Neural Computing and Applications (2023) 35:12979–12994
https://doi.org/10.1007/s00521-020-05243-5

S.I. : DEEP SOCIAL COMPUTING

Evolution of cooperation in malicious social networks with differential privacy mechanisms

Tao Zhang[1] · Dayong Ye[1] · Tianqing Zhu[1] · Tingting Liao[2] · Wanlei Zhou[1]

Received: 1 April 2020 / Accepted: 24 July 2020 / Published online: 3 August 2020

Abstract
Cooperation is an essential behavior in multi-agent systems. Existing mechanisms have two common drawbacks. The first drawback is that malicious agents are not taken into account. Due to the diverse roles in the evolution of cooperation, malicious agents can exist in multi-agent systems, and they can easily degrade the level of cooperation by interfering with agent's actions. The second drawback is that most existing mechanisms have a limited ability to fit in different environments, such as different types of social networks. The performance of existing mechanisms heavily depends on some factors, such as network structures and the initial proportion of cooperators. To solve these two drawbacks, we propose a novel mechanism which adopts differential privacy mechanisms and reinforcement learning. Differential privacy mechanisms can be used to relieve the impact of malicious agents by exploiting the property of randomization. Reinforcement learning enables agents to learn how to make decisions in various social networks. In this way, the proposed mechanism can promote the evolution of cooperation in malicious social networks.

Keywords Evolution of cooperation · Reinforcement learning · Differential privacy · Social network

1 Introduction

Designing mechanisms that promotes the level of cooperation among agents has been a challenge in multi-agent systems (MAS). Many tasks in artificial intelligence (AI) require multiple agents to cooperate. For example, the evolution of cooperation in social dilemmas [1], multi-agent games [2] multi-agent control [3], and community detection [4] all involve cooperation in MAS. As there is an increasing number of applications that involve interactions, the challenge of designing cooperative MAS has been a long-standing goal in AI.

The problem of evolution of cooperation in MAS is often illustrated as the issue in the iterated prisoner's dilemma (PD) game in social networks [5–7]. In the game, each agent has two actions: cooperate (C) and defect (D). When one agent chooses to cooperate, it has a higher risk to be utilized by other agents which may choose to defect. In terms of this logic, empirical research shows that "one-shot" PD game results in a low level of cooperation [8]. In MAS, agents interact in social networks, and interact with each other multiple times, which makes the evolution of cooperation extremely complex. The benefit of improving the level of cooperation is that a higher cooperation level is beneficial to the whole systems. Usually, the Nash equilibrium cannot provide a desirable learning target; the evolution of cooperation becomes difficult to sustain among multiple agents in the long term. Moreover, the evolution of cooperation is hard to adapt in different types of networks. Hence, self-interested interactions among

✉ Tianqing Zhu
Tianqing.Zhu@uts.edu.au

Tao Zhang
Tao.Zhang-3@student.uts.edu.au

Dayong Ye
Dayong.Ye@uts.edu.au

Tingting Liao
tingting.liao@uts.edu.au

Wanlei Zhou
Wanlei.Zhou@uts.edu.au

[1] Centre for Cyber Security and Privacy, School of Computer Science, University of Technology Sydney, Sydney, Australia

[2] Department of Computer Science, Wuhan Polytechnic University, Wuhan, China

agents require the design of incentive mechanisms that motivate agents to cooperate in different types of social networks.

Until now, a large body of mechanisms have been proposed to promote the evolution of cooperation. Recently, a redistribution mechanism was proposed to promote cooperation in the repeated PD game, in which some agents who have a higher payoff share a fraction of their income with neighbors [9]. The uncertain reputation was considered in dynamic networks in the repeated PD game [10]. Leibo et al. [1] studied sequential social dilemmas with Reinforcement learning (RL) where each agent learns how to choose actions in terms of its own deep Q-learning network. In [11], inequity aversion was considered in the evolution of cooperation, and the RL approach was used to promote the level of cooperation in a general-sum Markov game.

Despite many works exploring the ability to promote the evolution of cooperation among multiple agents in MAS, two issues are seldom considered. The first is that most previous works have assumed that all agents are rational when playing the iterative PD game. In fact, malicious agents or individuals are likely to exist in MAS and society [12, 13]. For example, coordinated attacks (e.g., DDoS) in cyberspace are launched via cooperative hackers. Malicious agents may impose negative effects on the evolution of cooperation, which lead to a decrease in the level of cooperation.

The second issue is that, previous mechanisms can achieve a high level of cooperation in some conditions, while exhibiting a lower level in other conditions. The conditions include many factors, such as the initial proportion of cooperators, the types of social networks, and the state update rule [5]. For example, using Win-stay lose-shift (WLSL) rule [14], cooperation evolves in scale-free network when the initial fraction of cooperators is larger than 0.5, while not in random networks. Hence, the evolution of cooperation is easily affected by different conditions. Generally, recent works have two challenges: (1) how should the evolution of cooperation resist the impact caused by malicious agents in MAS; (2) how should the evolution of cooperation adapt in various social networks with different initial cooperators.

To tackle these two challenges, we propose the differentially private reinforcement learning (DP–RL) mechanism to promote the evolution of cooperation in malicious social networks with the *stability* to resist the impact of malicious agents, and the *adaptivity* to fit various situations in static and dynamic social networks. RL can promote the evolution of cooperation by seeking direct benefits and finding the maximum expected reward in the process of continuous interaction with others in different types of social networks. Moreover, we apply differential privacy mechanisms and use its property of randomization to adjust the action of agents to reduce the impact of malicious agents.

We aim to provide a stable and adaptive mechanism to promote the evolution of cooperation in static and dynamic social networks. The contributions of this paper can be summarized as follows:

- To our best knowledge, this is the first work to consider malicious agents in the evolution of cooperation in the social networks. We apply a differential privacy mechanism to adjust the actions of agents, which helps agents make decisions to resist the impact of malicious agents.
- Second, we design a mechanism for the evolution of cooperation based on RL in static and dynamic social networks. This provides a better adaptivity to fit various conditions than other mechanisms in MAS.
- Third, we theoretically analyze our proposed mechanism and conduct experiments to validate the effectiveness of the DP–RL mechanism. Our code is open-sourced on GitHub.[1]

2 Related work

In this section, we first give a summary of related studies and then state the difference between our work and other related studies.

2.1 Review of related works

The PD game has received the most attention in the evolution of cooperation, which frequently occurs in real society, such as price competition, environmental protection. Many mechanisms have been proposed to promote the evolution of cooperation and these mechanisms can be basically divided into direct reciprocity, indirect reciprocity and network reciprocity. Here, reciprocity is defined as occurring when my actions toward you depend on your actions in the past. Also, RL recently became a popular method to promote the evolution of cooperation.

Direct mechanisms The key idea of direct mechanisms is that “I help you, you help me”. The most famous mechanism is Tit-for-tat (TFT) [15], which is a simple mechanism to imitate the action of its opponent in the former round. Later, some mechanisms of its variants were proposed to overcome the occasional mistakes, such as Tit-for-two-tats (TFTS) [16] and Generous TFT (GTFT) [17]. In [18], Novak presented the Imitate-best-neighbor (IBN), where agents imitate the action of the agent who received

[1] https://github.com/dasdsfdfdsfsd/iudfjksldnf.

the highest payoff in the last round. In [14], Nowak proposed another famous mechanism Win-stay, Lose-shift (WLSL) which had a better cooperative level than TFT in the repeated PD game. A WLSL play remains its action, only if its current payoff is higher than the payoff in the previous round.

Indirect mechanisms The key idea of indirect mechanisms is that “I help you, somebody help me” [19]. Recently, many indirect mechanisms were proposed to promote the evolution of cooperation. In [20], a new model of indirect reciprocity was designed to allow reputation building to be costly. In [9], a new redistribution mechanism was proposed to promote cooperation in MAS, in which some agents share a fraction of their income with neighbors.

Network reciprocity Many works were focused on the cooperation in spatially structured networks [21]. Some works have applied mechanisms of indirect reciprocity in complex networks [22]. In [23], Hofmann studied a survey on the topic of the evolution of cooperation in social networks in MAS with different direct mechanisms. Ye and Zhang [7] proposed a self-adaptation mechanism in the evolution of cooperation in social networks, which combines the advantages of different classical direct mechanisms. Chen et al. [24] designed a mechanism to render the individual reputation adaptively changed as the system proceeded. Previous studies show limited effects based on network structures on promoting the level of cooperation [25, 26]. Network dynamics allow agents to change their cooperation by creating or dissolving links with other agents [27, 28]. These models predict that rapid rewiring of the network supports cooperation. However, when the network updates too slowly, the threat of severed links cannot be carried out often enough to make defection maladaptive.

Reinforcement learning Since learning is a popular method applied in MAS, RL has also been investigated in the iterative PD game for the evolution of cooperation in MAS. Ezaki et al. used RL to explain conditional cooperation [29] and network reciprocity [30]. In [31], a simple RL model was applied to enable the evolution of cooperation with the analysis of the adaptive dynamics. Recently, [1] studied sequential social dilemmas with RL; each agent learn the policy with its own deep Q-network. In [32], deep reinforcement learning was used to study the decentralized MAS in high-dimensional environments.

2.2 Discussion of related works

Two issues have seldom been considered in the existing works: (1) the stability to resist the impact of malicious agents in MAS and, (2) the adaptivity to fit in different conditions in MAS.

