Privacy-preserved Fully Decentralized Multi-agent Reinforcement Learning for Networked Social Systems

Dan Qiao, Junge Zhang*, Yingwen Zhang, Shixiang Xiao, Hao Chen

***Abstract*—Privacy is critical to data security for machine learning algorithms and has attracted extensive attention from academia, industry, and users in recent years. Especially in a multi-agent reinforcement learning (MARL) system where multiple agents interact and communicate with each other, it is particularly indispensable to protect the private information of agents from being inferred by eavesdroppers and malicious nodes. In this paper, we develop Differentially Private *QD*-Learning (DP-*QD*-Learning) to provide privacy-preserving capabilities with theoretical guarantees for networked MARL algorithms under a *fully decentralized* framework. Each agent independently collects local personalized rewards and receives noise-disturbed DP-protected neighbor information over the network to cooperatively maximize team global rewards. Besides, we develop Differentially Private Consensus MARL (DPC-MARL) with the same DP mechanism for the scenario of high-dimensional state-action spaces. Then, we evaluate DP-*QD*-Learning in the environment of the Central Bank Monetary Policy (CBMP) and DPC-MARL in the environment of Cooperative Adaptive Platoon Control (CAPC), where there does not exist a center to collect all agent information for centralized training and the transmitted information is highly private. The results show that the decentralized agents can still reach the consensus of opinions in CMBP and high-quality cooperation in CAPC without transmitting accurate private information, which also conforms to the common sense of economics and the intuition of daily life. Our work appears to be the first theoretical study of the privacy-preserving fully decentralized MARL algorithm for networked agents.**

***Index Terms*—Multi-agent Reinforcement Learning, Differential Privacy, Networked Systems**

I. Introduction

With the remarkable achievements of reinforcement learning (RL) [1], multi-agent reinforcement learning (MARL) has been gradually showing great theoretical value and application potential in recent years. As a branch of RL, MARL is generally modeled as a Markov Game or a Stochastic Game [2], which is much more complicated than single-agent RL to learn optimal policies with the consideration of multiple agents coexistence and interactive decision-making in a common environment. In a variety of fields, MARL algorithms have achieved better performance than classical methods or human-level such as video games, transportation, infrastructure scheduling, and social welfare problem [3]–[7].

To address the non-stationary issue caused by the interaction of decision-making behaviors of multiple agents, the most popular framework of MARL algorithms is centralized training and decentralized execution (CTDE). In CTDE, it is usually assumed that there exists a powerful centralized controller during the training process to collect all the actions and observations of each agent, and then transfer a common reward back to each agent. During the execution stage, each agent makes decisions that only rely on individual observations. Therefore, an intuitive idea is to use centralized training to decompose joint team rewards into individual contributions of each member, and update their strategies according to individual contributions. This method is also called value decomposition, such as VDN, QMIX, and Qtran [3], [8], [9]. For countinous control tasks, CTDE is also very effective in the more complex cooperative and competitive environments, such as MADDPG and COMA [10], [11]. Moreover, noting the importance of information exchange in team cooperation, various communication based MARL algorithms have been developed, such as DIAL, BiCNet, NeurComm, ETCNet, and CommPPO [6], [12]–[15]. Agents use explicit or implicit communication protocols to get the knowledge of others' rewards, observations or strategies, which helps the agent understand the environment and teammate behavior better, and make actions that are more beneficial to the team.

However, most of the above CTDE algorithms are reluctant to handle the exponentially increasing state-action space, i.e., the curse of dimensionality in the center controller. Moreover, the overwhelming amount of exchanging information between the center and agents during the training process brings huge pressure to the communication channel, which also leads to the systemic risk of centralized structures. An alternative way to relax the limitations of CTDE is developing decentralized training decentralized execution (DTDE) MARL algorithms on distributed networked systems [16]. During the training process of DTDE, the avaliable information of the agent is limited to the local topological neighbors instead of all agents in CTDE, which avoids the potential information leakage and overfitting to other agents' policy. In the excution process, the use of neighbor information promotes agents to focus on the cooperative relationship between each other, rather than only making decisions based on their own observations in CTDE. This architecture of "local interaction and global coordination" can effectively alleviate the dimensional curse of single node

Dan Qiao, Junge Zhang, and Hao Chen are with the Institute of Automation, Chinese Academy of Sciences, Beijing 100190, P.R.China (e-mail: qiaodan.casia@gmail.com, jgzhang@nlpr.ia.ac.cn, chenhao2019@ia.ac.cn); Yingwen Zhang is with School of Transportation Science and Engineering, Beihang University, Beijing 102206, P.R.China (e-mail: zyw_iii@buaa.edu.cn); Shixiang Xiao is with School of Electronic, Electrical and Communication Engineering, University of Chinese Academy of Sciences, Beijing 101408, P.R.China (e-mail: xiaoshixiang16@mails.ucas.ac.cn)

in the training process, because the number of neighbors that each agent can affect is usually limited, and fit for common social systems such as cooperative autonomous driving (CAV), power grid and sensor networks.

DTDE can greatly improve the practicability and the flexibility of the MARL algorithms by using network to diffuse local information, but it also faces a unique problem, that is, the reliability of neighbor information under network communication. Most of the DTDE MARL algorithms suppose that the communication channels and members in the team are safe and trustworthy enough which ignored cyberattacks and adversarial behaviors. Recently, some research [17], [18] have showed that when there exist malicious attack nodes in the network, the privacy data of normal members may be monitored by *malicious agents*, and the team behavior will be misled to the wrong target, resulting in the serious crash of the system.

To tackle the data leaking problem, a feasible solution is to protect the privacy of the transmitted information by applying homomorphic encryption [19]. However, encryption technologies often require a large number of computing resources and are difficult to be deployed on mobile devices, for instance, autonomous vehicles and rescue robots. Compared with encryption methods, the differential privacy (DP) method [20] has been widely concerned as an alternative method and applied in Google or Amazon due to the low computing consumption and effective privacy protection ability. The DP technique protects the privacy of data by adding uncorrelated noises to the transmitted data in the networks. Its unique mapping method can ensure that the malicious agent cannot recover the original true value of the eavesdropped agents even if it accesses the inputs, outputs, and algorithm mechanism.

Motivated by the security necessary for the practical deployment of networked systems, in this article, we propose a differential privacy based fully decentralized *QD*-learning algorithm for networked systems by adding Laplace noise on the transimitted Q-value, which prevents the privacy information of normal agents from eavesdropping by malicious agents. The insight here is that even if people passes the noisy and ambiguous message, the group can still achieve the consensus of opinions and accomplish tasks cooperatively. However, the update process of Q-value with randomly noised neighbor information will bring more challenges to the convergence analysis. It is critical to design appropriate noise protection mechanisms carefully and analyze their impact on optimal convergence. To our best knowledge, this is the first work to introduce differential privacy technology into the field of networked decentralized MARL. Our main contributions are threefolds as follows.

1) To handle the difficulty of convergence, we propose the DP protected *QD*-Learning algorithm by designing decaying Laplace noises for the transmitted Q-value for each agent. Besides, without loss of generality, we develop DPC-MARL under the actor-critic framework as a privacy-preserving DTDE MARL algorithm suitable for broader tasks with high-dimensional state-action spaces.
2) Considering the influence of additive noise on the updating process, we prove the mean convergence of each agent's Q-values with the randomize of reward collection and provide a new distribution bound of average Q-value to describe the (p, r)-accuracy with optimal Q^* in our algorithm. The privacy analysis about privacy budgets shows the trade-off between algorithm accuracy and privacy level.
3) We develop a networked social environment named as Central Bank Monetary Policy (CBMP) and show the effectiveness of ϵ-privacy DP-*QD*-Learning in the case of outside eavesdropper agents. For the generality of the DP mechanism, we also extend it as DPC-MARL for solving a high-dimensional state-action space task, i.e., the Cooperative Adaptive Platoon Control (CAPC).

Notation: Let $\mathbb{R}$ and $\mathbb{R}^+$ denote the set of real numbers and positive real numbers. Let $\mathbb{N}$ and $\mathbb{N}_{\geq 0}$ denote the set of integer numbers and nonnegative integer numbers. $\mathbf{1}_N \in \mathbb{R}^{N*1}$ is column vector with all 1 elements. The probability space $(\Omega, \mathcal{F})$ supports all random objects, where $\mathbb{E}[\cdot]$ and $\mathbb{P}(\cdot)$ denote the expectation and probability on $(\Omega, \mathcal{F})$ respectively. The operator $||\cdot||$ and $||\cdot||_\infty$ represent the $\mathcal{L}_2$ norm and $\mathcal{L}_\infty$ norm of the vectors and matrixs. The eigenvalues for a matrix L is ordered as $\lambda_1(L) \leq \lambda_2(L) \leq \ldots \leq \lambda_N(L)$. $exp(x)$ represents the exponetial function e^x.

II. Related Works

A. Networked DTDE MARL

The first benchmark of DTDE algorithms can be traced back to *QD*-learning which originally introduced consensus and innovation update into Q-learning algorithm [21]. Afterward, a consensus-based actor-critic algorithm was firstly proposed in ConsensusMARL [22] for a group of networked agents with the theoretical proof of convergence rate under the linear approximation assumption and comparable simulations under deep neural networks. This work further inspired extensive research about sample efficiency and continuous control under the DTDE scheme [23], [24]. In [14], NeurComm was proposed for networked system control by formulating the problem as a spatiotemporal Markov decision process. A novel differentiable communication protocol and a spatial discount factor were introduced to improve learning efficiency and performance. In [25]–[27], authors investigated the exponentially decaying mechanism of agent influence with space in networked MARL systems, which provided theoretical guarantees for the local-interaction global-cooperation characteristic of DTDE framework. In *value propagation* algorithm [28], networked MARL was converted to a constrained distributed optimization problem and solved by primal-dual decentralized optimization method with a non-asymptotic convergence rate of $\mathcal{O}(1/T)$ with nonlinear function approximation. In [16], a fully decentralized MARL framework F2A2 was proposed for large-scale general cooperative MARL settings by designing primal-dual hybrid gradient decent updating. The theoretical analysis and empirical results demonstrated that the F2A2 framework can improve the flexibility and performance for a variety of on-policy and off-policy MARL algorithms, such as SAC, TD3 and COMA.

B. Differential Privacy

Very recently, the DP technique has been widely discussed in various communities such as machine learning, distributed optimization, multi-agent consensus control and so on [29]–[40]. In [29], the DP noise was introduced in single-agent RL to protect the private rewards of agents from being restored by algorithms such as inverse reinforcement learning. In [30], the theoretical analysis about the DP-stochastic gradient descent with gradient clipping and DP noises proved the effectiveness of DP in common deep learning algorithms and provided a guideline for balancing the privacy, fairness and accuracy in deep learning algorithms.

Besides, as an effective training method to improve user data privacy, the federated learning uploads the encrypted gradient instead of local user data, and obtains the aggregated gradient from the trusted central server to update their local model. However, the attackers can monitor the transimitted parameters or attack the server to obtain the private information. In [31]–[33], the DP mechanism was employed in parameter transmit and guaranteed the convergence performance and higher privacy levels of networked systems. Prochlo can further amplify the privacy protection degree of the central server in federated learning by randomly shuffling and anonymizing the transmission data sources. Moreover, the DP mechanism is also extensively investigated in multi-agent distributed optimization and distributed control community to protect the state trajectory of agents, such as [36]–[40]. However, it is still an open problem to protect the privacy of MARL algorithms with DP. For more details about DP refer to the survey [41].

III. Preliminaries

A. Networked Markov Decision Process

Consider a set of networked systems consisting of $N-agents$ running in a common environment, where the communication topology among agents is described by a graph $\mathcal{G}$. Different from *CTDE* methods, in this paper we focus on the **Decentralized-Training-Decentralized-Execution** setting, i.e., all the agents collect individual rewards from the environment and make decisions by themselves without a centralized information collection and distribution (Fig. 1).

For generalization, the communication topology is depicted as a randomly-switching and jointly-connected undirected graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}, A\}$. The node $i \in \mathcal{V}, i = 1, 2, \dots, N$ represents the i^{th} agent and the edge $e_{ij} \in \mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}, i, j = 1, 2, \dots, N$ represents the communication link between the nodes j and i. The adjacency matrix $A = (a_{ij})_{N \times N}$ denotes the structure of the graph, $a_{ij} = 1$ if $j, i \in \mathcal{E}$, and $a_{ij} = 0$ otherwise. Self-loop is not considered, i.e., $a_{ii} = 0$. Degree matrix is defined as $D = diag\{d_1, d_2, \dots, d_N\}$ with $d_i = \sum_{j=1}^{N} a_{ij}$. The Laplacian matrix satisfies $L = D - A$. Agent i can only exchange information with its neighbors $\mathcal{N}_i = \{j | a_{ij} = 1\}$.

