_\infty^{\bar{\gamma}\pi}(\boldsymbol{s}, \tilde{\boldsymbol{a}}) \quad \forall \boldsymbol{s} \in \mathcal{X}. \tag{9}$$

Lemma 1: Given an arbitrary initial policy, iterative application of (9) for policy update will eventually result in a policy π_s^* that satisfies $\mathcal{V}_\infty^{\bar{\gamma}\pi_s^*}(\boldsymbol{s}) \leq \mathcal{V}_\infty^{\bar{\gamma}\pi}(\boldsymbol{s})$, $\forall \boldsymbol{s} \in \mathcal{S}$, $\forall \pi$, $\forall \bar{\gamma} \in [0, 1)$.

Proof: See Appendix A of the Supplementary Material. ■

Proposition 1: For $\pi \in \{\pi \mid \mathcal{V}_\infty^{\bar{\gamma}\pi}(\boldsymbol{s}) < h, \forall \boldsymbol{s} \in \mathcal{S}_s, \forall \bar{\gamma} \in [0, 1)\}$, define $\epsilon = \inf_{\boldsymbol{s}, \bar{\gamma}} \{h - \mathcal{V}_\infty^{\bar{\gamma}\pi}(\boldsymbol{s})\}$ where $\epsilon \in (0, 1]$. When $\bar{\gamma} \geq (1 - \epsilon)^{1/(T-1)}$, then we have $\mathcal{V}_\infty^{\pi}(\boldsymbol{s}) \leq h$.

Proof: See Appendix B of the Supplementary Material. ■

Lemma 2: Given an initial policy π_s^0 and $\bar{\gamma} \geq (1 - \epsilon)^{1/(T-1)}$, in the obtained policy sequence $\{\pi_s^0, \pi_s^1, \ldots, \pi_s^*\}$ by applying (9) iteratively, exist a policy $\pi^\dagger$ that satisfies $\mathcal{V}_N^{\pi^\dagger}(\boldsymbol{s}) \leq h$, $\forall \boldsymbol{s} \in \mathcal{S}_s$.

Proof: See Appendix C of the Supplementary Material. ■

According to Lemma 2, as long as we make the value of $\bar{\gamma}$ close enough to 1, we can apply (9) iteratively to find a safe policy $\pi^\dagger$ that can satisfy the constraint $\mathcal{V}_N^{\pi^\dagger}(\boldsymbol{s}) \leq h, \forall \boldsymbol{s} \in \mathcal{S}_s$. We summarize the process as Algorithm 1.

2) Constrained PI: In the second stage of PI, we set $\pi^\dagger$ as the initial safe policy and aim to find a high-return policy that also meets the safety constraint. Thus, we focus on solving the following problem:

$$\max_{\pi} \mathbb{E}_\pi \left[\sum_{t=0}^{\infty} \gamma^t r(\boldsymbol{s}_t, \boldsymbol{a}_t) \right], \text{s.t.}\ \mathcal{V}_N^{\pi}(\boldsymbol{s}) \leq \mathcal{V}_N^{\pi^\dagger}(\boldsymbol{s}) \quad \forall \boldsymbol{s} \in \mathcal{S}_s. \tag{10}$$

In the policy improvement step, in order to make the new policy get a higher return while satisfying the safety constraint, we

Algorithm 2: Constrained PI.

Input: A risk horizon N. A reward discount factor γ. A set $\mathcal{K} = \{0, \ldots, N-1\}$. $\pi(\boldsymbol{s})$, $V^{\pi}(\boldsymbol{s})$, $\mathcal{V}^{\pi}_{N-t}(\boldsymbol{s}), t \in \mathcal{K}, \forall \boldsymbol{s} \in \mathcal{S}$. An initial safe policy $\pi^{\dagger}$.

Procedure:

Let $\pi \leftarrow \pi^{\dagger}$.
repeat
  Set $V^{\pi}(\boldsymbol{s}) = 0, \mathcal{V}^{\pi}_{N-t}(\boldsymbol{s}) = 0, t \in \mathcal{K}, \forall \boldsymbol{s} \in \mathcal{S}$.
  repeat {Policy evaluation}
    for $\boldsymbol{s}$ in $\mathcal{X}_s$ **do**
      $V^{\pi}(\boldsymbol{s}) \leftarrow \sum_{\boldsymbol{a}} \pi(\boldsymbol{a}|\boldsymbol{s}) \sum_{\boldsymbol{s}'} P(\boldsymbol{s}'|\boldsymbol{s}, \boldsymbol{a})[r + \gamma V^{\pi}(\boldsymbol{s}')]$.
      $\mathcal{V}^{\pi}_{N-t}(\boldsymbol{s}) \leftarrow \mathbb{E}_{\pi}[c + \mathcal{V}^{\pi}_{N-t-1}(\boldsymbol{s}')], t \in \mathcal{K}$.
    end for
  until $V^{\pi}(\boldsymbol{s})$ and $\mathcal{V}^{\pi}_{N}(\boldsymbol{s}), \forall \boldsymbol{s} \in \mathcal{X}_s$ converge.
  Set $\mathcal{V}^{\pi}_{N-t}(\boldsymbol{s}) = 1, \boldsymbol{s} \in \mathcal{S}_u, t \in \mathcal{K}$.
  Get $Q^{\pi}(\boldsymbol{s}, \boldsymbol{a})$, $\mathcal{Q}^{\pi}_{N}(\boldsymbol{s}, \boldsymbol{a})$ by $V^{\pi}(\boldsymbol{s})$, $\mathcal{V}^{\pi}_{N}(\boldsymbol{s})$.
  for $\boldsymbol{s}$ in $\mathcal{X}$ **do** {Constrained Policy Improvement}
    $\mathcal{I} = \{\tilde{\boldsymbol{a}} \mid \mathcal{Q}^{\pi}_{N-t}(\boldsymbol{s}, \tilde{\boldsymbol{a}}) \leq \mathcal{V}^{\pi}_{N-t}(\boldsymbol{s}), t \in \mathcal{K}\}$.
    $\pi(\boldsymbol{s}) \leftarrow \arg\max_{\tilde{\boldsymbol{a}} \in \mathcal{I}} Q^{\pi}(\boldsymbol{s}, \tilde{\boldsymbol{a}})$.
  end for
until π converge.

Output: π.

Algorithm 3: Fixed-horizon Constrained Policy Iteration (FHCPI).

Input: An initial policy π.

Procedure:

From π, get the initial safe policy $\pi^{\dagger}$ by Algorithm 1.
From $\pi^{\dagger}$, get the final policy π^{*} by Algorithm 2.