In the view of [33], everything can be agents in MAS, malicious behaviors in frequency and number of interactions emerges naturally in MAS, yet most existing mechanisms assume that agents are rational in the evolution of cooperation. In [11], the irrational is referred to as inequity aversion. The inequity aversion can change the effective payoff structure by overperforming the payoff or underperforming the payoff than others. The effect of inequity aversion is uncertain in different cases. In our paper, we consider malicious agents in MAS whose goal is to degrade the level of cooperation, and we use a differential privacy mechanism to defend against the interference of malicious agents.

Moreover, for classical mechanisms, the level of cooperation of previous mechanisms is easily affected by factors, such as the initial proportion of cooperators, the structure of networks and updated rules [23]. Results in [23] showed that different conditions could have a huge impact on the emergence of cooperation. In [7], a self-adaptation mechanism was proposed to improve the level of cooperation in different conditions. However, it is assumed that different strategies have been learned in each agent. In this paper, we improve the adaptivity of mechanisms with RL to fit in different conditions.

3 Model description

In this section, we first give an overview of the interactions among agents, then present the rules of PD game and the settings of the malicious agents in MAS.

3.1 Overview of the model

In MAS, we consider a finite set of N agents modeled by structured social networks. For simplicity, we present a simple social network in Fig. 1, showing how agents interact with each other in the structured social network, where agents correspond to nodes, and links correspond to the connections. Two types of agents are considered in the system: common agents (Circle) and malicious agents (Triangle). The goal of common agents is to finish some

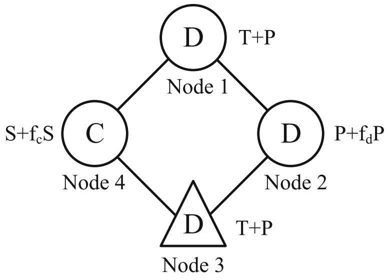

Fig. 1 Interactions between diverse agents in the PD game

tasks in MAS and the goal of malicious agents is to interfere common agents to finish tasks.

In the structured social network, agents interact with each other independently and simultaneously, and they decide whether to Cooperate (C) or Defect (D) in terms of RL strategies. In each interaction, agents will receive the payoff according to the PD rule from neighbors with regard to their actions. From Fig. 1, we know that the agent's actions are related to its types of neighbors and the actions of its neighbors.

3.2 The description of PD game rules

In Fig. 1, each agent makes decisions according to the rules in the repeated PD games and the payoff rule, as shown in Table 1. When agents cooperate with each other, they receive *Reward* payoff (R); when they defect with each other, they receive the *Punishment* payoff (P). When one agent chooses to cooperate, and the other chooses to defect, the first cooperator receives the *Sucker* payoff (S) and the second defector receives the *Temptation* payoff (T) . For example, when Node 1 interacts with its neighbors, Node 2 and Node 4, Node 1 receives the payoff $T + P$. In repeated PD games, payoffs satisfy two conditions: (1) $T > R > P > S$; and (2) $2R > T + S$. A higher value of T states more strict conditions for possible cooperation among agents in the long run.

3.3 The description of malicious agents

Figure 1 shows a malicious agent (node 3) and the impact of the malicious agent by interfering with other agents' payoffs in MAS. Malicious agents can be defined in many forms to destroy the cooperation between agents. Here, the behavior of malicious agents is defined in a simple setting. It is assumed that when a malicious agent interact with other agents, the malicious agent only shows a lower payoff, which is f_c of the real payoff to its neighbors who choose to cooperate. f_c is a smaller payoff parameter that the malicious agent shows to the neighboring agent who choose to cooperate. In contrast, a malicious agent shows a higher payoff, which is f_d of the real payoff to its neighbors who choose to defect. f_d is a higher payoff parameter that the malicious agent shows to the neighboring agent who choose to defect. The range of f_c is larger than 1 and the range of f_d is from 0 to 1. This assumption is reasonable because agents have control of its own payoff, and provide misleading payoff that can degrade the level of cooperation.

Table 1 Payoff matrix

	C	D
C	R	S
D	T	P

According to the PD rule, the neighbor to the left (Node 4) of the malicious agent (Node 3) in Figure 1 receives the real payoff $2S$ and shows the payoff $S + f_c * S$; the neighbor to the right (Node 2) of the malicious agent (Node 3) receives the real payoff $2P$ and shows the payoff $P + f_d * P$. Due to the interference of malicious agents, their neighbors who choose to cooperate receive a lower payoff; their neighbors who choose to defect receive a higher payoff. In this way, malicious agents tend to make neighboring agents defect during the interactions.

4 Background

4.1 Reinforcement learning

RL is a machine learning technique that enables agents to learn in an interactive environment by trial and error, using feedback from their own actions and experiences. From a biological point of view, it is consistent with the method of life cognition and learning the external environment. Basically, it is composed of agents, states, actions, and rewards. At each time step t, the agent enters into a state s^t along with a reward r in terms of the environment setting and chooses an action a^t from the action set. During learning, the agent's action is updated with the policy π. The idea behind RL is that agents will learn how to make actions from the environment via multiple interactions by receiving rewards for performing actions.

In this paper, the Q-learning algorithm is adopted to design the mechanism. It is a value-based RL algorithm that is used to find the optimal action-selection policy $\pi(s, a)$ using a Q-function [34]. In each step, the agent evaluates its action in terms of the payoff from other agents in the PD games and determines the value of taking an action a at a state s in terms of the Q-function. The Q-function can be given in an iterative manner as,

$$Q_{t+1}(s^t, a^t) = (1 - \alpha)Q_t(s^t, a^t) + \alpha[r(s^t, a^t) + \gamma \max_{a^t} Q_t(s^{t+1}, a^t)] \tag{1}$$

where $\alpha \in [0, 1]$ is the learning rate; $\gamma \in [0, 1]$ is the discount factor, which has the effect of valuing rewards received earlier higher than those received later; s^{t+1} is the next state and $\max_{a^t} Q_t(s^{t+1}, a^t)$ is a function that gives the maximum Q-value in state s^{t+1}. Before agents explore the environment, the Q-value gives the same arbitrary fixed value. As agents explore the environment, the Q-value can provide agents a better and better approximation.

4.2 Differential privacy

Differential privacy is a rigorous privacy model, which is widely used in data mining [35] and machine learning [36–38]. In brief, D is a dataset that contains a set of records. Two datasets D and D' are referred to as neighboring datasets when they differ in one record. A query f is a function that maps records $r \in D$ to abstract outputs $f(D) \in \Omega$, where Ω is the whole set of outputs.

Definition 1 (*ϵ-Differential privacy*) [39] A randomized algorithm $\mathcal{M}(D)$ satisfies *ϵ-differential privacy* if for any input pair of D and D', and for any possible outcome $\mathcal{M}(D) \in \Omega$,

$$Pr[\mathcal{M}(D) \in \Omega] \leq \exp(\epsilon) \cdot Pr[\mathcal{M}(D') \in \Omega] \tag{2}$$

where ϵ refers to the privacy budget that controls the privacy level. The lower ϵ represents the higher privacy level.

Definition 2 (*Sensitivity*). [40] For a query $f : D \underset{\mathcal{R}}{\longrightarrow}$, and neighboring datasets, the sensitivity Δf is defined as,

$$\Delta f = \max_{D,D'} ||f(D) - f(D')||_1 \tag{3}$$

Sensitivity describes the maximal difference between neighboring datasets, which is only related to the type of query$f$.

Mechanisms that are used to implement differential privacy algorithms are referred to as differential privacy mechanisms, such as exponential mechanism [41] and Laplace mechanism [42]. In our consideration, differential privacy mechanisms can not only be used to protect data privacy, but can also be used to adjust the scale of parameters [43, 44]. In this paper, we use exponential mechanism to adjust the scale of weights, in order to relieve the impact of malicious agents without identifying them. Exponential mechanism is a technique for designing differentially private algorithms, and the definition is given as,

Definition 3 (*Exponential mechanism*) Given score function $S(D, \phi)$ of a dataset D, the exponential mechanism M satisfies ϵ-differential privacy if

$$\mathcal{M}(D) = \left(\text{return } \phi \propto \exp(\frac{\epsilon S(D, \phi)}{2\Delta f})\right) \tag{4}$$

where score function $S(D, \phi)$ is used to evaluate the quality of an output ϕ and Δf is the sensitivity. This definition implies the fact that the probability of returning ϕ increases exponentially with the increase in the value of $S(D, \phi)$. Exponential noise is generated by exponential mechanism.

5 DP–RL mechanism in static social networks

5.1 Overview of DP–RL mechanism

Figure 2 shows a general description of how agents make decisions in terms of RL when interacting with other neighboring agents in the evolution of cooperation. In the DP–RL mechanism, we adopt the method of Q-learning and the differential privacy mechanism to promote the evolution of cooperation with good adaptivity and stability. The proposed mechanism mainly includes four steps:

(1) Agents explore the environment which is the state of its neighboring agents in the static social network. (2) At each time step, each agent calculates the Q-function with the knowledge learned from the last time step. (3) Differential privacy mechanism is applied to adjust the reward in the Q-function. This can adjust the action of agents in order to resist the interference of malicious agents. (4) Agents choose to cooperate or defect according to the policy, which is updated via the Q-function.