The networked Markov Decision Process (MDP) can be described by a 6-tuple $\{\mathcal{S}, \{\mathcal{A}^i\}, P, \{r_i\}_{i \in \mathcal{V}}, \{\mathcal{G}_t\}_{t \geq 0}, \{\mathcal{M}_{ij}\}_{ij \in \mathcal{E}}\}$: $\mathcal{S}$ is the finite global state space, A^i is the finite action space for each agent, and the reward $r_i(s, a) : \mathcal{S} \times \mathcal{A}^i \to \mathbb{R}$ is the random one-stage local reward of agent i whenever action $a^i \in \{\mathcal{A}^i\}$ is applied at state $s \in \mathcal{S}$. As the environment is influenced by the joint action $\mathcal{A}_{joint} = \prod_{i=1}^{N} \mathcal{A}^i$ of all the agents, the state transition is governed by the MDP probability $P : \mathcal{S} \times \mathcal{A}_{joint} \times \mathcal{S} \to [0, 1]$. The neighbor message $m_{\mathcal{N}_i i} = \{m_{ji}\}_{j \in \mathcal{N}_i} \subseteq \mathcal{M}_{ji}$ represents the information received by agent i from its neighbors $\mathcal{N}_i$.

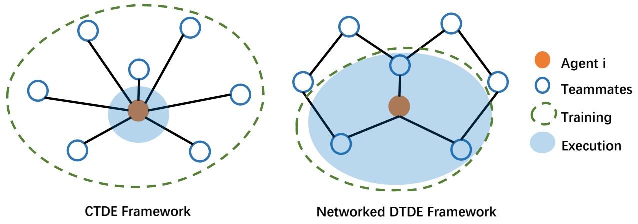


Fig. 1: In CTDE (left), agent i uses all agents' observation in centralized training and self observation in decentralized excution. In networked DTDE (right), agent i uses local neighbor observation in both of training and execution.

Definition 1. *(Networked Multi-agent MDP.) A networked Multi-Agent MDP is characterized as a 6-tuple $\{\mathcal{S}, \{\mathcal{A}^i\}, P, \{r_i\}_{i \in \mathcal{V}}, \{\mathcal{G}_t\}_{t \geq 0}, \{\mathcal{M}_{ij}\}_{ij \in \mathcal{E}}\}$. We assume that the states are globally observable and the individual rewards are local and different. At time t, each agent i chooses its own action and get reward $r_i(s, a)$ from the environment. At the same time, it also transmits and receives neighbor message $m_{\mathcal{N}_i i}$. Then the environment updates. All local rewards are **not** shared with each other.*

Besides, the centralized global reward is defined as $r_{center} = \frac{1}{N}\sum_{i=1}^{N} r_i$. Note that both the reward and execution are performed locally and individually, our model fully follow the DTDE framework.

For a policy $\pi^i : \mathcal{S} \times \mathcal{A}^i \to [0, 1]$ and initial state S_0, the infinite horizon discounted state-action value function of agent i is

$$Q^{i(\pi)}_{s,a} = \mathbb{E}_{\pi^i}[\sum_{t \geq 0}^{\infty} \gamma^t r_i(s(t), a(t)) | S_0 = s, A_0^i = a, \pi^i],$$

where $\gamma \in (0, 1)$ is the discounting factor. For clear context, we will write it as an abbreviation $Q^i_{s,a}$ below. The objective of *QD*-Learning is to minimize the Bellman innovation error and neighborhood consensus error

$$\frac{1}{2}(Q^i_{s,a} - \mathbb{E}[r_i + \gamma \max_{a'} Q(s', a')] - \sum_{j \in \mathcal{N}_i} a_{ij}(Q^i_{s,a} - Q^j_{s,a}))^2.$$

B. Differential Privacy

For the privacy information protection of datasets, differential privacy has been shown to be a powerful benchmark against many forms of hostile attacks and eavesdropping. The fundamental idea of differential privacy is that, considering an original dataset $\mathcal{D}$ and its adjacent dataset $\mathcal{D}'$, the output (observation) should be the same at the probability space after running certain mapping mechanism on both datasets (Fig. 2). In this way, the eavesdropper cannot restore the true value of the original dataset $\mathcal{D}$ by any means, even if it can access the operating mechanism and structure of the algorithm.

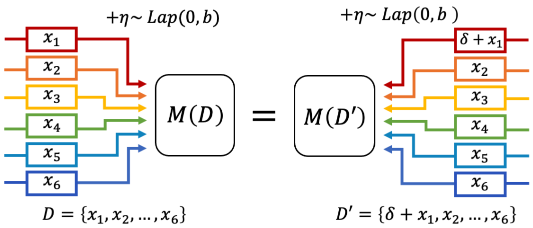


Fig. 2: The outputs of two δ-adjacency datasets $\mathcal{D}$ and $\mathcal{D}'$ with DP noise η equal in probability.

Definition 2. *$\epsilon-$Differential Privacy [20]. Let $\delta \in \mathbb{R}^+$, two datasets $\mathcal{D}$ and $\mathcal{D}'$ are $\delta-$adjacent which satisfy the following condition,*

$$|\mathcal{D}_i - \mathcal{D}'_i| \leq \begin{cases} \delta & if \quad i = i_0, \\ 0 & if \quad i \neq i_0, \end{cases}$$

for some $i_0, i \in \{1, \dots, n\}$. Then, the randomized mechanism $\mathcal{M} : \mathcal{D} \to \mathcal{O}$ is said to be $\epsilon-$Differential Privacy if for any subset $O \subseteq \mathcal{O}$, it holds that

$$\mathbb{P}(\mathcal{M}(\mathcal{D}) \in O) \leq exp(\epsilon)\mathbb{P}(\mathcal{M}(\mathcal{D}') \in O). \tag{1}$$

IV. Differential-Private *QD*-Learning

Considering potential eavesdroppers and malicious attacks in many social systems, it is necessary to protect the privacy message of each agent in the networks. At the same time, many social systems can be modeled as networked MDP problems with finite state space and finite action space, such as opinion networks, election, monetary policy, social welfare, and so on. In order to highlight the effectiveness of privacy protection mechanism in fully decentralized MARL, this paper uses tabular *QD*-Learning algorithm [21] as a benchmark to reduce the impact of various complex techniques on convergence, so as to focus on the theoretical analysis of differential privacy noise on Q-value convergence and optimality.

Here we propose our algorithm as DP-*QD*-Learning in Algorithm 1. The main improvement is that the original privacy data $Q^j_{s,a}$ transmitted as neighbor message m_{ij} has been replaced by the DP protected information $\hat{Q}^j_{s,a}$ in Algorithm 1 line 20. In order to facilitate the discussion of the theoretical results in section V-A, following the settings of the *QD*-Learning [21], we propose the following assumptions and properties.

Assumption IV.1. *For the complete probability space $(\Omega, \mathcal{F})$, denote the MDP filtration $\{\mathcal{F}_t\}$ with σ-algebra as $\mathcal{F}_t = \sigma(\{\mathcal{S}(s), \mathcal{A}(s)\}_{s \leq t}, \{r_i(s(s), a(s))\}_{s<t})$. Hence, the one-stage reward $r_i(s(t), a(t))$ is adapted to $\mathcal{F}_{t+1}$ for all $t > 0$. The state transition probability is $\mathbb{P}(s' = S'|\mathcal{F}_t) = p^a_{s,s'}$.*

Assumption IV.2. *The one-stage reward possess super-quadratic moments which means that there exists a positive constant $\varepsilon_1 > 0$ with $K_r \in \mathbb{R}^+$ satisfying the condition as follows,*

$$\mathbb{E}[r_i^{2+\varepsilon_1}] \leq K_r < \infty, \quad \forall i = 1, \dots, N.$$

Algorithm 1 Differential Privacy *QD*-Learning

```
1:  **Inputs**: the environment and the reward function $r(\cdot)$
2:  **Parameters**: agents number $N$, noise parameters $s$ and $q$,
    discount factor $\gamma$, learning rate parameters $a, b, \tau_1, \tau_2$
3:  **Output**: trained value function $Q(s, a)$
4:  Initialize $Q^i_{s,a}$ and $k = [k_1, \dots, k_N]$ to be all zero matrix;
    set start from random initial state $S_0$
5:  for $t = 0, \dots, T$ do
6:      for $i = 1, \dots, N$ do
7:          Sample the action $a^i \sim \pi^i(s, a^i)$
8:          Receive reward $r_i(s, a)$, and count the sampling times
            $k_i$ of the state-action pair $(s, a^i)$
9:          Update the gains $\alpha_k$ and $\beta_k$ with $k_i$ as (3)
10:     end for
11:     The environment receives the joint action $a_{joint} =
        \prod_{i=1}^N a^i$ and updates into the new state $s'$
12:     for $i = 1, \dots, N$ do
13:         Get state $s'$
14:         Compute $\iota_i = s_i q_i^t$, sample $\eta_i \sim Lap(0, \iota_i)$, accord-
            ing to Equation (5)
15:         Update $\hat{Q}^i_{s,a} \leftarrow Q^i_{s,a} + \eta_i$
16:     end for
17:     /* Message transmission.
18:     for $i = 1, \dots, N$ do
19:         Receive neighbor messages $\hat{Q}^j_{s,a}, j \in \mathcal{N}_i$
20:         Update $Q^i_{s,a} \leftarrow Q^i_{s,a} - \beta_k \sum_{v_j \in \mathcal{N}_i}(Q^i_{s,a} - \hat{Q}^j_{s,a}) +$
            $\alpha_k(r_i(s, a) + \gamma \max_{a'} Q^i_{s',a'} - Q^i_{s,a})$
21:     end for
22:     Update state $s \leftarrow s'$
23: end for
24: Return the trained $Q(s, a)$ function
```

Assumption IV.3. *The communication graph $\mathcal{G}(t) = \{\mathcal{V}, \mathcal{E}(t), A(t)\}$ is undirected connected and balance graph for all $t > 0$.*

Proposition IV.4. *For a connected undirected graph $\mathcal{G}$, the eigenvalues of the adjacency matrix A satisfy that $\lambda_1(A) = 0$, $\lambda_N(A) \geq \ldots \geq \lambda_2(A) > \lambda_1(A)$.*

The update process of the Q-value for each state-action pair (s, a) in our algorithm evolves as

$$\begin{aligned} Q^i_{s,a}(t+1) &= Q^i_{s,a}(t) - \beta_{s,a}(t) \sum_{v_j \in \mathcal{N}_i(t)} (Q^i_{s,a}(t) - \hat{Q}^j_{s,a}(t)) \\ &+ \alpha_{s,a}(t)(r_i(s_t, a_t) + \gamma \max_{a' \in \mathcal{A}} Q^i_{s',a'}(t) - Q^i_{s,a}(t)), \end{aligned} \tag{2}$$

where the weight sequences $\alpha_{s,a}(t)$ and $\beta_{s,a}(t)$ are $\mathcal{F}_t(t)$-adapted processes. For the state-action pair (s, a) at time t, they satisfy the following equations,

$$\alpha_{s,a}(k) = \begin{cases} \dfrac{a}{(k+1)^{\tau_2}} & if \quad t = T_{s,a}(k), \\ 0 & otherwise, \end{cases}$$

$$\beta_{s,a}(k) = \begin{cases} \dfrac{b}{(k+1)^{\tau_1}} & if \quad t = T_{s,a}(k), \\ 0 & otherwise, \end{cases} \tag{3}$$

where $a, b, \tau_1 \in (\frac{1}{2}, 1], \tau_2 \in (0, \tau_1 - \frac{1}{2+\varepsilon_1})$ are positive constants, and $t = T_{s,a}(k)$ represents the $k+1$-th sampling time which is finite almost surely for $k \in \mathbb{N}_{\geq 0}$. The transmitted message is designed as

$$\hat{Q}^i_{s,a}(t) = Q^i_{s,a}(t) + \eta_i(t), \tag{4}$$

where the additive Laplace noise is designed as

$$\eta_i(t) \sim Lap(0, \iota_i(t)), and \quad \iota_i(t) = s_i q_i^t, q_i \in (0,1), \tag{5}$$

with s_i and q_i being positive constants.

Remark 1. *Our algorithm update follows the form of innovation + consensus algorithm [21], which is a much more general framework. When $\beta_{s,a}(t) = 0$, the update 2 is reduced as the traditional single-agent TD-error reinforcement learning algorithm [42]. Moreover, if $\alpha_{s,a}(t) = 0$, the update (2) is reduced to the classical multi-agent consnesus control algorithm [38]. With the paramters τ_1 and τ_2 being selected to satisfy $\lim_{t\to\infty} \frac{\beta_{(s,a)}(t)}{\alpha_{(s,a)}(t)} = \infty$, the consensus factor will dominate the update process and leads the Q-table of all agents to the optimal convergence asymptotically as mentioned in (M.5) in [21].*

V. Theoretical Results

A. Convergence Analysis

In this section, we will show the asymptotical convergence of the DP-*QD*-Learning algorithm. First, we will demonstrate the mean square convergence of Q^i. Second, we will show the convergence of the expectation $\mathbb{E}[Q^i]$.

