Collective Learning for the Emergence of Social Norms in Networked Multiagent Systems

Chao Yu, Minjie Zhang, Senior Member, IEEE, and Fenghui Ren

Abstract—Social norms such as social rules and conventions play a pivotal role in sustaining system order by regulating and controlling individual behaviors toward a global consensus in large-scale distributed systems. Systematic studies of efficient mechanisms that can facilitate the emergence of social norms enable us to build and design robust distributed systems, such as electronic institutions and norm-governed sensor networks. This paper studies the emergence of social norms via learning from repeated local interactions in networked multiagent systems. A collective learning framework, which imitates the opinion aggregation process in human decision making, is proposed to study the impact of agent local collective behaviors on the emergence of social norms in a number of different situations. In the framework, each agent interacts repeatedly with all of its neighbors. At each step, an agent first takes a best-response action toward each of its neighbors and then combines all of these actions into a final action using ensemble learning methods. Extensive experiments are carried out to evaluate the framework with respect to different network topologies, learning strategies, numbers of actions, influences of nonlearning agents, and so on. Experimental results reveal some significant insights into the manipulation and control of norm emergence in networked multiagent systems achieved through local collective behaviors.

Index Terms—Consensus, distributed multiagent systems, emergent behaviors, ensemble learning, multiagent learning, social networks, social norms.

I. INTRODUCTION

O

NE of the most critical problems in the coordinated control of large-scale distributed multiagent systems (MASs) is to design efficient strategies that enable all the agents to reach an agreement in the areas of common interest [1]. The concept of social norms [2], originally used in the field of sociology to study human social behavior, is of great interest to MAS researchers as it can be used to help increase the predictability of agent behavior, and facilitate the coordination and cooperation among distributed agents to achieve a consensus in the whole system. There have been numerous theoretical investigations in the MAS literature of social norms under different assumptions about agent interaction protocols, societal topologies, and observation capabilities [3]. In empirical applications, social norms have been used as an efficient mechanism to regulate agent behaviors in large-scale distributed systems, such as electronic institutions [4], norm-supported computational societies [5], and normative ad hoc networks [6].

It has been well recognized that two distinct approaches are suitable for the establishment of a social norm in MASs [7]. The first one is the prescriptive approach that an omnipresent authority formulates, specifies, and enforces how the agents should behave according to the administrative incentives. The second one is the bottom-up approach that enables a norm to evolve and emerge on its own without relying on any centralized authority. The former is often based on the offline design, where every agent has the norms hard-wired at the beginning, while the latter is usually based on the online emergent design, where agents decide the most suitable conventions through their local interactions [8]. As the environments where agents are located become dynamic and changing all the time, and the system may involve a large number of distributed agents, it is expensive and inefficient to have a centralized enforcer to formulate and specify social norms in a prescriptive manner. As such, it is more desirable to enable social norms to evolve on their own automatically. Understanding what mechanisms contribute to the bottom-up emergence of social norms is of great interest in the coordinated control of distributed MASs [9]–[12].

Learning is a robust mechanism to facilitate the emergence of stable norms for the distributed MASs [7]. For this reason, some researchers have focused on equipping agents with the capacity to learn to establish a norm for a society. For example, Shoham and Tennenholtz [13] used a highest cumulative reward learning rule to study the norm emergence; Sen et al. [11], [14] proposed a framework for the emergence of norms through social learning based on private local interactions; and Savarimuthu et al. [3] discussed the three aspects of active learning of norm emergence in agent societies. All these investigations handled the issue of norms in the context of an unstructured agent society and showed that a random fashion of agent learning would be sufficient for the emergence of robust norms without a centralized authority.

In the real world, however, people usually interact with each other under some physical constraints. In other words, who meets whom is not random, but is determined by some spatial relationship or social networks [15]. Actually, social networks provide the topology and the infrastructure through
which the norms can be exchanged and influenced, and thus are fundamental in determining the process of norm emergence [8]. For this reason, a number of researchers have studied norm emergence by examining the underlying network topology of agents. For example, Sen and Sen [16] studied how varying topologies of social networks would affect norm emergence in scale-free, fully-connected and ring networks; Villatoro et al. [12], [17] investigated the effects of memory of past activities during learning on the emergence of social norms in different network structures, and used two social instruments to facilitate norm emergence in networked agent societies.

All these studies, however, are based on a simple interaction protocol; each agent must be paired for interaction with one of its neighbors, randomly or preferentially, so that this agent can directly learn from the interaction. This interaction protocol simplifies real-life situations when individuals can collectively make a decision from multiple alternatives. This collective decision making is inherent in human nature because people often seek several opinions before making a final decision [18]. To reach a group consensus, people often interact with others at the same time and learn simultaneously from all of these interactions. However, it is not clear, if, in this collective decision making context, a norm will still emerge successfully, and what impacts this collective decision making will have on norm emergence under different conditions (e.g., different network topologies or heterogeneity of agents).

To answer these questions, this paper proposes a collective learning framework to study the emergence of social norms in networked MASs. In this framework, norms evolve as agents learn over repeated interactions with all their neighbors using multiagent reinforcement learning (MARL) [19] algorithms. Each interaction is framed as a stage game, which has multiple equilibria making coordination between agents uncertain. At each time step, an agent chooses a best-response action regarding each of its neighbors and aggregates all of these actions into an overall action using ensemble learning techniques. The agent then plays the aggregated action with all its neighbors and receives the corresponding reward. Finally, the learning information toward each neighbor will be updated by using the reward. The proposed collective learning framework itself does not require the agent to observe its neighbors’ payoffs and actions. Whether the agent is able to observe its neighbors’ actions is determined by the specific learning strategies adopted for agent interaction with its neighbors. The proposed framework is based on agents’ collective learning from past experience. This means that our work differs from most existing work on the norm emergence in the framework of evolutionary game theory [20], [21], in which the focus is either on the macro-level population dynamics using replicator functions or on the agent-level strategy dynamics using predefined imitation rules, such as imitate-best-neighbor. We investigate a number of key issues such as neighborhood and population size, agent cognitive deficiency, and learning strategy, and their influences on norm emergence under the collective learning framework.