Algorithm 1 describes our DP–RL method in static social networks. The goal of our algorithm is to learn a policy that tells agents whether to cooperate or defect with others in the iterated PD game. Before agents explore the environment, the Q-table gives an initial fixed value. Then, agents explore their neighbor's states, choose an action a, get a reward r, and predict the maximum future reward (Line 4–7). Next, noise is added to the reward in the Q-function to adjust the action of agents (Line 10–11). Thereafter, the policy is updated with the difference of the expected Q-value on the state and its average reward (Line 12). Finally, the limit function is applied to normalize $\pi(s)$ such that it sums to 1 and agents make decisions with the normalized policy π (Line 14–15). The design of the Q-function in Step 2 and the application of differential

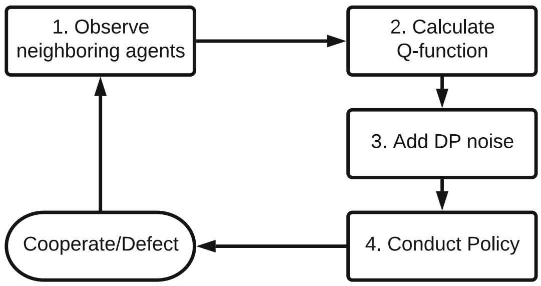

Fig. 2 General description of the proposed method

privacy mechanism in Step 3 are two crucial steps, and we will explain these two steps in detail in the following subsections.

Algorithm 1 DP-RL in static networks

```
Input: $\alpha, \gamma, \epsilon$;
Output: Policy $\pi$;
Initialize: $Q_{value} = 0$;
for $t \in T$ do
    for each agent $i \in N$ do
        Observe neighbors and choose action $a$ at state $s$;
        Calculate the reward $r$ according to Eq (7);
        Predict $Q(s^{t+1}, a^t)$ according to Eq (10);
        Calculate Q-function with $\langle s, a \rangle$;
        Add noise to adjust the reward in Q-function;
        $w_i'' \leftarrow exp(w_i' \epsilon / 2\Delta f)$ according to Eq (3);
        $\bar{r} \leftarrow$ average reward $= \sum_{a \in A} \pi(s,a) Q_t(s,a)$;
        for action $a \in \mathcal{A}$: do
            $\pi(s,a) \leftarrow \pi(s,a) + \gamma(Q_t(s,a) - \bar{r})$;
        end for
        $\pi \leftarrow Normalize(\pi(s))$;
    end for
end for
```

5.2 The design and calculation of the Q-function

In RL, the goal of agents is to maximize the expected cumulative reward. Since the action of agent i is related to its neighbors, the reward depends on the payoff of agent i, denoted as $p_{i,0}$ and the payoff of its neighbors, denoted as $p_{i,nei}$. Furthermore, the reward is related to the number of neighbors of agent i, denoted as $|\beta_i|$, which includes the number of cooperators $n_{i,C}$ and defectors $n_{i,D}$.

Contrary to the agents' goal, the goal of the system is to maximize the level of cooperation. These two goals might not be aligned with each other. Hence, the design of the reward in Q-function is the key to achieve both goals. The updated Q-function contains three parts, as shown in Fig. 3. The first part is the current Q-value $Q(s^t, a^t)$. The second part is the reward r and the third part is the maximum expected future reward. With a carefully designed Q-function, agents will learn how to choose actions to satisfy their own goal, while achieving the system goal. In the following sections, we will explain these parts in detail.

The first part At each step, agent i selects an action a_i^t, observes its surrounding s_i^t, receives the reward, and then enters into a new state s_i^{t+1}. The move depends on both the previous state and the next predicted action.

The second part Reward r consists of two components.

(1) The first component $p_{i,0}$ is the payoff that the agent i receives from its neighboring agents, which can be expressed as,

$$p_{i,0} = \sigma_i(n_{i,D}S + n_{i,C}R) + (1 - \sigma_i)(n_{i,D}P + n_{i,C}T) \tag{5}$$

where σ_i equals 1 if agent i is a cooperator and 0 otherwise.

(2) The second component $p_{i,nei}$ is the reward that neighbors of the agent i receives, which can be expressed as,

$$p_{i,nei} = p_{i,1} + p_{i,2} + \cdots + p_{i,|\beta_i|} \tag{6}$$

Here, the sequence of $p_{i,k}$ is in an increasing order in terms of the values. We use weight w_i to denote the importance of each neighbor's payoff to the reward. When one neighboring agent has a higher payoff, then the corresponding weight is higher. Also, weight can balance the scale of the reward. The weight is expressed as,

$$w_{i,k} = p_{i,k} / \sum_{k=0}^{|\beta_i|} p_{i,k} \tag{7}$$

With the scaling of weight, reward r includes the payoff of the agent itself and its neighbors, which can be denoted as,

$$r_i = p_{i,0} + p_{i,nei} = \sum_{k=0}^{|\beta_i|} w_{i,k} p_{i,k} \tag{8}$$

Since the goal of each agent is to maximize its total reward, agents may not evolve to cooperate in the desire way. A reward for cooperation is essential to achieve the system's goal.

The third part The maximum expected future Q-value is given by the new state s^{t+1} and all possible actions at that state. The agent believes that the actions of neighboring agents are related to historic behaviors. Hence, we compute the empirical frequency of opponent actions over past moves. The estimated probability of choosing to cooperate or defect in the next action can be presented as,

$$P(a_{i,C}^{t+1}) = \frac{1}{t} \sum_{t=1}^{t} [a_i^t = C] \tag{9}$$

$$P(a_{i,D}^{t+1}) = 1 - P(a_{i,C}^{t+1}) \tag{10}$$

The maximum expected reward can be formulated as,

$$\max_{\sigma_i} Q(s_i^{t+1}, a_i^t) = \max_{\sigma_i} \sum_{k=1}^{|\beta_i|} \{P(a_{k,C}^{t+1}) \sigma_i (S + R) + P(a_{k,D}^{t+1})(1 - \sigma_i)(P + T)\} \tag{11}$$

$$Q_t(s^t, a^t) = (1 - \alpha) \underbrace{Q_t(s^t, a^t)}_{\text{The first part}} + \alpha[\underbrace{r(s^t, a^t)}_{\text{The second part}} + \gamma \max_{a^t} \underbrace{Q_{t+1}(s^{t+1}, a^t)}_{\text{The third part}}]$$

Fig. 3 Q-function

Intuitively, each agent has a reputation based on its previous behaviors. Hence, each agent calculates the expected payoff from cooperation or defection, and then makes a decision based on which action achieves a higher expected Q-value. After this calculation, the agent will know what is the best-response action for each state.

5.3 Adding noise with a differential privacy mechanism

We use a exponential mechanism to adjust the actions of agents in the proposed DP–RL method. In the system, agents are not aware of who are malicious agents in the neighborhood, which makes it difficult to mitigate the impact of malicious agents. Differential privacy is a privacy model which ensures that changing one record in the dataset will not affect too much of the output. This model is achieved by differential mechanisms which calibrate some noise to the output. We use this property of differential privacy to calibrate noise to the weights, ensuring that the reward of agents is not affected by malicious agents too much.

First, we define two variants of the weight: (1) interference weight w'_i, and (2) adjusted weight w''_i. Interference weight w'_i is the weight w_i after the interference of malicious agents. Due to the malicious agent's impact, the payoff of neighboring agents change, and the weight will change from weight w_i to interference weight w'_i. Adjusted weight w''_i is the interference weight w'_i after adding differential privacy noise. The calculation of w'_i and w''_i is similar to the calculation of w_i in Eq. (7).

The exponential mechanism is given in Eq. (3) and includes three variables: score function $S(D, \phi)$, sensitivity ΔS, and privacy budget ϵ. In the following, we will explain how to apply these variables to add exponential noise to the weights in the reward.

Score function Exponential noise is added to interference weight w'_i, and thus the score function is

$$S(D, \phi) = w'_i \tag{12}$$

Here, the input D in the exponential mechanism is interference weight w'_i and the output ϕ is adjusted weight w''_i. Assume that agent i has a number of $(n+1)$ common agents and $(|\beta_i| - n)$ malicious agents in its neighborhood. The interference reward can be denoted as,

$$r'_i = \sum_{k=0}^{n} w'_{i,k} p_{i,k} + \sum_{k=n+1}^{|\beta_i|} w'_{i,k} p'_{i,k} \tag{13}$$

where $r'_{i,k}$ is the interference payoff given by the neighboring malicious agent k. Malicious agents can show a higher payoff to defectors and a lower payoff to cooperators, which will mislead agents to make decisions. After applying differential privacy mechanism, we obtain the adjusted reward r'' denoted as,

$$r''_i = \sum_{k=0}^{n} w''_{i,k} p_{i,k} + \sum_{k=n+1}^{|\beta_i|} w''_{i,k} p'_{i,k} \tag{14}$$

To relieve the impact of malicious agents on the interference reward r', the goal of differential privacy mechanism is make the adjusted reward r'' closer to the original reward r_i.

Sensitivity The sensitivity Δf equals 1. In Algorithm 1 each agent in MAS makes decisions according to the policy π, which is updated via the Q-function. We add exponential noise to the weights in r in the Q-function. In exponential mechanism, since the input of interference weight is $w'_i \in [0, 1]$, and the output of adjusted weight w''_i is also in the range of [0, 1] with normalization. Hence, the maximal change between input, and the output is 1 and sensitivity Δf equals 1.