To this end, we define the local *QD* operator for the (s,a)-pair of agent i as

$$\mathcal{Y}^i_{s,a}(Q) = \mathbb{E}[r_i(s,a)] + \gamma \sum_{s' \in \mathcal{S}} p^a_{s,s'} \max_{a' \in \mathcal{A}} Q_{s',a'},$$

which has the fixed point defined as

$$Q^{i*}_{s,a} = \mathbb{E}[r_i(s,a)] + \gamma \sum_{s' \in \mathcal{S}} p^a_{s,s'} \max_{a' \in \mathcal{A}} Q^{i*}_{s',a'},$$

where $Q^{i*} = [Q^{i*}_{s,a}] \in \mathbb{R}^{|\mathcal{S} \times \mathcal{A}|}$. Then the updating law (2) can be rewritten as

$$\begin{aligned} Q^i_{s,a}(t+1) = Q^i_{s,a}(t) - \beta_{s,a}(t) \sum_{v_j \in \mathcal{N}_i(t)} (Q^i_{s,a}(t) - \hat{Q}^j_{s,a}(t)) \\ + \alpha_{s,a}(t)(\mathcal{Y}^i_{s,a}(Q^i(t)) - Q^i_{s,a}(t) + v^i_{S,A}(Q^n_t)), \end{aligned} \tag{6}$$

where

$$\begin{aligned} v^i_{S,A}(Q^n_t) &= r_i(s_t, a_t) + \gamma \max_{a' \in \mathcal{A}} Q^i_{s',a'}(t) - \mathcal{Y}^i_{s,a}(Q^i(t)) \\ &= r_i(s_t, a_t) - \mathbb{E}[r_i(s,a)] \\ &\quad + \gamma \max_{a' \in \mathcal{A}} Q^i_{s',a'}(t) - \gamma \sum_{s' \in \mathcal{X}} p^a_{s,s'} \max_{a' \in \mathcal{A}} Q^{i*}_{s',a'} \end{aligned}$$

is a magnifying operator which satisfies

$$\begin{aligned} \mathbb{E}[v^i_{S,A}(Q^n_t) | \mathcal{F}_t] = 0, \\ \mathbb{E}[||v^i_{S,A}(Q^n_t)||^2 | \mathcal{F}_t] \leq K_v < \infty. \end{aligned}$$

Define the piecewise updating operator $z^i_{s,a}$ for agent i at the updating time $T^i_{s,a}(k)$ as

$$z^i_{s,a}(k) = Q^i_{s,a}(t), \quad for \quad t = T^i_{s,a}(k), k = 1, 2, \ldots.$$

Then from (6) we have

$$\begin{aligned} z^i_{s,a}(k+1) = & z^i_{s,a}(k) + \beta_{s,a}(k) \sum_{v_j \in \mathcal{N}_i(k)} (z^i_{s,a}(k) - z^j_{s,a}(k)) \\ & + \alpha_{s,a}(k)(\mathcal{Y}^i_{s,a}(z^i_{s,a}(k) - z^i_{s,a}(k) + v^i_{S,A}(Q^n_k)) \\ & + \beta_{s,a}(k) \sum_{v_j \in \mathcal{N}_i(k)} \eta_j(k). \end{aligned}$$

It can be written in the vector form as

$$\begin{aligned} z_{s,a}(k+1) = (I_N - \beta(k)L(t) - \alpha(k)I_N) z_{s,a}(k) \\ + \alpha(k)(\mathcal{Y}_{s,a}(z_{s,a}(k)) + v(k)) + \beta(k)A(k)\eta(k), \end{aligned} \tag{7}$$

where

$$\begin{aligned} \mathcal{Y}_{s,a}(z_{s,a}(k)) = col[\mathcal{Y}^i_{s,a}(z^i_{s,a}(k))], \quad i = 1, \ldots, N, \\ v(k) = col[v^i_{S,A}(Q^n_k)], \quad i = 1, \ldots, N. \end{aligned}$$

Then we will show the mean square convergence theorem for the DP-*QD*-Learning algorithm.

Theorem V.1 (Consensus in mean square a.s.). *Consider the undirected connected graph $\mathcal{G}$, under Assumptions IV.1-IV.3, for every (s,a)-pair, the Q-value of DP-QD-Learning can achieve asymptotically consensus in mean square almost surely (a.s.) as*

$$\lim_{t\to\infty} \mathbb{E}[(Q^i_{s,a}(t) - Q^j_{s,a}(t))^2] = 0, i, j = 1, \ldots, N. \tag{8}$$

Proof. To prove Theorem V.1, we first establish the following lemma for a general case. The proof of Lemma 1 can be found in Appendix IX.

Lemma 1. *For each state-action pair (s,a), let $y_{s,a}(t)$ denote the $\{\mathcal{F}_t\}$ adapted process evolving as*

$$\begin{aligned} y_{s,a}(t+1) = (I_N - \beta(t)L(t) - \alpha(t)I_N) y_{s,a}(t) \\ + \alpha(t)v(t) + \beta(t)A(t)\eta(t), \end{aligned} \tag{9}$$

where the weight sequences $\{\alpha(t)\}$ and $\{\beta(t)\}$ are given by (3) and $\{v(t)\}$ is an $\{\mathcal{F}_{t+1}\}$ adapted magnifying process. Then we have $\lim_{t\to\infty} \mathbb{E}[(y^i_{s,a}(t) - y^j_{s,a}(t))^2] = 0, i, j = 1, \ldots, N$ as $t \to \infty$ a.s.

Define $\varpi^i_{s,a}(k) = z^i_{s,a}(k) - y^i_{s,a}(k)$, with (7) and (29), we have

$$\begin{aligned} \varpi_{s,a}(k+1) = & (I_N - \beta(k)L(k) - \alpha(k)I_N) \varpi_{s,a}(k) \\ & + \alpha_{s,a}(k) \mathcal{Y}^i_{s,a}(z^i_{s,a}(k)). \end{aligned} \tag{10}$$

As $\mathcal{Y}^i_{s,a}(z^i_{s,a}(k)) \leq G < \infty$ is bounded, then we have

$$\varpi_{s,a}(k) \leq (1 - w_3 \alpha_{s,a}(k)) \varpi_{s,a}(k) + \alpha_{s,a}(k) G.$$

With Lemma 3.1 (Polyak, 1987) in [40] , we can conclude that $\lim_{k\to\infty}\varpi_{s,a}(k)=0$, which implies that

$$\begin{aligned}\lim\mathbb{E}[(z^i_{s,a}(k)-z^j_{s,a}(k))^2]=&\lim\mathbb{E}[(y^i_{s,a}(k)-y^j_{s,a}(k))^2]\\=&0\end{aligned}\tag{11}$$

as $k\to\infty$. Since $z^i_{s,a}(k)$ is the piecewise process of $Q^i_{s,a}(k)$ and $t\to\infty$ as $k\to\infty$, it is clear that the DP-*QD*-Learning can achieve consensus in mean square a.s. □

Then we will show that the expectation of $Q^i_{s,a}(t)$ will achieve consensus. Because only the Laplacian noises η with the expectation of 0 is introduced into the transmission information $\hat{Q}_{s,a}(t)$, it is easy to get the conclusion of convergence of the expectation $\mathbb{E}[Q^i]$ as Lemma 5.2 in [21].

Theorem V.2 (**Consensus in expectation**). *Consider the undirected connected graph $\mathcal{G}$, under Assumptions IV.1-IV.3, for every (s,a)-pair, the Q-value of DP-QD-Learning can achieve consensus in expectation almost surely as*

$$\lim_{t\to\infty}\mathbb{E}[Q^i_{s,a}(t)-\bar{Q}_{s,a}(t)]=0, i,j=1,\dots,N,\tag{12}$$

where $\bar{Q}_{s,a}(t)=\frac{1}{N}\sum_{i=1}^N Q^i_{s,a}(t)$ is the average value of $Q^i_{s,a}$ for all $i\in\mathcal{N}$ at time t.

The proof of Theorem V.2 can be found in Appendix X.

B. The (p,r)-accuracy Analysis

Due to the disturbance of random noise in the DP mechanism, the convergence of Q-value in DP-*QD*-Learning may not precisely converge to the optimal value Q^*, but to its neighborhood. The accuracy of convergence value in DP-*QD*-Learning algorithm can be described by its statistical law. In this section, we will demonstrate its distribution character with theoretical analysis for the bound of expectation and variance.

In order to compare the convergence performance of fully decentralized algorithms and centralized algorithms, we assume that there exists a center that can collect all agent information and conduct centralized training. Define the centralized *QD* operator $\bar{\mathcal{Y}}:\mathbb{R}^{|\mathcal{S}\times\mathcal{A}|}\mapsto\mathbb{R}^{|\mathcal{S}\times\mathcal{A}|}$, whose (s,a)-th component $\bar{\mathcal{Y}}_{s,a}:\mathbb{R}^{|\mathcal{S}\times\mathcal{A}|}\mapsto\mathbb{R}$ is

$$\bar{\mathcal{Y}}_{s,a}(Q)=\frac{1}{N}\sum_{i=1}^N\mathbb{E}[r_i(s,a)]+\gamma\sum_{s'\in\mathcal{S}}p^a_{s,s'}\max_{a'\in\mathcal{A}}Q_{s',a'}.$$

for all $Q\in\mathbb{R}^{|\mathcal{S}\times\mathcal{A}|}$. Informally, $\bar{\mathcal{Y}}(\cdot)$ is the average of the local *QD* operators, i.e., for each $Q\in\mathbb{R}^{|\mathcal{S}\times\mathcal{A}|}$ and state-action pair (s,a), we have

$$\bar{\mathcal{Y}}_{s,a}(Q)=\frac{1}{N}\sum_{i=1}^N\mathcal{Y}^i_{s,a}(Q).\tag{13}$$

Lemma 2 (**Contraction Lemma [21]**). *The centralized QD operator is a contraction process. Specifically, we have*

$$||\bar{\mathcal{Y}}(Q)-\bar{\mathcal{Y}}(Q')||_\infty\le\gamma||Q-Q'||_\infty,\forall Q,Q'\in\mathbb{R}^{|\mathcal{S}\times\mathcal{A}|}.$$

Define the unique fixed point of $\bar{\mathcal{Y}}(\cdot)$ as Q^*. Define the average value of $Q_{s,a}(t)$ as $\bar{Q}_{s,a}(t)=\frac{1}{N}\mathbf{1}_N^TQ_{s,a}(t)$.

From the updating law (6), for each state-action pair (s,a) we have

$$\begin{aligned}\bar{Q}_{s,a}(t+1)=&(1-\alpha_{s,a}(t))\bar{Q}_{s,a}(t)+\alpha_{s,a}(t)[\bar{v}_{s,a}(t)\\&+\frac{1}{N}\sum_{i=1}^N\mathcal{Y}^i_{s,a}(Q^i_t)]+\frac{\beta_{s,a}(t)}{N}\mathbf{1}_N^TA(t)\eta(t),\end{aligned}\tag{14}$$

where $\bar{v}_{s,a}(t)=\frac{1}{N}\mathbf{1}_N^Tv_{s,a}(t)$. Note that the connected graph $\mathcal{G}$ satisfies $\mathbf{1}_N^TL=0$. Then the equation (14) can be written as

$$\begin{aligned}\bar{Q}_{s,a}(t+1)=&(1-\alpha_{s,a}(t))\bar{Q}_{s,a}(t)+\alpha_{s,a}(t)[\bar{v}_{s,a}(t)+\bar{\varepsilon}_{s,a}(t)\\&+\bar{\mathcal{Y}}_{s,a}(\bar{Q}_t)]+\frac{\beta_{s,a}(t)}{N}\mathbf{1}_N^TA(t)\eta(t),\end{aligned}$$

where the residual term is defined as

$$\bar{\varepsilon}_{s,a}(t)=\frac{1}{N}\sum_{i=1}^N(\mathcal{Y}^i_{s,a}(Q^i_t)-\mathcal{Y}^i_{s,a}(\bar{Q}_t)).\tag{15}$$

As the operators $\mathcal{Y}^i_{s,a}(\cdot)$ are Lipschitz, we have

$$|\bar{\varepsilon}_{s,a}(t)|\le l_1\sum_{i=1}^N||Q^i_t-\bar{Q}_t||,\tag{16}$$

where l_1 is the Lipschitz constant. Moreover, we have the following equations as:

$$\begin{aligned}\mathbb{E}[\bar{\varepsilon}^2_{s,a}(t)]\le&l_1^2\mathbb{E}[(\sum_{i=1}^N(Q^i_{s,a}(t)-\bar{Q}_{s,a}(t)))^2]\\=&l_1^2\mathbb{E}[\sum_{i=1}^NQ^i_{s,a}(t)Q^j_{s,a}(t)+C_N^2\bar{Q}^2_{s,a}(t)\\&-2\sum_{i=1}^N(Q^i_{s,a}(t)+Q^j_{s,a}(t))\bar{Q}_{s,a}(t)]\\&+\mathbb{E}[\sum_{i=1}^N(Q^i_{s,a}(t)-\bar{Q}_{s,a}(t))^2]\\=&l_1^2\mathbb{E}[\sum_{i=1}^NQ^i_{s,a}(t)Q^j_{s,a}(t)+(C_N^2-2N(N-1))\bar{Q}^2_{s,a}(t)]\\&+\mathbb{E}[\sum_{i=1}^N(Q^i_{s,a}(t)-\bar{Q}_{s,a}(t))^2].\end{aligned}$$

As $Q^i_{s,a}(t)$ and $\bar{Q}_{s,a}(t)$ are bounded which is proved in Lemma 5.1 in [21], consider the mean square convergence $\mathbb{E}[(Q^i_{s,a}(t)-\bar{Q}_{s,a}(t))^2]=0$, then we have $\mathbb{E}[\bar{\varepsilon}^2(t)]=K'<\infty$ is bounded.