The remainder of the paper is organized as follows. Section II discusses the related work. Section III gives formal definitions of networked MASs and social norms. Section IV presents the collective learning framework. Section V shows the experimental studies. Finally, Section VI concludes the paper, and gives some directions for future research.

II. RELATED WORK

The emergence of social norms has gained increasing attention in the area of MASs. Much work on norms has been done using simulation models, analysis methods, and theoretical concepts from MASs [7]. However, most of the work focuses on the logic, rule-based specification, and declaration of norms by assuming a centralized authority and complete knowledge [11]. Comparatively, not much work has been done on the decentralized emergence of social norms via learning from local interactions. This section first examines some eminent studies on norm emergence using learning techniques in unstructured as well as structured MASs. As agents in our framework interact with each other using MARL and ensemble learning methods, this section also compares our work with the prolific work that has been done along the multiagent learning and ensemble learning research paradigm.

A. Norm Emergence in Unstructured MASs

Many researchers have studied emergence of social norms in unstructured multiagent systems, i.e., no network topology of agents is assumed. Shoham and Tennenholtz [13] proposed an approach based on the highest cumulative reward (HCR) rule to study the emergence of social norms. According to this rule, an agent chooses the strategy that has yielded the highest reward in the past m iterations. History of the strategies chosen and the rewards for each strategy are stored in a memory of a certain size. The experiments showed that the rate of updating strategy and the interval between memory flushes had a significant impact on the efficiency of norm emergence. Sen et al. [11], [14] proposed a mechanism for the emergence of norms through social learning from private interactions. They experimented with three reinforcement learning algorithms and studied the influence of the size of population and actions, and heterogeneity of the population on norm emergence. More recently, Savarimuthu et al. [3] discussed the three aspects of active learning (i.e., experiential, observational, and communication-based learning) of norm emergence, and demonstrated the usefulness of combining these three aspects of learning to boost the convergence of social norms.

All these studies handled the issue of norm emergence via learning in the context of an agent population, in which each agent can interact randomly with other agents. Our work differs from all these studies because we focus on the emergence of norms in a networked MAS. In addition, agents in our model learn simultaneously and collectively with all their neighbors by using ensemble learning techniques. This is in contrast to the sequential learning process in all previous studies, in which each agent is selected sequentially to interact with another agent in the population.

B. Norm Emergence in Networked MASs

A number of researchers have studied norm emergence by considering the underlying network topology of agents.
Sen and Sen [16] evaluated how varying topologies of social networks affected the emergence of norms through social learning. Three different kinds of network topologies (i.e., scale-free, fully-connected, and ring networks) were studied to show how quickly norms converged in social networks depending on parameters such as the topology of the network, the population size, and the number of actions available. Villatoro et al. [17] investigated the effects of memory and the history of past activities during learning on the success and rate of emergence of social norms in different network structures. The authors confirmed that different characteristics of the topology in which agents are located could produce different convergence rates for reaching a social norm. Later, Villatoro et al. [12] used social instruments to facilitate norm emergence in networked agent societies. Two social instruments (i.e., rewiring and observation) were proposed to effectively address the frontier effect problem caused by the subnorms in the network. Recently, Mahmoud et al. [22] further extended Axelrod’s seminal model [23] by considering the topological structures, in particular, the scale-free networks.

All these studies, however, were based on a simple interaction protocol; each agent must be paired for interaction with one of its neighbors so that this agent can learn directly from this interaction. This interaction protocol simplifies real-life situations when individuals can collectively make a decision from multiple alternatives. In our study, an agent interacts with all of its neighbors simultaneously and learns from these interactions collectively. The focus of this paper is to study the impact of local collective behaviors on the overall emergence of norms in a number of different conditions. This focus differentiates our work from all these previous studies.

C. Multiagent Learning and Ensemble Learning

In our framework, norms evolve as agents learn from the local interactions with their neighbors using MARL algorithms [19]. In this point of view, our work can fall in the normative multiagent learning paradigm [24], with the focus on understanding equilibrium (norm) arising through the use of game theory tools. Agents must choose from multiple optimal joint actions to avoid an uncoordinated joint action, causing the so-called equilibrium selection problem. One paradigm of research in multiagent learning involves two agents playing a stage game iteratively to learn policies to reach a preferred equilibrium in a single-shot game [25] or a repeated game [26]. Another line of research focuses on learning when coordination is necessary between loosely coupled agents [27], or on learning to maximize an objective global function for the entire population through local interactions [28]. Unlike the work following these two paradigms, our work studies the equilibrium selection problem in networked MASs, in which each agent’s decision can be affected by its neighbors as well. Ensemble learning methods have been extensively studied in supervised learning to make more accurate predictions or classifications [18]. In reinforcement learning, ensemble learning methods have also been widely used to boost the learning efficiency and improve the learning performance. Sun and Peterson [29] developed several techniques using genetic algorithms to partition the spaces of a task. Agents applied the Q-learning algorithm [30] to learn the action-values in subspaces and all the values were combined through a weighting scheme to a single agent. Jiang [31] developed a system in which each reinforcement learning algorithm learns individually in a learning module and provides its output to an intelligent aggregation module. The aggregation module aggregates these outputs dynamically using some aggregation methods and provides an action decision to deal with dynamic learning problems. Wiering and van Hasselt [32] used a number of ensemble methods to combine multiple independent reinforcement learning algorithms to choose the best action. All these studies employed ensemble learning in reinforcement learning with the aim of enhancing learning efficiency and improving final performance. In our work, however, the ensemble learning is used to combine the focal agent’s best-response actions toward each of its neighbors to make a final decision. The focus is not on learning efficiency, but on the different patterns of norm emergence achieved through varied local collective behaviors in different structural or topological settings.