Privacy budget Privacy budget in differential privacy controls the privacy level, and in this paper it controls the scale of weights. At each time step t, the privacy budget added to the agent i is $\frac{\epsilon}{(2\Delta f|\beta_i|t)}$. For each agent, the whole privacy budget is ϵ, and it is averagely distributed to its neighboring agents at each step t. In Sect. 6.1, we will provide a theoretical analysis on how to calculate the privacy budget.

5.4 Discussion

Intuitively, RL is a proper method for the evolution of cooperation, because it is able to establish conditional reflex, seeking benefits and avoiding disadvantages, and finding the best strategy for survival in the process of continuous interaction with others in the environment.

In the design of the reward in the Q-function, agents with the immediate reward of cooperation would learn towards cooperation policies to maximize the expected accumulated benefits, and eventually, maximizing only their own reward would learn more selfish policies. Hence, the action of an agent should be relevant to its neighbors. In other words, the reward for an action should not only consider the agent itself, but should also consider its neighbor's actions. Based on this, we design the reward in the Q-function to consider the agent's payoff and its neighbors' payoff. Moreover, the maximum expected future Q-value relates to the agents historical actions. One agent who has a higher probability of choosing to cooperate will receive a higher expected Q-value.

When malicious agents exist in MAS, they interfere with the cooperation among agents via the reward of agents. This enlarges the scale of reward and weights.

Hence, one method of relieving malicious agents' impact is to adjust the scale of interference weight. The key to exponential mechanism in differential privacy is that it adjusts the scale of weights by varying the privacy budget. Based on this observation, we adopt exponential mechanism to add noise to adjust the scale of interference weight.

6 DP–RL mechanism in dynamic social networks

6.1 Overview

In this section, we apply the DP–RL in dynamic social networks. In dynamic networks, the connection among agents changes over time. Agents may leave the network and new agents may enter; the links between agents can be formed and dismissed as well. Hence, a dynamic network is a more general case in the evolution of cooperation.

The idea for designing the Q-function in dynamic networks is similar to the design in static networks. The difference is that the agent needs to learn more knowledge of its neighbor's states. In dynamic networks, neighbors of agents are changing with the network update. In other words, the number of neighboring agents and the state of neighboring agents changes. One agent needs more time to experience more possible situations of its neighbors, which takes more time to learn.

Algorithm 2 DP-RL in dynamic networks

```
Input: $\alpha, \gamma, \epsilon$;
Output: Policy $\pi$;
Initialize: $Q_{value} = 0$;
for $t \in T$ do
    for each agent $i \in N$ do
        Observe neighbors and choose action $a$ at state $s$;
        if the network updates then
            $Q^{new}_{value} \leftarrow \overline{Q}^{current}_{value}$;
        end if
        Calculate the reward $r$ according to Eq (7);
        Predict $Q_t(s^{t+1}, a^t)$ according to Eq (10);
        Calculate Q-function with $\langle s, a \rangle$;
        Add noise to adjust the reward in Q-function;
        $w''_i \leftarrow exp(w'_i\epsilon/2\Delta f)$ according to Eq (3);
        $\bar{r} \leftarrow$ average reward= $\sum_{a \in A} \pi(s,a)Q_t(s,a)$;
        for action $a \in \mathcal{A}$: do
            $\pi(s,a) \leftarrow \pi(s,a) + \gamma(Q_t(s,a) - \bar{r})$;
        end for
        $\pi \leftarrow Normalize(\pi(s))$;
    end for
end for
```

6.2 The design of the Q-function

In general, the DP–RL mechanism in dynamic networks also has four steps as shown in Algorithm 2 . The goal of Algorithm 2 is to learn a policy, which tells agents how to cooperate with other agents in the iterated PD game when their connections with others are dynamic. In step 1, an initial fixed value is given and agents start to explore the environment. As the dynamic network varies, some agents appear new neighbors, and some neighbors disappear. Agents will learn how to make decisions in new surroundings with RL. In step 2, the Q-function provides better and better approximations by continuously updating the Q-values from each interaction among agents. In step 3, to resist the impact of malicious agents, the exponential mechanism in differential privacy is used to adjust the scale of weight. In the final step, the normalized policy p_i is learned with limited function.

In order to learn more efficiently, we adjust the initial Q-value for new states. When new states appear, the corresponding new Q-value is given. Initially, all the Q-values are set to 0. After learning via some epochs, agents learn how to interact with neighboring agents according to the Q-table. When the network updates, it is likely that some new neighbors will emerge, which means new knowledge needs to be learned to fit in the new surroundings (new states). If the network updates frequently, the learning process for new surroundings might be slow. Intuitively, we set the Q-value for new states as the mean of all other current Q-values. This is because the knowledge of old neighbors might be more useful than no knowledge of new neighbors. The new Q-value is defined as

$$Q^{new}_{value} = \overline{Q}^{current}_{value} \tag{15}$$

For the historical action records, new neighbor's historical action records are set to an equal probability to cooperate or defect.

The application of exponential mechanism in dynamic networks is similar to that of static networks. The malicious agents still have an impact on the level of cooperation in dynamic networks and we use exponential mechanism in differential privacy to adjust weights to resist the impact of malicious agents.

6.3 Discussion

The level of cooperation can be sustained in dynamic structured social networks. The key knowledge in our proposed method that agents have learned is how to choose actions when there are a number of cooperators and defectors in the neighborhood. The number of neighbors and the state of those neighbors is more significant knowledge to learn; who the neighbors are is less important because the weight for the expected Q-value is smaller. Therefore, our proposed mechanism can improve the level of cooperation in dynamic social networks.

7 Theoretical analysis

7.1 The analysis of differential privacy

In this paper, we apply differential privacy mechanism to adjust the scale of weights to relieve the impact of malicious agents. The privacy budget determines the extent of the adjustment on weights.

Theorem 1 *The proposed DP–RL method satisfies ϵ-differential privacy.*

Proof The key to proving that the proposed method satisfies ϵ-differential privacy is to analyze the privacy budget that is consumed in the exponential mechanism. Two compositions are involved: *sequentialcomposition* [41] and *parallelcomposition* [45].

Lemma 1 *Sequential composition: Suppose that a set of privacy mechanisms $\mathcal{M} = \{\mathcal{M}_1, ..., \mathcal{M}_m\}$, gives ϵ_i differential privacy $(i = 1, 2..., m)$, and these mechanisms are sequentially performed on a dataset. $\mathcal{M}$ will provides $(\sum_i \epsilon_i)$-differential privacy for this dataset.*

Lemma 2 *Parallel composition: Suppose that a set of privacy mechanism $\mathcal{M} = \{\mathcal{M}_1, ..., \mathcal{M}_m\}$, gives ϵ_i differential privacy $(i = 1, 2..., m)$, and these mechanisms are performed on the disjoint subsets of a entire dataset. $\mathcal{M}$ will provides max(ϵ_i)-differential privacy for this dataset.*

We analyze the privacy budget ϵ. For each agent i at time step t, $\mathrm{Exp}(i, t) = \exp(\frac{\epsilon s(D,r)}{2\Delta f|\beta_i|t})$ is calculated according to Eq. (3), adding to agent i's weight. The amount of noise added to agent i's weight depends on the number of its neighbors $|\beta_i|$. Since the noise is averagely distributed to neighbors according to Lemma 1, the weight at each time t satisfies $\frac{\epsilon}{t}$-differential privacy. In addition, there are t steps involved; the payoff for each agent satisfies ϵ-differential privacy in terms of Lemma 2.

As each player guarantees ϵ-differential privacy, the proposed method again guarantees overall ϵ-differential privacy according to Lemmas 1 and 2. The proof is applicable for the DP–RL mechanism in static and dynamic networks. □

7.2 The analysis of resistance on malicious agents

In the MAS, malicious agents mislead neighboring agents' actions by interfering the reward of agents. Hence, to relieve the impact of malicious agents, we need to make the malicious reward closer to the original reward. The malicious agent has much higher impact when the interactive agent choose to defect than choose to cooperate. This is because the higher reward encourages the defection and thus degrade the level of cooperation. Our analysis of resistance on malicious agents focuses on how to reduce the interference reward that encourages the defection. Before analysis, we define two types of utility loss to measure the impact on malicious agents, and the impact on differential privacy mechanisms.

Definition 4 (*Malicious utility loss*) The malicious utility loss is defined as the difference between the original reward and the malicious reward, which is presented as,

$$\begin{aligned}\mathcal{U}_m &= r'_i - r_i \\ &= \sum_{k=0}^{n}(w'_{i,k} - w_{i,k})p_{i,k} + \sum_{k=n+1}^{|\beta_i|}(w'_{i,k}f_d - w_{i,k})p_{i,k}\end{aligned} \tag{16}$$

Malicious utility loss indicates the impact of malicious agents on the reward. A higher malicious utility loss means a higher malicious impact.

Definition 5 (Adjusted utility loss) The adjusted utility loss is defined as the difference between the original reward and the adjusted reward, which is presented as,

$$\begin{aligned}\mathcal{U}_a =& r''_i - r_i \\ &= \sum_{k=0}^{n}(w''_{i,k} - w_{i,k})p_{i,k} + \sum_{k=n+1}^{|\beta_i|}(w''_{i,k}f_d - w_{i,k})p_{i,k}\end{aligned} \tag{17}$$

Adjusted utility loss indicates the impact of exponential mechanism on the reward. A lower adjusted utility loss has a better resistance to malicious agents' interference. The application of exponential mechanism is to reduce the malicious utility loss.