Consider the auxiliary $\{\mathcal{F}_t\}$ adapted process $\{z_{s,a}(t)\}$, such that for all t,

$$\begin{aligned}z_{s,a}(t+1)=&\big(1-\alpha_{s,a}(t)\big)z_{s,a}(t)\\&+\alpha_{s,a}(t)\big(\bar{v}_{s,a}(t)+\bar{\varepsilon}_{s,a}(t)\big).\end{aligned}\tag{17}$$

Define the error as $\tilde{Q}_{s,a}(t)=\bar{Q}_{s,a}(t)-z_{s,a}(t)-Q^*_{s,a}$, with (14) and (17) we have

$$\begin{aligned}\tilde{Q}_{s,a}(t+1)=&\big(1-\alpha_{s,a}(t)\big)\tilde{Q}_{s,a}(t)+\frac{\beta_{s,a}(t)}{N}\mathbf{1}_N^TA(t)\eta(t)\\&+\alpha_{s,a}(t)(\bar{\mathcal{Y}}_{s,a}(\bar{Q}_t)-\bar{\mathcal{Y}}_{s,a}(Q^*)).\end{aligned}\tag{18}$$

Theorem V.3. *Consider the undirected connected graph $\mathcal{G}$, under Assumptions IV.1-IV.3, for every (s,a)-pair, if the average value $\bar{Q}_{s,a}(t)$ evolves as follows,*

$$\begin{aligned}\bar{Q}_{s,a}(t+1) =& (1-\alpha_{s,a}(t))\bar{Q}_{s,a}(t) + \alpha_{s,a}(t)[\bar{v}_{s,a}(t) \\ &+ \frac{1}{N}\sum_{i=1}^{N}\mathcal{Y}^i_{s,a}(Q^i_t)] + \frac{\beta_{s,a}(t)}{N}\boldsymbol{1}_N^T A(t)\eta(t),\end{aligned}$$

the expectation of $\bar{Q}_{s,a}(t)$ will converge to the optimal value Q^*, i.e., $\lim_{t\to\infty}\sup\mathbb{E}[||\bar{Q}_{s,a}(t) - Q^*||_\infty] = 0$ a.s. for all (s,a)-pairs.

Proof. Computing the expectation of (18) for both sides implies that

$$\begin{aligned}\mathbb{E}[\tilde{Q}_{s,a}(t+1)] =& \big(1-\alpha_{s,a}(t)\big)\mathbb{E}[\tilde{Q}_{s,a}(t)] \\ &+ \alpha_{s,a}(t)\mathbb{E}[\big(\bar{\mathcal{Y}}_{s,a}(\bar{Q}_t) - \bar{\mathcal{Y}}_{s,a}(Q^*)\big)].\end{aligned}$$

As $Q_{s,a}(t)$ is bounded and Q^* is a fixed constant, there must exist an a.s finite random variable R, such that

$$R = \lim_{t\to\infty}\sup\mathbb{E}[||\bar{Q}_{s,a}(t) - Q^* - z_{s,a}(t)||_\infty]. \tag{19}$$

Then we verify that $R = 0$ by constructing a contrary. Assume that $R \neq 0$, then there exists an event $\mathcal{C}$ of positive measure such that $R > 0$ on $\mathcal{C}$. Let $\delta > 0$ be a constant satisfying $\gamma(1+\delta) < 1$. Then there exists a random time t_δ satisfying

$$\mathbb{E}[||\bar{Q}_t - Q^* - z_{s,a}(t)||_\infty] < R(1+\delta)$$

on the event $\mathcal{C}$ a.s for all $t \geq t_\delta$. Consequently, we have

$$|\mathbb{E}[\tilde{Q}_{s,a}(t+1)]| \leq \big(1-\alpha_{s,a}(t)\big)|\mathbb{E}[\tilde{Q}_{s,a}(t)]| + \alpha_{s,a}(t)\gamma(1+\delta)R.$$

Clearly, with the pathwise application of Proposition 4.1 in [21], it means that

$$\mathbb{P}\Big(\lim_{t\to\infty}\sup|\mathbb{E}[\tilde{Q}_{s,a}(t)]| \leq \gamma(1+\delta)R\Big) \geq \mathbb{P}(\mathcal{C}) > 0.$$

Since $\gamma(1+\delta) < 1$, we have

$$\lim_{t\to\infty}\sup|\mathbb{E}[\tilde{Q}_{s,a}(t)]| < R \quad a.s. \quad on \quad event \quad \mathcal{C}. \tag{20}$$

Clearly, (19) and (20) are contradictory. Then we have

$$\lim_{t\to\infty}\sup\mathbb{E}[||\bar{Q}_t - Q^* - z_{s,a}(t)||_\infty] = 0, \tag{21}$$

which means that the expectation of $\tilde{Q}_{s,a}(t)$ will converge to 0 as $t \to \infty$ a.s. □

Then we will investigate the variance of $\tilde{Q}_{s,a}(t)$. With (18), we have

$$\begin{aligned}&var\big(\tilde{Q}_{s,a}(t+1)\big) = \mathbb{E}[\big(\tilde{Q}_{s,a}(t+1)\big)^2] - \big(\mathbb{E}[\tilde{Q}_{s,a}(t+1)]\big)^2 \\ &\leq \mathbb{E}[\big(\tilde{Q}_{s,a}(t+1)\big)^2] \\ &\leq \mathbb{E}[\Bigg(\big(1-\alpha_{s,a}(t)+\gamma\alpha_{s,a}(t)\big)\tilde{Q}_{s,a}(t) + \frac{\beta_{s,a}(t)}{N} * \mathbf{1}_N^T A(t)\eta(t)\Bigg)^2] \\ &= \big(1-\alpha_{s,a}(t)+\gamma\alpha_{s,a}(t)\big)^2\mathbb{E}[\big(\tilde{Q}_{s,a}(t)\big)^2] \\ &+ \frac{\beta_{s,a}(t)^2}{N^2}\mathbb{E}[\big(\mathbf{1}_N^T A(t)\eta(t)\big)^2] \\ &+ \frac{2\big(1-\alpha_{s,a}(t)+\gamma\alpha_{s,a}(t)\big)\beta_{s,a}(t)}{N}\mathbb{E}[\tilde{Q}_{s,a}(t)\mathbf{1}_N^T A(t)\eta(t)] \\ &= \big(1-\alpha_{s,a}(t)+\gamma\alpha_{s,a}(t)\big)^2\mathbb{E}[\big(\tilde{Q}_{s,a}(t)\big)^2] \\ &+ \frac{\beta_{s,a}(t)^2}{N^2}\mathbb{E}[\big(\mathbf{1}_N^T A(t)\eta(t)\big)^2],\end{aligned} \tag{22}$$

where $A(t) = D(t) - L(t)$. Then we can conclude that $var\big(\tilde{Q}_{s,a}(t)\big) \leq W_0 s_i^2 q_i^{2t-2}\frac{1-(\frac{M_t}{q_i^2})^t}{1-\frac{M_t}{q_i^2}}$, where $M_t = (1 - \alpha_{s,a}(t) + \gamma\alpha_{s,a}(t))^2 \in (0,1)$ and $W_0 = \frac{\beta_{s,a}(0)^2}{N^2}\lambda_N(\bar{D})$. More details refer to Appendix XII.

Theorem V.4 (The (p,r)-accuracy). *The average Q-value $\bar{Q}_{s,a}(t)$ of each state-action pair (s,a) in the DP-QD-Learning algorithm achieves (p,r)-accuracy with its expectaion equals to Q^* and $r = \frac{\sqrt{var(\tilde{Q}_{s,a}(t))}}{\sqrt{p}}$.*

Proof. Let $\breve{Q}_{s,a}(t) = \bar{Q}_{s,a}(t) - Q^*$. As $t \to \infty$, we have the expectation of $\breve{Q}_{s,a}(t)$ as

$$\mathbb{E}[\breve{Q}_{s,a}(t)] = \mathbb{E}[\tilde{Q}_{s,a}(t) + z_{s,a}(t)] = 0. \tag{23}$$

Similarly, the variance of $\breve{Q}_{s,a}(t)$ is

$$\begin{aligned}var(\breve{Q}_{s,a}(t)) &= \mathbb{E}[\breve{Q}^2_{s,a}(t)] - \big(\mathbb{E}[\breve{Q}_{s,a}(t)]\big)^2 \\ &\leq \mathbb{E}[\big(\tilde{Q}_{s,a}(t) + z_{s,a}(t)\big)^2] \\ &\leq 2\mathbb{E}[\tilde{Q}^2_{s,a}(t)] + 2\mathbb{E}[z^2_{s,a}(t)] \\ &= 2var\big(\tilde{Q}_{s,a}(t)\big)\end{aligned} \tag{24}$$

By using Chebyshev's inequality, we obtain

$$\begin{aligned}\mathbb{P}(|\breve{Q}_{s,a}(t)| \leq r) &= 1 - \mathbb{P}(|\breve{Q}_{s,a}(t)| > r) \\ &\geq 1 - \frac{2var(\tilde{Q}_{s,a}(t))}{r^2}.\end{aligned} \tag{25}$$

Choosing $r = \frac{\sqrt{2var(\tilde{Q}_{s,a}(t))}}{\sqrt{p}}$, we have $1 - \mathbb{P}(|\breve{Q}_{s,a}(t)| > r) \geq 1 - p$, which implies that the DP-*QD*-Learning algorithm achieves (p,r)-accuracy with 0. As Q^* is a constant, we can finally get the distribution of $\bar{Q}_{s,a}(t)$ follows the (p,r)-accuracy with its expectaion equals to Q^* and $r = \frac{\sqrt{2var(\tilde{Q}_{s,a}(t))}}{\sqrt{p}}$. □

C. Privacy Analysis

Consider that there exists an external eavesdropper which can access all the transmission information $\hat{Q}^i$ among agents. The DP-*QD*-Learning algorithm aims to protect the real information Q^i of each agent from being restored by the external eavesdropper using observations $\hat{Q}^i$.

Theorem V.5. *The DP-QD-Learning with the decaying Laplace noise η designed as (5) is ϵ-differentially private. The privacy degree satisfies $\epsilon = \max\{\epsilon_i\} = \max\{\frac{|\delta|}{s_i q_i^t}\}, i = 1, \ldots, N$, where δ is the adjacency bound in Defination 2.*

Proof. Consider a pair of δ-adjacent sets $Q_{s,a}(t) = [Q_{s,a}^1(t), \ldots, Q_{s,a}^N(t)]$ and $Q'_{s,a}(t) = [Q_{s,a}^{1'}(t), \ldots, Q_{s,a}^{N'}(t)]$ as the private dataset. Let the observation set be $\hat{Q}_{s,a}(t)$. Then the allowable set for any $t > 0$ can be defined as

$$R_t^{(l)} = \{\eta_t \in \Omega_t | M^{(l)}(Q_{s,a}(t)) \in O_t\}, l = 1, 2, \tag{26}$$

where $\Omega_t = \mathbb{R}^n$ is the sample space at time t, and $O_t \subseteq \mathbb{R}^n$ is a subset of the observation set $\mathcal{O}$. As the random virable observation is only determined by the noise $\eta(t)$, then we have

$$\mathbb{P}\{\eta_t \in \Omega_t | M^{(l)}(Q_{s,a}(t)) \in O_t\} = \int_{R_t^{(l)}} f_{n,t}(\eta_t^{(l)}) d\eta_t^{(l)}, l = 1, 2,$$

where $f_{n,t}$ is the n-dimensional joint Laplace p.d.f at time t given by

$$f_{n,t}(\eta_t) = \prod_{i=1}^{N} \mathcal{L}(\eta_i; \iota_i). \tag{27}$$

Here we define a bijection between $R_t^{(1)}$ and $R_t^{(2)}$ to simplify the analysis. Without loss of generality, we define two sets of noise $\eta_t^{(1)} = \{\eta_1^{(1)}(t), \ldots, \eta_1^{(N)}(t)\}$ and $\eta_t^{(2)} = \{\eta_2^{(1)}(t), \ldots, \eta_2^{(N)}(t)\}$ as

$$\eta_i^{(2)}(t) = \begin{cases} \eta_i^{(1)}(t) + \Delta\eta_i(t) & if \quad i = i_0, \\ 0 & if \quad i \neq i_0, \end{cases}$$

Clearly, we have $Q_{s,a}(t) + \eta_t^{(1)} = Q'_{s,a}(t) + \eta_t^{(2)}$, i.e., $M(Q) = M'(Q')$, which is the strictest case satisfying $R_k^{(1)} = R_k^{(2)}$. Then we can conclude that for any noise selection $\eta_t^{(1)}$, there always exists a unique bijection $\eta_t^{(2)}$ satisfing $\eta_t^{(2)} \in R_k^{(2)}$. The converse argument is also true. As $\Delta\eta_i(t)$ is fixed and independent with $\eta_t^{(2)}$, we have

$$\mathbb{P}\{\eta_t \in \Omega_t | M^{(2)}(Q_{s,a}(t)) \in O_t\} = \int_{R_t^{(1)}} f_{n,t}(\eta_t^{(1)} + \Delta\eta_i(t)) d\eta_t^{(1)}.$$

Then we have

$$\begin{aligned} \frac{\mathbb{P}\{M^{(1)}(Q_{s,a}(t)) \in O_t\}}{\mathbb{P}\{M^{(2)}(Q_{s,a}(t)) \in O_t\}} &= \frac{\int_{R_t^{(1)}} f_{n,t}(\eta_t^{(1)}) d\eta_t^{(1)}}{\int_{R_t^{(1)}} f_{n,t}(\eta_t^{(1)} + \Delta\eta_i(t)) d\eta_t^{(1)}} \\ &= \frac{\int_{R_t^{(1)}} \prod_{i=1}^{N} \mathcal{L}(\eta_i^{(1)}(t); \iota_i(t))}{\int_{R_t^{(1)}} \prod_{i=1}^{N} \mathcal{L}(\eta_i^{(1)}(t) + \Delta\eta_i(t); \iota_i(t))} \\ &\leq e^{\frac{|\Delta\eta_i(t)|}{\iota_i(t)}}, \end{aligned} \tag{28}$$

which means that $\epsilon_i = \frac{|\Delta\eta_i(t)|}{\iota_i(t)}$.