Output: π^{*}.

update the policy by the following form:

$$\pi'(\boldsymbol{s}) = \arg\max_{\tilde{\boldsymbol{a}}} Q^{\pi}(\boldsymbol{s}, \tilde{\boldsymbol{a}}) \quad \forall \boldsymbol{s} \in \mathcal{X}$$
$$\text{s.t. } \mathcal{Q}^{\pi}_{N-t}(\boldsymbol{s}, \tilde{\boldsymbol{a}}) \leq \mathcal{V}^{\pi}_{N-t}(\boldsymbol{s}), \quad t = 0, \ldots, N-1. \tag{11}$$

Note that the above optimization problem has at least one feasible solution, which is $\pi^{\dagger}$. The updated form can lead to an improved policy in terms of $V^{\pi}(\boldsymbol{s})$, which means $V^{\pi}(\boldsymbol{s}) \leq V^{\pi'}(\boldsymbol{s}), \forall \boldsymbol{s} \in \mathcal{S}$.

Lemma 3 (Constrained policy improvement): If a new policy π' is the optimal feasible solution of the maximization problem defined in (11). Then, we have $V^{\pi}(\boldsymbol{s}) \leq V^{\pi'}(\boldsymbol{s}), \forall \boldsymbol{s} \in \mathcal{S}$.

Proof: See Appendix D of the Supplementary Material. ■

Lemma 4 (Safety guarantee): If a new policy π' is the optimal feasible solution of the maximization problem defined in (11). Then, we have $\mathcal{V}^{\pi}_{N}(\boldsymbol{s}) \geq \mathcal{V}^{\pi'}_{N}(\boldsymbol{s}), \forall \boldsymbol{s} \in \mathcal{S}$.

Proof: See Appendix E of the Supplementary Material. ■

Thus, by updating the policy using (11) iteratively, the resulting policy will also satisfy the safety constraint (i.e., $\mathcal{V}^{\pi'}_{N}(\boldsymbol{s}) \leq \mathcal{V}^{\pi^{\dagger}}_{N}(\boldsymbol{s}) \leq h, \forall \boldsymbol{s} \in \mathcal{S}$), and the cumulative reward of the new policy will be greater than or equal to that of the old policy. We summarize the constrained PI process as Algorithm 2. If

Algorithm 4: Fixed-horizon Constrained Policy Optimization (FHCPO).

Input: Initialize parameters $\boldsymbol{\theta}$ of the policy network π_{θ}. Initialize parameters $\boldsymbol{\varphi}_1, \boldsymbol{\varphi}_2, \boldsymbol{\varphi}_3$ of three critic networks $\hat{V}_{\varphi_1}, \hat{\mathcal{V}}_{\varphi_2}, \hat{\mathcal{V}}_{\varphi_3}$, respectively. Initialize a data buffer $\mathcal{D}$.

Procedure:

for each epoch **do**
  Collect a series of trajectories under policy π_{θ}.
  Store the trajectories in $\mathcal{D}$.
  Compute $A^{\pi_{\theta}}$ with $\hat{V}_{\varphi_1}$ and data in $\mathcal{D}$.
  Compute $\mathcal{A}^{\pi_{\theta}}_{N}$ with $\hat{\mathcal{V}}_{\varphi_2}, \hat{\mathcal{V}}_{\varphi_3}$ and data in $\mathcal{D}$.
  Estimate KL divergence $KL = \mathbb{E}_{\boldsymbol{s}}[D(\pi_k(\cdot|\boldsymbol{s})||\pi_{\theta}(\cdot|\boldsymbol{s}))]$ with data in $\mathcal{D}$.
  Estimate $\boldsymbol{g}, \boldsymbol{b}, H$ with $A^{\pi_{\theta}}, \mathcal{A}^{\pi_{\theta}}_{N}$ and KL.
  Update $\boldsymbol{\theta}$ by solving the problem defined in (17).
  Update $\boldsymbol{\varphi}_1, \boldsymbol{\varphi}_2, \boldsymbol{\varphi}_3$ by gradient descent by MSE loss on reward-to-go and cost-to-go.
  Clear buffer $\mathcal{D}$.
end for

Output: π_{θ}.

we combine Algorithms 1 and 2, we can get the final two-stage FHCPI algorithm, shown in Algorithm 3. Moreover, we can get the following theorem.

Theorem 1: Starting from an arbitrary policy π_0, the proposed FHCPI algorithm can eventually make the policy converge to a local optimal policy whose safety is guaranteed.

Proof: See Appendix F of the Supplementary Material. ■

It should be noted that Algorithm 3 will converge to only a local optimal policy of the problem presented in (8), which means the learned policy does not necessarily equal to an optimal policy π^{*} which satisfies

$$V^{\pi^{*}}(\boldsymbol{s}) \geq V^{\pi}(\boldsymbol{s}) \quad \forall \boldsymbol{s} \in \mathcal{S}$$
$$\forall \pi \in \{\pi \mid \mathcal{V}^{\pi}_{N}(\boldsymbol{s}) \leq h \quad \forall \boldsymbol{s} \in \mathcal{S}_s\}. \tag{12}$$

However, when we set $h = 0$, the proposed two-stage FHCPI algorithm can guarantee the global optimal and the safety of the learned policy. We summarize this content in Appendix G of the Supplementary Material. In addition, the two-stage FHCPI algorithm is also suitable for model-free situations. Instead of the policy evaluation step by iteration, the Monte Carlo method or the temporal-difference method can be utilized to estimate all value functions [27], i.e., $V^{\pi}, \mathcal{V}^{\pi}_{N-t}, Q^{\pi}$ and $\mathcal{Q}^{\pi}_{N-t}, t = 0, \ldots, N$.

B. FHCPL for Continuous Situation

For the continuous state and action space, a function or a neural network, parameterized by $\boldsymbol{\theta}$, can be used to approximate the policy π. Then, for the problem presented in (11), the value functions can be approximated by neural networks, and the policy can be updated using normal constrained optimization methods. However, it is usually intractable to optimize the neural network with N constraints in (11) efficiently. Therefore, for the continuous situation with the fixed-horizon constraint,

we propose the FHCPO algorithm. To enable the proposed fixed-horizon constraint to be used in direct policy optimization methods, we first give the approximate performance difference lemma w.r.t. the risk state value function.