Theorem 2 *Exponential mechanism help to reduce malicious utility loss when* $\frac{1}{w'_{i,|\beta_i|} - w'_{i,n+1}} \ln(\frac{1}{(|\beta_i|-n)w'_{i,n+1}}) \leq \epsilon \leq \frac{\ln(\frac{|\beta_i|+1}{|\beta_i|-n}((w'_{i,n+1} + \cdots + w'_{i,|\beta_i|}) + \frac{C}{f_d p_{i,n+1}}))}{w'_{i,n+1} - w'_{i,|\beta_i|}}$.

Proof To prove that exponential mechanism is useful to reduce the malicious utility loss, we need to prove $\mathcal{U}_m - \mathcal{U}_a \geq 0$ so that the adjusted reward is closer to the original reward. First, $\mathcal{U}_m - \mathcal{U}_a$ can be calculated as follows,

$$\begin{aligned}&\mathcal{U}_m - \mathcal{U}_a \\ &= \sum_{k=0}^{n}(w'_{i,k} - w''_{i,k})p_{i,k} + \sum_{k=n+1}^{|\beta_i|}(w'_{i,k} - w''_{i,k})f_d p_{i,k}\end{aligned} \tag{18}$$

Note that Eq. (18) has two parts. In the first part, the reward is not affected by malicious agents because not all agents have malicious neighbors. The reward in this part does not change and the weight is affected by malicious agents because the sum of weight equals one. In the second part, the reward and weight are affected by malicious agents. Exponential mechanism is applied to all weights because malicious agents are unknown for common agents. The goal of exponential mechanism is to reduce the interference weight corresponding to the malicious reward. In other words, $w''_{i,k}$ should be smaller than $w'_{i,k}$ in the second part in Eq. (18). Hence, to reduce the malicious utility loss with differential privacy mechanisms, two conditions need to be satisfied in Eq. (18): 1) $\mathcal{U}_m - \mathcal{U}_a \ge 0$; 2) $\sum_{k=n+1}^{|\beta_i|}(w'_{i,k} - w''_{i,k}) \ge 0$.

To satisfy the first condition, for simplicity, we denote the minimum of the first part $\sum_{k=0}^{n}(w'_{i,k} - w''_{i,k})p_{i,k}$ in E (18) as $-C(C > 0)$. Then, we can rewrite Eq. (18) as,

$$\sum_{k=n+1}^{|\beta_i|}(w'_{i,k} - w''_{i,k})f_d p_{i,k} \ge C \tag{19}$$

When using $e^{\epsilon w'_{i,k}}$ to replace $w''_{i,k}$ and $p_{i,n+1}$ to replace $p_{i,k}$, we can obtain Inequation (20).

$$(w'_{i,n+1} + \cdots + w'_{i,|\beta_i|}) + \frac{C}{f_d p_{i,n+1}} \ge \frac{e^{\epsilon w'_{i,n+1}} + \cdots + e^{\epsilon w'_{i,|\beta_i|}}}{e^{\epsilon w'_{i,1}} + \cdots + e^{\epsilon w'_{i,|\beta_i|}}} \tag{20}$$

With shrinking the numerator and enlarging the denominator in the right of Inequation (20), we can transform Inequation (20) to Inequation (21).

$$(w'_{i,n+1} + \cdots + w'_{i,|\beta_i|}) + \frac{C}{f_d p_{i,n+1}} \ge \frac{(|\beta_i| - n)e^{\epsilon w'_{i,n+1}}}{(|\beta_i| + 1)e^{\epsilon w'_{i,|\beta_i|}}} \tag{21}$$

After simplification, we can obtain the upper bound as,

$$\epsilon \le \frac{\ln(\frac{|\beta_i|+1}{|\beta_i|-n}((w'_{i,n+1} + \cdots + w'_{i,|\beta_i|}) + \frac{C}{f_d p_{i,n+1}}))}{w'_{i,n+1} - w'_{i,|\beta_i|}} \tag{22}$$

Now we need to prove the second part in Eq. (18). The goal is to prove $(w'_{i,k} - w''_{i,k}) \ge 0$ for $k \in [n+1, |\beta_i|]$

$$\begin{aligned}
&(w'_{i,k} - w''_{i,k}) \ge 0 \\
&\quad = (w'_{i,k} - \frac{e^{\epsilon w'_{i,k}}}{e^{\epsilon w'_{i,k}} + \cdots + e^{\epsilon w'_{i,|\beta_i|}}}) \ge 0 \\
&\quad = \frac{w'_{i,k}(e^{\epsilon w'_{i,k}} + \cdots + e^{\epsilon w'_{i,|\beta_i|}}) - e^{\epsilon w'_{i,k}}}{e^{\epsilon w'_{i,k}} + \cdots + e^{\epsilon w'_{i,|\beta_i|}}} \ge 0
\end{aligned} \tag{23}$$

Note that the denominator is always positive in Inequation (23). To ensure the numerator positive, the lower bound of the Inequation (23) is achieved when we use $e^{\epsilon w'_{i,|\beta_i|}}$ to replace all $e^{\epsilon w'_{i,k}}$. Then Inequation (23) can be written as,

$$w'_{i,k}((|\beta_i| - n)e^{\epsilon w'_{i,|\beta_i|}}) \ge e^{\epsilon w'_{i,k}} \tag{24}$$

When using $k = n+1$ in the Inequation (24), we can obtain the lower bound as,

$$\epsilon \ge \frac{1}{w'_{i,|\beta_i|} - w'_{i,n+1}} \ln\left(\frac{1}{(|\beta_i| - n)w'_{i,n+1}}\right) \tag{25}$$

□

We prove that exponential mechanism can help to reduce malicious utility loss when ϵ is in the range denoted in Eqs. (22) and (25). The range of ϵ is important to resist the malicious impact. This is because ϵ determines the extent of the adjustment on weights. When privacy budget ϵ is large, the exponential mechanism can have a larger adjust for the weights affected by malicious agents, while having a smaller adjust for the weights of common agents, and vice versa.

7.3 The analysis of convergence

Theorem 3 *DP–RL mechanism is convergent (1) the reward is set properly;* (2) learning rate α *is decayed gradually.*

Proof To prove the convergence of the proposed DP–RL mechanism, we need to prove that the Q-value is convergent, i.e., $\lim_{x\to\infty}[Q_{t+1}(s^t, a^t) - Q_t(s^t, a^t)] \to 0$. According to Eq. (1), we have

$$\begin{aligned}
&Q_{t+1}(s^t, a^t) - Q_t(s^t, a^t) \\
&\quad = Q_{t+1}(s^t, a^t) - Q_t(s^t, a^t) \\
&\quad = (1-\alpha)Q_t(s^t, a^t) + \alpha[r(s^t, a^t) + \gamma Q_t(s^{t+1}, a^t) - Q_t(s^t, a^t) \\
&\quad = -\alpha Q_t(s^t, a^t) + \alpha[r(s^t, a^t) + \gamma Q_t(s^{t+1}, a^t)]
\end{aligned} \tag{26}$$

To prove the convergence, we need to prove

$$\begin{aligned}
&\lim_{t\to\infty}[Q_{t+1}(s^t, a^t) - Q_t(s^t, a^t)] \to 0 \\
&\quad \lim_{t\to\infty}[-\alpha[Q_t(s^t, a^t) - r(s^t, a^t) - \gamma Q_t(s^{t+1}, a^t)]] \to 0
\end{aligned} \tag{27}$$

When α is decayed gradually, the mechanism is obviously convergent [46]. When α is not decayed gradually, the mechanism can also be convergent. Note that $Q_t(s^{t+1}, a^t)$ is the maximum expected future Q-value, which is calculated according to agents' historical behaviors. After many steps, agents realize which action is likely to receive a higher benefit in its situation (neighboring agents' historical behaviors) and neighboring agents' historical behaviors tend to be easier to predict correctly with a high probability. Therefore, $Q_t(s^{t+1}, a^t)$ and $Q_t(s^t, a^t)$ begins to converge gradually. When the reward $r(s^t, a^t) = Q_t(s^t, a^t) - \gamma Q_t(s^{t+1}, a^t)$, then the algorithm is convergent. □

7.4 The analysis of complexity

Theorem 4 *The overall time complexity of the proposed DP–RL method is $O(T \cdot N)$, where N is the number of agents in MAS and T is the number of time steps.*

Proof Algorithm 1 has three loop bodies, two loops in line 2–16 and one loop in line 11–13. The number of iterations of the first and second loop is T and N and the third loop is 2 because there are only 2 actions available. Therefore, the overall time complexity of Algorithm 1 is $O(T \cdot N)$. Algorithm 2 also has three loop bodies, two loops in line 2–19 and one loop in line 14–16. The complexity analysis is similar to the Algorithm 1. The overall time complexity of two algorithms is $O(T \cdot N)$. The difference is that the iterations in dynamic network is larger than the iterations in static networks due to the fluctuations in dynamic networks. Hence, the time complexity in dynamic networks is larger than the time complexity in static networks. □

8 Experiments

In this section, we first describe the experimental setup, including baseline methods, network structures, and parameters. Then, we present the proposed method in static and dynamic networks.

