IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 5 April 2024, accepted 9 May 2024, date of publication 17 May 2024, date of current version 24 May 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3402351

RESEARCH ARTICLE

Adversarial Reinforcement Learning Against Statistic Inference on Agent Identity

YUE TIAN, QI JIANG, ZUXING LI, (Member, IEEE), AND CHAO WANG, (Member, IEEE)
School of Electronics and Information Engineering, Tongji University, Shanghai 201804, China

Corresponding author: Zuxing Li (zuxing@tongji.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 62006173 and Grant 62171322, in part by the National Key Research and Development Program of China under Grant 2023YFE0112500, and in part by the 2021–2023 China–Serbia Inter-Governmental Science and Technology (S&T) Cooperation Project under Grant 6.

ABSTRACT This paper considers an agent identity privacy problem in Markov decision process. There are two types of agents with different instantaneous control reward functions, e.g., two types of energy consumption activities in smart grid. An eavesdropper is assumed to intercept the observations of agent and make a statistic inference on the agent identity, which is privacy-sensitive and can be utilized by the eavesdropper to further make corresponding malicious attacks. With regard to the agent identity privacy problem, a privacy-preserving Markov decision process is formulated and a novel adversarial reinforcement learning algorithm is further proposed by exploiting the ideas of deep reinforcement learning and variational method to design the agent policies with the aim to optimally tradeoff improving cumulative control reward and preventing agent identity privacy leakage. Experiments in a modified OpenAI Gym environment show different training process patterns and justify the effectiveness of the proposed algorithm.

INDEX TERMS Kullback-Leibler divergence, Markov decision process, privacy-by-design, variational method.

I. INTRODUCTION

The cyber-physical systems (CPS) employ the computation, communication, and control technologies to achieve the desired performance of physical processes, and pave a way to more efficiently and reliably interact with the physical environment [1], e.g., smart grid [2] and Internet of vehicles (IoV) [3] and [4]. For CPS, a decision-making scheme is optimally designed to process a large sequence of monitoring data and generate a sequence of corresponding actions, which can be modeled as a Markov decision process (MDP) and solved through the model-based optimal control theory [5] or model-free reinforcement learning (RL) [6]. Since the breakthrough of deep reinforcement learning (DRL) in 2015, a large number of DRL algorithms, e.g., deep Q network (DQN) [7], deep deterministic policy gradient (DDPG) [8], twin delayed deep deterministic policy gradient algorithm (TD3) [9], and proximal policy optimization (PPO) [10], have been developed to exploit the strong representative capability of deep neural network (DNN) and the sequential decision-making optimization methods of RL, and have been successively applied to a wide range of CPS. With the evolution of technologies, the efficiency and reliability of CPS have been significantly improved.

Nowadays, CPS have been deployed in highly-complex application scenarios, where a large volume of data is collected, transmitted, and processed. When the data is intercepted, the eavesdropper can infer on privacy-sensitive information via data analysis methods, e.g., detection and estimation theory [11], and deep learning algorithms [12]. Therefore, the research on privacy problems in CPS has attracted increasing attention recently. Smart meter privacy [13] is a typical CPS privacy problem, in which privacy-preserving technologies are briefly introduced as follows. Note that the ideas of those technologies can be applied to most CPS to prevent privacy from leakages.

In smart grid, smart meters monitor the real-time energy data and feed the data back to the energy provider for

The associate editor coordinating the review of this manuscript and approving it for publication was Xiong Luo.

prediction of future energy demand, adjustment of energy production, and adaptive billing. Meanwhile, smart meter privacy problem arises since the energy data can leak the private energy consumption behaviors of user.

Regarding the smart meter privacy problem, obfuscation is a commonly used privacy-preserving method. Many works focused on employing local energy resources, e.g., battery, solar panel, and wind turbine, to distort the smart meter reading from the real energy data to reduce information-theoretic measure of privacy risk, and characterized the utility-privacy trade-off [14], [15], [16], [17], [18].

Cryptography is a classical and widely-used computational-security technology. For smart grid, Reference [19] proposed a privacy-aware authenticated key agreement scheme to secure communication between smart meters and the energy provider based on lightweight cryptography primitives such as one-way hash function. In [20], an authentication scheme using the elliptic curve cryptography was presented for secure communications in smart grid. Recently, the emerging homomorphic encryption [21], [22] is seen as a promising solution to the information security problems since the encrypted data can be directly processed by the service provider without revealing the original data. Based on homomorphic cryptosystem, privacy-preserving energy data aggregation can be realized [23], [24].

The notion of differential privacy (DP) was originally formulated to model the privacy problem of neighboring datasets through query releases [25] and has been applied to CPS [26]. In [27], a DP scheme was proposed to add noise obtained from a virtual chargeable battery to the energy data while maintaining billing accuracy. By jointly utilizing a battery and a renewable energy source with small storage, near Laplace distributed random noise is generated to realize cost-friendly DP of smart meters [28].

Federated learning (FL) [29], [30] provides a privacy-preserving distributed learning framework, where clients share locally-trained models instead of their privacy-sensitive raw data with a server for model aggregation. To enhance privacy, FL is commonly implemented jointly with other privacy-preserving technologies. For instance, Reference [31] applied an inner-product functional encryption scheme to encrypt the local model parameters during the FL to realize privacy-preserving energy prediction. In [32], a differentially private FL-based framework was developed for residential short term load forecasting and ensuring the privacy of the smart meter readings.

From the brief recapitulation, it can be noticed that most efforts have been paid to preserve the privacy in CPS through additive distortion or cryptography.

The rest of this paper is organized as follows. In Section II, the related works and our main contributions are presented. In Section III, the agent identity privacy problem in an MDP is introduced. The considered privacy problem is formulated as a privacy-preserving MDP in Section IV. An adversarial RL algorithm is developed to efficiently optimize the privacy-preserving policy in Section V. In Section VI, experiments are conducted to show the effectiveness of the proposed algorithm. Section VII concludes this paper.

II. RELATED WORKS

As presented in Section I, a diversity of privacy problems have been formulated and studied in CPS from different perspectives. To realize optimal interactions between the cyber system and the physical environment, RL algorithms are implemented as efficient methods to solve the corresponding MDPs, e.g., [33], [34], [35], [36], [37]. Therefore, studying privacy problems in MDP and developing privacy-preserving RL algorithms can provide general and practical solutions for CPS privacy problems. However, not many related works have been reported.

In the literature, the privacy-sensitive information in an MDP model can be the environment state [38], [39], [40] and the reward function of the agent [41], [42]. With regard to the two main types of privacy-sensitive information, privacy-preserving RL have been developed. Homomorphic encryption has been employed to develop privacy-preserving RL algorithms to protect the environment states and rewards [43], [44]. References [45], [46], [47], and [48] formulate the DP problem of sharing privacy-sensitive environment states in a multi-agent control model and study the privacy-preserving control, where each agent applies the DP method, e.g., Laplace mechanism and Gaussian mechanism, to distort the local states before sharing them with other agents while guaranteeing the control system network to operate well. In RL, the value function is closely related with the privacy-sensitive reward function and therefore an eavesdropper can infer on the reward function by intercepting the learned value function. Regarding this privacy problem, the differentially private Q-learning algorithm [49] adds functional noise to the value function such that the neighbor reward functions become indistinguishable.

Although [43], [44], [45], [46], [47], [48], and [49] study the privacy problems in the MDP context, the proposed privacy-preserving RL algorithms mainly rely on the additive DP noise or cryptography mechanism. Due to the stricter requirements of privacy, e.g., GDPR in Europe [50], the principle of privacy-by-design has attracted increasing attention in recent years. Regarding privacy problem in MDP, privacy-by-design means that privacy-preserving RL algorithm can optimize the control policy with both objectives of improving reward and preserving privacy.

In this paper, we consider a quite different privacy problem in an MDP, where the agent has two candidates and the agent identity is the privacy-sensitive information. Different from the existing works to exploit the DP and cryptography methods, we model the agent identity privacy problem as a statistic inference attack, employ an information theoretic privacy measure, formulate a privacy-preserving MDP, and develop a novel adversarial RL algorithm from the privacy-by-design perspective, which can efficiently solve

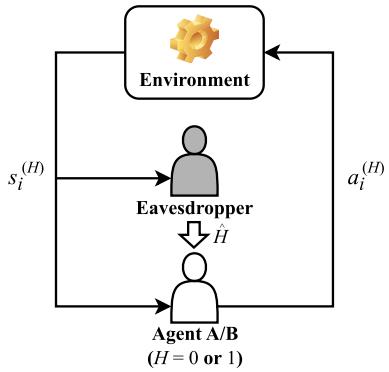


FIGURE 1. Agent identity privacy leakage through intercepted environment states in an MDP.

the optimal privacy-preserving policies. The effectiveness of the proposed adversarial RL algorithm is justified in a modified OpenAI Gym environment. In our previous work [51], we consider the agent identity privacy problem in a linear quadratic Gaussian control and derive closed-form expressions of the privacy-preserving policies. Here we extend the previous work to study the agent identity privacy problem in a general MDP context.

III. AGENT IDENTITY PRIVACY PROBLEM IN MARKOV DECISION PROCESS

The considered privacy problem is shown in Figure 1. In the MDP, the agent identity is a binary hypothesis H, which can be Agent A corresponding to $H = 0$, or Agent B corresponding to $H = 1$. Let $\mathcal{S}$ denote the environment state set and $\mathcal{A}$ denote the action set of both agents. For the environment, $P_{\mathrm{T}} : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to \mathbb{R}$ represents the state transition probability density or mass function. The major difference between the agents lays in their instantaneous control reward functions. Let $r_0 : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to \mathbb{R}$ denote the instantaneous control reward function of Agent A and $r_1 : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to \mathbb{R}$ denote the instantaneous control reward function of Agent B. For both agents, $0 < \gamma \leq 1$ represents the discount factor of the instantaneous control reward. Therefore, the MDP can be represented by two alternative tuples: $\langle \mathcal{S}, \mathcal{A}, P_{\mathrm{T}}, r_0, \gamma \rangle$ when Agent A is present, and $\langle \mathcal{S}, \mathcal{A}, P_{\mathrm{T}}, r_1, \gamma \rangle$ when Agent B is present. Under hypothesis H, the N-step MDP operates as follows: At Step i, the agent observes the environment state $s_i^{(H)} \in \mathcal{S}$, makes a decision of action $a_i^{(H)} \in \mathcal{A}$, feeds the action back to the environment, which evolves to the next state $s_{i+1}^{(H)} \in \mathcal{S}$ with the probability density or mass $P_{\mathrm{T}}\left(s_{i+1}^{(H)} \middle| s_i^{(H)}, a_i^{(H)}\right)$, and receives a reward $r_i^{(H)} = r_H\left(s_i^{(H)}, a_i^{(H)}, s_{i+1}^{(H)}\right)$.