Lemma 5 (Approximate performance difference): If we let $\mathbb{E}_{\boldsymbol{s}}[D(\pi'(\cdot|\boldsymbol{s})||\pi(\cdot|\boldsymbol{s}))] \leq \delta$, where δ is a small positive real number that is close to 0, then we can have that $\mathcal{V}_N^{\pi'}(\boldsymbol{s}) - \mathcal{V}_N^{\pi}(\boldsymbol{s}) \approx \mathbb{E}_{\pi'}[\sum_{k=0}^{N-1} \mathcal{A}_N^{\pi}(\boldsymbol{s}_{t+k}, \boldsymbol{a}_{t+k})|\boldsymbol{s}_t = \boldsymbol{s}]$, where $\mathcal{A}_N^{\pi}(\boldsymbol{s}, \boldsymbol{a}) = \mathcal{Q}_N^{\pi}(\boldsymbol{s}, \boldsymbol{a}) - \mathcal{V}_N^{\pi}(\boldsymbol{s})$, and $D(\cdot||\cdot)$ is some kind of distance metric.

Proof: See Appendix H of the Supplementary Material. ■

Then, we denote $\mathbb{E}_{\boldsymbol{s}\sim\phi(\cdot)}[\mathcal{V}_N^{\pi}(\boldsymbol{s})]$ as $J_c(\pi)$. According to Lemma 5, we have

$$\begin{aligned} J_c(\pi') - J_c(\pi) &= \mathbb{E}_{\boldsymbol{s}\sim\phi(\cdot)}\left[\mathcal{V}_N^{\pi'}(\boldsymbol{s}) - \mathcal{V}_N^{\pi}(\boldsymbol{s})\right] \\ &\approx \mathbb{E}_{\boldsymbol{s}\sim\phi(\cdot)}\left[\mathbb{E}_{\pi'}\left[\sum_{k=0}^{N-1} \mathcal{A}_N^{\pi}(\boldsymbol{s}_{t+k}, \boldsymbol{a}_{t+k})\middle|\boldsymbol{s}_t = \boldsymbol{s}\right]\right] \\ &= \mathbb{E}_{\boldsymbol{s}\sim\phi(\cdot)}\left[\mathbb{E}_{\pi}\left[\sum_{k=0}^{N-1} \rho_{\pi}^{\pi'} \mathcal{A}_N^{\pi}(\boldsymbol{s}_{t+k}, \boldsymbol{a}_{t+k})\middle|\boldsymbol{s}_t = \boldsymbol{s}\right]\right] \end{aligned} \tag{13}$$

where $\rho_{\pi}^{\pi'} = \prod_{k=0}^{N-1} \frac{\pi'(\boldsymbol{a}_{t+k}|\boldsymbol{s}_{t+k})}{\pi(\boldsymbol{a}_{t+k}|\boldsymbol{s}_{t+k})}$. Then, we also denote $J(\pi) = \mathbb{E}_{\pi}[\sum_{t=0}^{\infty} \gamma^t r(\boldsymbol{s}_t, \boldsymbol{a}_t)]$. Thus, the problem presented in (8) can be written as

$$\max_{\pi} \quad J(\pi), \quad \text{s.t. } J_c(\pi) \leq h. \tag{14}$$

To solve the problem defined in (14), the trust region optimization approach is introduced. Specifically, in each step of the policy optimization, we aim to tackle the following problem:

$$\begin{aligned} \pi_{k+1} = \arg\max_{\pi} \quad & J(\pi) - J(\pi_k) \\ \text{s.t.} \quad & J_c(\pi) \leq h \\ & D(\pi, \pi_k) \leq \delta \end{aligned} \tag{15}$$

where D is some kind of distance metric. In addition, by [32], we have, $J(\pi) - J(\pi_k) \approx 1/(1-\gamma)\mathbb{E}_{\boldsymbol{s}\sim d^{\pi_k}, \boldsymbol{a}\sim\pi}[A^{\pi_k}(\boldsymbol{s}, \boldsymbol{a})]$, where $d^{\pi}(\boldsymbol{s}) = (1-\gamma)\sum_t \gamma^t P_{\pi}(\boldsymbol{s}_t = \boldsymbol{s})$. By introducing Kullback–Leibler (KL) divergence, we can approximate the above optimization problem as follows:

$$\begin{aligned} \pi_{k+1} \approx \arg\max_{\pi} \quad & \frac{1}{1-\gamma}\mathbb{E}_{\boldsymbol{s}\sim d^{\pi_k}, \boldsymbol{a}\sim\pi}[A^{\pi_k}(\boldsymbol{s}, \boldsymbol{a})] \\ \text{s.t. } J_c(\pi_k) + \mathop{\mathbb{E}}_{\substack{\boldsymbol{s}\sim d^{\pi_k} \\ \boldsymbol{a}\sim\pi_k}} & \left[\sum_{k=0}^{N-1} \rho_{\pi_k}^{\pi} \mathcal{A}_N^{\pi_k}(\boldsymbol{s}_{t+k}, \boldsymbol{a}_{t+k})\middle|\boldsymbol{s}_t = \boldsymbol{s}\right] \leq h \\ & \mathbb{E}_{\boldsymbol{s}}[D(\pi_k(\cdot|\boldsymbol{s})||\pi(\cdot|\boldsymbol{s}))] \leq \delta \end{aligned} \tag{16}$$

where $A^{\pi_k}(\boldsymbol{s}, \boldsymbol{a})$ is the advantage function in terms of reward, and $\rho_{\pi_k}^{\pi} = \prod_{k=0}^{N-1} \frac{\pi(\boldsymbol{a}_{t+k}|\boldsymbol{s}_{t+k})}{\pi_k(\boldsymbol{a}_{t+k}|\boldsymbol{s}_{t+k})}$.

According to Taylor's expansions on the objective function and constraint function of the above problem, the final approximate optimization problem can be written as

$$\boldsymbol{\theta}_{k+1} \approx \arg\max_{\boldsymbol{\theta}} \quad \boldsymbol{g}^T(\boldsymbol{\theta} - \boldsymbol{\theta}_k) \tag{17a}$$

$$\text{s.t.} \quad J_c(\pi_k) - h + \boldsymbol{b}^T(\boldsymbol{\theta} - \boldsymbol{\theta}_k) \leq 0 \tag{17b}$$

$$\frac{1}{2}(\boldsymbol{\theta} - \boldsymbol{\theta}_k)^T H(\boldsymbol{\theta} - \boldsymbol{\theta}_k) \leq \delta \tag{17c}$$

where $\boldsymbol{\theta}_k$ is the current parameters of the policy net. $\boldsymbol{g}$ is the gradient of the objective function at $\boldsymbol{\theta}_k$, and $\boldsymbol{b}$ is the gradient of the constraint function. H denotes the Hessian matrix of the KL-divergence.