The general aim of our proposed method is to improve the evolution of cooperation with a desirable stability and adaptivity in static and dynamic networks. Stability is tested with two variables—the proportion of malicious agents and the scale of DP noise. Adaptivity is tested with two variables—the initial cooperation level and structured social networks. Both of them will be evaluated by the final level of cooperation. This is defined as the final number of cooperators n^C of the whole number of agents N in MAS, which can be expressed as,

$$c_{\text{final}} = n^C / N \tag{28}$$

8.1 Experimental setup

Proposed mechanisms We perform the proposed RL mechanism in MAS where malicious agents exist and do not exist, and DP–RL mechanism in MAS where malicious agents exist.

- **DP-RL mechanism**, where agents make decisions with the proposed RL and the DP mechanism in MAS where malicious agents exist.
- **RL mechanism**, where agents make decisions with the proposed RL in MAS where no malicious agent exists.
- **RL-Malicious mechanism**, where agents make decisions with the proposed RL in MAS where malicious agents exist.

Baselines Three suitable baselines are considered for the experiments. IBS and IBN are representative mechanisms since many other mechanism are developed on the basis of these schemes; redistribution scheme was proposed recently, which had a good performance in the evolution of cooperation.

- **Imitate–best–neighbor** (IBN), where each agent imitates the action of the wealthiest agent (including itself) in the next round [18].
- **Imitate–best–strategy** (IBS), where each agent adopts the strategy that accumulates the highest payoff in its neighborhood [23].
- **Local–redistribution–strategy** (LRS), where wealthy agents in MAS share a fraction of their income with neighbors [9].

We also perform these mechanisms in MAS where malicious agents exist, called IBN-Malicious, IBS-Malicious, LRS-Malicious mechanisms.

Network structure Three types of social networks are considered in the experiments.

- **Homogeneous network** In a homogeneous network, each node has the same number of connections n. The number of connections each node has is the same. Here, we set each node to have $n = 4$ connections with other nodes.
- **Random network** In a random network, the link between nodes is set with a connected probability p. The probability denotes that a node has k connections following a binomial distribution $B(n-1, p)$, where n is the number of nodes. When n is large and $p \leq 0.5$, the distribution of node degree can be modeled by a Poisson distribution $P(x = k) = e^k \frac{\lambda^k}{k!}$ with $\lambda = np$. The connected probability is set to 0.015 to obtain an average degree 4 in the random network.
- **Scale-free network** The distribution of node degree in scale-free networks follows a power law, $n_d \propto d^{-\tau}$, where n_d denotes the number of nodes of degree d and $\tau \in [2, 3]$ denotes a constant. A scale-free network has the property that a minority of nodes have many connections, while a majority have few connections. we set $n_d = 4$ to have an average degree 4 in the scale-free network.

Parameters Each network consists of 1000 nodes, and the average degree of the networks is set as 4. We show 600 rounds of training epochs in the experiments because the level of cooperation becomes steady for all methods before 600 rounds. In the dynamic networks, we update the network links in every 100 rounds. The results of the experiments are calculated by averaging the results of 1000 runs. The parameters for the experiments are set as, $T = 1.2, R = 1, P = 0.1, S = 0, \alpha = 0.7, \gamma = 0.1$. The f_c is set to 0.2 and f_d is set to 3 in the experiments. The percentage of malicious agents in MAS is set to 0.2. The privacy budget is set to 0.1.

8.2 Experiments in static social networks

8.2.1 Experiments for stability

One aim of our mechanism is to improve the stability for the evolution of cooperation. Stability means the ability to resist the impact on the level of cooperation caused by malicious agents. The simulation starts the game with an equal proportion of cooperators and defectors.

Figure 4 shows that the cooperation level decreases remarkably for all mechanisms when malicious agents are involved in MAS. For example, the final cooperation is around 98% when no malicious agent exists in the homogeneous network, and it decreases to 70% when malicious agents are involved. This means that malicious agents can have a huge negative effect on the evolution of cooperation.

The DP–RL mechanism can improve the level of cooperation, compared with the proposed RL mechanism. As shown in Fig. 4, the level of cooperation increases by 5–7% in three types of networks after applying DP to the RL mechanism. This indicates that the DP–RL mechanism can resist the impact of malicious agents to some degree. This is because when adding exponential noise to the weight, the reward is adjusted towards a more fair direction. Consequently, the negative effect of malicious agents is relieved.

Figure 5 shows how much privacy budget is needed to defend against different levels of malicious agents in static networks. It is observed that with the increase in the proportion of malicious agents, the level of cooperation decreases dramatically. This is reasonable because more malicious agents will interfere more with the evolution of cooperation. Also, it is observed that a small amount of noise can have a desirable effect on the level of cooperation. This is because a larger privacy budget ϵ in the exponential mechanism may have a limited effect on the scale of the weight.

For other mechanisms, malicious agents can destroy the evolution of cooperation in the system. For the most situations, other mechanisms end up with a cooperation level of around 20% in the three types of networks. This indicates that malicious agents can have a huge negative impact on the evolution of cooperation and previous mechanisms have failed to resist this negative impact.

8.2.2 Experiments for adaptivity

The second aim of our proposed mechanism is to improve adaptivity for the evolution of cooperation. Adaptivity means the ability to promote the level of cooperation in different situations, such as different structured social networks and the initial proportion of cooperators. Thus, we test the adaptivity of the RL mechanism in MAS where no malicious agents exist. We set two variables—initial cooperation level and structured social networks to test the final cooperation level to assess adaptivity. Also, the performance is evaluated with the final level of cooperation in MAS, and the desirable adaptivity can sustain the evolution of cooperation in various conditions.

In Fig. 6, the clearest point is that the proposed RL mechanism can achieve almost the same final proportion of cooperators when the initial proportion of cooperators is different. Specifically, in a homogeneous network, when the initial proportion of cooperators is less than 50%, the proportion of cooperators cannot be improved with the mechanisms of IBN, IBS and LRS. When the initial

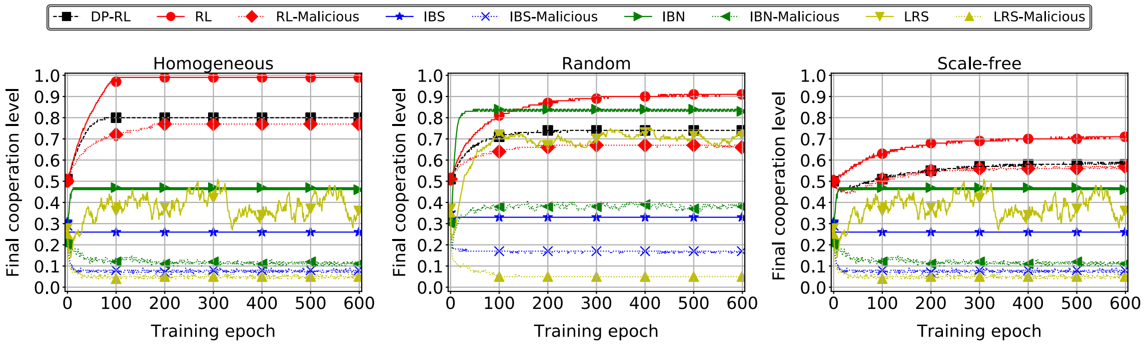

Fig. 4 The stability to resist the impact of malicious agents for different mechanisms in static networks

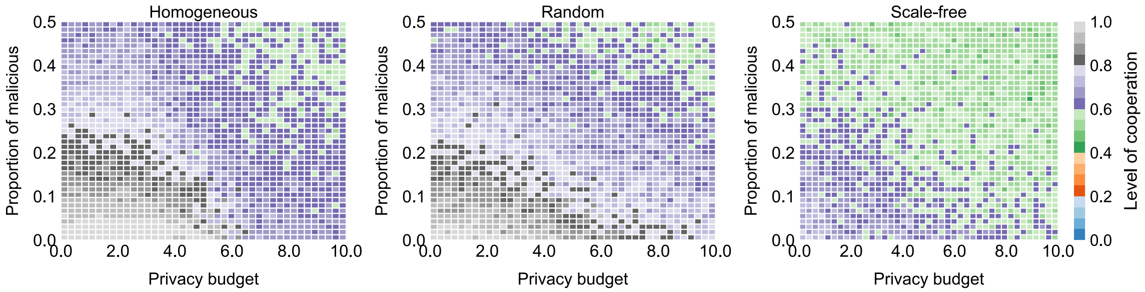

Fig. 5 The impact of varying privacy budgets and the proportion of malicious agents in static networks

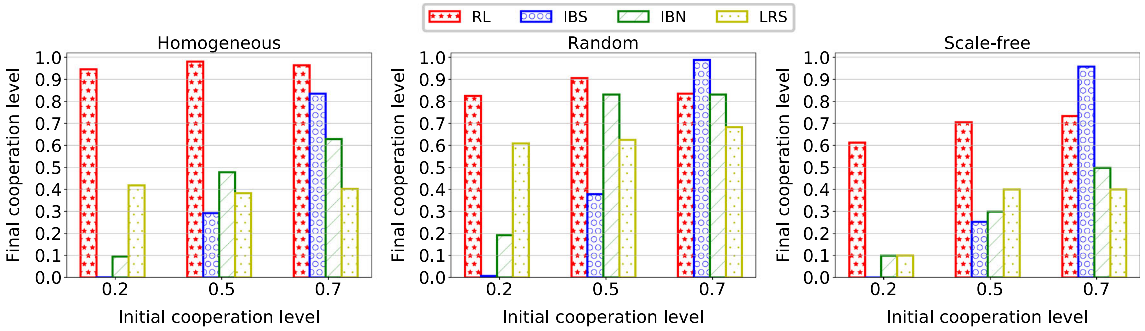

Fig. 6 The adaptivity of different mechanisms in static networks

proportion of cooperators is more than 50%, the proportion of cooperators increases for all methods. However, the proposed method can achieve the highest level of cooperation (around 98%). It is also observed that the proposed RL mechanism can achieve excellent levels of cooperation in homogeneous (around 95%) and random networks (over 80%), and a good level of cooperation in Scale-free networks (around 65%).