We assume that an eavesdropper intercepts the observations of agent, i.e., the environment states $s_{1:N+1}^{(H)} := \left(s_1^{(H)}, s_2^{(H)}, \ldots, s_{N+1}^{(H)}\right)$, and makes an adversarial binary hypothesis testing on the agent identity $\hat{H}$. To evaluate the privacy risk, we employ the following Kullback-Leibler (KL) divergence:

$$D\left(P_{S_{1:N+1}^{(1)}} \middle\| P_{S_{1:N+1}^{(0)}}\right) := E\left[\log \frac{P_{S_{1:N+1}^{(1)}}\left(S_{1:N+1}^{(1)}\right)}{P_{S_{1:N+1}^{(0)}}\left(S_{1:N+1}^{(1)}\right)}\right]. \quad (1)$$

From the information theoretic perspective, a larger value of $D\left(P_{S_{1:N+1}^{(1)}} \middle\| P_{S_{1:N+1}^{(0)}}\right)$ means that the random state sequences $S_{1:N+1}^{(0)}$ and $S_{1:N+1}^{(1)}$ induced by Agent A and Agent B are more statistically different, and therefore the adversarial hypothesis testing on the agent identity can be more precise, i.e., there is a higher privacy risk.

Remark 1. *Agent identity privacy problem exists in many CPS. Here we take a smart meter privacy scenario as an example, where the smart meter reading of energy data (environment state) can be generated due to user activity (Agent A) or without user activity (Agent B). An eavesdropper intercepts the energy data, infers on if the user is at home, and decides the next adversarial behaviors.*

IV. PRIVACY-PRESERVING MARKOV DECISION PROCESS

We assume that Agent A is privacy-unaware while Agent B is privacy-aware. This assumption is reasonable in practice. In the smart meter privacy example, the user activity (Agent A) does not need to change while the energy data generated without user activity (Agent B) can be reconfigured such that the eavesdropper cannot precisely determine if the user is at home based on the intercepted energy data.

Let $\phi_i^{(0)} : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ denote the random policy of Agent A at Step i. Agent A optimizes policies $\phi_{1:N}^{(0)} := \left(\phi_1^{(0)}, \phi_2^{(0)}, \ldots, \phi_N^{(0)}\right)$ with the aim to maximize cumulative control reward as

$$\phi_{1:N}^{(0)*} = \arg\max_{\phi_{1:N}^{(0)}} E\left[\sum_{i=1}^{N} \gamma^{i-1} r_0\left(S_i^{(0)}, A_i^{(0)}, S_{i+1}^{(0)}\right)\right].$$

The optimal policies $\phi_{1:N}^{(0)*}$ can be efficiently solved by using the established DRL algorithms. When Agent A employs the optimal policies, we denote the induced random environment states by $S_{1:N+1}^{(0)*}$.

Let $\phi_i^{(1)} : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ denote the random policy of Agent B at Step i. Due to the privacy-aware assumption, Agent B optimizes policies $\phi_{1:N}^{(1)} := \left(\phi_1^{(1)}, \phi_2^{(1)}, \ldots, \phi_N^{(1)}\right)$ with two objectives: maximizing cumulative control reward and minimizing privacy risk. We formulate a privacy-preserving MDP for Agent B, which can be represented by the tuple $\langle \mathcal{S}, \mathcal{A}, P_{\mathrm{T}}, r_1, \gamma, \lambda \rangle$ with an additional privacy-preserving weight $\lambda \geq 0$. Specifically, Agent B optimizes policies $\phi_{1:N}^{(1)}$ to maximize the following weighted-sum objective as

$$\phi_{1:N}^{(1)*} = \arg\max_{\phi_{1:N}^{(1)}} E\left[\sum_{i=1}^{N} \gamma^{i-1} r_1\left(S_i^{(1)}, A_i^{(1)}, S_{i+1}^{(1)}\right)\right] - \lambda D\left(P_{S_{1:N+1}^{(1)}} \middle\| P_{S_{1:N+1}^{(0)*}}\right). \quad (2)$$

As the privacy-preserving weight λ increases, Agent B concerns more about reducing the privacy risk, and vice versa. In the special case of $\lambda = 0$, the design of policies $\phi_{1:N}^{(1)*}$ only aims to maximize the cumulative control reward of Agent B. The other special case with $\lambda \to \infty$ means that the design of policies $\phi_{1:N}^{(1)*}$ only aims to minimize the privacy risk. Actually, $\phi_{1:N}^{(1)*} = \phi_{1:N}^{(0)*}$ is an optimal solution of Agent B in the second special case since both agents adopting the same policies leads to identical probability distributions as $P_{S_{1:N+1}^{(1)*}} = P_{S_{1:N+1}^{(0)*}}$ and achieves the minimum value of the non-negative KL divergence term as $D\left(P_{S_{1:N+1}^{(1)*}} \middle|\middle| P_{S_{1:N+1}^{(0)*}}\right) = 0$.

V. ADVERSARIAL REINFORCEMENT LEARNING

In the following, we focus on developing an adversarial RL algorithm for the privacy-preserving MDP of Agent B with non-discounted instantaneous control rewards, i.e., $\gamma = 1$, a finite discrete action set, i.e., $||\mathcal{A}|| < \infty$, and deterministic policy, i.e., $\phi_i^{(H)} : \mathcal{S} \to \mathcal{A}$. The idea and algorithm in this work can be extended to more general cases.

A. MODIFIED PRIVACY-PRESERVING MARKOV DECISION PROCESS

To solve the optimal privacy-preserving policies by adversarial RL, the weighted-sum objective (2) of Agent B should be expressed as a cumulative formulation of instantaneous privacy-preserving rewards. Based on the chain rule of KL divergence [52, Theorem 2.5.3], the privacy risk term can be equivalently decomposed as

$$\begin{aligned} &D\left(P_{S_{1:N+1}^{(1)}} \middle|\middle| P_{S_{1:N+1}^{(0)*}}\right) \\ &= D\left(P_{S_1^{(1)}} \middle|\middle| P_{S_1^{(0)}}\right) + \sum_{i=1}^{N} D\left(P_{S_{i+1}^{(1)}|S_i^{(1)}} \middle|\middle| P_{S_{i+1}^{(0)*}|S_i^{(0)*}}\right) \\ &= \sum_{i=1}^{N} D\left(P_{S_{i+1}^{(1)}|S_i^{(1)}} \middle|\middle| P_{S_{i+1}^{(0)*}|S_i^{(0)*}}\right) \\ &= \sum_{i=1}^{N} E\left[\log \frac{P_{S_{i+1}^{(1)}|S_i^{(1)}}\left(S_{i+1}^{(1)}|S_i^{(1)}\right)}{P_{S_{i+1}^{(0)*}|S_i^{(0)*}}\left(S_{i+1}^{(1)}|S_i^{(1)}\right)}\right], \end{aligned} \tag{3}$$

where the second equality holds by assuming that the random initial state is independent of the agent identity, i.e., $p_{S_1^{(0)}} = p_{S_1^{(1)}}$ and $D\left(P_{S_1^{(1)}} \middle|\middle| P_{S_1^{(0)}}\right) = 0$. Then the weighted-sum objective in (2) with $\gamma = 1$ can be rewritten as the cumulative privacy-preserving reward:

$$\begin{aligned} &E\left[\sum_{i=1}^{N} r_1\left(S_i^{(1)}, A_i^{(1)}, S_{i+1}^{(1)}\right)\right] - \lambda D\left(P_{S_{1:N+1}^{(1)}} \middle|\middle| P_{S_{1:N+1}^{(0)*}}\right) \\ &\quad = E\left[\sum_{i=1}^{N} r_{1,\lambda}\left(S_i^{(1)}, A_i^{(1)}, S_{i+1}^{(1)}\right)\right], \end{aligned}$$

where the instantaneous privacy-preserving reward of Agent B at Step i is defined as

$$\begin{aligned} &r_{1,\lambda}\left(s_i^{(1)}, a_i^{(1)}, s_{i+1}^{(1)}\right) \\ &\quad = r_1\left(s_i^{(1)}, a_i^{(1)}, s_{i+1}^{(1)}\right) - \lambda \log \frac{P_{S_{i+1}^{(1)}|S_i^{(1)}}\left(s_{i+1}^{(1)}|s_i^{(1)}\right)}{P_{S_{i+1}^{(0)*}|S_i^{(0)*}}\left(s_{i+1}^{(1)}|s_i^{(1)}\right)}. \end{aligned} \tag{4}$$

Therefore, the privacy-preserving MDP can be equivalently represented by the tuple $\langle \mathcal{S}, \mathcal{A}, P_T, r_{1,\lambda}, \gamma = 1\rangle$. Since $r_{1,\lambda}$ is a function of the conditional probability densities/masses $P_{S_{i+1}^{(1)}|S_i^{(1)}}$ and $P_{S_{i+1}^{(0)*}|S_i^{(0)*}}$, it is impossible to evaluate the instantaneous privacy-preserving reward in the adversarial RL, where the knowledge of the statistical models is not available. A modified instantaneous privacy-preserving reward, which can be evaluated without the knowledge of probability models, is needed.