III. NETWORKED MASs AND SOCIAL NORMS

This section gives formal descriptions of networked MASs and social norms.

A. Networked MASs

Definition 1: A networked MAS can be represented as an undirected graph \( G = (V, E) \), where \( V = \{v_1, ..., v_N\} \) is a set of vertices (agents), and \( E \subseteq V \times V \) represents a set of edges, each of which connects two interacting vertices (agents).

Definition 2: Given a networked MAS \((V, E)\), the neighbors of agent \(i\), which are denoted as \(N(i)\), are a set of agents such that \(N(i) = \{v_j \mid (v_i, v_j) \in E\}\), and \(N(i) \subset V\).

This paper focuses primarily on the following three types of topologies to represent a networked MAS.

1) Grid networks: A grid network is a 2-D lattice with four neighbors for each inner node, three neighbors for each boundary node, and two neighbors for each corner node. In reality, parallel computing clusters and multicore processors are usually organized as a grid network. We use \(GR_N\) to denote a grid network \((N\text{ is the number of nodes)}\).

2) Small-world networks: This kind of network is to represent the small-world phenomenon in many natural, social, and computer networks, where each node has only a small number of neighbors, and yet can reach any other node in a small number of hops. This kind of networks appears in many social networks such as the collaboration networks of film actors and academic researchers, and the friendship networks of high school students [33]. We use \(SW_{N}^{k,\rho}\) to denote a small-world network, where \(k\) is the average size of the neighborhood of a node, \(\rho\) is the rewiring probability to indicate the different orders of randomness of the network, and \(N\) is the number of nodes.

3) Scale-free networks: This kind of network is characterized by the power law of degree distribution of
nodes, which means that a few rich nodes have high connectivity degrees, while the remaining nodes have low connectivity degrees. The probability that a node has $k$ neighbors is roughly proportional to $k^{-\gamma}$. Examples of scale-free networks include the network of citations of scientific papers [34], and links between web pages on the World Wide Web [33]. We use $SF_{N}^{\alpha, \gamma}$ to denote a scale-free network (N is the number of nodes).

### B. Coordination Games and Social Norms

Social norms, such as driving on a particular side of the road, not littering in parks, or tipping in restaurants, are ubiquitous in daily life, and play a pivotal role in regulating and guiding individual behaviors in human societies. Conformity to norms can reduce social conflict, mitigate cognitive load, and thus sustain social order in the whole society [11]. This research is to study how agents can learn to establish a social norm in networked MASs via local collective decision making. Here, a social norm is defined as a restriction on the set of actions available to the agents, as given by Definition 3.

**Definition 3:** A social norm is a convention or law that restricts agents’ behaviors to one particular action [13].

A social norm is said to have been established when all (or at least the majority of) agents in a society have complied with the same action. One performance measure that can be used to evaluate how fast social norms arise in a society is the convergence time $T_c$ for a given level of convergence $c$ so that $R_l \geq R_r$ holds for $c \geq T_c$, where $R_r$ is the convergence ratio of a society at time $t$ (i.e., the fraction of agents adopting the majority action). In this paper, following most of previous studies, we will focus on the study of the average time $T_{90\%}$.

We use learning rules of the road [11], [35] as a metaphor to study the emergence of norms. In this scenario, agents strive to establish a convention of driving either on the left (L) or on the right (R) of the road. This interaction can be viewed as a 2-person 2-choice symmetric coordination game [13], [35], with the payoff matrix displayed in Table I.

<table>
<thead>
<tr>
<th></th>
<th>Left (L)</th>
<th>Right (R)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Left (L)</td>
<td>$x, x$</td>
<td>$u, u$</td>
</tr>
<tr>
<td>Right (R)</td>
<td>$v, v$</td>
<td>$y, y$</td>
</tr>
</tbody>
</table>

Although its payoff matrix appears simple, the coordination game poses a very challenging puzzle for human beings to solve efficiently. The game has two pure Nash-equilibria, that is, both agents drive on the left or both agents drive on the right. Classical game theory, however, does not give a coherent account of how people would play a game like this. The problem is that there is nothing in the structure of the game itself that allows the players (even purely rational players) to infer what they ought to do. In reality, people can play such games because they can rely on some contextual cues to agree on a particular equilibrium [35]. One such contextual cue is social norms (i.e., conventions and laws) that can be used to guide human behaviors when moral or rational reasoning does not provide a clear guidance because of the myopic behavior and the limited processing ability of individuals [36].

### IV. COLLECTIVE LEARNING IN NETWORKED MASs

This section introduces the proposed collective learning framework in networked MASs.

#### A. Proposed Learning Framework

The sketch of the collective learning framework is given by Algorithm 1. All agents in a system interact repeatedly and simultaneously with their neighbors. At each time step, each agent needs to determine an action to play with its neighbors. For each of its neighbors, agent $i$ chooses the best-response action using a specific learning strategy (Line 5). The actions for all the neighbors are then aggregated into an overall action $a_i$ using ensemble learning methods (Line 6).