As $Q_{s,a}(t)$ and $Q'_{s,a}(t)$ are assumed to be δ-adjacent, we have $|\Delta\eta_i(t)| = |\delta|$, which implies that $\epsilon_i = \frac{|\delta|}{s_i q_i^t}$. Since the smaller the degree of privacy, the better the privacy protection performance, the privacy degree of the system ϵ equals to the maximum of all agents' privacy degree $\max\{\epsilon_i\}$. □

VI. Experimental Setup

In recent years, some benchmark environments have been developed for MARL experiment such as the particle environment [6], Hanabi [43], and the StarCraft Multi-agent Challenge (SMAC) [3]. However, few of them are developed for networked social systems. For this reason, we design a toy environment called the CBMP environment to evaluate the convergence and the privacy preserving of DP-*QD*-Learning. Moreover, in order to show the generalization potential of the DP mechanism in a broader scope of MARL algorithms, we extend a deep MARL algorithm ConsensusMARL [22] to DPC-MARL, which employs our DP mechanism in the transmitted parameters of the neural networks of the critic. The empirical experiments of DPC-MARL was implemented on a modified CAPC environment [14], [15] with a traffic simulator.

A. The Central Bank Monetary Policy Environment

In this scenario, we consider a common networked social system: the central bank monetary policy regulation. The central bank adjusts the fiscal strategy to urge citizens to change their savings, so as to achieve the expected purpose of tightening or increasing monetary liquidity. Intuitively, when each citizen perceives the monetary policy, due to their own private information such as age, job, health status, and education level, they have different scores (opinions) for the actions and receive the personalized rewards after taking actions. Moreover, people do not only update their strategies based on their rewards but are also influenced by their neighbors' opinions. Finally, after sufficient update steps and exchange of opinions, everyone's strategies tend to be consistent, so that they show stable group behavior.

The globally observable environment $\mathcal{S}$ is the fiscal strategy of the central bank, and the agent is a citizen. The actions $\mathcal{A}^i$ that each agent can take is to adjust its own savings amount, and the personalized and different rewards $r_i(s, a)$ are long-term expected interest income. The Q-table represents citizens' opinions on monetary policy. The eavesdroppers are assumed to be external members of the team and can monitor the communication channel between agents to obtain the Q value. Clearly, there is no centralized controller like CTDE algorithms to collect information and send specific instructions to each agent in CMBP. Besides, the information exchanged and transmitted between citizens is highly related to personal privacy information, which obviously needs to be protected.

From a structuralist perspective, any group system such as business relations and international relations can be described as the sum of individual relationships, which can be illustrated by weight topologies. Therefore, we adopt the network to formalize the exchange and influence of opinions among people. Commonly, this opinion relationship of human communities is modeled as small-world networks or scale-free networks, most of which are randomly connected graphs with specific degree distribution. We provide Fig. 3 to illustrate the CBMP environment.

Although the real monetary policy regulation usually involves a variety of benchmark interest rates, hundreds of millions of participating members, and various influencing factors, we can

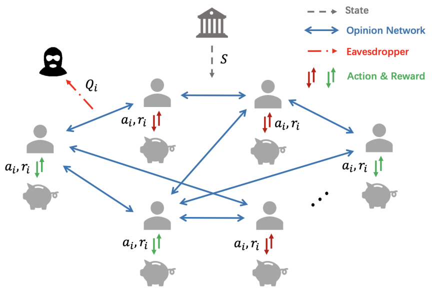


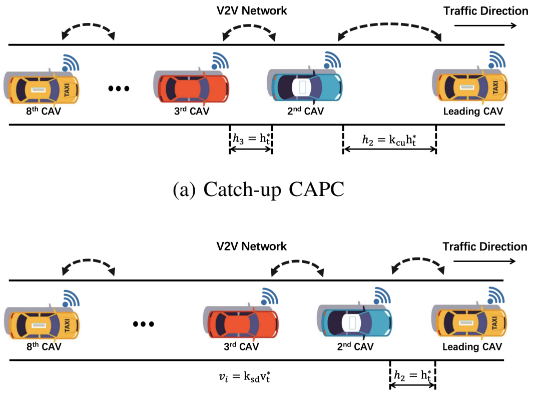
Fig. 3: CMBP Environments.

reasonably simplify the environmental state into interest rate increasing or reducing channel with finite state space $|\mathcal{S}| = 2$. And the action of agents can be simplified as increasing or decreasing the amount of their savings with finite action space $|\mathcal{A}^i| = 2$. Without the loss of generality, we randomly generate a connected graph of $N = 20$ agents as the communication topology in each episode. Therefore, the number of state-action pairs is 4 and Q^i is a $\mathbb{R}^{2\times 2}$ matrix. 8 state transition probabilities were chosen independently by uniformly sampling the interval $[0, 1]$ satisfying that $\sum p(\cdot|s, a) = 1$. For each agent i, the random one stage reward $r_i(s, a)$ is assumed to follow a Gaussian distribution with variance 20, whose expectation $\mathbb{E}[r_i(s, a)]$ for each (s, a) pair is ramdomly sampled over the interval $[50, 400]$. The discounting factor was taken to be $\gamma = 0.7$. The weight sequence was set as $\tau_1 = 1$ and $\tau_2 = 0.2$. We set the decaying Laplace noise in (5) as $s_i = 10$ and $q_i = 0.99$, while the undecaying noise are set as $s_i = 10$ and $q_i = 1$.

B. The Cooperative Adaptive Platoon Control Environment

In the above theoretical analysis of DP-*QD*-Learning, for highlighting the impact of privacy protection, we have reasonably simplified the networked social problem into a tabular Q-learning task. In this subsection, we consider a more concrete networked social system problem with high-dimensional state-action space to illustrate the generalization of the proposed DP mechanism.

Note that with the rapid development of vehicle perception equipments, vehicle to vehicle (V2V) communication and computer vision algorithms, cooperative autonomous vehicles (CAV) have shown great potential in reducing congestion and energy consumption [44], [45]. Here we consider a common scenario of multiple CAVs running in a single lane platoon and formalize it as a CAPC environment. The main objective of CAPC is to simultaneously minimize the car-following headway and the velocity fluctuation of each CAV in the platoon, which can improve road traffic rate and reduce fuel consumption caused by frequent acceleration changes. In the CAPC problem, the communication range is limited on the spatial scale and difficult to afford centralized control strategy. Meanwhile, the information transmitted by CAVs may implicitly or explicitly expose their private information, such as destination, communication protocol, algorithm mechanism, etc., which need to be carefully protected. The experiment platform is based on the simulation of urban mobility (SUMO), which is wildly used in the field of traffic research to simulate various car-following models and visualize the dynamics of the CAV platoon.


V2V Network
Traffic Direction
8th CAV
3rd CAV
2nd CAV
Leading CAV
$h_3 = \mathrm{h}_\mathrm{t}^*$
$h_2 = \mathrm{k}_\mathrm{cu}\mathrm{h}_\mathrm{t}^*$
$v_i = \mathrm{k}_\mathrm{sd}\mathrm{v}_\mathrm{t}^*$
$h_2 = \mathrm{h}_\mathrm{t}^*$

(a) Catch-up CAPC

(b) Slow-down CAPC

Fig. 4: CAPC Environments.

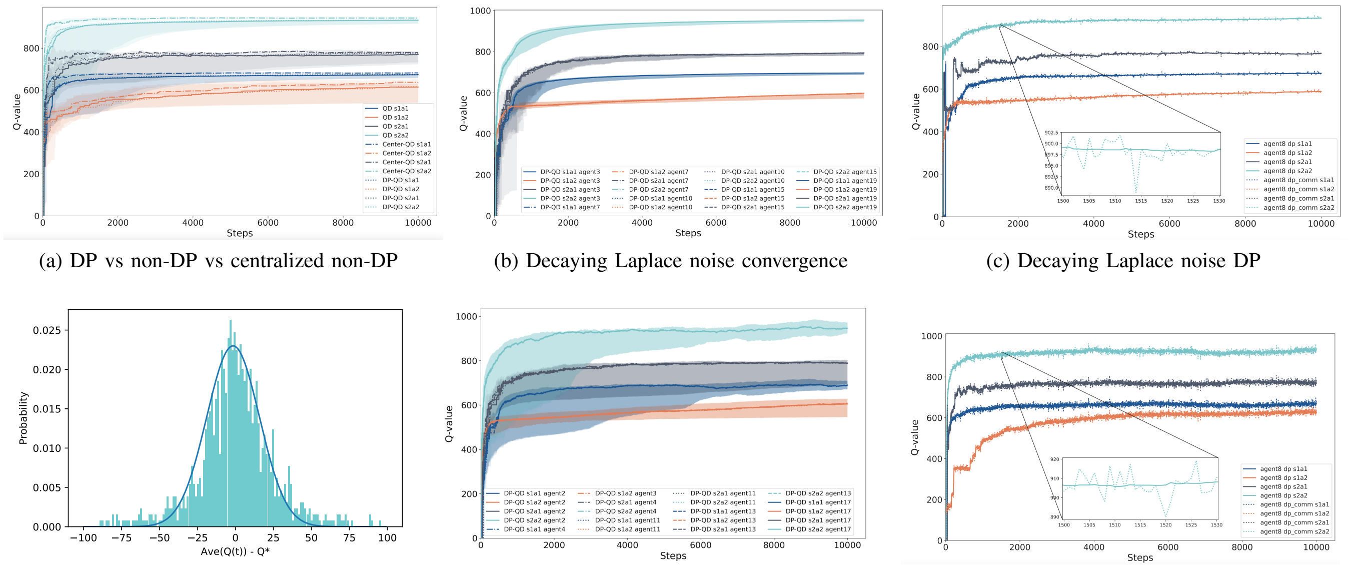
TABLE I: HYPERPARAMETERS IN CAPC

Hyperparameter	Value
Safety headway	$h_i \geq 1m$
Safety velocity	$v_i \leq 30m/s$
Safety acceleration	$\|a_i\| \leq 2.5m/s^2$
Stop headway in OVM	$h_{stop} = 5m$
Full-speed headway in OVM	$h_{full} = 35m$
Collision penalty ($h_{i,t} < 1m$)	1000
Additional cost for collisions	$5(2h_{stop} - h_{i,t})^2$

The V2V communication topology is set as bidirectional following [46], which means that each CAV can share messages with both neighbors in front and back. Following the optimal velocity model (OVM) adopted in [14], the states of CAV $\mathcal{S}_i$ are consist of its headway $h_{i,t} \in \mathbb{R}^+$, velocity $v_{i,t} \in \mathbb{R}^+$, and acceleration $a_{i,t} \in \mathbb{R}^+$. The longitudinal control actions $\mathcal{A}_i$ are the pairs of headway gain α_i° and velocity gain β_i° with four optional levels $\{(0, 0), (0, 0.5), (0.5, 0), (0.5, 0.5)\}$. The control interval is set as $0.1s$ to avoid too frequent accelaration switches, and the overall control time is $60s$. The cost is set as $(h_{i,t} - h_t^*)^2 + (v_{i,t} - v_t^*)^2 + 0.1a_{i,t}^2$, where h_t^* and v_t^* are the target headway and velocity of the CAV platoon. Considering the safety of practical driving, we set additional safety constraints and collision costs as shown in Table I. Here we consider two scenarios, the catch-up CAPC and the slow-down CAPC.

In the catch-up CAPC, the initial states of all CAV except the second CAVs are equal to the target states, which means that $v_{i,0} = v_t^* = 15m/s$ and $h_{i,0} = h_t^* = 20m, \forall i \neq 2$. The states of the third CAV are set as $v_{2,0} = 10m/s$ and $h_{2,0} = k_{cu}h_t^*$ with $k_{cu} \in [3, 4]$. The objective of the catch-up CAPC is to make the last seven CAVs left behind catch up with the first car in the platoon.