In [53], the authors studied estimation of f-divergence as solving an equivalent Bayes decision problem by variational methods. Note that KL divergence is a special case of f-divergence with regard to the convex function $f(u) := u \log u$. Based on [53], we give a lower bound on the conditional KL divergence $D\left(P_{S_{i+1}^{(1)}|S_i^{(1)}} \middle|\middle| P_{S_{i+1}^{(0)*}|S_i^{(0)*}}\right)$ in terms of random environment states.

Theorem 1. *A lower bound on the conditional KL divergence term can be*

$$\begin{aligned} &D\left(P_{S_{i+1}^{(1)}|S_i^{(1)}} \middle|\middle| P_{S_{i+1}^{(0)*}|S_i^{(0)*}}\right) \\ &\quad \geq \sup_{\theta_i} E_{\sim P_{S_i^{(1)}}}\left[E_{\sim P_{S_{i+1}^{(1)}|S_i^{(1)}}}\left[\theta_i\left(S_{i+1}^{(1)}\right)\right]\right] \\ &\qquad - E_{\sim P_{S_i^{(1)}}}\left[E_{\sim P_{S_{i+1}^{(0)*}|S_i^{(0)*}}}\left[\exp\left(\theta_i\left(S_{i+1}^{(0)*}\right) - 1\right)\right]\right], \end{aligned} \tag{5}$$

where $\theta_i : \mathcal{S} \to \mathbb{R}$ is a discriminator function at Step i.

Proof: Given $s \in \mathcal{S}$, it follows from [53] that

$$\begin{aligned} &D\left(P_{S_{i+1}^{(1)}|S_i^{(1)}=s} \middle|\middle| P_{S_{i+1}^{(0)*}|S_i^{(0)*}=s}\right) \\ &\quad = E_{\sim P_{S_{i+1}^{(1)}|S_i^{(1)}=s}}\left[\log \frac{P_{S_{i+1}^{(1)}|S_i^{(1)}=s}\left(S_{i+1}^{(1)}\right)}{P_{S_{i+1}^{(0)*}|S_i^{(0)*}=s}\left(S_{i+1}^{(1)}\right)}\right] \\ &\quad \geq \sup_{\theta_{i,s}} E_{\sim P_{S_{i+1}^{(1)}|S_i^{(1)}=s}}\left[\theta_{i,s}\left(S_{i+1}^{(1)}\right)\right] \\ &\qquad - E_{\sim P_{S_{i+1}^{(0)*}|S_i^{(0)*}=s}}\left[\exp\left(\theta_{i,s}\left(S_{i+1}^{(0)*}\right) - 1\right)\right], \end{aligned} \tag{6}$$

where $\theta_{i,s} : \mathcal{S} \to \mathbb{R}$ is a discriminator function at Step i and by assuming the environment states of agents $S_i^{(1)} = S_i^{(0)*} = s$. From the definition of conditional KL divergence

and the inequality (6), we have

$$
\begin{aligned}
&D\left(P_{S_{i+1}^{(1)}|S_i^{(1)}} \middle|\middle| P_{S_{i+1}^{(0)*}|S_i^{(0)*}}\right) \\
&= \int_{\mathcal{S}} P_{S_i^{(1)}}(s) D\left(P_{S_{i+1}^{(1)}|S_i^{(1)}=s} \middle|\middle| P_{S_{i+1}^{(0)*}|S_i^{(0)*}=s}\right) ds \\
&\geq \int_{\mathcal{S}} P_{S_i^{(1)}}(s) \Bigg[\sup_{\theta_{i,s}} E_{\sim P_{S_{i+1}^{(1)}|S_i^{(1)}=s}} \left[\theta_{i,s}\left(S_{i+1}^{(1)}\right)\right] \\
&\qquad - E_{\sim P_{S_{i+1}^{(0)*}|S_i^{(0)*}=s}} \left[\exp\left(\theta_{i,s}\left(S_{i+1}^{(0)*}\right) - 1\right)\right] \Bigg] ds \\
&\geq \sup_{\theta_i} \int_{\mathcal{S}} P_{S_i^{(1)}}(s) \left[E_{\sim P_{S_{i+1}^{(1)}|S_i^{(1)}=s}} \left[\theta_i\left(S_{i+1}^{(1)}\right)\right] \right. \\
&\qquad \left. - E_{\sim P_{S_{i+1}^{(0)*}|S_i^{(0)*}=s}} \left[\exp\left(\theta_i\left(S_{i+1}^{(0)*}\right) - 1\right)\right] \right] ds \\
&= \sup_{\theta_i} E_{\sim P_{S_i^{(1)}}} \left[E_{\sim P_{S_{i+1}^{(1)}|S_i^{(1)}}} \left[\theta_i\left(S_{i+1}^{(1)}\right)\right] \right] \\
&\qquad - E_{\sim P_{S_i^{(1)}}} \left[E_{\sim P_{S_{i+1}^{(0)*}|S_i^{(0)*}}} \left[\exp\left(\theta_i\left(S_{i+1}^{(0)*}\right) - 1\right)\right] \right].
\end{aligned} \tag{7}
$$

■

Note that the proposed lower bound on the conditional KL divergence $D\left(P_{S_{i+1}^{(1)}|S_i^{(1)}=s} \middle|\middle| P_{S_{i+1}^{(0)*}|S_i^{(0)*}=s}\right)$ does not depend on the conditional probability densities/masses $P_{S_{i+1}^{(1)}|S_i^{(1)}}$ and $P_{S_{i+1}^{(0)*}|S_i^{(0)*}}$, and can be evaluated by the observations of environment states through the Monte Carlo (MC) method or temporal difference (TD) method. On the other hand, the proposed lower bound needs optimization of the discriminator θ_i, which can be approximately represented by a DNN and optimized through gradient ascent. Therefore, we employ the proposed lower bound as an estimation of the conditional KL divergence term and formulate a modified privacy-preserving MDP, where Agent B optimizes policies to maximize the following modified cumulative objective:

$$
\begin{aligned}
\phi_{1:N}^{(1)*} = \arg\max_{\phi_{1:N}^{(1)}} \inf_{\theta_{1:N}} \sum_{i=1}^{N} & E_{\sim P_{S_i^{(1)}}} \Bigg[- \lambda E_{\sim P_{S_{i+1}^{(1)}|S_i^{(1)}}} \left[\theta_i\left(S_{i+1}^{(1)}\right)\right] \\
& + \lambda E_{\sim P_{S_{i+1}^{(0)*}|S_i^{(0)*}}} \left[\exp\left(\theta_i\left(S_{i+1}^{(0)*}\right) - 1\right)\right] \\
& + E_{\sim P_{A_i^{(1)}, S_{i+1}^{(1)}|S_i^{(1)}}} \left[r_1\left(S_i^{(1)}, A_i^{(1)}, S_{i+1}^{(1)}\right)\right] \Bigg].
\end{aligned} \tag{8}
$$

Focusing on the modified privacy-preserving MDP, we propose a novel adversarial reinforcement learning algorithm in the next sub-section. Note that the modified cumulative objective depends on the optimal discriminators. The adversarial reinforcement learning algorithm should efficiently solve the optimal privacy-preserving policies of Agent B as well as the optimal discriminators, which are players in a dynamic max-min game as shown in (8).

B. PRIVACY-PRESERVING DEEP Q NETWORK

Regarding the dynamic max-min game (8), it cannot be solved by directly implementing DRL algorithms for standard MDP problems. The max-min or min-max games have been considered in many generative models, e.g., generative adversarial network (GAN) [54], and the common idea of those works is to obtain an approximate solution of a max-min or min-max game through iterative maximization and minimization of the design objective. It is worth noting that the iterative optimization scheme does not necessarily lead to the solution of the original max-min or min-max game. Here, we employ the iterative optimization idea and exploit the sequential decision optimization method of DQN to propose a novel privacy-preserving DQN (PPDQN) algorithm and efficiently solve the modified privacy-preserving MDP problem.

The PPDQN model consists of three DNNs: the discriminator network $\theta(\cdot;\vartheta)$ with parameters ϑ, the action-value network of Agent B $Q^{(1)}(\cdot,\cdot;\varphi)$ with parameters φ, and the target action-value network $\hat{Q}(\cdot,\cdot;\hat{\varphi})$ with parameters $\hat{\varphi}$, where $Q^{(1)}(\cdot,\cdot;\varphi)$ and $\hat{Q}(\cdot,\cdot;\hat{\varphi})$ have the same network structures. The diagram of the PPDQN algorithm is shown in Figure 2 and the pesudocode is presented in Algorithm 1. Based on the iterative optimization idea, the PPDQN algorithm consists of two key steps: update of discriminator network and update of action-value network of Agent B. To aid the update of networks, data sampling is carried out.

1) UPDATE OF DISCRIMINATOR NETWORK

Given the policies of Agent B $\phi_{1:N}^{(1)}$, the discriminator functions are updated with respect to the following minimization objective:

$$
\begin{aligned}
\inf_{\theta_{1:N}} \sum_{i=1}^{N} E_{\sim P_{S_i^{(1)}}} \Bigg[& - E_{\sim P_{S_{i+1}^{(1)}|S_i^{(1)}}} \left[\theta_i\left(S_{i+1}^{(1)}\right)\right] \\
& + E_{\sim P_{S_{i+1}^{(0)*}|S_i^{(0)*}}} \left[\exp\left(\theta_i\left(S_{i+1}^{(0)*}\right) - 1\right)\right] \Bigg].
\end{aligned} \tag{9}
$$

In PPDQN, the discriminator network $\theta(\cdot;\vartheta)$ represents the discriminator functions. Given a mini-batch of K randomly-sampled transitions $\left\{\left(s_j^{(1)}, a_j^{(1)}, r_j^{(1)}, s_{j+1}^{(1)}, s_{j+1}^{(0)*}\right)\right\}_{j\in\mathcal{B}_i}$, similar to GAN, the discriminator network is then updated by performing gradient descents for T times on the following loss function of the network parameters ϑ:

$$
J(\vartheta) = -\frac{1}{K} \sum_{j\in\mathcal{B}_i} \left(\theta\left(s_{j+1}^{(1)};\vartheta\right) - \exp\left(\theta\left(s_{j+1}^{(0)*};\vartheta\right) - 1\right)\right). \tag{10}
$$

Note that the loss function $J(\vartheta)$ in the PPDQN algorithm approximates the objective of (9) through MC method.