These ensemble learning methods will be described in detail in Section IV-B. After determining the final action $a_i$, agent $i$ then plays this action with each of its neighbors and receives a reward regarding each neighbor (Line 8). Finally, agent $i$ updates the learning information regarding each neighbor using $a_i$ and corresponding reward (Line 9).

Different learning strategies can be used for the interaction with each neighbor to determine the best-response action. This paper focuses on reinforcement learning (RL) approaches [37], in which an agent learns a policy through trial-and-error interactions with its environment so as to maximize the expected discounted reward for each state in a sequential decision-making problem. One of the most important and widely used RL approach is Q-learning [30], in which an agent makes a decision through the estimation of a set of $Q$ values. Its one-step updating rule is given as follows:

$$Q(s, a) \leftarrow Q(s, a) + \alpha[R(s, a) + \gamma \max_{a'} Q(s', a') - Q(s, a)]$$

(1)

where $\alpha \in (0, 1)$ is a learning rate, $\gamma \in [0, 1)$ is a discount factor, $R(s, a)$ and $Q(s, a)$ are the immediate and expected reward of choosing action $a$ in state $s$ at time step $t$, respectively, and...
$Q(s', a')$ is the expected discounted reward of choosing action $a'$ in state $s'$ at time step $t+1$. $\max_{a'} Q(s', a')$ indicates the maximum value of the expected discounted reward in the new state $s'$ at time step $t+1$.

Every $Q$ value of a state-action pair can be stored in a table for a discrete state-action space. In a state $s$, an agent chooses the best-response action $a$ based on the $Q$ values. The agent then transits to a new state $s'$ and receives a reward $R(s, a)$ from the environment. The agent then updates the $Q$ value of state-action pair [i.e., $Q(s, a)$] according to (1).

It is proved that this tabular Q-learning method converges to the optimal $Q^*(s, a)$ w.p.1 when all the state-action pairs are visited infinitely and an appropriate learning rate is chosen [30].

During learning, an agent needs to make a balance between the exploitation of learnt knowledge and the exploration of unexplored environment in order to learn an optimal policy [37]. The $\epsilon$-greedy exploration policy is an efficient mechanism to tradeoff exploitation and exploration during learning, which can be given as follows [37]:

$$\pi(s, a) = \begin{cases} 1 - \epsilon & \text{if } a = \arg \max_{a'} Q(s, a'), \\ \epsilon & \text{otherwise} \end{cases}$$

(2)

where $\epsilon \in [0, 1]$ is the exploration rate.

Equation (2) means that an agent chooses the action with the highest $Q$-value with a probability of $1-\epsilon$ (i.e., exploitation of the learnt knowledge) and chooses other actions randomly with a probability of $\epsilon$ (i.e., exploration of the unknown environment). The value of $\epsilon$ is often set small to indicate a small probability of exploration.

Fig. 1 gives a vivid illustration of the collective learning framework based on Q-learning. Each focal agent $i$ keeps learning information in terms of a $Q$-value table $Q_j(s, a)$ for each neighbor $j$. At each time step, regarding each neighbor $j$, agent $i$ chooses the best-response action with the highest $Q$-value based on the corresponding $Q$-value table with a probability of $1-\epsilon$ (i.e., exploitation), or chooses other actions randomly with a probability of $\epsilon$ (i.e., exploration). Agent $i$ then collectively makes a decision by aggregating all the actions $a_{i,j}$ for each neighbor into a final action $a_i$ and interacts with each of its neighbors using action $a_i$. The $Q$-value table $Q_j(s, a)$ regarding neighbor $j$ then can be updated according to (1) after the agent receives the immediate reward $r_{i,j}$.

In Algorithm 1, the exploration process is conducted during an agent’s local interactions with each of its neighbors. This means that the agent conducts exploration for each neighbor before aggregating all the actions into a final deterministic action. We call this kind of exploration local exploration, and the framework based on this kind of exploration is denoted as collective learning-1. In order to study the impact of different levels of uncertainties caused by exploration on the learning performance, we also propose another exploration mode in Algorithm 2, in which an agent determines a greedy action using a learning strategy regarding each of its neighbors (Line 5), and then aggregates these actions for all the neighbors into an overall action $a'_i$ (Line 6). Exploration is then conducted when the agent chooses the final action $a_i$ based on the aggregated overall action $a'_i$ (Line 7). We call this kind of exploration global exploration, and the framework based on this kind of exploration is denoted as collective learning-g.

The collective learning framework proposed here is significantly different from the paired learning framework that has been adopted in most previous studies [12], [16], [17]. In the paired learning framework, at each time step, each agent is randomly paired with one of its neighbors for interaction and the agent directly learns from the interaction either through a best response rule [16] or a memory-based rule [17]. The collective learning framework, however, imitates the opinion aggregation process in human decision making because people usually seek several opinions before making a final decision [18]. As the final decision of an agent is affected by all its neighbors, this collective learning can have a significant influence on the emergence of social norms in the whole society in different conditions (e.g., different network topologies, ensemble methods, or heterogeneities of agents).