In addition, the problem in (17) is a convex optimization problem whose analytical solution can be obtained directly through the KKT condition. Therefore, we can solve the problem defined in (14) by iteratively updating $\boldsymbol{\theta}$ using (17). Notably, compared with the constrained optimization problem in the case of infinite horizons, (17b) has a more relaxed constraint. This is because, when we apply the infinite horizon constraint, the final optimization problem shown in (17b) will be rewritten as

$$J_{c,\infty}(\pi_k) - h + \boldsymbol{b}^T(\boldsymbol{\theta} - \boldsymbol{\theta}_k) \leq 0 \tag{18}$$

where $J_{c,\infty}(\pi_k)$ is defined as $\mathbb{E}_{\boldsymbol{s}\sim\phi(\cdot)}[\mathcal{V}_{\infty}^{\pi_k}(\boldsymbol{s})]$. From the definitions of J_c and $J_{c,\infty}$, we have $J_c(\pi_k) \leq J_{c,\infty}(\pi_k)$. Therefore, in the fixed-horizon setting, the constraint becomes more lenient, and the agent has a greater possibility of getting higher returns.

In the practical algorithm, according to the definition of A^{π_θ} and $\mathcal{A}_N^{\pi_\theta}$, we can get $A^{\pi_\theta}(\boldsymbol{s}, \boldsymbol{a})$ and $\mathcal{A}_N^{\pi_\theta}(\boldsymbol{s}, \boldsymbol{a})$ by the following formulas:

$$A^{\pi_\theta}(\boldsymbol{s}, \boldsymbol{a}) = r(\boldsymbol{s}, \boldsymbol{a}) + \gamma V^{\pi_\theta}(\boldsymbol{s}') - V^{\pi_\theta}(\boldsymbol{s}) \tag{19}$$

$$\mathcal{A}_N^{\pi_\theta}(\boldsymbol{s}, \boldsymbol{a}) = c(\boldsymbol{s}, \boldsymbol{a}) + \mathcal{V}_{N-1}^{\pi_\theta}(\boldsymbol{s}') - \mathcal{V}_N^{\pi_\theta}(\boldsymbol{s}) \tag{20}$$

where V^{π_θ} and $\mathcal{V}_N^{\pi_\theta}$ can be obtained by

$$\begin{aligned} V^{\pi_\theta}(\boldsymbol{s}) &= \mathbb{E}_{\pi_\theta}\left[\sum_{k=0}^{\infty} \gamma^k r_{t+k+1}\middle|\boldsymbol{s}\right] \approx \frac{1}{M}\sum_{i=1}^{M}\sum_{k=0}^{\infty} \gamma^t r_{t+k+1} \\ \mathcal{V}_N^{\pi_\theta}(\boldsymbol{s}) &= \mathbb{E}_{\pi_\theta}\left[\sum_{k=0}^{N-1} c_{t+k+1}\middle|\boldsymbol{s}\right] \approx \frac{1}{M}\sum_{i=1}^{M}\sum_{k=0}^{N-1} c_{t+k+1} \end{aligned} \tag{21}$$

where M is the number of occurrences of $\boldsymbol{s}$ in the sampled data buffer $\mathcal{D}$.

Therefore, in order to estimate the advantage functions, we can use three critic neural networks $\hat{V}_{\varphi_1}$, $\hat{\mathcal{V}}_{\varphi_2}$, and $\hat{\mathcal{V}}_{\varphi_3}$ to approximate V^{π_θ}, $\mathcal{V}_N^{\pi_\theta}$, and $\mathcal{V}_{N-1}^{\pi_\theta}$, respectively. We denote the reward-to-go $\sum_{k=0}^{\infty} \gamma^t r_{t+k+1}$ as $\hat{R}_t$. We also denote the cost-to-go $\sum_{k=0}^{N-1} c_{t+k+1}$ and $\sum_{k=0}^{N-2} c_{t+k+1}$ as $\hat{C}_t^N$, $\hat{C}_t^{N-1}$, respectively. Then, the loss functions of these three critic networks can be defined as the mean square error (MSE) on the reward-to-go $\hat{R}_t$ or the corresponding cost-to-go $\hat{C}_t^N$, $\hat{C}_t^{N-1}$. We summarize the FHCPO in Algorithm 4.

V. Experiments

In this section, we evaluate the proposed FHCPI algorithm in a discrete grid world environment. In addition, we conduct extensive experiments on several risk-sensitive environments to evaluate our FHCPO algorithm in continuous situations. We compare FHCPO with three widely cited and state-of-the-art constrained RL algorithms, i.e., CPO [17], IPO [29], and PCPO [30], and one classic deep RL algorithm, i.e., PPO [16]. We aim to investigate the following **r**esearch **q**uestions.

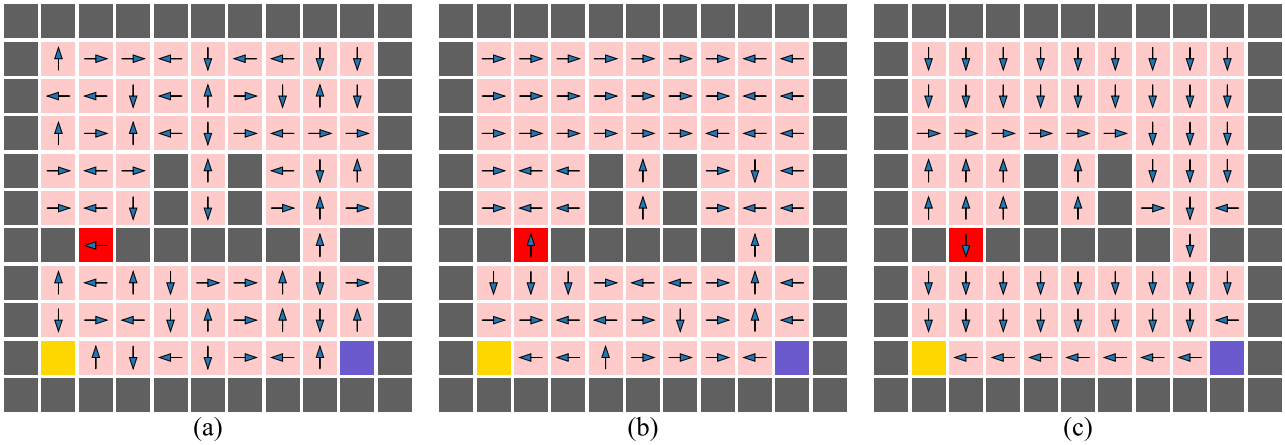

Fig. 3. Experimental results of FHCPI in the discrete grid world environment. Three images represent (a) the random initialized policy, (b) the safe policy $\pi^{\dagger}$ via safe PI (stage 1), and (c) the final learned policy π^{*} by the constrained PI (stage 2), respectively.