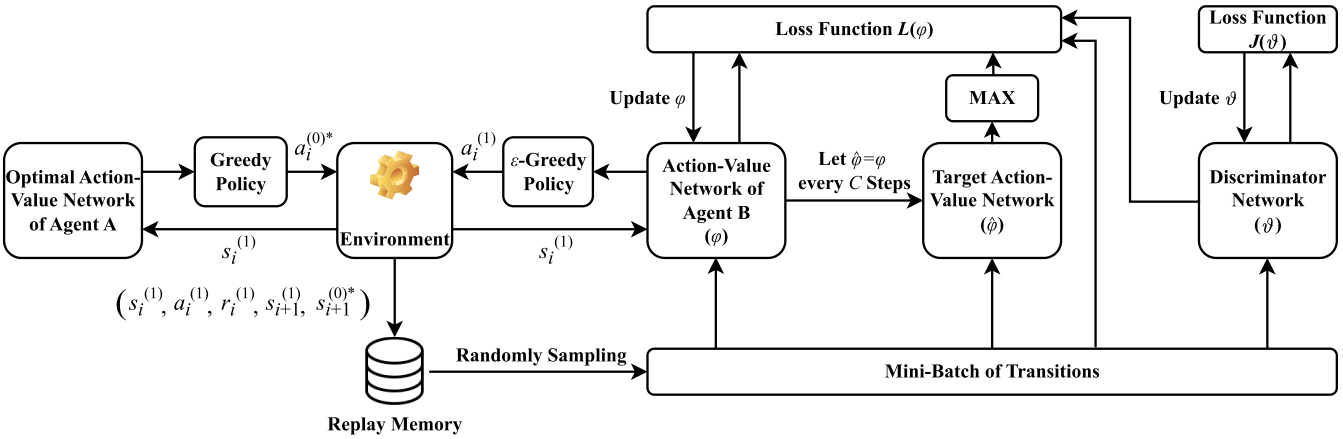


FIGURE 2. Diagram of the PPDQN algorithm.

2) UPDATE OF ACTION-VALUE NETWORK OF AGENT B

Given the discriminator functions $\theta_{1:N}$, the privacy-preserving policies of Agent B are updated with respect to the following maximization objective:

$$\max_{\phi_{1:N}^{(1)}} \sum_{i=1}^{N} E_{\sim P_{S_i^{(1)}}}\Big[-\lambda E_{\sim P_{S_{i+1}^{(1)}|S_i^{(1)}}}\left[\theta_i\left(S_{i+1}^{(1)}\right)\right] + \lambda E_{\sim P_{S_{i+1}^{(0)*}|S_i^{(0)*}}}\left[\exp\left(\theta_i\left(S_{i+1}^{(0)*}\right)-1\right)\right] + E_{\sim P_{A_i^{(1)},S_{i+1}^{(1)}|S_i^{(1)}}}\left[r_1\left(S_i^{(1)}, A_i^{(1)}, S_{i+1}^{(1)}\right)\right]\Big]. \tag{11}$$

The optimization problem (11) corresponds to a standard MDP represented by the tuple $\langle \mathcal{S}, \mathcal{A}, P_{\mathrm{T}}, r'_{1,\lambda}, \gamma = 1\rangle$, and the instantaneous reward $r'_{1,\lambda}$ can be identified in (11) as

$$r'_{1,\lambda}\left(s_i^{(1)}, a_i^{(1)}, s_{i+1}^{(1)}, s_{i+1}^{(0)*}\right) = r_1\left(s_i^{(1)}, a_i^{(1)}, s_{i+1}^{(1)}\right) - \lambda\theta_i\left(s_{i+1}^{(1)}\right) + \lambda\exp\left(\theta_i\left(s_{i+1}^{(0)*}\right)-1\right), \tag{12}$$

where the next environment states $s_{i+1}^{(0)*}$ and $s_{i+1}^{(1)}$ are respectively obtained by feeding the optimal action of Agent A $a_i^{(0)*} = \phi_i^{(0)*}(s_i^{(0)*})$ and action of Agent B $a_i^{(1)} = \phi_i^{(1)}(s_i^{(1)})$ back to the environment given the current environment states $s_i^{(0)*} = s_i^{(1)}$.

The DQN algorithm is applicable for solving the optimization problem (11) and therefore is implemented in the PPDQN algorithm to update the privacy-preserving policy of Agent B when the discriminator network is fixed. Regarding the problem (11), the main idea of DQN is briefly introduced as follows. In the asymptotic regime, given an initial state $s \in \mathcal{S}$ and an initial action $a \in \mathcal{A}$, define the optimal action-value function $Q^{(1)*}(s, a)$ as the maximum cumulative objective of Agent B by employing the optimal privacy-preserving policies from the second step. It follows from the Bellman equation [6] that

$$Q^{(1)*}(s,a) = \int_{\mathcal{S}\times\mathcal{S}} P_{\mathrm{T}}\left(s'|s,a\right) P_{\mathrm{T}}\left(s''\,\middle|\, s, \phi_i^{(0)*}(s)\right) \left[r'_{1,\lambda}\left(s,a,s',s''\right) + \max_{a'\in\mathcal{A}} Q^{(1)*}\left(s',a'\right)\right] ds' ds''. \tag{13}$$

Note that the optimal privacy-preserving policy of Agent B can be implemented as $\phi_i^{(1)*}(s) = \arg\max_{a\in\mathcal{A}} Q^{(1)*}(s,a)$. Therefore, the DQN algorithm aims to obtain the optimal action-value function instead of solving the optimal policy directly.

In the PPDQN algorithm, the action-value network $Q^{(1)}(\cdot,\cdot;\varphi)$ represents the action-value function of Agent B, and the target action-value network $\hat{Q}(\cdot,\cdot;\hat{\varphi})$ represents the target action-value function. Given a mini-batch of randomly-sampled transitions $\left\{\left(s_j^{(1)}, a_j^{(1)}, r_j^{(1)}, s_{j+1}^{(1)}, s_{j+1}^{(0)*}\right)\right\}_{j\in\mathcal{B}_i}$ and the discriminator network, the instantaneous rewards are estimated based on (12) as: For $j \in \mathcal{B}_i$,

$$r'_j = r_j^{(1)} - \lambda\theta\left(s_{j+1}^{(1)};\vartheta\right) + \lambda\exp\left(\theta\left(s_{j+1}^{(0)*};\vartheta\right)-1\right). \tag{14}$$

Similar to DQN, the action-value network of Agent B is then updated by performing gradient descent on the following loss function of the network parameters φ:

$$L(\varphi) = \sum_{j\in\mathcal{B}_i}\left(r'_j + \max_{a\in\mathcal{A}}\hat{Q}\left(s_{j+1}^{(1)}, a;\hat{\varphi}\right) - Q^{(1)}\left(s_j^{(1)}, a_j^{(1)};\varphi\right)\right)^2. \tag{15}$$

Intuitively, $r'_j + \max_{a\in\mathcal{A}}\hat{Q}\left(s_{j+1}^{(1)}, a;\hat{\varphi}\right)$ can be seen as the label of $Q^{(1)}\left(s_j^{(1)}, a_j^{(1)};\varphi\right)$.

3) UPDATE OF TARGET ACTION-VALUE NETWORK

After every C updates of the action-value network of Agent B, the updated action-value network parameters are assigned to the target action-value network parameters, i.e., $\hat{\varphi} = \varphi$.

4) DATA SAMPLING

The update of discriminator network and action-value network of Agent B needs sampling of environment states, actions, and instantaneous control rewards. The data sampling of the PPDQN algorithm is carried out as follows. At Step i, Agent B observes the environment state $s_i^{(1)}$ and selects an action $a_i^{(1)}$ based on the current action-value network $Q^{(1)}(\cdot,\cdot;\varphi)$ using the ε-greedy algorithm. Agent B then executes the selected action $a_i^{(1)}$ in emulator, observes the next environment state $s_{i+1}^{(1)}$, and receives the instantaneous control reward $r_i^{(1)}$. Agent A also observes the same environment state $s_i^{(1)}$ and selects an action $a_i^{(0)*}$ based on the optimal action-value function $Q^{(0)*}$ using the greedy algorithm. Agent A then executes the selected action $a_i^{(0)*}$ in emulator and observes the next environment state $s_{i+1}^{(0)*}$. The observations at this step form a transition $\left(s_i^{(1)}, a_i^{(1)}, r_i^{(1)}, s_{i+1}^{(1)}, s_{i+1}^{(0)*}\right)$, which is stored in a replay memory.

5) TIME COMPLEXITY ANALYSIS

We make an analysis on the time complexity of the proposed PPDQN algorithm. Assume that the discriminator network consists of K_d fully-connected layers and the i-th layer contains $N_{d,i}$ neurons; the action-value network of Agent B consists of K_b fully-connected layers and the j-th layer contains $N_{b,j}$ neurons. In the training phase of the PPDQN algorithm, the time complexity is dominated by the updates of the discriminator network and the action-value network of Agent B. As shown in Algorithm 1, at every training step, the discriminator network is updated for T times while the action-value network of Agent B is updated for one time. In the inference phase of the PPDQN algorithm, only the action-value network of Agent B is operated to determine the optimal actions of Agent B. Following from [55], the time complexity of the training phase is

$$C_{\text{training}} = \mathcal{O}\left(T \sum_{2\le i\le K_d-1} N_{d,i-1}N_{d,i} + N_{d,i}N_{d,i+1} + \sum_{2\le j\le K_b-1} N_{b,j-1}N_{b,j} + N_{b,j}N_{b,j+1}\right), \quad (16)$$

and the time complexity of the inference phase is

$$C_{\text{inference}} = \mathcal{O}\left(\sum_{2\le j\le K_b-1} N_{b,j-1}N_{b,j} + N_{b,j}N_{b,j+1}\right). \quad (17)$$

We take the DQN algorithm as a benchmark method. Note that the DQN model consists of two DNNs with identical network structures: the action-value network and the target action-value network. We assume that the action-value network of the DQN model has the same network structure as the action-value network of the PPDQN model. In the training phase of the DQN algorithm, the time complexity is dominated by the updates of the action-value network. In the inference phase of the DQN algorithm, only the action-value network is operated to determine the optimal actions. Following from [55], the training and inference phases of the DQN algorithm have the same level of time complexity as

$$C'_{\text{training}} = C'_{\text{inference}} = \mathcal{O}\left(\sum_{2\le j\le K_b-1} N_{b,j-1}N_{b,j} + N_{b,j}N_{b,j+1}\right). \quad (18)$$

In the training phase, the PPDQN algorithm has a higher time complexity than that of the DQN algorithm, which is due to the updates of the discriminator network in the PPDQN algorithm. In the inference phase, the PPDQN and DQN algorithms have the same level of time complexity because only the action-value network is operated for each algorithm. The comparison reveals the cost of time complexity to train privacy-preserving policy.