In the slow-down CAPC, the initial headways of all CAVs are equal to the target headway and the velocities are equal to a multiple of the target velocity, which means that $h_{i,0} = h_t^* = 20m$ and $v_{i,0} = k_{sd}v_t^*$ with $k_{sd} \in [1.5, 2.5]$ and $v_t^* = 15m/s$. The objective of the slow-down CAPC is to reduce the CAV



(a) DP vs non-DP vs centralized non-DP

(b) Decaying Laplace noise convergence

(c) Decaying Laplace noise DP

(d) Statistic of $\tilde{Q}_{s2,a2}(t)$ over 1000 runs

(e) Undecaying Laplace noise convergence

(f) Undecaying Laplace noise DP

Fig. 5: The learning results in the CMBP task.

platoon from the initial fast driving to the target speed in the first 30s and keep it. The collision avoidance is more important in the slow-down CAPC than the catch-up CAPC.

To illustrate the tradeoff between privacy and performance, we set three levels of DP noise gain as $s_{gain} = \{0.01, 0.1, 1\}$. In each update, the average weight of the critic neural network is calculated as the base scale s_i. Then, the base scale s_i is multiplied by the noise gain s_{gain} and decaying gain q_i^t as the variance of DP noise in this round of communication, i.e., $\eta_i(t) \sim Lap(0, \iota_i(t))$ with $\iota_i(t) = s_{gain} s_i(t) q_i^t$. Here we set $q_i = 1$ as undecaying noise for better privcacy performance.

C. Baselines and algorithm setup

CMBP Task. Since there is no previous article to study the cross-field of networked MARL and privacy protection, we set the baselines of CMBP as a centralized Q-Learning algorithm and the non-privacy protection *QD*-Learning proposed in [21]. We aim for proving that the DTDE framework and the DP mechanism can improve the robustness of MARL algorithms without losing performance, rather than obtaining a higher score in the test.

The centralized algorithm is named as *Center-QD* following the update as

$$Q_{s,a}^{cen}(t+1) = Q_{s,a}^{cen}(t) + \alpha_{s,a}(t)(\frac{1}{N}\sum_{i=1}^{N} r_i(s_t, a_t) + \gamma \max_{a' \in \mathcal{A}} Q_{s',a'}^{cen}(t) - Q_{s,a}^{cen}(t)),$$

where $Q_{s,a}^{cen}$ represents the Q-value of the state-action pair (s, a) of the centralized algorithm Q^{cen}. Here *Center-QD* can be regarded as the existence of a center to collect the information of all agents for centralized training, and select the same action for all agents. All parameters settings of DP-*QD*-Learning, *QD*-Learning, and *Center-QD* are the same. For each execution, we set the maximum step of updates as 10000.

TABLE II: HYPERPARAMETERS OF ALGORITHM 2

Hyperparameter	Value
Training steps	$1Million$
Discount factor	$\gamma = 0.99$
Random seeds	5
Learning rate of actor	5×10^{-4}
Learning rate of critic	2.5×10^{-4}
Batch size	$\lvert\mathcal{B}\rvert = 60$
DP noise gain	$s_{gain} = \{0.01, 0.1, 1\}$

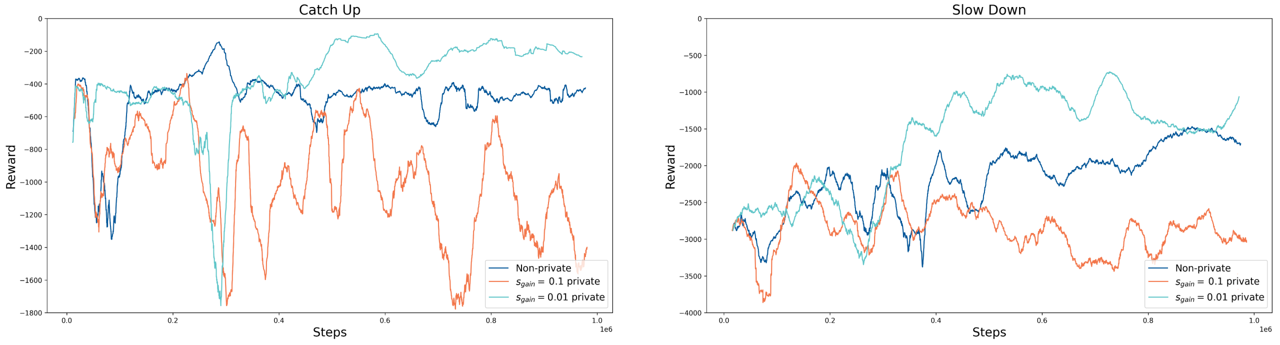
CAPC Task. In ConsensusMARL [22], Zhang considered the similar settings of a DTDE algorithm and proposed an actor-critic MARL algorithm with linear function approximation to handle the high-dimensional state-action pairs. The critic follows the consensus update by passing the parameters $\tilde{\mu}_t^i$ and $\tilde{v}_t^i$ of the critic networks instead of directly passing Q-values. However, when the malicious agent obtains the structure of the critic network, it can still recover the privacy information of the agents.

Motivated by this, we propose DPC-MARL (Alg. 2 in Appendix XIII) by adding Laplace DP noise to the transmitted messages of ConsensusMARL. Here we set the baseline algorithm in CAPC as ConsensusMARL [22]. Both algorithms are applied to A2C agents. The critic networks use 3 layer fully-connected networks, which have 64 units in the hidden layer. The model is trained over 1M steps with discount factor $\gamma = 0.99$ in each episode. The hyperparameter details are in Table II.

VII. Results

A. Results of the CMBP task

First, we investigate the performance of DP-*QD*-Learning, *QD*-Learning, and *Center-QD* in the CMBP task. Fig. 5 shows the Q-values of each state-action pair (s, a) at 5 randomly



(a) CAPC catch up

(b) CAPC slow down

Fig. 6: CAPC learning curves.

selected agents over the networks. Note that the lines and shade show the average value and standard deviation across 5 execution episodes. Each agent collects different rewards and evolves in different trajectories, but eventually converges to the same value under the update mechanism of $innovation + consensus$.

In particular, we compare Q-values of a random agent when it follows the DP-*QD*-Learning with *QD*-Learning and *Center-QD* in Fig. 5a. It can be observed that the DP-protected fully decentralized algorithm reaches a similar performance with unprotected *QD*-Learning and the centralized algorithm. Fig. 5b shows that the proposed DP-*QD*-Learning algorithm successfully ensures the stability and mean square convergence of Q values of all state-action pairs (s, a) with decaying DP noises. With 10^3 times running under the same setup in DP-*QD*-Learning, the empirical results of the distribution of the convergence error $\tilde{Q}$ has been provided in the form of histogram. Due to space limitation, we only show the distribution histogram of $Q^i_{s2,a2}$ in the Fig. 5d. The Q value of other state-action pairs also conforms to a similar normal distribution. Besides, to illustrate the effectiveness of privacy protection, we compare the transmitted information $\hat{Q}^i_{s,a}(t)$ and the real privacy information $Q^i_{s,a}(t)$ in Fig. 5c, which are plotted as the dotted line and the solid line in the figure repectively. Clearly, eavesdroppers cannot restore the true privacy Q-value of the agents according to the one-step transmission Q-value in communication channels.

However, decaying DP noise may decrease to 0 after massive updates, resulting in the risk of privacy leakage. Consequently, we revise the decaying scalar q_i in the Laplace noise (5) into constant 1 as the undecaying DP mechanism. As shown in Fig 5e and Fig.5f, the nondecaying DP mechanism does not change the property of convergence in mean square, but significantly improves the effect of privacy protection by selecting the appropriate noise variance s_i. In addition to the Laplace noise paradigm used in this paper, other kinds of noise mechanisms in the DP community are also suitable for DP-MARL algorithms, such as Gaussian mechanism, exponential mechanism, and so on [30]. The choice of the type of privacy noises will not bring significant changes to the convergence analysis of this paper but only affect the ϵ-privacy degree.

TABLE III: Final Rewards of the CAPC Tasks

Noise Level	Rewards	
	CAPC Catch-up	CAPC Slow-down
Non-DP MARL $s_{gain} = 0$	-519.68 ± 58.20	-1585.82 ± 190.37
DP-MARL $s_{gain} = 0.01$	-187.77 ± 64.82	-1450.16 ± 386.11
DP-MARL $s_{gain} = 0.1$	-1296.81 ± 347.86	-2875.79 ± 291.53
DP-MARL $s_{gain} = 1$	None	None

B. *The results of CAPC*

In CAPC scenarios, we take the rewards of 1 Million training steps as the evaluation. Fig. 6 demonstrates the learning curves of rewards with different noise levels. The training rewards of both the CAPC catch-up task and the CAPC slow-down task can converge to stable values. In 6a, DPC-MARL with the noise level $s_i = 0.01$ outperforms other noise levels and gets the closest performance to the non-protected baseline algorithm. In 6b, the DP-protected MARL algorithm with the noise level $s_i = 0.01$ also gets a similar performance to the baseline. The above results show that the DP mechanism can also effectively protect the privacy of networked agents in complex tasks. In addition, we have been surprised to find that appropriate noise can provide additional rewards improvement slightly. This kind of performance improvement may be due to the random noise in the update, which helps exploration and avoids the local optimum as investigated in [47], [48].

In addition, we can observe that the noise levels s_{gain} significantly affect the variance of the rewards in steps. Both CAPC tasks showed the same trend, that is, the greater the privacy noise, the greater the variance of rewards. Note that the curve with noise level $s_{gain} = 1$ vanished because the large noise disturbs the real parameters of the critic value seriously, resulting in the collapse of the training. Meanwhile, the algorithms with a higher DP noise level $s_{gain} = 0.1$ obtain fewer rewards than the algorithms with lower DP noise. It is due to the fact that the DP noise is significantly larger than the approximation parameters of the critic, which leads to the dominant position of DP noise in the transmitted information and the bias. However, since the expectation of DP noise is 0, it can be found that the reward curves are still stable and converge after smoothing the rewards of multiple steps. We also evaluate the rewards of both CAPC tasks in Tab. III. It can be observed that the CAPC catch-up task usually collects more rewards than the slow-down task. Perhaps because the CAV in the slow-down task suffers more collision risks and penalties.

VIII. Conclusion

We have studied networked MDP for social system problems under the fully decentralized MARL framework. Particularly, we consider potential eavesdroppers and attackers in the environment and design an ϵ-differential private DP-*QD*-Learning algorithm to protect each agent's private information with the decaying additive Laplace noise. We have proved the convergence and distribution with theoretical analysis and demonstrated the promising performance of the algorithm in a newly developed social problem environment CBMP. Besides, by extending our DP mechanism to ConsensusMARL, we verify the effectiveness of the DP-based MARL algorithm on complex task CAPC with high-dimensional state-action space. Simulation results show that the DP mechanism can ensure the convergence of the MARL algorithm with privacy protection and improve the performance with appropriate noise level. Our attempt is the first theoretical work to consider privacy in fully decentralized MARL, hoping to provide a rethink for developing resilient and secured MARL algorithms.

IX. Proof of Lemma 1

Lemma 3. *For each state-action pair (s,a), let $y_{s,a}(t)$ denote the $\{\mathcal{F}_t\}$ adapted process evolving as*

$$\begin{aligned} y_{s,a}(t+1) = (I_N - \beta(t)L(t) - \alpha(t)I_N)y_{s,a}(t) \\ + \alpha(t)v(t) + \beta(t)A(t)\eta(t), \end{aligned} \tag{29}$$

where the weight sequences $\{\alpha(t)\}$ and $\{\beta(t)\}$ are given by equation (3) in the full paper and $\{v(t)\}$ is an $\{\mathcal{F}_{t+1}\}$ adapted magnifying process. Then we have $\lim_{t\to\infty}\mathbb{E}[(y^i_{s,a}(t) - y^j_{s,a}(t))^2] = 0, i,j = 1,\ldots,N$ as $t \to \infty$ a.s.

Proof. Define a positive definite function as

$$P(t) = \sum_{i=1}^{N} \sum_{v_j \in \mathcal{N}_i(t)} a_{ij}(t)(y^i_{s,a}(t) - y^j_{s,a}(t))^2. \tag{30}$$

Then the matrix form of (30) is $P(t) = y^T_{s,a}(t)L(t)y_{s,a}(t)$. Following by (29), we have:

$$\begin{aligned} &P(t+1) \\ =&y^T_{s,a}(t+1)L(t)y_{s,a}(t+1) \\ =&y^T_{s,a}(t)\Psi^T(t)L(t)\Psi(t)y_{s,a}(t) + \alpha(t)y^T_{s,a}(t)\Psi^T(t)L(t)\bar{\mathbf{v}} \\ &+ \alpha(t)\bar{\mathbf{v}}^T L(t)\Psi(t)y_{s,a}(t) + \beta(t)y^T_{s,a}(t)\Psi^T(t)L(t)A(t)\eta(t) \\ &+ \alpha^2(t)\bar{\mathbf{v}}^T L(t)\bar{\mathbf{v}} + \alpha(t)\beta(t)\bar{\mathbf{v}}^T L(t)A(t)\eta(t) \\ &+ \beta(t)\eta^T(t)A^T(t)L(t)\Psi(t)y_{s,a}(t) + \alpha(t)\beta(t)\eta^T(t)A^T(t)L(t)\bar{\mathbf{v}} \\ &+ \beta^2(t)\eta^T(t)A^T(t)L(t)A(t)\eta(t), \end{aligned} \tag{31}$$

where $\Psi(t) = I_N - \beta(t)L(t) - \alpha(t)I_N$.