The reason for the improved performance in our mechanism is that the proposed RL mechanism can combine the advantages of direct and indirect mechanisms. The reward r for each step can be regarded as the direct payoff in the direct mechanism. The maximum expected Q-value is calculated according to the neighbor's historic actions, which corresponds to the reputation in the indirect mechanism. Hence, the proposed RL mechanism can provide desirable adaptivity in different scenarios.

The other mechanisms, i.e., IBN and IBS, encourage the defecting agent's neighbors to switch to defect. When the initial proportion of cooperators is set to a low value (0.2), it is less likely that defectors can find a wealthy neighboring cooperator and then imitate the cooperator's action, because cooperators surrounded by many defectors cannot obtain high payoffs. This situation is improved when the initial proportion of cooperators is increased.

8.3 Experiments in dynamic social networks

8.3.1 Experiments for stability

In dynamic networks, one aim is also to improve the stability for the evolution of cooperation when malicious agents exist. The network is updated every 100 epochs and the initial proportions of cooperators and defectors are equal.

As shown in Fig. 7, when malicious agents are in MAS, it is clearly noted that the level of cooperation decreases notably for all mechanisms in dynamic networks. For the proposed mechanisms, the RL mechanism decreases notably with the impact of malicious agents in dynamic networks and the DP–RL mechanism can resist the impact of malicious agents to some degree.

Figure 7 shows that the beginning level of cooperation in RL, RL-Malicious and DP–RL mechanisms increases and when the network is just updated, the level of cooperation decreases some (depends on different networks) and then rises until the next network updates. Compared to static networks, the level of cooperation for the RL-Malicious mechanism fluctuates with the update of network and does not converge before 600 training epochs. This indicates that malicious agents and dynamic networks can interfere with the cooperation and can lead to a slower learning rate.

In homogeneous networks, the network update has limited fluctuations. In random networks, the fluctuation

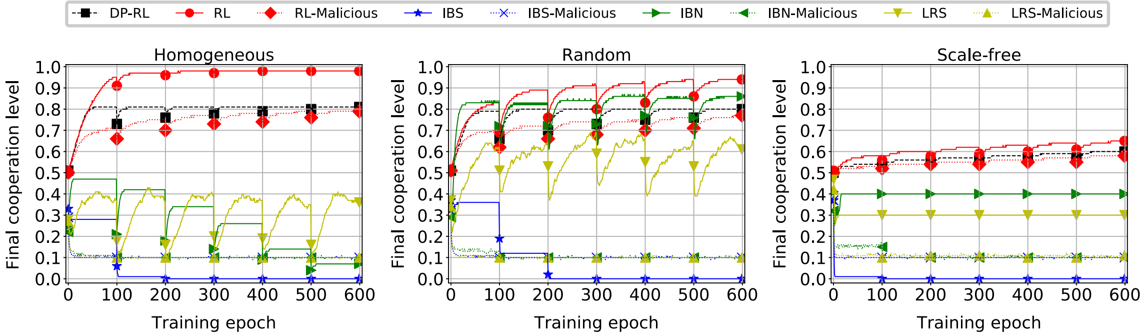

Fig. 7 The stability to resist the impact of malicious agents for different mechanisms in dynamic networks

becomes clearer for RL, RL-Malicious and DP–RL mechanisms and the tendency to cooperate increasing after fluctuations. In scale-free networks, the level of cooperation is also not as desirable as in the homogeneous and random networks, and the level of cooperation becomes quite closed for RL, RL-Malicious and DP–RL mechanisms in the end. Still, the performance is significantly higher than other mechanisms.

For other mechanisms, it is noted that the level of cooperation of the IBS mechanism is undesirable, decreasing to zero in three dynamic networks. The IBN mechanism has a high level of cooperation (around 88%) in random networks. The LRS mechanism fluctuates remarkably in dynamic homogeneous and random networks and reaches a middle cooperation level in three dynamic networks.

Figure 8 shows how much privacy budget is needed in the exponential mechanism to defend against the impact of different proportions of malicious agents in dynamic networks. Similar to the static networks, it is noted that a higher proportion of malicious agents in MAS will degrade the level of cooperation. It is also noted that a smaller privacy budget has a better effect on the level of cooperation. This is due to the property of exponential mechanism, a larger privacy budget has a limited effect on the scale of importance weight.

8.3.2 Experiments for adaptivity

In Fig. 9, it is noted that the RL mechanism can achieve a similar and much higher level of cooperation than other schemes when the initial proportion of cooperators is different, which shows an excellent adaptivity. The RL mechanism can achieve the cooperation level of around 90% in homogeneous and random networks, and around 65% in scale-free networks.

For other mechanisms, when the initial proportion of cooperators is less than 50%, the level of cooperation decreases severely with the IBN, IBS and LRS mechanisms. When the initial proportion of cooperators is more than 50%, the IBN mechanism can perform better than IBS and LRS mechanisms. It is also noted that the IBS mechanism can achieve a higher level of cooperation with a higher initial proportion of cooperators.

8.4 Discussion and summary

8.4.1 Discussion

In our proposed mechanism, agents can learn how to make decisions when agents are in different social networks. We can note that the proposed mechanism have a remarkable performance in homogeneous and random networks and a good performance in scale-free networks. This could be inferred that the calculation of the reward in Q-function is

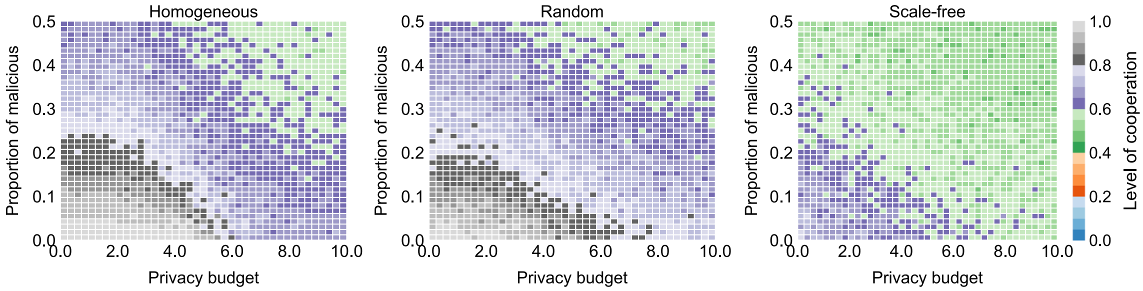

Fig. 8 The impact of varying privacy budgets and the proportion of malicious agents in dynamic networks

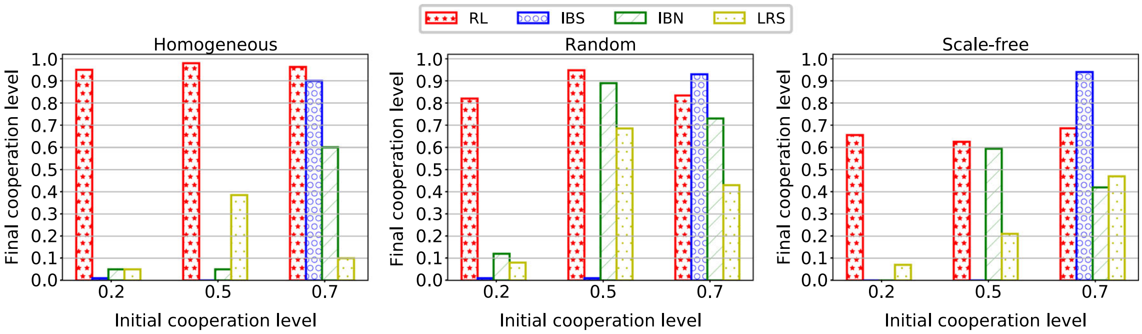

Fig. 9 The adaptivity of different mechanisms in dynamic networks

related to one agent and its neighboring agents. In scale-free networks, few agents may have a large number of neighboring agents and an enormous number of neighboring agents' neighbors. The interaction among agents becomes complicated and calculation of reward becomes complex, which may degrade performance in scale-free networks. In addition, the privacy budget in differential privacy can determine the extent of resisting the impact malicious agents, and thus the choice of privacy budget is important.

8.4.2 Summary

According to the experimental results, the proposed DP–RL method can achieve a desirable stability and adaptivity for the evolution of cooperation in static and dynamic social networks. In terms of stability, the DP–RL mechanism can resist the impact of malicious agents with an increase of 5–7% in the level of cooperation in static and dynamic networks. In terms of adaptivity, the evolution of cooperation can adapt to different initial proportion of cooperators and different types of static and dynamic social networks; the level of cooperation is significantly higher than other three mechanisms in most situations.