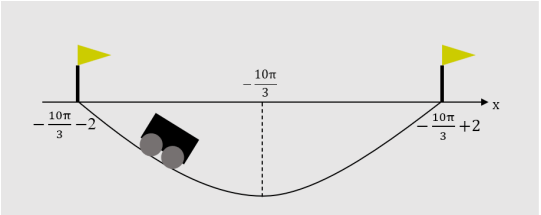


FIGURE 3. The modified "Mountain Car" game environment.

VI. EXPERIMENTS

We implement the proposed PPDQN algorithm in the OpenAI Gym environment [56] and justify its effectiveness to improve the cumulative control reward while preventing agent identity privacy leakage.

A. EXPERIMENT SETTINGS

As shown in Figure 3, the experiments are done in a modified "Mountain Car" game environment, where a car is moving in a sinusoidal valley. The symmetric valley spans the range $[-\frac{10\pi}{3}-2, -\frac{10\pi}{3}+2]$ in the x-axis, i.e., coordinates of the center (bottom), the left boundary, and the right boundary are $-\frac{10\pi}{3}$, $-\frac{10\pi}{3}-2$, and $-\frac{10\pi}{3}+2$ in the x-axis, respectively. At each step, the environment state is a two-dimensional vector $s_i^{(H)} := \left(x_i^{(H)}, v_i^{(H)}\right)$, which consists of the coordinate of the car in the x-axis, i.e., $-\frac{10\pi}{3}-2 \le x_i^{(H)} \le -\frac{10\pi}{3}+2$, and the car velocity with respect to the x-axis $-0.7 \le v_i^{(H)} \le 0.7$. At the initial step, the car is randomly placed on the left slope of the sinusoidal valley with the coordinate $-\frac{10\pi}{3} - 1 \le x_1^{(H)} \le -\frac{10\pi}{3} - \frac{3}{4}$ and has a zero initial velocity

Algorithm 1 Privacy-Preserving Deep Q Network

Initialize replay memory $\mathcal{D}$
Initialize discriminator network $\theta(\cdot;\vartheta)$ with random network parameters ϑ
Initialize action-value network of Agent B $Q^{(1)}(\cdot,\cdot;\varphi)$ with random network parameters φ
Initialize target action-value network $\hat{Q}(\cdot,\cdot;\hat{\varphi})$ with network parameters $\hat{\varphi}=\varphi$
Optimize action-value function of Agent A $Q^{(0)*}$ using the DQN algorithm
for episode $= 1, 2, \ldots, M$ **do**
  Randomly initialize $s_1^{(1)} \in \mathcal{S}$
  for step $i = 1, 2, \ldots, N$ **do**
    With probability ε, Agent B selects an action $a_i^{(1)}$ uniformly and randomly from $\mathcal{A}$
    Otherwise, Agent B selects the action $a_i^{(1)} = \arg\max_{a\in\mathcal{A}} Q^{(1)}\left(s_i^{(1)}, a; \varphi\right)$
    Given $s_i^{(1)}$, Agent B executes the action $a_i^{(1)}$ in emulator, observes the next state $s_{i+1}^{(1)}$, and receives the instantaneous control reward $r_i^{(1)}$
    Agent A selects the action $a_i^{(0)*} = \arg\max_{a\in\mathcal{A}} Q^{(0)*}\left(s_i^{(1)}, a\right)$
    Given $s_i^{(1)}$, Agent A executes the action $a_i^{(0)*}$ in emulator and observes the next state $s_{i+1}^{(0)*}$
    Store transition $\left(s_i^{(1)}, a_i^{(1)}, r_i^{(1)}, s_{i+1}^{(1)}, s_{i+1}^{(0)*}\right)$ in $\mathcal{D}$
    Randomly sample a mini-batch $\mathcal{B}_i$ of K transitions $\left\{\left(s_j^{(1)}, a_j^{(1)}, r_j^{(1)}, s_{j+1}^{(1)}, s_{j+1}^{(0)*}\right)\right\}_{j\in\mathcal{B}_i}$ from $\mathcal{D}$
    For $j \in \mathcal{B}_i$, evaluate the instantaneous reward r_j'
    Update the discriminator network parameters ϑ by performing gradient descents on the loss function $J(\vartheta)$ for T times
    Update the action-value network parameters φ by performing gradient descent on the loss function $L(\varphi)$
    Set $\hat{\varphi}=\varphi$ every C steps
  end for
end for

$v_1^{(H)} = 0$. At each step, there are three action options for the car: acceleration to the left denoted by $a_i^{(H)} = 0$, no acceleration denoted by $a_i^{(H)} = 1$, and acceleration to the right denoted by $a_i^{(H)} = 2$. The environment dynamics are described as follows.

$$\check{v}_{i+1}^{(H)} = v_i^{(H)} + \left(a_i^{(H)} - 1\right) f - \cos\left(0.15 x_i^{(H)}\right) g + z_i,$$
$$\tilde{v}_{i+1}^{(H)} = \max\left\{\min\left\{\check{v}_{i+1}^{(H)}, 0.7\right\}, -0.7\right\},$$
$$\tilde{x}_{i+1}^{(H)} = x_i^{(H)} + \tilde{v}_{i+1}^{(H)},$$

where z_i is randomly and independently generated following a Gaussian distribution $\mathcal{N}(0, 0.0003)$ to guarantee the validity of the KL divergence terms, $f = 0.6$ denotes the force, and $g = 0.75$ denotes the gravity. It can be observed that the gravity does not affect the car velocity at the bottom of the valley. When $\tilde{x}_{i+1}^{(H)} \le -\frac{10\pi}{3} - 2$ or $\tilde{x}_{i+1}^{(H)} \ge -\frac{10\pi}{3} + 2$, the car crashes into the left or right boundary and then rebounds, i.e., the velocity direction changes. Taking into account the crash events, the state transition is set as

$$v_{i+1}^{(H)} = \tilde{v}_{i+1}^{(H)} \left(2\mathbb{I}\left(-\frac{10\pi}{3} - 2 < \tilde{x}_{i+1}^{(H)} < -\frac{10\pi}{3} + 2\right) - 1\right),$$
$$x_{i+1}^{(H)} = \max\left\{\min\left\{\tilde{x}_{i+1}^{(H)}, -\frac{10\pi}{3} + 2\right\}, -\frac{10\pi}{3} - 2\right\},$$

where $\mathbb{I}(\cdot)$ denotes an indicator function. More specifically, when $\tilde{x}_{i+1}^{(H)} \le -\frac{10\pi}{3} - 2$ or $\tilde{x}_{i+1}^{(H)} \ge -\frac{10\pi}{3} + 2$, the coordinate of the car is truncated as $x_{i+1}^{(H)} = -\frac{10\pi}{3} - 2$ or $x_{i+1}^{(H)} = -\frac{10\pi}{3} + 2$, and the velocity direction (sign) is inversed as $v_{i+1}^{(H)} = -\tilde{v}_{i+1}^{(H)}$; otherwise, $x_{i+1}^{(H)} = \tilde{x}_{i+1}^{(H)}$ and $v_{i+1}^{(H)} = \tilde{v}_{i+1}^{(H)}$. In the experiment, one modified "Mountain Car" game consists of 500 steps.

We consider three typical agents in the experiments. For Agent A, the instantaneous control reward is set as

$$r_i^{(0)} = r_0\left(s_i^{(0)}, a_i^{(0)}, s_{i+1}^{(0)}\right) = \left|v_{i+1}^{(0)}\right| - 10^2\mathbb{I}\left(x_{i+1}^{(0)} = -\frac{10\pi}{3} - 2 \text{ or } x_{i+1}^{(0)} = -\frac{10\pi}{3} + 2\right),$$

i.e., Agent A aims to maximize the speed of the car and reduce the number of crashes. With respect to Agent A, we study privacy-preserving policies of Agent B and Agent B′ by the proposed PPDQN algorithm, respectively. For Agent B, the instantaneous control reward is set as

$$r_i^{(1)} = r_1\left(s_i^{(1)}, a_i^{(1)}, s_{i+1}^{(1)}\right) = \left|v_{i+1}^{(1)}\right|,$$

i.e., Agent B aims to maximize the speed of the car. For Agent B′, the instantaneous control reward is set as

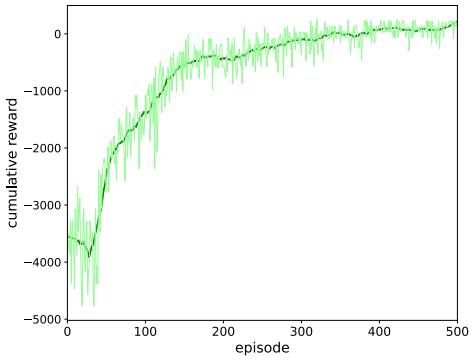
$$r_i^{(1')} = r_{1'}\left(s_i^{(1')}, a_i^{(1')}, s_{i+1}^{(1')}\right) = -\left|v_{i+1}^{(1')}\right|,$$

i.e., Agent B′ aims to minimize the speed of the car.