Taking expectation of both sides of (31) and applying the inequality technology to every term yield that

$$\begin{aligned} \mathbb{E}[P(t+1)] =&\mathbb{E}[y^T_{s,a}(t)\Psi^T(t)L(t)\Psi(t)y_{s,a}(t) \\ &+ \alpha^2(t)\bar{\mathbf{v}}^T L(t)\bar{\mathbf{v}} + \beta^2(t)\eta^T(t)A^T(t)L(t)A(t)\eta(t)] \\ \leq&(1 - \beta(t)\lambda_2(L) - \alpha(t))^2\mathbb{E}[P(t)] + \alpha^2(t)\mathbb{E}[||\bar{\mathbf{v}}||^2|\mathcal{F}_t] \\ &+ \beta^2(t)\lambda_2(A^T L)\mathbb{E}[\eta^2(t)|\mathcal{F}_t] \end{aligned} \tag{32}$$

As $\eta(t)$ is independent with $\{\mathcal{F}_t\}$, then we have $\mathbb{E}[\eta^2(t)|\mathcal{F}_t] = \mathbb{E}[\eta^2(t)] = var(\eta(t)) = 2c_i^2 q_i^{2t}$. Since $var(\eta(t)) \to 0$ as $t \to \infty$ is an decaying noise, then we have

$$\begin{aligned} \mathbb{E}[P(t+1)] \leq&(1 - \beta(t)\lambda_2(L) - \alpha(t))^2\mathbb{E}[P(t)] + \alpha^2(t)\mathbb{E}[||\bar{\mathbf{v}}||^2|\mathcal{F}_t] \\ =&(1 - 2\alpha(t) + \alpha^2(t))\mathbb{E}[P(t)] + \alpha^2(t)K \\ &- (2\beta(t)\lambda_2(L(t)) - \beta^2(t)\lambda_2^2(L(t)) \\ &- 2\alpha(t)\beta(t)\lambda_N(L(t)))\mathbb{E}[P(t)]. \end{aligned} \tag{33}$$

With the parameters $\alpha(t)$ and $\beta(t)$ defined by (3) in the full paper, there exists a positive integer k_0 and a constant $w_3 > 0$, such that for $t \geq T_{s,a}(k_0)$ implies a.s.

$$\begin{gathered} 2\beta(t)\lambda_2(L(t)) - \beta^2(t)\lambda_2^2(L(t)) - 2\alpha(t)\beta(t)\lambda_N(L(t))) \geq 0, \\ 0 < 1 - 2\alpha(t) + \alpha^2(t) \leq 1 - w_3\alpha(t). \end{gathered}$$

Then for every $\varepsilon > 0$, there exists $t_\varepsilon > 0$ such that

$$\mathbb{P}(T_{s,a}(k_0) > t_\varepsilon) < \varepsilon.$$

Now, for a given $\varepsilon > 0$, construct the process $\{P^\varepsilon(t)\}$ as follows

$$P^\varepsilon(t) = \ddagger(T_{s,a}(k_0) \leq t_\varepsilon)P(t), \forall t, \tag{34}$$

where $\ddagger$ denote the corresponding indicator random variable, i.e., $\ddagger(B)$ takes the value one on the event $\mathcal{B}$ and zero otherwise. Then we have

$$\begin{aligned} \mathbb{E}[P^\varepsilon(t+1)] \leq& \ddagger\,(T_{s,a}(k_0) \leq t_\varepsilon)[(1 - c_3\alpha(t))\mathbb{E}[P(t)] + \alpha^2(t)K] \\ \leq&(1 - w_3\alpha(t))\mathbb{E}[P^\varepsilon(t)] + \alpha^2(t)K. \end{aligned} \tag{35}$$

Clearly, (35) satisifies the purview of the Proposition 4.1 in [21], then we have $\mathbb{P}(\lim_{t\to\infty}\mathbb{E}[P^\varepsilon(t)] = 0) = 1$ a.s.

Finally, we have the conclusion that the process $\{\mathbb{E}[P(t)]\}$ converges to zero on the event $\{T_{s,a}(k_0) \leq \varepsilon\}$, which means that

$$\mathbb{P}(\lim_{t\to\infty} P(t) = 0) > 1 - \varepsilon.$$

Since $\varepsilon > 0$ is chosen arbitrary, the mean square convergence of $y^i_{s,a}$ follows by taking ε to zero. □

X. Proof of Theorem 2

Theorem X.1. *Consider the undirected connected graph $\mathcal{G}$, under Assumptions 1-3 in the full paper, for every (s,a)-pair, the Q-value of DP-QD-Learning can achieve consensus in expectation almost surely as*

$$\lim_{t\to\infty}\mathbb{E}[Q^i_{s,a}(t) - \bar{Q}_{s,a}(t)] = 0, i,j = 1,\ldots,N, \tag{36}$$

where $\bar{Q}_{s,a}(t) = \frac{1}{N}\sum_{i=1}^N Q^i_{s,a}(t)$ is the average value of $Q^i_{s,a}$ for all $i \in \mathcal{N}$ at time t.

Proof. Following the process of Lemma 5.2 in [21], we have

$$\tilde{z}_{k+1} =(I_N - \beta(t)L(t) - \alpha(t)I_N)\tilde{z}_k + \alpha_k(\tilde{U}_k + \tilde{J}_k) + \beta_k A(t)\tilde{\eta}_k,$$

where $\tilde{z}_k = z_{s,a}(k) - \frac{1}{N}\sum_{i=1}^N z^i_{s,a}(k)$ and $\tilde{\eta}_k = (1 - \frac{1}{N})\eta_k$. Obviously, this is the same form as Lemma 5.2 of [21], except for the average value of the noise disturbance in the last term. Therefore, we can easily conclude that its expectation is not affected by the noise with the expectation of 0, so we have

$$\begin{aligned} \mathbb{E}[\tilde{z}_{k+1}] =& (I_N - \beta(t)L(t) - \alpha(t)I_N)\mathbb{E}[\tilde{z}_k] + \alpha_k(\mathbb{E}[\tilde{U}_k] + \mathbb{E}[\tilde{J}_k]) \\ \leq& (1 - c_2 r_k)\mathbb{E}[||\tilde{z}_k||] + \alpha_k(||\tilde{U}_k|| + \mathbb{E}[||\tilde{J}_k||]). \end{aligned}$$

Then we get the same conclusion with Lemma 5.2 in [21] that $(k+1)^\tau \mathbb{E}[\tilde{z}_k] \to 0$ as $k \to \infty$ a.s. for all $\tau \in (0, \tau_1 - \tau_2 - 1/(2+\varepsilon_1))$. In particular, $\mathbb{E}[\tilde{z}_k] \to 0$ as $k \to \infty$ a.s. □

XI. The bound of auxiliary process

Lemma 4. *Consider the auxiliary $\{\mathcal{F}_t\}$ adapted process $\{z_{s,a}(t)\}$, such that, for all t,*

$$z_{s,a}(t+1) = (1-\alpha_{s,a}(t))z_{s,a}(t) + \alpha_{s,a}(t)(\bar{\mathbf{v}}_{s,a}(t) + \bar{\varepsilon}_{s,a}(t)) \tag{37}$$

With $\mathbb{E}[\bar{\varepsilon}^2] = K' < \infty$ and $\mathbb{E}[\bar{\varepsilon}] \to 0$, we have $\mathbb{E}[z_{s,a}(t)] \to 0$ and $var(z_{s,a}(t)) \to 0$ as $t \to \infty$ a.s.

Proof: Define a positive function $H_t = z^2_{s,a}(t)$. Then we have as $t \to \infty$

$$\begin{aligned}\mathbb{E}[H_{t+1}] =&\mathbb{E}[((1-\alpha_{s,a}(t))z_{s,a}(t) + \alpha_{s,a}(t)(\bar{\mathbf{v}}_{s,a}(t) + \bar{\varepsilon}_{s,a}(t)))^2]\\ =&(1-\alpha_{s,a}(t))^2\mathbb{E}[z^2_{s,a}(t)] + \alpha^2_{s,a}(t)\mathbb{E}[\bar{\mathbf{v}}^2_{s,a}(t) + \bar{\varepsilon}^2_{s,a}(t)]\\ &+ 2\alpha_{s,a}(t)(1-\alpha_{s,a}(t))\mathbb{E}[z_{s,a}(t)(\bar{\mathbf{v}}_{s,a}(t) + \bar{\varepsilon}_{s,a}(t))]\\ &+ 2\alpha^2_{s,a}(t)\mathbb{E}[\bar{\mathbf{v}}_{s,a}(t)\bar{\varepsilon}_{s,a}(t)]\\ =&(1-a_t)\mathbb{E}[z^2_{s,a}(t)] + b_t(K_r + K')\end{aligned}$$

where $a_t = 2\alpha_{s,a}(t) - \alpha^2_{s,a}(t) \in (0,1)$, $b_t = \alpha^2_{s,a}(t) > 0$, $\sum_{t=1}^{\infty} a_t = \infty$, $\frac{b_t}{a_t} \to 0$ as $t \to 0$, and $K = \mathbb{E}[\bar{\mathbf{v}}^2(t)] < \infty$ and $K' = \mathbb{E}[\bar{\varepsilon}^2(t)] < \infty$.

By using Lemma 3.1 (Polyak, 1987) in [40], we have $\mathbb{E}[H_t] = 0$ as $t \to \infty$. Besides, we have $\mathbb{E}[z_{s,a}(t+1)] = (1-\alpha_{s,a}(t))\mathbb{E}[z_{s,a}(t)] + \alpha_{s,a}(t)\mathbb{E}[\bar{\mathbf{v}}_{s,a}(t) + \bar{\varepsilon}_{s,a}(t)] = 0$ as $t \to \infty$. Then we have $var(z_{s,a}(t)) = \mathbb{E}[z^2_{s,a}(t)] - (\mathbb{E}[z_{s,a}(t)])^2 = 0$.

XII. The bound of the variance

From (22) we have

$$\begin{aligned}\mathbb{E}[(\tilde{Q}_{s,a}(t+1))^2] \leq&(1-\alpha_{s,a}(t) + \gamma\alpha_{s,a}(t))^2\mathbb{E}[\tilde{Q}^2_{s,a}(t)]\\ &+ \frac{\beta_{s,a}(t)^2}{N^2}\mathbb{E}[(\mathbf{1}_N^T D(t)\eta(t))^2]\\ =&M_t\mathbb{E}[\tilde{Q}^2_{s,a}(t)] + W_t\mathbb{E}[\eta^2(t)]\end{aligned} \tag{38}$$

where $M_t = (1-\alpha_{s,a}(t) + \gamma\alpha_{s,a}(t))^2 \in (0,1)$ and $W_t = \frac{\beta_{s,a}(t)^2}{N^2}\lambda_N(\bar{D})$.

Let $\Omega_t = \mathbb{E}[(\tilde{Q}_{s,a}(t))^2]$ and $Z_t = \mathbb{E}[\eta^2(t)] = s_i^2 q_i^{2t}$ Then we have the iteration of (38) as

$$\begin{aligned}\Omega_t \leq&M_{t-1}\Omega_{t-1} + W_{t-1}Z_{t-1}\\ \leq&M_{t-1}M_{t-2}\Omega_{t-2} + M_{t-1}W_{t-2}Z_{t-2} + W_{t-1}Z_{t-1}\\ \leq&\dots\\ \leq&M_{t-1}\dots M_1M_0\Omega_0 + W_{t-1}Z_{t-1} + M_{t-1}W_{t-2}Z_{t-2}\\ &+ \dots + M_{t-1}M_{t-2}\dots M_1W_0Z_0\end{aligned}$$

As $M_t \in (0,1)$, we have $M_tM_{t-1}\dots M_0 \to 0$ as $t \to \infty$. Besides, $M_0 < M_1 < \ldots < M_t$ and $W_0 > W_1 > \ldots > W_t$, then we have

$$\begin{aligned}\Omega_t &\leq W_0Z_{t-1} + M_tW_0Z_{t-2} + M_t^2W_0Z_{t-3} + \ldots + M_t^{t-1}W_0Z_0\\ &\leq W_0(s_i^2q_i^{2t-2} + M_ts_i^2q_i^{2t-4} + M_t^2s_i^2q_i^{2t-6} + \ldots)\\ &= W_0s_i^2q_i^{2t-2}(1 + \frac{M_t}{q_i^2} + \frac{M_t^2}{q_i^4} + \ldots + \frac{M_t^{t-1}}{q_i^{2t-2}})\\ &= W_0s_i^2q_i^{2t-2}\frac{1-(\frac{M_t}{q_i^2})^t}{1-\frac{M_t}{q_i^2}}\end{aligned} \tag{39}$$