**Algorithm 2:** Collective Learning Framework (Global Exploration Mode)

1. Initialize network and learning parameters;
2. for each step $t$ ($t=1,...,T$) do
   for each agent $i$ ($i=1,...,n$) do
     for each neighbor $j \in N(i)$ of agent $i$ do
       Agent $i$ chooses greedy action $a_{i,j}$ regarding neighbor $j$ using a learning strategy;
       Agent $i$ aggregates all the actions $a_{i,j}$ into action $a'_i$;
       Agent $i$ chooses final action $a_i$ based on $a'_i$ with exploration;
     for each neighbor $j \in N(i)$ of agent $i$ do
       Agent $i$ plays action $a_i$ with neighbor $j$ and receives corresponding reward $r_{i,j}$;
     Agent $i$ updates learning information regarding neighbor $j$ using $(a_i, r_{i,j})$;
   end for
   end for
end for
B. Ensemble Methods in Agent Learning

The idea behind ensemble learning methods is to weigh several individual classifiers and then combine them to make a final decision that will be better than the one made by each of them separately [18]. Although ensemble learning is used for the aim of increasing learning speed and improving final performance, it has been employed in different forms in a reinforcement learning setting. The different ways in which ensemble learning is used lies in the different ways of defining the action choices. For example, the action choices can be defined as different learning algorithms [32], diversified function approximations in terms of neural network topologies and weights [38], or state-value functions [39]. In our framework, however, the actions that need to be aggregated are the focal agent’s best-response actions toward each of its neighbors in every interaction. The agent needs to combine all these actions to make a final decision by considering each neighbor’s position (e.g., degree of connectivity) as well as the neighbor’s performance in past interactions. The ensemble learning imitates the human collective decision making process in that people usually consult with many others before making a final decision. People consider not only others’ individual characteristics, such as intelligence and knowledge, but also their reputation, social position and power [40].

Formally, let \( a^*_j \) be the best-response action for neighbor \( j \) at time \( t \), and \( a_t \) be the aggregated final action. We enumerate the set of actions for each agent as \( A = \{a[1], ..., a[M]\} \), where \( M \) is the number of actions available. The selection of this final action is then denoted as \( \pi_t(a[h]), a[h] \in A \). The value of \( p_t(a[h]) \) represents the focal agent’s preference for action \( a[h] \). The final action \( a_t \) can then be determined as follows:

\[
\pi_t(a[h]) = \begin{cases} 1 & \text{if } a[h] = \arg \max_a p_t(a), \\ 0 & \text{otherwise}. \end{cases}
\]

We use the following methods to calculate \( p_t(a[h]) \).

1) Majority voting: The preference values are calculated by the majority voting ensemble method as follows:

\[
p_t(a[h]) = \sum_{j=1}^{[N(i)]} I(a[h], a^*_j) \tag{4}
\]

where \([N(i)]\) is the number of neighbors of focal agent \( i \), and \( I(a[h], a^*_j) \) is an indicator function defined as

\[
I(a[h], a^*_j) = \begin{cases} 1 & \text{if } a[h] = a^*_j, \\ 0 & \text{otherwise}. \end{cases}
\]

The most preferred action is simply the one that is suggested by most of the neighbors. The principle of this method reflects the fact that people are social beings and can be influenced by each other so that people are more prone to accept the opinion that is adopted by the majority of their neighbors [40].

2) Weighted voting: The majority voting method simply counts the number of each action as the preference for corresponding action. However, each agent in the network can occupy different positions, and thus can play a different role in shaping the norms of the whole society. For example, in scale-free networks, the power law distribution of the degree of connectivity of the nodes means that a few rich nodes can have high degrees of connectivity. The decisions of these powerful nodes are pivotal in the society. Assume that the decision regarding neighbor \( j \) is weighed by weight \( w_{t,j} \). The weighted voting method can be given as

\[
p_t(a[h]) = \sum_{j=1}^{[N(i)]} w_{t,j} I(a[h], a^*_j). \tag{6}
\]

Several different ways can be used to determine weights \( w_{t,j} \). Here, two approaches are proposed as follows.

1) Structure-based approach: This approach considers the different structural position of each agent in the network. A straightforward way of defining the structure-based weight of each agent is to use the agent’s degree of connectivity. Therefore, the weight \( w_{t,j} \) of neighbor \( j \) can be calculated as follows:

\[
w_{t,j} = \frac{|N(j)|}{|N(i)|} \tag{7}
\]

where \(|N(j)|\) is the number of neighbors of neighbor \( j \), \(|N(i)|\) is the of neighbors of focal agent \( i \), and \(|N(l)|\) is the number of neighbors of neighbor \( l \).

2) Performance-based approach: This approach determines each neighbor’s weight according to past interaction experience between this neighbor and the focal agent. If a neighbor’s action is always consistent with the agent’s own action, the agent will then consider the neighbor to be more trustworthy and accordingly assign a higher weight to this neighbor. This is driven by the fact that in the real world a person who has a higher reputation will have a greater influence on biasing the opinions in a society. So, we have

\[
w_{t,j} = w_{t-1,j} + \beta(s - w_{t-1,j}) \tag{8}
\]

where \( w_{0,j} = \frac{1}{|N(j)|} \), \( \beta \) is a learning rate; and \( s = 1 \) if interaction at time \( t - 1 \) is successful, otherwise \( s = 0 \). The interaction is successful if the interaction brings a positive reward to the agent, namely, the actions of the interacting agents are consistent with each other.

V. EXPERIMENTS AND ANALYSIS

This section evaluates the proposed framework in a number of different settings.