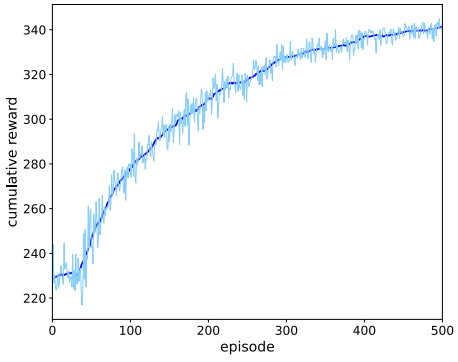
Regarding the PPDQN model, the action-value and target action-value networks have the same structure, which in the experiment consists of two fully connected hidden layers (each hidden layer has 60 neurons) and uses the rectified linear unit (ReLU) activation function. The discriminator network in the experiment consists of three fully connected hidden layers, which respectively have 60, 32, 32 neurons, and also uses the ReLU activation function. The learning rates of the action-value network and the discriminator network are 0.02 and 0.0003. The probability ε is initially set to be 0.9 and decays with an exponential rate 0.9851 until the lower bound 0.01. The target action-value network is updated every 200 training steps.

B. EXPERIMENT RESULTS

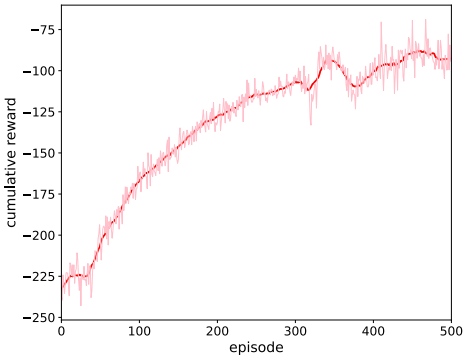
We firstly train the optimal action-value networks of Agent A, Agent B, and Agent B′ by using the DQN algorithm. Note that DQN training of Agent B and Agent B′ is equivalent to



(a) Training process of Agent A



(b) Training process of Agent B



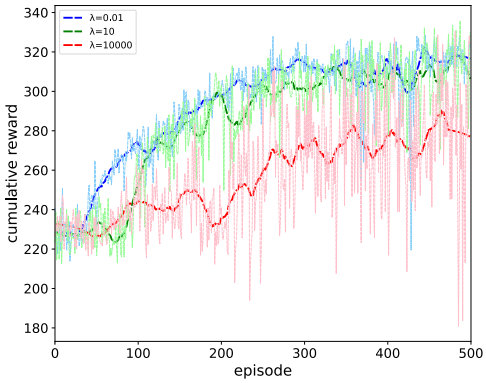
(c) Training process of Agent B′

FIGURE 4. Training processes of Agent A, Agent B, and Agent B′ without privacy-preserving concerns by using the DQN algorithm.

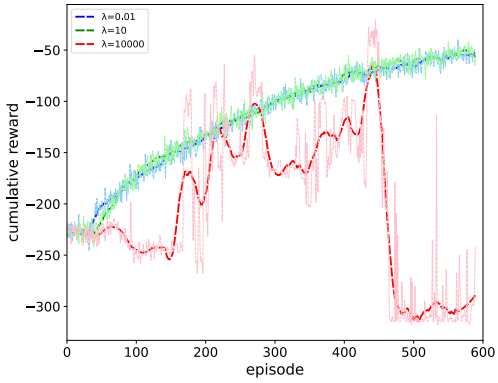
PPDQN training of Agent B and Agent B′ without privacy-preserving concerns, i.e., the privacy-preserving weight is set to $\lambda = 0$. As shown in Figure 4, the DQN training processes of Agent A, Agent B, and Agent B′ can converge within 500 episodes.

For $\lambda = 0.01$, 10, and 10000, we then train the optimal action-value networks of Agent B and Agent B′ by using the PPDQN algorithm and the optimal action-value network of Agent A trained by the DQN algorithm. As shown in Figure 5, the PPDQN training processes of Agent B and Agent B′ can converge within 600 episodes.

Compared with the DQN training process of Agent B in Figure 4(b), the PPDQN training processes of Agent B in



(a) Training processes of Agent B by using the PPDQN algorithm for different values of λ



(b) Training processes of Agent B′ by using the PPDQN algorithm for different values of λ

FIGURE 5. Training processes of Agent B and Agent B′ by using the PPDQN algorithm for $\lambda = 0.01$, 10, and 10000.

Figure 5(a) are notably unstable. It is mainly because of the conflicting design objectives of the privacy-preserving policy and the discriminator in the PPDQN algorithm. We have a similar observation in the comparison between the DQN training process of Agent B′ in Figure 4(c) and the PPDQN training processes of Agent B′ in Figure 5(b).

When $\lambda = 0.01$ and 10, the PPDQN training processes of Agent B and Agent B′ in Figure 5 have similar patterns as the DQN training processes of Agent B and Agent B′ in Figure 4. This observation makes sense because the privacy risk term does not take much effect when the value of privacy-preserving weight λ is small, i.e., the objective of PPDQN in these cases is dominated by maximizing cumulative control reward and therefore is similar to the objective of DQN.

When $\lambda = 10000$, the PPDQN training processes of Agent B and Agent B′ in Figure 5 have quite different patterns from the DQN training processes of Agent B and Agent B′ in Figure 4. Especially, when $\lambda = 10000$, the PPDQN training process of Agent B′ in Figure 5(b) shows significant instability in the first 500 episodes and then starts to converge. This distinct training process pattern mainly results from the intense dynamic max-min game given a large value for the privacy-preserving weight λ.

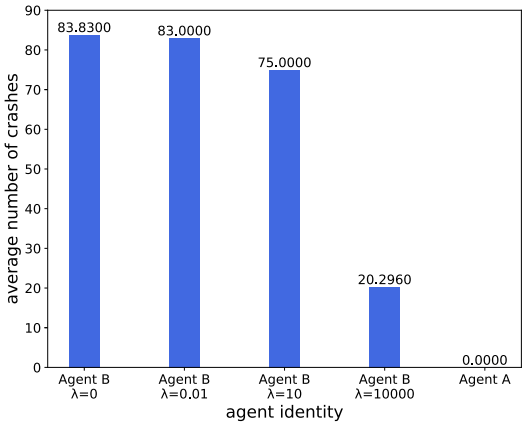
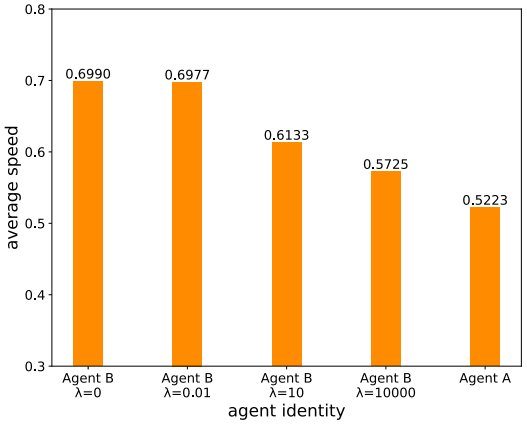
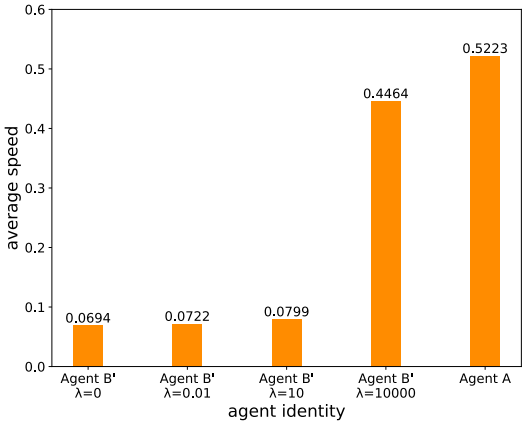


FIGURE 6. Comparison of average numbers of crashes between Agent A and Agent B for different values of λ.



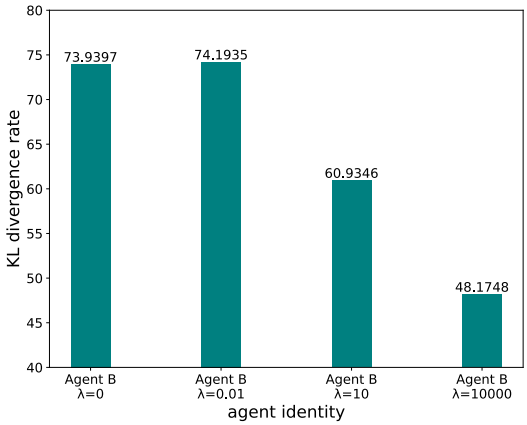
(a) Comparison of average speeds between Agent A and Agent B for different values of λ



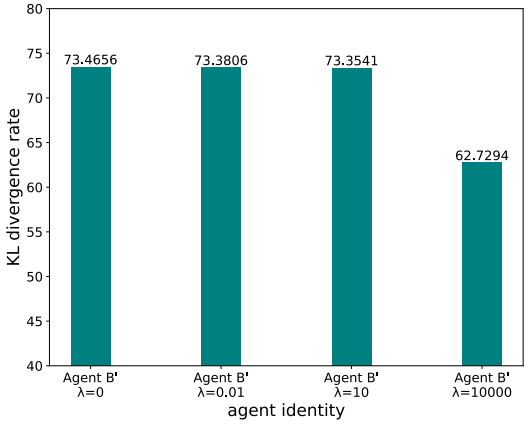
(b) Comparison of average speeds between Agent A and Agent B′ for different values of λ

FIGURE 7. Comparison of average speeds between Agent A, Agent B, and Agent B′ for different values of λ.

We next test the trained policies of Agent A, Agent B, and Agent B′. Each policy is implemented to play the modified "Mountain Car" game for 1000 times. In Figure 6 and



(a) Comparison of estimated KL divergence rates for Agent B with respect to Agent A given different values of λ



(b) Comparison of estimated KL divergence rates for Agent B′ with respect to Agent A given different values of λ

FIGURE 8. Comparison of estimated KL divergence rates for Agent B with respect to Agent A and for Agent B′ with respect to Agent A given different values of λ.