1) *RQ1:* Can the proposed two-stage FHCPI algorithm effectively encourage the agent to avoid risks and get high returns for the discrete case?
2) *RQ2:* For continuous cases, can the proposed FHCPO algorithm attain better performance w.r.t. total reward and risk avoidance than existing constrained RL algorithms and normal deep RL algorithms?
3) *RQ3:* How does the fixed-horizon N affect the performance of the proposed FHCPO algorithm?
4) *RQ4:* When we change the threshold C and the cost discount $\bar{\gamma}$ in the typical cost constraint in existing constrained RL algorithms (e.g., CPO), can we achieve similar performances to that in our FHCPO methods?

A. Evaluation Metrics and Platform Configuration

To compare different algorithms in both discrete and continuous cases, we adopt the following evaluation metrics: 1) Total reward, $\mathbb{E}_{\pi}[\sum_t r(\boldsymbol{s}_t, \boldsymbol{a}_t)]$; 2) average collision rate, $\mathbb{E}_{\pi}[\sum_t c(\boldsymbol{s}_t, \boldsymbol{a}_t)/l]$, where l is the length of each trajectory. In addition to maximizing the total reward like in typical RL, we also aim to make average collision rate (unsafe state ratio) within certain thresholds. All experiments were conducted on a PC equipped with an Intel Core i7-9700 CPU, NVIDIA RTX 3060 GPU, and 32 GB RAM. The machine learning framework PyTorch was employed to facilitate the training of neural networks [33].

B. Discrete Grid World Experiment

1) Description of the Grid World Environment: In our risk-sensitive discrete grid world environment, shown in Fig. 3(a), the agent is aimed at reaching the target state without going through unsafe states. The agent will receive a positive reward (+5) when it enters the target terminal state (gold square). In addition, there is an interference terminal state (blue square) in the environment, which is unsafe, but the agent will receive a reward (+5) when it enters this state. This sort of interference state often occurs in environments where rewards are not perfectly designed [34]. An unsafe transient state (red square) also occurs in this environment.

TABLE I
TOTAL REWARD, AND COLLISION RATE COMPARISON IN DISCRETE ENVIRONMENT

Metric	FHCPI	PI	Discrete-PPO	Discrete-CPO
Total reward	5.0	5.0	4.9515	0.0
Collision rate	0.0	0.1288	0.1672	0.0031

2) Experimental Results: Fig. 3 represents the learned policies in the course of the proposed two-stage FHCPI method. Fig. 3(a) shows a random initialized policy, and Fig. 3(b) represents the safe policy $\pi^{\dagger}$ obtained by the safe PI algorithm, in which the agent does not enter any unsafe state, but it does not achieve high returns either. Fig. 3(c) represents the final learned policy π^{*} by the constrained PI algorithm, where the risks are fully considered, and the agent also reaches the target. In addition, we compare the FHCPI with normal PI in RL [27], the discrete version of PPO (Discrete-PPO) [16], and the discrete version of the CPO (Discrete-CPO) [17]. In addition, we set $h = 0.01$ in each experiment. The corresponding experiment result is given in Table I. It can be observed that the proposed FHCPI algorithm achieves the optimal solution in this environment. PI and Discrete-PPO obtained similar performance. They can get high returns but high collision rates. However, Discrete-CPO can achieve low collision rates in this environment, but it can hardly achieve high returns, which demonstrates that the policy obtained by Discrete-CPO is conservative. Therefore, for *RQ1*, we can conclude that the proposed two-stage FHCPI algorithm can encourage the agent to avoid risks and get high returns in discrete environments.

C. Continuous Experiments

To answer *RQ2*, we compared the proposed FHCPO algorithm with three state-of-the-art safe RL algorithms, i.e., CPO, IPO,

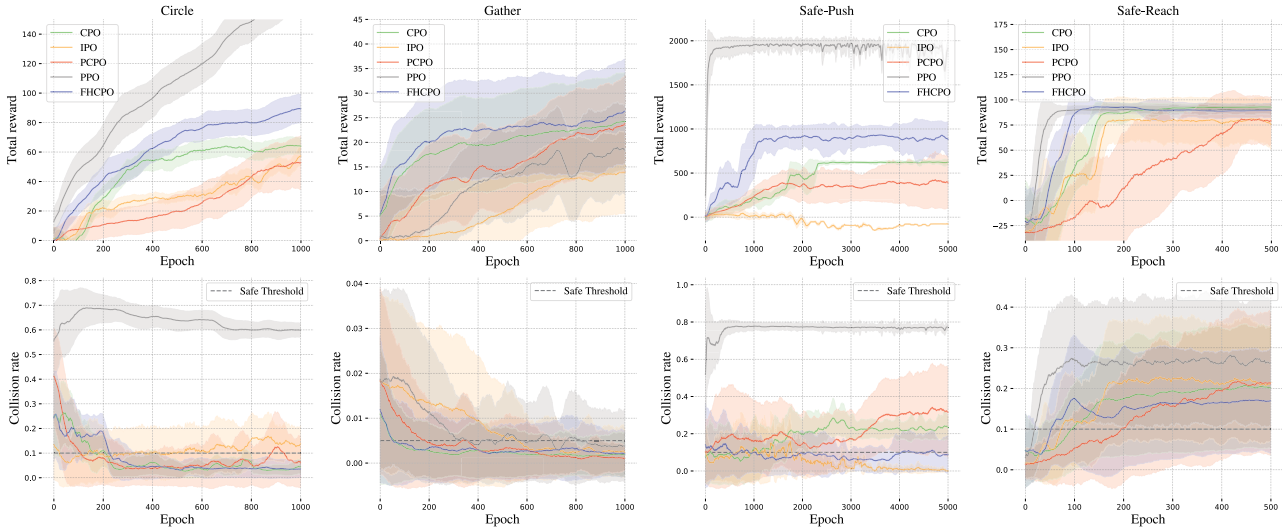

Fig. 4. Total reward, and collision rate, along with policy updates of PPO, CPO, IPO, PCPO, and the proposed FHCPO in circle, gather, safe-push, and safe-reach environments. Each curve is the average of the results under ten random seeds.

and PCPO, and a deep RL algorithm, i.e., PPO, in four risk-sensitive environments, which are illustrated in Appendix I of the Supplementary Material.