A. Experimental Setting

The pure coordination game [35] is used to formulate the interaction two neighboring agents, with \( x = y = +1 \) and \( u = v = -1 \) in Table I. The Watts–Strogatz model [41] is used to generate a small-world network \( SW^{N,M,u,v} \), and the Barabasi–Albert model [33] is used to generate a scale-free network \( SF^{N,M,u,v} \). To use the Barabasi–Albert model, \( m_0 = 5 \) agents initialize the population and a new agent with \( l = 1 \) edge is added to the network at every time step. This network evolves...
into a scale-free network $S^{k^3}_{N}$ following a power law with an exponent $\gamma = 3$ [8]. In the investigation, unless specifically stated otherwise, the small-world network is used as the default network topology due to the variety of this kind of network, and majority voting is used as the ensemble method due to its simplicity. After generating the interaction networks, we investigate the following important issues.

1) The first and most fundamental issue is to test whether a social norm can successfully emerge in the whole society in the proposed collective learning framework. If a norm does emerge, what is the rate of such an emergence?

2) To have a better understanding of the merits of the collective learning framework compared with the paired learning framework in previous studies, we also test both frameworks in heterogeneous societies where agents have varying cognitive capabilities by receiving noisy feedbacks from the environment. Each agent has a probability of $p_c$ to receive a transformed payoff $r \pm \sigma^2$ ($r = \pm 1$ is the original payoff). Probability $p_c$ and noise $\sigma^2$ indicate the different cognitive capabilities of the agents. We set $p_c$ to 0.2 to indicate a small probability of receiving noisy payoffs, and choose $\sigma^2$ from $\{1, 3, 4, 5\}$ to indicate different levels of cognitive deficiencies.

3) We are also interested in whether the proposed framework is robust enough for different learning strategies adopted by the agents. Three basic learning strategies are adopted for agent interaction: Q-learning [30] with $\epsilon$-greedy exploration; win-or-learn-fast with policy-hill-climbing (WoLF-PHC) [25]; and fictitious play (FP) [42]. Q-learning has been widely used in MASs, but converges only to pure strategies. An FP player uses the historical frequency count of its opponent’s past actions and tries to maximize the expected payoff by playing a best response to that mixed strategy represented by this frequency distribution. Learning rate $\alpha_w$ is set to 0.1 in Q-learning and FP to make small adaptation of learning parameter. $\epsilon$ is set to 0.04 when the agent is winning and learning rate $\alpha_l$ is set to 0.01 when the agent is losing.

4) The population size, the number of neighbors and actions, and the randomness of the network are important factors which influence the emergence of social norms [11], [16]. We vary agent number $N$ in network $SW^{12,0.8}_N$ in the range of $\{50, 1000\}$, number of neighbors $k$ in network $SW^{k,0.8}_{100}$ in the set of $\{4, 6, 8, 12, 20\}$, randomness of network $\rho$ in network $SW^{12,0.8}_N$ in the set of $\{0, 0.2, 0.4, 0.6, 0.8, 1\}$, number of actions in network $SW^{12,0.8}_{100}$ in the set $N_a = \{2, 4, 6, 10, 20\}$. When agents have more than two action choices, only when the agents choose the same action will they receive a payoff of 1. Otherwise, they receive a payoff of $-1$.

5) Due to the different features of the three kinds of networks, we also test the collective learning framework in these networks to discover the impact of different ensemble learning methods on norm emergence.

6) A small proportion of agents with fixed strategies in a population can significantly influence norm emergence in the whole system [10]. To study the effect of non-learning agents in shaping the emergence of norms, we place different numbers of agents with fixed actions or strategies in the society to discover these agents’ influence on the emergence of social norms. We are also interested in whether some non-learning agents playing contradictory strategies in different areas of the network can cause subnorms in the whole society.

The parameter settings in the experimental study are summarized in Table II for clarity.

### Table II

<table>
<thead>
<tr>
<th>Parameters in the experiments</th>
<th>Values</th>
<th>Meanings</th>
</tr>
</thead>
<tbody>
<tr>
<td>$N$</td>
<td>$[50, 1000]$</td>
<td>number of agents</td>
</tr>
<tr>
<td>$k$</td>
<td>${4, 6, 8, 12, 20}$</td>
<td>average size of neighborhoods in small-world networks</td>
</tr>
<tr>
<td>$\rho$</td>
<td>$[0, 1]$</td>
<td>rewiring probability in small-world networks</td>
</tr>
<tr>
<td>$m_0$</td>
<td>5</td>
<td>initial agents to generate small-world networks</td>
</tr>
<tr>
<td>$\gamma$</td>
<td>3</td>
<td>the exponent value of power law in scale-free networks</td>
</tr>
<tr>
<td>Learning algorithm</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$\alpha$</td>
<td>0.1</td>
<td>learning rate in Q-learning and fictitious play</td>
</tr>
<tr>
<td>$\alpha_w$</td>
<td>0.1</td>
<td>learning rate when losing in WoLF-PHC</td>
</tr>
<tr>
<td>$\alpha_l$</td>
<td>0.04</td>
<td>learning rate when winning in WoLF-PHC</td>
</tr>
<tr>
<td>$\epsilon$</td>
<td>0.1</td>
<td>$\epsilon$-exporation rate</td>
</tr>
<tr>
<td>Others</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$\beta$</td>
<td>0.8</td>
<td>learning rate to adjust the weight</td>
</tr>
<tr>
<td>$p_c$</td>
<td>0.2</td>
<td>probability of having cognitive deficiency</td>
</tr>
<tr>
<td>$\sigma^2$</td>
<td>${1, 3, 4, 5}$</td>
<td>noise of payoff</td>
</tr>
</tbody>
</table>

### Table III

**Norm Emergence After 1000 Runs in Network $SW^{12,0.8}_{100}$**

<table>
<thead>
<tr>
<th>Method</th>
<th>$N$</th>
<th>$k$</th>
<th>$\rho$</th>
<th>Success (%)</th>
<th>Reward</th>
<th>Speed($f_{SW(5)}$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Paired learning</td>
<td>506</td>
<td>494</td>
<td>100</td>
<td>0.80</td>
<td>10</td>
<td>0.99</td>
</tr>
<tr>
<td>Collective learning-I</td>
<td>493</td>
<td>567</td>
<td>100</td>
<td>0.80</td>
<td>10</td>
<td>0.99</td>
</tr>
<tr>
<td>Collective learning-g</td>
<td>522</td>
<td>478</td>
<td>100</td>
<td>0.80</td>
<td>10</td>
<td>0.99</td>
</tr>
</tbody>
</table>

larger than the value of the game’s equilibrium, and losing otherwise. Learning rate $\alpha_w$ is set to 0.04 when the agent is winning and learning rate $\alpha_l$ is set to 0.01 when the agent is losing.