Figure 7, different policies are compared in terms of the average number of crashes and the average speed in one game, respectively. It is worth noting that no crashes happen in the tests of Agent A policy and Agent B′ policies for different values of λ. The test performances of benchmark policies are listed as follows: The average number of crashes and the average speed for Agent A policy are 0 and 0.5223; the average number of crashes and the average speed for Agent B policy with $\lambda = 0$ (trained by the DQN algorithm) are 83.83 and 0.699; and the average number of crashes and the average speed for Agent B′ policy with $\lambda = 0$ (trained by the DQN algorithm) are 0 and 0.0694.

As shown in Figure 6 and Figure 7(a), both the average number of crashes and average speed for Agent B decrease as the value of privacy-preserving weight λ increases. These observations are reasonable because a larger value of the privacy-preserving weight λ makes the environment state sequence induced by Agent B more statistically similar to that induced by Agent A, while a smaller value of λ means Agent B optimizes the policy with more concerns about

increasing the cumulative control reward, i.e., increasing the speed. Similar arguments can be employed to justify the following observations regarding Agent B′: No crashes happen in the tests of Agent A policy and Agent B′ policies for different values of λ; and the average speed for Agent B′ increases as the value of λ increases in Figure 7(b).

To explicitly show the impact of privacy-preserving weight λ on agent identity privacy risk, the KL divergence rate for Agent B with respect to Agent A $\frac{1}{N}D\left(P_{S^{(1)*}_{1:N+1}}\middle\|P_{S^{(0)*}_{1:N+1}}\right)$ and the KL divergence rate for Agent B′ with respect to Agent A $\frac{1}{N}D\left(P_{S^{(1')*}_{1:N+1}}\middle\|P_{S^{(0)*}_{1:N+1}}\right)$ are estimated by using the environment states obtained in the tests of policies of Agent A, Agent B, and Agent B′. More specifically, the continuous coordinate space and velocity space are discretized equally into 20 coordinate states and 25 velocity states, which form a discrete state space with 500 environment states; discretize environment states of Agent A, Agent B, and Agent B′ obtained in the tests; estimate the joint probability mass functions $\hat{P}_{S^{(1)*}_{i+1},S^{(1)*}_{i}}$, $\hat{P}_{S^{(1')*}_{i+1},S^{(1')*}_{i}}$ and conditional probability mass functions $\hat{P}_{S^{(1)*}_{i+1}|S^{(1)*}_{i}}$, $\hat{P}_{S^{(1')*}_{i+1}|S^{(1')*}_{i}}$, $\hat{P}_{S^{(0)*}_{i+1}|S^{(0)*}_{i}}$ by counting the corresponding joint and conditional frequencies based on the discretized environment states; under the assumption of stationary MDP, estimate the KL divergence rate for Agent B with respect to Agent A by $D\left(\hat{P}_{S^{(1)*}_{i+1}|S^{(1)*}_{i}}\middle\|\hat{P}_{S^{(0)*}_{i+1}|S^{(0)*}_{i}}\right)$ and the KL divergence rate for Agent B′ with respect to Agent A by $D\left(\hat{P}_{S^{(1')*}_{i+1}|S^{(1')*}_{i}}\middle\|\hat{P}_{S^{(0)*}_{i+1}|S^{(0)*}_{i}}\right)$.

In Figure 8, the estimated KL divergence rates for Agent B with respect to Agent A and for Agent B′ with respect to Agent A are compared given different values of the privacy-preserving weight λ. As the value of λ becomes larger, the overall trend of the estimated KL divergence rate is to decrease, i.e., the agent identity privacy risk becomes smaller, since the policy designs of Agent B and Agent B′ concern more about preventing privacy leakage.

From Figure 7 and Figure 8, we can also observe the trade-off between control reward and privacy-preserving performance. As the value of λ increases, Figure 7(a) shows a decrease of average speed of Agent B, i.e., a degradation of the control reward of Agent B, while Figure 8(a) shows a decrease of KL divergence rate, i.e., an improvement of privacy-preserving performance. Similarly, as the value of λ increases, Figure 7(b) shows an increase of average speed of Agent B′, i.e., a degradation of the control reward of Agent B′, while Figure 8(b) shows a decrease of KL divergence rate, i.e., an improvement of privacy-preserving performance. The trade-off can be justified as follows. A larger value of λ means the policy design of Agent B or Agent B′ focuses more on improving privacy-preserving performance such that the induced environment states are more statistically similar to those induced by Agent A. Since the instantaneous control reward of Agent A is different from that of Agent B or Agent B′, the control reward of Agent B or Agent B′ degrades as the induced environment states are more statistically similar to those induced by Agent A.

VII. CONCLUSION

We consider the agent identity privacy problem in MDP and formulate it as a privacy-preserving MDP. By jointly exploiting the ideas of DRL and variational method, we further propose a novel PPDQN algorithm to efficiently solve the optimal privacy-preserving policy, which tradeoffs the objectives of improving cumulative control reward and preventing agent identity privacy leakage. We implement the PPDQN algorithm in a modified "Mountain Car" game by considering three typical agents with different control rewards. In the experiment, we can identify different training process patterns of the PPDQN algorithm. When the value of privacy-preserving weight is small, the training process of PPDQN is similar to that of DQN. When the value of privacy-preserving weight is large, the training process of PPDQN suffers from strong instability in the beginning due to the intense dynamic max-min game. The test results show the effectiveness of the PPDQN algorithm to tradeoff two objectives in the agent policy design. The proposed adversarial RL framework can be extended to more complex scenarios, e.g., applications with continuous actions or discrete-continuous hybrid actions, which can be our future work.

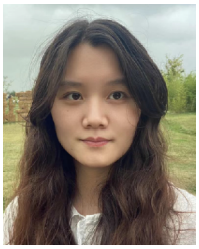
ACKNOWLEDGMENT

The authors thank the Sino-German Center of Intelligent Systems, Tongji University, for their support.

REFERENCES

[1] P. H. J. Nardelli, *Cyber-Physical Systems: Theory, Methodology, and Applications*. Hoboken, NJ, USA: Wiley, 2022.
[2] A. Keyhani and A. Chatterjee, "Automatic generation control structure for smart power grids," *IEEE Trans. Smart Grid*, vol. 3, no. 3, pp. 1310–1316, Sep. 2012.
[3] F. Yang, S. Wang, J. Li, Z. Liu, and Q. Sun, "An overview of Internet of Vehicles," *China Commun.*, vol. 11, no. 10, pp. 1–15, Oct. 2014.
[4] B. Ji, X. Zhang, S. Mumtaz, C. Han, C. Li, H. Wen, and D. Wang, "Survey on the Internet of Vehicles: Network architectures and applications," *IEEE Commun. Standards Mag.*, vol. 4, no. 1, pp. 34–41, Mar. 2020.
[5] D. P. Bertsekas and S. E. Shreve, *Stochastic Optimal Control: The Discrete-Time Case*. New York, NY, USA: Academic, 1978.
[6] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. Cambridge, MA, USA: MIT Press, 2018.
[7] V. Mnih et al., "Human-level control through deep reinforcement learning," *Nature*, vol. 518, pp. 529–533, Feb. 2015.
[8] T. P. Lillicrap et al., "Continuous control with deep reinforcement learning," in *Proc. ICLR*, 2016, pp. 1–10.
[9] S. Fujimoto, H. V. Hoof, and D. Meger, "Addressing function approximation error in actor-critic methods," in *Proc. ICML*, 2018., pp. 1587–1596.
[10] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," 2017, *arXiv:1707.06347*.
[11] H. L. Van Trees, K. L. Bell, and Z. Tian, *Detection, Estimation, and Modulation Theory—Part I: Detection, Estimation, and Filtering Theory*. Hoboken, NJ, USA: Wiley, 2013.
[12] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, MA, USA: MIT Press, 2016.
[13] M. R. Asghar, G. Dán, D. Miorandi, and I. Chlamtac, "Smart meter data privacy: A survey," *IEEE Commun. Surveys Tuts.*, vol. 19, no. 4, pp. 2820–2835, 4th Quart., 2017.