1) Experimental Results: Fig. 4 shows the experimental results of the comparison between the proposed FHCPO algorithm and the typical constrained RL methods with metrics of total reward, and collision rate in four risk-sensitive environments. The details parameters of the experiment are shown in Appendix J of the Supplementary Material. *Circle:* Compared with the CPO, IPO, and PCPO, the proposed FHCPO algorithm achieves a higher total reward and a more efficient training process while maintaining a low or comparable collision rate. In addition, the proposed FHCPO and the other two counterparts (CPO and PCPO) satisfy the safety constraints. On the other hand, since PPO does not consider safety in the environment, it achieves the highest return, but it also obtains the highest collision rate. *Gather:* The learned policies of all algorithms satisfy the safety constraints. However, our proposed FHCPO algorithm achieves the highest return, and fastest converge speed. It is worth noting that PPO can also satisfy the safety constraints in this environment. This is because the rewards in this environment penalize the agent's unsafe behavior. *Safe-Push:* Due to a lack of consideration for safety, PPO achieves almost optimal total reward but also has the highest collision rate. IPO has achieved a nearly 0 collision rate, but it cannot achieve a relatively high total reward. However, our FHCPO method can still achieve high returns while satisfying the constraints. *Safe-Reach:* In this environment, the collision rate of these methods is gradually increasing. This is because unsafe regions surround the target point, and the agent risks rushing to the goal. All methods fail to satisfy the constraints, but our FHCPO algorithm maintains a relatively low level of collision rate. In addition, in terms of total reward, our FHCPO can still achieve a faster convergence speed than its counterparts. These observations suggest that, with the advantage of the fixed-horizon risk, the proposed FHCPO methods can attain better returns while meeting the safety constraints.

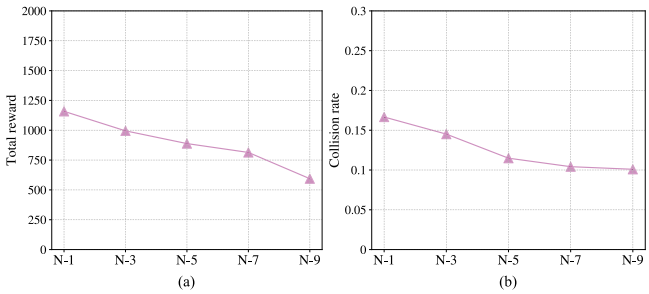

Fig. 5. (a) Total reward (b) and collision rate of different fixed-horizon N in safe-push environment with the proposed FHCPO.

D. FHCPO With Different Fixed-Horizon N

To answer the *RQ3*, we also conducted an additional experiment to explore how the fixed-horizon N affects the agent performance in our FHCPO algorithm. The result is shown in Fig. 5. Except for the difference in the fixed-horizon N, other parameters are the same as in the part of continuous experiments. From Fig. 5, we can observe that as N becomes larger, the return becomes lower, and the collision rate correspondingly becomes smaller. This is because when the fixed-horizon becomes larger, the agent will consider risks in further future, causing it to be more conservative. Conversely, as N gets smaller, the return becomes higher and the collision rate increases accordingly. However, the collision rate does not change drastically and remains around 0.12. Thus, the agent still maintains the risk-sensitive behavior. Therefore, by adjusting the value of the fixed-horizon N, the proposed FHCPO algorithm can trade off between high rewards and low risks.

E. Sensitivity Analysis of CPO

Considering the fact that typical safe RL methods, such as CPO, can also adjust the value of the discount $\bar{\gamma}$ or the cost

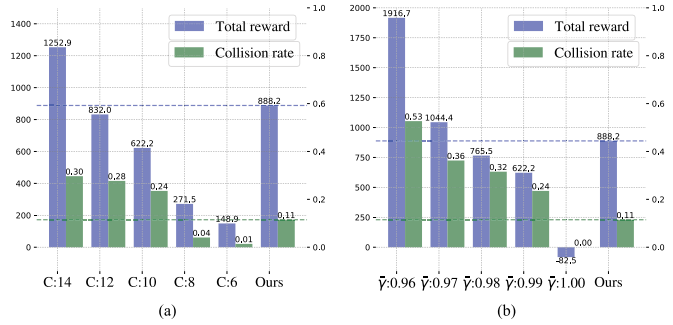

Fig. 6. Comparison between FHCPO and CPO in terms of total reward and collision rate on different (a) C and (b) $\bar{\gamma}$ in Safe-Push environment.

threshold C of the constraint to make a tradeoff between high rewards and low risks. To verify whether the CPO can achieve an equivalent performance to the proposed FHCPO, we performed a series of experiments in the safe-push environment using CPO by changing the cost threshold C and cost discount factor $\bar{\gamma}$. Fig. 6 shows the experimental result in different settings. It can be observed in Fig. 6 that as the cost threshold increases, the total reward also increases, but the collision rate will increase accordingly. When the value of C is small ($C = 6$), the agent will hardly enter unsafe areas, e.g., the collision rate is 0.01. But the return will be low due to being too conservative. However, in our FHCPO method, the agent can get a high return while maintaining a low collision rate. A similar situation also happens when we adjust $\bar{\gamma}$. In CPO, the algorithm is sensitive to the change of $\bar{\gamma}$. When $\bar{\gamma} = 0.96$, the agent ignores any risks and seeks to maximize the total reward. Thus, for normal safe RL methods, it is difficult to reach the performance of our FHCPO method by modifying C or $\bar{\gamma}$.

F. Industrial Applications

Safety is a critical concern in many industrial applications, and RL algorithms must be designed to ensure that each agent behaves in a safe and reliable manner. Thus, safe RL is important for such industrial applications. In the field of intelligent transportation, safe RL can be used to train self-driving cars or traffic signal control systems to help agents make decisions in complex traffic environments, which can maximize traffic efficiency without causing car accidents. In addition, in manufacturing, safe RL can be used to train industrial collaborative robots to perform assembly tasks in complex scenarios while avoiding damaging equipment or harming workers. Even in the field of energy management, such as a smart grid, safe RL can be used for power dispatching of the grid, and the trained agent needs to maximize the energy utilization efficiency of an area while ensuring that the current on the wire does not exceed the load. In addition, the proposed FHCPL can also be applied to other risk-sensitive industrial scenarios [35], [36], [37], [38], [39], [40], such as the field of Big Data and IoT systems, in which it is often necessary to ensure stable communication between devices. Therefore, the proposed FHCPL framework will effectively empower many risk-sensitive industrial scenarios and be applied to enhance efficiencies while ensuring the safety of industrial systems, as shown in Fig. 2. To better apply the FHCPL to different industrial scenarios, some tips are as follows.