**B. Results and Analysis**

1) **Convergence of Social Norms:** We first test the proposed framework in network $SW^{12,0.8}_{100}$ and compare it with the paired learning framework that has been adopted in most previous studies [12], [16], [17]. Table III presents the frequency and success ratio of converging to a social norm, the final average agent reward of the society, and the speed to evolve a norm. Table III shows that a social norm can successfully emerge in the whole population with 100% probability, and that agents converge to norm (Left) and norm (Right) with an almost equal probability. This makes sense because the game structure itself does not give any preference for any particular action to be the norm. The norm to which the whole population of agents converges depends on which norm is detected by the agents at first glance during the dynamics of agent interaction. The average reward of all the agents in the society using collective learning-I is almost 1, which means that nearly
all the agents have reached a consensus on which action should be the norm. The rewards using collective learning-g and paired learning are much lower than that using collective learning-l because agents are exploring the environment with a probability of $\epsilon = 0.1$. However, because agents using collective learning-l explore the environment locally and make a final decision collectively, the uncertainties caused by the exploration decrease. The time steps needed for 90% of the agents to choose the same action as a social norm under three frameworks differ dramatically. The collective learning framework is able to evolve a social norm much faster than the paired learning framework.

Fig. 2 shows the dynamics of the average reward of the whole population and Fig. 3 shows the frequency of each action adopted by the agents when norm (Left) emerges in the population. Initially, each agent randomly chooses an action, so there are about 50% of the agents choosing action L and the other 50% choosing action R. This results in the average payoff of 0 in the whole population. As the learning process moves on, however, the number of agents who choose action L as the norm increases. This means that more and more agents have reached a consensus on that action L should be the norm, and this consensus correspondingly increases the average payoff dramatically. From the results, we can also see that the norm emerges faster under the collective learning framework than under the paired learning framework. This is because agents using collective learning can learn directly from all of their neighbors at the same time to decrease the diversity among the agents. These results confirm that our collective learning framework is a more efficient mechanism for the emergence of social norms.

2) Influence of Agents’ Cognitive Deficiencies: Fig. 4 shows the dynamics of the action frequency in network $SW_{100}^{12,0.8}$ with different levels of cognitive deficiency $\sigma^2$. From the results, we can see that a society in the collective learning framework is able to maintain a higher convergence ratio and a quicker convergence speed, compared with a society in the paired learning framework. In both cases, as the cognitive deficiency $\sigma^2$ increases, the convergence speed slows down. This is because it is more difficult for the agents in societies with larger values of $\sigma^2$ to distinguish the effects of their actions on the environment, and these uncertainties can hinder the agents from reaching consensus with others in the society. However, the difference in the convergence speed using collective learning is not as significant as that using paired learning. This indicates that our collective learning framework can mitigate the uncertainties caused by the agents’ cognitive deficiencies, and is more efficient and robust for norm emergence compared with the paired learning framework.

3) Influence of Agents’ Learning Strategies: We test the three different learning strategies (i.e., Q-learning, WoLF-PHC, and fictitious play) in both a homogeneous and a heterogeneous society of 100 agents to study their influences on norm emergence. In the homogeneous society, all agents use the same learning strategy, while in the heterogeneous society, agents are equally divided according to the strategies they adopt. The heterogeneity of society models the real-life situation, when people have different learning capabilities in the same circumstances. Fig. 5 shows the dynamics of the average payoff using different learning strategies in network $SW_{100}^{12,0.8}$. As we can see, the collective learning framework can successfully evolve a social norm using all three learning
strategies. The quickest one is using Q-learning, followed by WoLF-PHC, and fictitious play. The norm evolves the most slowly using fictitious play because agents need a great deal of time to estimate the frequency distribution of neighbors’ past actions. In the heterogeneous society, the time to evolve a norm falls between the time taken by the corresponding homogeneous societies. These results are consistent with the previous study [11], in which agents learn randomly in an unstructured population, and further demonstrate that our collective learning framework is robust for the emergence of social norms in homogenous as well as heterogenous societies.

It can be seen that Q-learning agents are the fastest to evolve a social norm. Hence, we are interested in the impact of different proportions of these fast learning agents on norm emergence in the whole population. Fig. 6 shows the convergence time when different proportions of Q-learning agents are deployed in a population of Fictitious Play learners and in a population of WoLF-PHC learners. The results show that when there is only a small proportion (e.g., 10%) of Q-learning agents in the population, the convergence time is steeply reduced from that in the original homogeneous population, and further increasing the proportion of Q-learning agents steadily decreases the convergence time. These results illustrate that even a small proportion of fast learning learners can greatly facilitate the norm convergence in a large-scale agent society.