[14] L. Sankar, S. R. Rajagopalan, S. Mohajer, and H. V. Poor, "Smart meter privacy: A theoretical framework," *IEEE Trans. Smart Grid*, vol. 4, no. 2, pp. 837–846, Jun. 2013.
[15] G. Giaconi, D. Gündüz, and H. V. Poor, "Smart meter privacy with renewable energy and an energy storage device," *IEEE Trans. Inf. Forensics Security*, vol. 13, no. 1, pp. 129–142, Jan. 2018.
[16] S. Li, A. Khisti, and A. Mahajan, "Information-theoretic privacy for smart metering systems with a rechargeable battery," *IEEE Trans. Inf. Theory*, vol. 64, no. 5, pp. 3679–3695, May 2018.
[17] M. Shateri, F. Messina, P. Piantanida, and F. Labeau, "Real-time privacy-preserving data release for smart meters," *IEEE Trans. Smart Grid*, vol. 11, no. 6, pp. 5174–5183, Nov. 2020.
[18] Y. You, Z. Li, and T. J. Oechtering, "Energy management strategy for smart meter privacy and cost saving," *IEEE Trans. Inf. Forensics Security*, vol. 16, pp. 1522–1537, 2021.
[19] P. Gope and B. Sikdar, "Privacy-Aware authenticated key agreement scheme for secure smart grid communication," *IEEE Trans. Smart Grid*, vol. 10, no. 4, pp. 3953–3962, Jul. 2019.
[20] S. A. Chaudhry, K. Yahya, S. Garg, G. Kaddoum, M. M. Hassan, and Y. B. Zikria, "LAS-SG: An elliptic curve-based lightweight authentication scheme for smart grid environments," *IEEE Trans. Ind. Informat.*, vol. 19, no. 2, pp. 1504–1511, Feb. 2023.
[21] C. Gentry, "Fully homomorphic encryption using ideal lattices," in *Proc. 41st Annu. ACM Symp. Theory Comput.*, May 2009, pp. 169–178.
[22] Z. Brakerski, C. Gentry, and V. Vaikuntanathan, "(Leveled) fully homomorphic encryption without bootstrapping," in *Proc. 3rd Innov. Theor. Comput. Sci. Conf.*, Jan. 2012, pp. 309–325.
[23] A. Abdallah and X. S. Shen, "A lightweight lattice-based homomorphic privacy-preserving data aggregation scheme for smart grid," *IEEE Trans. Smart Grid*, vol. 9, no. 1, pp. 396–405, Jan. 2018.
[24] A. Mohammadali and M. S. Haghighi, "A privacy-preserving homomorphic scheme with multiple dimensions and fault tolerance for metering data aggregation in smart grid," *IEEE Trans. Smart Grid*, vol. 12, no. 6, pp. 5212–5220, Nov. 2021.
[25] C. Dwork, "Differential privacy," in *Proc. Int. Colloq. Autom. Lang. Program.*, 2006, pp. 1–12.
[26] M. U. Hassan, M. H. Rehmani, and J. Chen, "Differential privacy techniques for cyber physical systems: A survey," *IEEE Commun. Surveys Tuts.*, vol. 22, no. 1, pp. 746–789, 1st Quart., 2020.
[27] F. Kserawi, S. Al-Marri, and Q. Malluhi, "Privacy-preserving fog aggregation of smart grid data using dynamic differentially-private data perturbation," *IEEE Access*, vol. 10, pp. 43159–43174, 2022.
[28] M. B. Hossain, I. Natgunanathan, Y. Xiang, and Y. Zhang, "Cost-friendly differential privacy of smart meters using energy storage and harvesting devices," *IEEE Trans. Services Comput.*, vol. 15, no. 5, pp. 2648–2657, Sep. 2022.
[29] H. B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Proc. AISTATS*, 2017, pp. 1273–1282.
[30] Y. Li, X. Wang, R. Zeng, P. K. Donta, I. Murturi, M. Huang, and S. Dustdar, "Federated domain generalization: A survey," 2023, *arXiv:2306.01334*.
[31] M. M. Badr, M. M. E. A. Mahmoud, Y. Fang, M. Abdulaal, A. J. Aljohani, W. Alasmary, and M. I. Ibrahem, "Privacy-preserving and communication-efficient energy prediction scheme based on federated learning for smart grids," *IEEE Internet Things J.*, vol. 10, no. 9, pp. 7719–7736, May 2023.
[32] M. A. Husnoo, A. Anwar, N. Hosseinzadeh, S. N. Islam, A. N. Mahmood, and R. Doss, "A secure federated learning framework for residential short term load forecasting," *IEEE Trans. Smart Grid*, vol. 15, no. 2, pp. 2044–2055, Mar. 2024.
[33] T. Qian, C. Shao, X. Wang, and M. Shahidehpour, "Deep reinforcement learning for EV charging navigation by coordinating smart grid and intelligent transportation system," *IEEE Trans. Smart Grid*, vol. 11, no. 2, pp. 1714–1723, Mar. 2020.
[34] F. Sangoleye, J. Jao, K. Faris, E. E. Tsiropoulou, and S. Papavassiliou, "Reinforcement learning-based demand response management in smart grid systems with prosumers," *IEEE Syst. J.*, vol. 17, no. 2, pp. 1797–1807, Oct. 2023.
[35] A. Jarwan and M. Ibnkahla, "Edge-based federated deep reinforcement learning for IoT traffic management," *IEEE Internet Things J.*, vol. 10, no. 5, pp. 3799–3813, Mar. 2023.
[36] N. Qu, C. Wang, Z. Li, and F. Liu, "A transmission design in dynamic heterogeneous V2V networks through multi-agent deep reinforcement learning," *China Commun.*, vol. 20, no. 7, pp. 273–289, Jul. 2023.
[37] M. Bansal, I. Chana, and S. Clarke, "UrbanEnQoSPlace: A deep reinforcement learning model for service placement of real-time smart city IoT applications," *IEEE Trans. Services Comput.*, vol. 16, no. 4, pp. 3043–3060, Oct. 2023.
[38] P. Venkitasubramaniam, "Privacy in stochastic control: A Markov decision process perspective," in *Proc. 51st Annu. Allerton Conf. Commun., Control, Comput. (Allerton)*, Oct. 2013, pp. 381–388.
[39] X. Pan, W. Wang, X. Zhang, B. Li, J. Yi, and D. Song, "How you act tells a loT: Privacy-leakage attack on deep reinforcement learning," in *Proc. AAMAS*, 2019, pp. 368–376.
[40] L. Wang, I. R. Manchester, J. Trumpf, and G. Shi, "Initial-value privacy of linear dynamical systems," in *Proc. 59th IEEE Conf. Decis. Control (CDC)*, Dec. 2020, pp. 3108–3113.
[41] A. Y. Ng and S. Russell, "Algorithms for inverse reinforcement learning," in *Proc. Int. Conf. Mach. Learn. (ICML)*, 2000, pp. 663–670.
[42] B. D. Ziebart, A. L. Maas, J. A. Bagnell, and A. K. Dey, "Maximum entropy inverse reinforcement learning," in *Proc. 23rd AAAI Conf. Artif. Intell.*, vol. 8, Chicago, IL, USA, Jul. 2008, pp. 1433–1438.
[43] A. B. Alexandru and G. J. Pappas, "Encrypted LQG using labeled homomorphic encryption," in *Proc. 10th ACM/IEEE Int. Conf. Cyber-Phys. Syst.*, Apr. 2019, pp. 129–140.
[44] J. Suh and T. Tanaka, "Encrypted value iteration and temporal difference learning over leveled homomorphic encryption," in *Proc. Amer. Control Conf. (ACC)*, May 2021, pp. 2555–2561.
[45] J. Le Ny and G. J. Pappas, "Differentially private filtering," *IEEE Trans. Autom. Control*, vol. 59, no. 2, pp. 341–354, Feb. 2014.
[46] M. T. Hale and M. Egerstedt, "Cloud-enabled differentially private multiagent optimization with constraints," *IEEE Trans. Control Netw. Syst.*, vol. 5, no. 4, pp. 1693–1706, Dec. 2018.
[47] M. Hale, A. Jones, and K. Leahy, "Privacy in feedback: The differentially private LQG," in *Proc. Annu. Amer. Control Conf. (ACC)*, Jun. 2018, pp. 3386–3391.
[48] C. Hawkins and M. Hale, "Differentially private formation control," in *Proc. 59th IEEE Conf. Decis. Control (CDC)*, Dec. 2020, pp. 6260–6265.
[49] B. Wang and N. Hegde, "Privacy-preserving Q-learning with functional noise in continuous spaces," in *Proc. NeurIPS*, 2019, pp. 11327–11337.
[50] (2022). *General Data Protection Regulation (GDPR)*. Accessed: Dec. 6, 2023. [Online]. Available: https://gdpr-info.eu/
[51] E. Ferrari, Y. Tian, C. Sun, Z. Li, and C. Wang, "Privacy-Preserving design of scalar LQG control," *Entropy*, vol. 24, no. 7, p. 856, Jun. 2022.
[52] T. M. Cover and J. A. Thomas, *Elements of Information Theory*. Hoboken, NJ, USA: Wiley, 2006.
[53] X. Nguyen, M. J. Wainwright, and M. I. Jordan, "Estimating divergence functionals and the likelihood ratio by convex risk minimization," *IEEE Trans. Inf. Theory*, vol. 56, no. 11, pp. 5847–5861, Nov. 2010.
[54] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Proc. NeurIPS*, 2014, pp. 2672–2680.
[55] J. Tian, Q. Liu, H. Zhang, and D. Wu, "Multiagent deep-reinforcement-learning-based resource allocation for heterogeneous QoS guarantees for vehicular networks," *IEEE Internet Things J.*, vol. 9, no. 3, pp. 1683–1695, Feb. 2022.
[56] G. Brockman, V. Cheung, L. Pettersson, J. Schneider, J. Schulman, J. Tang, and W. Zaremba, "OpenAI gym," 2016, *arXiv:1606.01540*.



YUE TIAN received the B.Eng. degree from Xidian University, Xi'an, China, in 2021. She is currently pursuing the M.Sc. degree with the School of Electronics and Information Engineering, Tongji University, Shanghai, China. Her research interests include information security and adversarial learning.

QI JIANG received the B.Eng. degree in communication engineering from Tongji University, Shanghai, China, in 2022, where he is currently pursuing the M.Sc. degree with the School of Electronics and Information Engineering. His research interests include information security and semantic communication.

ZUXING LI (Member, IEEE) received the B.Eng. degree in information security from Shanghai Jiao Tong University, Shanghai, China, in 2009, the M.Sc. degree in electrical engineering from the Technical University of Catalonia, Barcelona, Spain, and the KTH Royal Institute of Technology, Stockholm, Sweden, in 2013, and the Ph.D. degree in electrical engineering from the KTH Royal Institute of Technology, in 2017. He was a Postdoctoral Researcher with CentraleSupelec, Paris, France, from 2018 to 2019, and the KTH Royal Institute of Technology, from 2019 to 2020. He has been an Assistant Professor with the School of Electronics and Information Engineering, Tongji University, Shanghai, since June 2020. His research interests include statistical inference, information theory, reinforcement learning, information security, and privacy.

CHAO WANG (Member, IEEE) received the B.Eng. degree from the University of Science and Technology of China, Hefei, China, in 2003, and the M.Sc. and Ph.D. degrees from The University of Edinburgh, Edinburgh, U.K., in 2005 and 2009, respectively. In 2008, he was a Visiting Student Research Collaborator with Princeton University, Princeton, USA. From 2009 to 2012, he was a Postdoctoral Research Associate with the KTH Royal Institute of Technology, Stockholm, Sweden. From 2018 to 2020, he was a Marie Curie Fellow with the University of Exeter, Exeter, U.K. He is currently a Professor with Tongji University, Shanghai, China. His research interests include information theory and signal processing for wireless communication networks, and data-driven research and applications for smart city and intelligent transportation systems.