1) Select a suitable fixed-horizon N according to the industrial scenario. In general, for problems with long-delayed or sparse costs, the value of N should be larger. On the contrary, it should be smaller.
2) Design appropriate reward and cost functions. They should preferably be designed to be dense and reflect the good or bad of the current state-action pair. This means an agent can get an accurate response from the reward and cost after taking an action.
3) Design a suitable state space so that a state can provide enough information for the agent to make decisions.

VI. Conclusion and the Future Work

In this article, for risk-sensitive industrial scenarios, we proposed the FHCPL system, in which a fixed-horizon constraint was presented that relaxes existing infinite horizon constraints and makes the agent's exploration more efficient. Meanwhile, a sense of risk avoidance in exploration is also maintained. Thus, better performance can be achieved by better exploration. For discrete cases, a two-stage FHCPI algorithm was given, in which the safety is guaranteed. A grid world experiment demonstrates that the FHCPI algorithm can help the agent avoid risks while obtaining high returns. For continuous cases, we proposed the FHCPO algorithm, which achieves superior performance to the typical safe RL methods in terms of total reward and collision rate in extensive empirical experiments. Nevertheless, we only considered the case where the horizon N is a fixed value. In the future, we will cogitate about making the fixed-horizon a dynamically adjustable value, which may enable the agent to explore better.

Appendix

The appendix is in the supplementary materials and also in a URL https://github.com/SChrisLin/FHCPL-APPENDIX/blob/main/Appendix.pdf.

References

[1] D. Silver et al., "Mastering the game of Go with deep neural networks and tree search," *Nature*, vol. 529, pp. 484–489, Jan. 2016.
[2] C. Yan et al., "Deep reinforcement learning of collision-free flocking policies for multiple fixed-wing UAVs using local situation maps," *IEEE Trans. Ind. Inform.*, vol. 18, no. 2, pp. 1260–1270, Feb. 2022.
[3] M. K. M. Rabby, A. Karimoddini, M. A. Khan, and S. Jiang, "A learning-based adjustable autonomy framework for human-robot collaboration," *IEEE Trans. Ind. Inform.*, vol. 18, no. 9, pp. 6171–6180, Sep. 2022.
[4] O. Vinyals et al., "Grandmaster level in StarCraft II using multi-agent reinforcement learning," *Nature*, vol. 575, pp. 350–354, Nov. 2019.
[5] K. Łukasz et al., "Model based reinforcement learning for atari," in *Proc. Int. Conf. Learn. Representations*, 2020. [Online]. Available: https://openreview.net/pdf?id=S1xCPJHtDB
[6] X. Luo, Y. Zhou, Z. Liu, and M. Zhou, "Fast and accurate non-negative latent factor analysis of high-dimensional and sparse matrices in recommender systems," *IEEE Trans. Knowl. Data Eng.*, vol. 35, no. 4, pp. 3897–3911, Apr. 2023.
[7] X. Luo, Y. Yuan, S. Chen, N. Zeng, and Z. Wang, "Position-transitional particle swarm optimization-incorporated latent factor analysis," *IEEE Trans. Knowl. Data Eng.*, vol. 34, no. 8, pp. 3958–3970, Aug. 2022.