XIII. DPC-MARL

Algorithm 2 DPC-MARL

```
Inputs: Initial values of the parameters $\mu_0^i, \omega_0^i, \tilde{\omega}_0^i, \theta_0^i, \forall i \in \mathcal{N}$; the initial state of the environment $s_0$ and the stepsizes $\{\beta_{\omega,t}\}_{t\geq 0}$ and $\{\beta_{\theta,t}\}_{t\geq 0}$. Each agent executes action $a_0^i \sim \pi^i(\cdot|s_0;\theta_0^i)$ and observes joint actions $a_0 = \{a_0^1, \ldots, a_0^N\}$. Initialize the iteration counter $t \leftarrow 0$.
for $t = 0, 1, 2, \ldots,$ do
    for $i = 1, \ldots, N$ do
        Observe the state $s_{t+1}$ and reward $r_{t+1}^i$.
        Update $\mu_{t+1}^i \leftarrow (1-\beta_{\omega,t}) \cdot \mu_t^i + \beta_{\omega,t} \cdot r_{t+1}^i$.
        Select and execute action $a_{t+1}^i \sim \pi^i(\cdot|s_{t+1};\theta_t^i)$.
    end for
    Observe joint actions $a_{t+1} = \{a_{t+1}^1, \ldots, a_{t+1}^N\}$
    for $i = 1, \ldots, N$ do
        TD error: $\delta_t^i = r_{t+1}^i - \mu_t^i + Q_{t+1}(\omega_t^i) - Q_t(\omega_t^i)$.
        Critic Step: $\tilde{\omega}_t^i \leftarrow \omega_t^i + \beta_{\omega,t} \cdot \delta_t^i \cdot \nabla_\omega Q_t(\omega_t^i)$.
        Update $A_t^i \leftarrow Q_t(\omega_t^i) - \sum_{a^i \in \mathcal{A}^i} \pi^i(s_t, a^i; \theta_t^i) \cdot Q(s_t, a^i, a^{-i}; \omega_t^i)$.
        Update $\psi_t^i \leftarrow \nabla_{\theta^i} \log \pi^i(s_t, a^i; \theta_t^i)$.
        Actor Step: $\theta_{t+1}^i \leftarrow \theta_t^i + \beta_{\theta,t} \cdot A_t^i \cdot \psi_t^i$.
        DP Step: $\hat{\omega}_t^i = \tilde{\omega}_t^i + \eta_i(t), \eta_i(t) \sim Lap(0, \iota_i(t))$
    end for
    /* Exchange $\hat{\omega}_t^i$ over the communication network $\mathcal{G}$.
    for $i = 1, \ldots, N$ do
        Consensus Step: $\omega_t^i \leftarrow Average(\hat{\omega}_t^j), j \in \mathcal{N}_i$.
    end for
    Update the iteration counter $t \leftarrow t + 1$.
end for
```

References

[1] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski *et al.*, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529–533, 2015.

[2] M. L. Littman, "Markov games as a framework for multi-agent reinforcement learning," in *Proceedings of the Eleventh International Conference on International Conference on Machine Learning*, 1994, p. 157–163.

[3] T. Rashid, M. Samvelyan, C. Schroeder, G. Farquhar, J. Foerster, and S. Whiteson, "Qmix: Monotonic value function factorisation for deep multi-agent reinforcement learning," in *International Conference on Machine Learning*. PMLR, 2018, pp. 4295–4304.

[4] S. El-Tantawy, B. Abdulhai, and H. Abdelgawad, "Multiagent reinforcement learning for integrated network of adaptive traffic signal controllers (marlin-atsc): methodology and large-scale application on downtown toronto," *IEEE Transactions on Intelligent Transportation Systems*, vol. 14, no. 3, pp. 1140–1150, 2013.

[5] A. Pretorius, S. Cameron, E. van Biljon, T. Makkink, S. Mawjee, J. du Plessis, J. Shock, A. Laterre, and K. Beguir, "A game-theoretic analysis of networked system control for common-pool resource management using multi-agent reinforcement learning," in *Advances in Neural Information Processing Systems*, vol. 33, 2020, pp. 9983–9994.

[6] G. Hu, Y. Zhu, D. Zhao, M. Zhao, and J. Hao, "Event-triggered communication network with limited-bandwidth constraint for multi-agent reinforcement learning," *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–13, 2021.

[7] Z. Sui, Z. Pu, J. Yi, and S. Wu, "Formation control with collision avoidance through deep reinforcement learning using model-guided demonstration," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 32, no. 6, pp. 2358–2372, 2021.

[8] P. Sunehag, G. Lever, A. Gruslys, W. M. Czarnecki, V. Zambaldi, M. Jaderberg, M. Lanctot, N. Sonnerat, J. Z. Leibo, K. Tuyls, and T. Graepel, "Value-decomposition networks for cooperative multi-agent learning based on team reward," in *Proceedings of the 17th International Conference on Autonomous Agents and Multiagent Systems*, 2018, p. 2085–2087.

[9] K. Son, D. Kim, W. J. Kang, D. E. Hostallero, and Y. Yi, "Qtran: Learning to factorize with transformation for cooperative multi-agent reinforcement learning," in *International Conference on Machine Learning*. PMLR, 2019, pp. 5887–5896.

[10] R. Lowe, Y. WU, A. Tamar, J. Harb, O. Pieter Abbeel, and I. Mordatch, "Multi-agent actor-critic for mixed cooperative-competitive environments," in *Advances in Neural Information Processing Systems*, vol. 30, 2017.

[11] J. Foerster, G. Farquhar, T. Afouras, N. Nardelli, and S. Whiteson, "Counterfactual multi-agent policy gradients," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 32, no. 1, 2018.

[12] S. Sukhbaatar, A. Szlam, and R. Fergus, "Learning multiagent communication with backpropagation," in *Advances in Neural Information Processing Systems*, vol. 29, 2016.

[13] P. Peng, Y. Wen, Y. Yang, Q. Yuan, Z. Tang, H. Long, and J. Wang, "Multiagent bidirectionally-coordinated nets: Emergence of human-level coordination in learning to play starcraft combat games," *arXiv preprint arXiv:1703.10069*, 2017.

[14] T. Chu, S. Chinchali, and S. Katti, "Multi-agent reinforcement learning for networked system control," in *International Conference on Learning Representations*, 2020.

[15] M. Li, Z. Cao, and Z. Li, "A reinforcement learning-based vehicle platoon control strategy for reducing energy consumption in traffic oscillations," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 32, no. 12, pp. 5309–5322, 2021.

[16] W. Li, B. Jin, X. Wang, J. Yan, and H. Zha, "F2a2: Flexible fully-decentralized approximate actor-critic for cooperative multi-agent reinforcement learning," *arXiv preprint arXiv:2004.11145*, 2020.

[17] M. Figura, K. C. Kosaraju, and V. Gupta, "Adversarial attacks in consensus-based multi-agent reinforcement learning," *arXiv preprint arXiv:2103.06967*, 2021.

[18] Y. Xie, S. Mou, and S. Sundaram, "Towards Resilience for Multi-agent QD-Learning," *arXiv preprint arXiv:2104.03153*, 2021.

[19] F. D. Garcia and B. Jacobs, "Privacy-friendly energy-metering via homomorphic encryption," in *International Workshop on Security and Trust Management*. Springer, 2010, pp. 226–238.

[20] C. Dwork, F. McSherry, K. Nissim, and A. Smith, "Calibrating noise to sensitivity in private data analysis," in *Theory of Cryptography Conference*, 2006.

[21] S. Kar, J. M. Moura, and H. V. Poor, "*QD*-learning: A collaborative distributed strategy for multi-agent reinforcement learning through *Consensus + Innovations*," *IEEE Transactions on Signal Processing*, vol. 61, no. 7, pp. 1848–1862, 2013.

[22] K. Zhang, Z. Yang, H. Liu, T. Zhang, and T. Başar, "Fully decentralized multi-agent reinforcement learning with networked agents," in *International Conference on Machine Learning*. PMLR, 2018, pp. 5872–5881.

[23] K. Zhang, Z. Yang, and T. Başar, "Networked multi-agent reinforcement learning in continuous spaces," in *2018 IEEE conference on decision and control (CDC)*. IEEE, 2018, pp. 2771–2776.

[24] K. Zhang, Z. Yang, H. Liu, T. Zhang, and T. Başar, "Finite-sample analysis for decentralized batch multiagent reinforcement learning with networked agents," *IEEE Transactions on Automatic Control*, vol. 66, no. 12, pp. 5925–5940, 2021.

[25] G. Qu, A. Wierman, and N. Li, "Scalable reinforcement learning of localized policies for multi-agent networked systems," in *Learning for Dynamics and Control*. PMLR, 2020, pp. 256–266.

[26] G. Qu, Y. Lin, A. Wierman, and N. Li, "Scalable multi-agent reinforcement learning for networked systems with average reward," *Advances in Neural Information Processing Systems*, vol. 33, pp. 2074–2086, 2020.

[27] Y. Lin, G. Qu, L. Huang, and A. Wierman, "Distributed reinforcement learning in multi-agent networked systems," *arXiv preprint arXiv:2006.06555*, 2020.

[28] C. Qu, S. Mannor, H. Xu, Y. Qi, L. Song, and J. Xiong, "Value propagation for decentralized networked deep multi-agent reinforcement learning," in *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[29] B. Wang and N. Hegde, "Privacy-preserving q-learning with functional noise in continuous spaces," in *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[30] T. Zhang, T. Zhu, K. Gao, W. Zhou, and P. S. Yu, "Balancing learning model privacy, fairness, and accuracy with early stopping criteria," *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–13, 2021.

[31] K. Wei, J. Li, M. Ding, C. Ma, H. H. Yang, F. Farokhi, S. Jin, T. Q. S. Quek, and H. V. Poor, "Federated learning with differential privacy: Algorithms and performance analysis," *IEEE Transactions on Information Forensics and Security*, vol. 15, pp. 3454–3469, 2020.

[32] Y. Zhao, J. Zhao, M. Yang, T. Wang, N. Wang, L. Lyu, D. Niyato, and K.-Y. Lam, "Local differential privacy-based federated learning for internet of things," *IEEE Internet of Things Journal*, vol. 8, no. 11, pp. 8836–8853, 2021.

[33] B. Jiang, J. Li, H. Wang, and H. Song, "Privacy-preserving federated learning for industrial edge computing via hybrid differential privacy and adaptive compression," *IEEE Transactions on Industrial Informatics*, pp. 1–1, 2021.

[34] A. Bittau, Ú. Erlingsson, P. Maniatis, I. Mironov, A. Raghunathan, D. Lie, M. Rudominer, U. Kode, J. Tinnes, and B. Seefeld, "Prochlo: Strong privacy for analytics in the crowd," in *Proceedings of the 26th Symposium on Operating Systems Principles*, 2017, pp. 441–459.

[35] D. Bacciu and D. Numeroso, "Explaining deep graph networks via input perturbation," *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–12, 2022.

[36] Z. Huang, S. Mitra, and G. Dullerud, "Differentially private iterative synchronous consensus," in *Proceedings of the 2012 ACM workshop on in the electronic society*, 2012, pp. 81–90.

[37] J. Le Ny and G. J. Pappas, "Differentially private filtering," *IEEE Transactions on Automatic Control*, vol. 59, no. 2, pp. 341–354, 2014.

[38] E. Nozari, P. Tallapragada, and J. Cortés, "Differentially private average consensus: Obstructions, trade-offs, and optimal algorithm design," *Automatica*, vol. 81, pp. 221–231, 2017.

[39] J. Jordon, J. Yoon, and M. Van Der Schaar, "Pate-gan: Generating synthetic data with differential privacy guarantees," in *International conference on learning representations*, 2018.

[40] X.-K. Liu, J.-F. Zhang, and J. Wang, "Differentially private consensus algorithm for continuous-time heterogeneous multi-agent systems," *Automatica*, vol. 122, p. 109283, 2020.

[41] M. Gong, Y. Xie, K. Pan, K. Feng, and A. K. Qin, "A survey on differentially private machine learning," *IEEE Computational Intelligence Magazine*, vol. 15, no. 2, pp. 49–64, 2020.

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

[43] N. Bard, J. N. , S. Chandar, N. Burch, M. Lanctot, H. F. Song, E. Parisotto, V. Dumoulin, S. Moitra, E. Hughes *et al.*, "The hanabi challenge: A new frontier for ai research," *Artificial Intelligence*, vol. 280, p. 103216, 2020.

[44] X. Huang, X. Cheng, Q. Geng, B. Cao, D. Zhou, P. Wang, Y. Lin, and R. Yang, "The apolloscape dataset for autonomous driving," in *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, 2018, pp. 954–960.

[45] M. Zhu, X. Wang, and Y. Wang, "Human-like autonomous car-following model with deep reinforcement learning," *Transportation research part C: emerging technologies*, vol. 97, pp. 348–368, 2018.

[46] Q. Luo, A.-T. Nguyen, J. Fleming, and H. Zhang, "Unknown input observer based approach for distributed tube-based model predictive control of heterogeneous vehicle platoons," *IEEE Transactions on Vehicular Technology*, vol. 70, no. 4, pp. 2930–2944, 2021.

[47] M. Fortunato, M. G. Azar, B. Piot, J. Menick, M. Hessel, I. Osband, A. Graves, V. Mnih, R. Munos, D. Hassabis, O. Pietquin, C. Blundell, and S. Legg, "Noisy networks for exploration," in *International Conference on Learning Representations*, 2018.

[48] M. Plappert, R. Houthooft, P. Dhariwal, S. Sidor, R. Y. Chen, X. Chen, T. Asfour, P. Abbeel, and M. Andrychowicz, "Parameter space noise for exploration," in *International Conference on Learning Representations*, 2018.