9 Conclusion and future work

In this paper, we mainly focused on two problems (1) the impact of malicious agents in MAS; (2) the impact of the structured social networks and the initial proportion of cooperators in MAS. To overcome these problems, we designed the DP–RL mechanism to enable the evolution of cooperation in static and dynamic social networks. The RL method can learn the benefits of direct mechanisms and indirect mechanisms by updating the designed Q-function. More importantly, we applied differential privacy mechanisms to adjust the agents' action in order to resist the impact of malicious agents. The experimental results prove that our proposed mechanism maintains a higher level of cooperation with malicious agent's interference. Also, the proposed mechanism can have a desirable level of cooperation in different social networks and different initial proportion of cooperators.

In the future, we intend to improve our method in two ways. First, malicious agents are considered in MAS, and malicious agents can interact with other agents repeatedly. One intuitive method to resist the impact of malicious agents is to detect who are malicious agents in the system. In this way, the impact of malicious agent can be eliminated in a thorough way. Second, we have not set a particular task to achieve in MAS. In real applications, one task or several tasks will be assigned for multiple agents to complete in MAS.

Acknowledgements This work is supported by an ARC Discovery Project (DP200100946) from Australian Research Council, Australia.

Compliance with ethical standards

Conflict of interest The authors declare that there are no conflicts of interest regarding the publication of this paper.

References

1. Leibo JZ, Zambaldi V, Lanctot M, Marecki J, Graepel T (2017) Multi-agent reinforcement learning in sequential social dilemmas. In: Proceedings of the 16th conference on autonomous agents and multiagent systems
2. Peng P, Wen Y, Yang Y, Yuan Q, Tang Z, Long H, Wang J (2017) Multiagent bidirectionally-coordinated nets: emergence of human-level coordination in learning to play starcraft combat games, ArXiv preprint. arXiv:1703.10069
3. Matignon L, Jeanpierre L, Mouaddi A-I (2012) Coordinated multi-robot exploration under communication constraints using decentralized markov decision processes. In: AAAI conference on artificial intelligence
4. Liu F, Xue S, Wu J, Zhou C, Hu W, Paris C, Nepal S, Yang J, Yu PS (2020) Deep learning for community detection: progress, challenges and opportunities. ArXiv preprint. arXiv:2005.08225
5. Hofmann L-M, Chakraborty N, Sycara K (2011) The evolution of cooperation in self-interested agent societies: a critical study. In: The 10th international conference on autonomous agents and multiagent systems, pp 685–692

6. Ranjbar-Sahraei B, Ammar HB, Bloembergen D, Tuyls K, Weiss G (2014) Theory of cooperation in complex social networks. In: Proceedings of the 25th AAAI conference on artificial intelligence
7. Ye D, Zhang M (2015) A self-adaptive strategy for evolution of cooperation in distributed networks. IEEE Trans Comput 64(4):899–911
8. Charness G, Rigotti L, Rustichini A (2016) Social surplus determines cooperation rates in the one-shot prisoner's dilemma. Games Econ Behav 100:113–124
9. Pinheiro FL, Santos FP (2018) Local wealth redistribution promotes cooperation in multiagent systems. In: Proceedings of the 17th international conference on autonomous agents and multiagent systems, pp 786–794
10. Lozano P, Antonioni A, Tomassini M, Sánchez A (2018) Cooperation on dynamic networks within an uncertain reputation environment. Sci Rep 8:1–9
11. Hughes E, Leibo JZ, Phillips M, Tuyls K, Dueñez-Guzman E, Castañeda AG, Dunning I, Zhu T, McKee K, Koster R et al (2018) Inequity aversion improves cooperation in intertemporal social dilemmas. In: Advances in neural information processing systems, pp 3326–3336
12. Santos FC, Pinheiro FL, Lenaerts T, Pacheco JM (2012) The role of diversity in the evolution of cooperation. J Theor Biol 299:88–96
13. Liu L, De Vel O, Han Q-L, Zhang J, Xiang Y (2018) Detecting and preventing cyber insider threats: a survey. IEEE Commun Surv Tutor 20(2):1397–1417
14. Nowak M, Sigmund K (1993) A strategy of win-stay, lose-shift that outperforms tit-for-tat in the prisones dilemma game. Nature 364:56
15. Axelrod R, Hamilton W (1981) The evolution of cooperation. Science 211(4489):1390–1396
16. Axelrod R, Dion D (1988) The further evolution of cooperation. Science 242(4884):1385–1390
17. Nowak MA, Sigmund K (1992) Tit for tat in heterogeneous populations. Nature 355(6357):250
18. Nowak MA, May RM (1992) Evolutionary games and spatial chaos. Nature 359:826
19. Novak MA, Sigmund K (2005) Evolution of indirect reciprocity. Nature 437(7063):1291
20. Santos FP, Santos FC, Pacheco JM (2018) Social norm complexity and past reputations in the evolution of cooperation. Nature 555(7695):242
21. Pinheiro FL, Pacheco JM, Santos FC (2012) From local to global dilemmas in social networks. PloS One 7(2):e32114
22. Fu F, Hauert C, Nowak MA, Wang L (2008) Reputation-based partner choice promotes cooperation in social networks. Phys Rev E 78(2):026117
23. Hofmann L-M, Chakraborty N, Sycara K. The evolution of cooperation in self-interested agent societies: a critical study. In: The 10th international conference on autonomous agents and multiagent systems, pp 685–692
24. Chen M-H, Wang L, Sun S-W, Wang J, Xia C-Y (2016) Evolution of cooperation in the spatial public goods game with adaptive reputation assortment. Phys Lett A 380(1–2):40–47
25. Grujić J, Fosco C, Araujo L, Cuesta JA, Sánchez A (2010) Social experiments in the mesoscale: humans playing a spatial prisoner's dilemma. PloS One 5(11):e13749
26. Traulsen A, Semmann D, Sommerfeld RD, Krambeck H-J, Milinski M (2010) Human strategy updating in evolutionary games. Proc Natl Acad Sci 107(7):2962–2966
27. Fehl K, van der Post DJ, Semmann D (2011) Co-evolution of behaviour and social network structure promotes human cooperation. Ecol Lett 14(6):546–551
28. Rand DG, Arbesman S, Christakis NA (2011) Dynamic social networks promote cooperation in experiments with humans. Proc Natl Acad Sci 108(48):19 193–19 198
29. Ezaki T, Horita Y, Takezawa M, Masuda N (2016) Reinforcement learning explains conditional cooperation and its moody cousin. PLoS Comput Biol 12(7):e1005034
30. Ezaki T, Masuda N (2017) Reinforcement learning account of network reciprocity. PloS One 12(12):e0189220
31. Tanabe S, Masuda N (2012) Evolution of cooperation facilitated by reinforcement learning with adaptive aspiration levels. J Theor Biol 293:151–160
32. Tampuu A, Matiisen T, Kodelja D, Kuzovkin I, Korjus K, Aru J, Aru J, Vicente R (2017) Multiagent cooperation and competition with deep reinforcement learning. PloS One 12(4):e0172395
33. Kubera Y, Mathieu P, Picault S (2010) Everything can be agent! In: Proceedings of the 9th international conference on autonomous agents and multiagent systems, pp 1547–1548
34. Watkins CJCH, Dayan P (1992) Q-learning. Mach Learn 8(3):279–292
35. Zhu T, Li G, Zhou W, Philip SY (2017) Differentially private data publishing and analysis: a survey. IEEE Trans Knowl Data Eng 29(8):1619–1638
36. Abadi M, Chu A, Goodfellow I, McMahan HB, Mironov I, Talwar K, Zhang L (2016) Deep learning with differential privacy. In: Proceedings of the 2016 ACM SIGSAC conference on computer and communications security, pp 308–318
37. Zhu T, Xiong P, Li G, Zhou W, Philip SY (2018) Differentially private model publishing in cyber physical systems. Future Gen Comput Syst 108:1297–1306
38. Zhang T, Zhu T, Xiong P, Huo H, Tari Z, Zhou W (2019) Correlated differential privacy: feature selection in machine learning. IEEE Trans Ind Inform 16:2115–2124
39. Dwork C, Kenthapadi K, McSherry F, Mironov I, Naor M . Our data, ourselves: privacy via distributed noise generation. In: Advances in cryptology—EUROCRYPT 2006, pp 486–503
40. Dwork C (2011) A firm foundation for private data analysis. Commun ACM 54(1):86–95
41. McSherry F, Talwar K (2007) Mechanism design via differential privacy. In: 48th annual IEEE symposium on foundations of computer science
42. Dwork C, McSherry F, Nissim K, Smith A (2006) Calibrating noise to sensitivity in private data analysis. In: Theory of cryptography, pp 265–284
43. Ye D, Zhu T, Zhou W, Philip SY (2019) Differentially private malicious agent avoidance in multiagent advising learning. IEEE Trans Cybern
44. Zhu T, Philip SY (2019) Applying differential privacy mechanism in artificial intelligence. In: 2019 IEEE 39th international conference on distributed computing systems (ICDCS). IEEE, pp 1601–1609
45. McSherry FD (2009) Privacy integrated queries: an extensible platform for privacy-preserving data analysis. In: Proceedings of the 2009 ACM SIGMOD international conference on management of data, pp 19–30
46. Melo FS (2001) Convergence of q-learning: a simple proof, Institute Of Systems and Robotics, Tech. Rep, pp 1–4