[8] H.-M. Chung, S. Maharjan, Y. Zhang, and F. Eliassen, "Distributed deep reinforcement learning for intelligent load scheduling in residential smart grids," *IEEE Trans. Ind. Inform.*, vol. 17, no. 4, pp. 2752–2763, Apr. 2021.
[9] D. Qiu, T. Chen, G. Strbac, and S. Bu, "Coordination for multienergy microgrids using multiagent reinforcement learning," *IEEE Trans. Ind. Inform.*, vol. 19, no. 4, pp. 5689–5700, Apr. 2023.
[10] D. Liu, Q. Wei, D. Wang, X. Yang, and H. Li, *Adaptive Dynamic Programming With Applications in Optimal Control*. Cham, Switzerland: Springer, 2017.
[11] J. Yu, X. Dong, Q. Li, J. Lü, and Z. Ren, "Adaptive practical optimal time-varying formation tracking control for disturbed high-order multi-agent systems," *IEEE Trans. Circuits Syst. I: Reg. Papers*, vol. 69, no. 6, pp. 2567–2578, Jun. 2022.
[12] Q. Wei, F.-Y. Wang, D. Liu, and X. Yang, "Finite-approximation-error-based discrete-time iterative adaptive dynamic programming," *IEEE Trans. Cybern.*, vol. 44, no. 12, pp. 2820–2833, Dec. 2014.
[13] J. García and F. Fernández, "A comprehensive survey on safe reinforcement learning," *J. Mach. Learn. Res.*, vol. 16, pp. 1437–1480, 2015.
[14] J. Schulman, P. Moritz, S. Levine, M. Jordan, and P. Abbeel, "High-dimensional continuous control using generalized advantage estimation," in *Proc. Int. Conf. Learn. Representations*, 2016. [Online]. Available: https://arxiv.org/pdf/1506.02438.pdf
[15] J. Schulman et al., "Trust region policy optimization," in *Proc. Int. Conf. Mach. Learn.*, 2015, pp. 1889–1897.
[16] J. Schulman et al., "Proximal policy optimization algorithms," 2017, *arXiv:1707.06347*.
[17] J. Achiam et al., "Constrained policy optimization," in *Proc. Int. Conf. Mach. Learn.*, 2017, pp. 22–31.
[18] Y. Zhang, Q. Vuong, and K. Ross, "First order constrained optimization in policy space," in *Proc. Adv. Neural Inf. Process. Syst.*, 2020, pp. 15338–15349.
[19] Q. Yang, T. D. Simao, S. H. Tindemans, and M. T. Spaan, "WCSAC: Worst-case soft actor critic for safety-constrained reinforcement learning," in *Proc. AAAI Conf. Artif. Intell.*, 2021, pp. 10639–10646.
[20] Y. Chow, O. Nachum, E. Duenez-Guzman, and M. Ghavamzadeh, "A Lyapunov-based approach to safe reinforcement learning," in *Proc. Adv. Neural Inf. Process. Syst.*, 2018, pp. 8103–8112.
[21] G. Thomas, Y. Luo, and T. Ma, "Safe reinforcement learning by imagining the near future," in *Proc. Adv. Neural Inf. Process. Syst.*, 2021, pp. 13859–13869.
[22] K. W. Ross, "Randomized and past-dependent policies for Markov decision processes with multiple constraints," *Operations Res.*, vol. 37, no. 3, pp. 474–477, Jun. 1989.
[23] E. Altman, *Constrained Markov Decision Processes*. London, U.K.: Routledge, 2021.
[24] J. Achiam and D. Amodei, "Benchmarking safe exploration in deep reinforcement learning," 2019. Accessed: Dec. 7, 2023. [Online]. Available: https://cdn.openai.com/safexp-short.pdf
[25] Y. Zhang and K. Ross, "On-policy deep reinforcement learning for the average-reward criterion," in *Proc. Int. Conf. Mach. Learn.*, 2021, pp. 12535–12545.
[26] K. De Asis et al., "Fixed-horizon temporal difference methods for stable reinforcement learning," in *Proc. AAAI Conf. Artif. Intell.*, 2020, pp. 3741–3748.
[27] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. Cambridge, MA, USA: MIT Press, 2018.
[28] Y. Chow, M. Ghavamzadeh, L. Janson, and M. Pavone, "Risk-constrained reinforcement learning with percentile risk criteria," *J. Mach. Learn. Res.*, vol. 18, pp. 6070–6120, 2017.
[29] Y. Liu, J. Ding, and X. Liu, "IPO: Interior-point policy optimization under constraints," in *Proc. AAAI Conf. Artif. Intell.*, 2020, pp. 4940–4947.
[30] T.-Y. Yang et al., "Projection-based constrained policy optimization," in *Proc. Int. Conf. Learn. Representations*, 2020. [Online]. Available: https://openreview.net/pdf?id=rke3TJrtPS
[31] T. Xu, Y. Liang, and G. Lan, "CRPO: A new approach for safe reinforcement learning with convergence guarantee," in *Proc. Int. Conf. Mach. Learn.*, 2021, pp. 11480–11491.
[32] S. Kakade and J. Langford, "Approximately optimal approximate reinforcement learning," in *Proc. Int. Conf. Mach. Learn.*, 2002. [Online]. Available: https://people.eecs.berkeley.edu/~pabbeel/cs287-fa09/readings/KakadeLangford-icml2002.pdf
[33] A. Paszke et al., "PyTorch: An imperative style, high-performance deep learning library," in *Proc. Adv. Neural Inf. Process. Syst.*, 2019, pp. 8024–8035.
[34] Z. Wang et al., "Reward-constrained behavior cloning," in *Proc. Int. Joint Conf. Artif. Intell.*, 2021, pp. 3169–3175.
[35] J. Wang, C. Zhao, S. He, Y. Gu, O. Alfarraj, and A. Abugabah, "LogUAD: Log unsupervised anomaly detection based on word2vec," *Comput. Syst. Sci. Eng.*, vol. 41, no. 3, pp. 1207–1222, Jun. 2022.
[36] J. Wang, Y. Yang, T. Wang, R. S. Sherratt, and J. Zhang, "Big data service architecture: A survey," *J. Internet Technol.*, vol. 21, no. 2, pp. 393–405, Mar. 2020.
[37] J. Chen, K. Li, K. Li, P. S. Yu, and Z. Zeng, "Dynamic planning of bicycle stations in dockless public bicycle-sharing system using gated graph neural network," *ACM Trans. Intell. Syst. Technol.*, vol. 12, no. 2, pp. 2501–2522, Mar. 2021.
[38] J. Zhang, S. Zhong, T. Wang, H.-C. Chao, and J. Wang, "Blockchain-based systems and applications: A survey," *J. Internet Technol.*, vol. 21, no. 1, pp. 1–14, Jan. 2020.
[39] C. Chen, K. Li, S. G. Teo, X. Zou, K. Li, and Z. Zeng, "Citywide traffic flow prediction based on multiple gated spatio-temporal convolutional neural networks," *ACM Trans. Knowl. Discov. Data*, vol. 14, no. 4, pp. 4201–4223, May 2020.
[40] J. Wang, Y. Zou, P. Lei, R. S. Sherratt, and L. Wang, "Research on recurrent neural network based crack opening prediction of concrete dam," *J.Internet Technol.*, vol. 21, no. 4, pp. 1161–1169, Jul. 2020.

Ke Lin (Graduate Student Member, IEEE) received the B.E. degree in mechanical engineering from the China University of Geosciences, Wuhan, China, in 2017, and the M.E. degree in mechanical engineering from Shanghai Jiao Tong University, Shanghai, China, in 2020. He is currently working toward the Ph.D. degree in control science and engineering with the Harbin Institute of Technology (Shenzhen), Shenzhen, China.

His research interests include safe reinforcement learning, constrained reinforcement learning, and their applications in various risk-sensitive industrial scenarios.

Duantengchuan Li is currently working toward the Ph.D. degree in software engineering with the School of Computer Science, Wuhan University, Wuhan, China.

His research interests include reinforcement learning, recommendation system, representation learning, natural language processing, large language model, intention recognition, pattern recognition, computer vision, and their applications in software engineering and industry.

Mr. Li has frequently been a Reviewer for serveral international journals, including IEEE TRANSACTIONS ON INDUSTRIAL INFORMATICS, IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS, etc.

Yanjie Li (Member, IEEE) received the B.S. degree from Qingdao University, Qingdao, China, in 2001 and the Ph.D. degree from the University of Science and Technology of China, Hefei, China, in 2006.

From 2006 to 2008, he was a Research Associate with the Department of Electronic and Computer Engineering, Hong Kong University of Science and Technology, Hong Kong. In September 2008, he joined Harbin Institute of Technology Shenzhen (HITSZ), Shenzhen, China. Now, he is an Associate Professor with HITSZ. His research interests include stochastic optimization and learning, Markov decision process (MDP), partially observable MDP and reinforcement learning.

Dr. Li was the recipient of Ho-Pan-Ching-Yi best paper award in 2014.

Shiyu Chen received the B.E. degree in automation from the Anyang Institute of Technology, Anyang, China, in 2014, and the M.E. degree in control science and engineering in 2017 from the Harbin Institute of Technology Shenzhen, Shenzhen, China, where he is currently working toward the Ph.D. degree in control science and engineering.

His research interests include deep reinforcement learning and agile trajectory planning.

Xindong Wu (Fellow, IEEE) received the bachelor's and master's degrees in computer science from the Hefei University of Technology, Hefei, China, and the Ph.D. degree in artificial intelligence from the University of Edinburgh, Edinburgh, U.K., in 1993.

He is a Senior Research Scientist with the Research Center for Knowledge Engineering, Zhejiang Lab, China. His research interests include Big Data analytics, data mining, knowledge engineering, and knowledge-based systems. Dr. Wu was an Editor-in-Chief of IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING (2005–2008) and Knowledge and Information Systems, and is the Steering Committee chair of the IEEE International Conference on Data Mining. He is a Foreign Member of the Russian Academy of Engineering and a Fellow of American Association for the Advancement of Science.