4) Influence of Population Size, Number of Neighbors and Actions, and Randomness of Networks: The dynamics of average agent reward with different population sizes is shown in Fig. 7, from which we can see that the larger the agent population, the longer it takes for the entire society to converge to a social norm. This is because the larger the society, the more difficult to diffuse the effect of local learning to the whole society. This phenomenon can be seen in human societies where small groups and clans can more easily establish social norms than is the case in larger societies [11].

Fig. 8 shows the dynamics of average agent reward with different neighborhood size \( k \) in a network with a degree distribution that is similar to a random graph. The results show that when the average number of neighbors is increased, the convergence time is steadily reduced. This effect is due to the clustering coefficient of the network. Clustering coefficient is a measure of degree to which nodes in a graph tend to cluster together [12]. When the average number of neighbors increases, the clustering coefficient also increases, and therefore agents located in different parts of the network only need a smaller number
of interactions to reach a consensus. On the other hand, when agents have a small neighborhood size, they only interact with their neighbors, which account for a small proportion of the whole population. This results in diverse subnorms formed at different regions of the network. Such subnorms conflict with each other in the network, and thus more interactions are needed to solve these conflicts and achieve a uniform norm for the whole society. Another interesting phenomenon is that when the average neighborhood size is small, a minor increase in the size (e.g., from four neighbors to eight neighbors) can bring about significant improvement of the emergence speed, while further increasing the neighborhood size (e.g., from 8 neighbors to 20 neighbors) cannot cause a further significant improvement. In other words, the relation between the decrease of convergence time and the increase of neighborhood size is nonlinear, but follows a logarithmical distribution, which can be further seen from Fig. 9.

Fig. 10 shows the influence of different numbers of potential actions on norm emergence. Note that the different initial values are due to the payoff matrix of corresponding coordination games. As can be seen from Fig. 10, a larger number of available actions results in a delayed convergence of norms. This is because a larger number of actions are more likely to produce local subnorms, leading to diversity across the society. It thus takes a longer time for the agents to eliminate this diversity and achieve a final consensus, and thus the norm emergence process is prolonged throughout the network.

Fig. 11 shows the influence of network randomness on norm emergence. When \( \rho = 0 \), network \( SW^{k,\rho}_{100} \) is reduced to a regular ring lattice. Increasing rewiring probability \( \rho \) produces a small network with increasing randomness. When \( \rho = 1 \), the network becomes a fully random network. The results indicate that it is more efficient for a norm to emerge in a network with higher randomness. This is because the increase in randomness can reduce the network diameter (i.e., the largest number of hops to traverse from one vertex to another [17]), and the smaller a network diameter is, the more efficient for the network to evolve a social norm [8].

5) Influence of Ensemble Methods: Fig. 12 shows the influence of different ensemble methods as well as paired learning method on norm emergence in three different kinds of networks. In the grid and small-world network, the majority voting method and structure-based method outperform the paired learning method throughout the whole learning period. Norms using the performance-based method converge very slowly at the beginning and then quickly outperform those using the paired learning method afterwards. In the scale-free network, however, the three methods under the collective learning framework have almost the same performance, and all outperform the paired learning method throughout the whole learning period. These results show that the proposed ensemble methods can bring about different patterns of norm emergence in the three different kinds of networks, and further confirm that our collective learning framework is more efficient for norm emergence than the paired learning framework.

6) Influence of Nonlearning Agents: As reported in [10], [11], and [14], a small proportion of non-learning (NL) agents can significantly influence norm emergence in the whole society. Some agents may not have learning capabilities but always choose a predefined action, or choose an action according to some fixed strategy. The existence of these NL agents can bias, facilitate, or even impede norm emergence in the society. We
Fig. 12. Emergence of social norms in three networks using different ensemble methods (averaged over 100 runs). (a) Grid network $GR_{100}$. (b) Small-world network $SW_{4,0.8}$. (c) Scale-free network $SF_{100}^2$.

Fig. 13. Convergence of social norms with different proportions $p_f$ of NL agents in the whole population in network $GR_{100}$ (averaged over 50 runs).

Fig. 14. Influence of different proportions of non-learning agents (driving on the left) in the whole population on biasing the norm in network $GR_{100}$.

now study the influence of these NL agents on the emergence of social norms in three different scenarios.

Scenario 1: Nonlearners with the same fixed action: In this experiment, we replace some learning agents by NL agents, who use the same fixed action (i.e., driving on the left), in network $GR_{100}$. Fig. 13 presents the convergence process when there are 0%, 1%, 5%, 10%, 50% NL agents in the whole population. The results show that when there are no NL agents, the norm converges very slowly because all the agents are learners and must explore all the possible actions equally before they can arrive at a mutual consistent action. When there are only 1% NL agents, the norm does not emerge any faster because the local effect of a single NL agent is insufficient to expedite convergence to a norm in a 100 agent society. When there are more NL agents in the population, however, the norm emergence process is greatly accelerated. This is because an agent that has an NL agent in its neighborhood can observe a bias toward the fixed action always chosen by the NL agent. When the agent starts to exploit this knowledge, it takes advantage of this bias and consequently plays the action played by the NL agent. This effect of bias then can be cascaded into the whole population faster when there are more NL agents.

We are also interested in the impact of the proportion of NL agents on biasing the whole population toward a particular norm. In previous studies when all agents are learners, it was observed that the norms evolved roughly with an equal frequency over multiple runs. This is reasonable because the payoff matrix of the coordination game itself has no preference for one norm over the other. When NL agents are present, however, the norm can be biased. The influence of different proportions of NL agents (driving on the left) on biasing the population to norm $Left$ in network $GR_{100}$ is given by Fig. 14, in which the $x$-axis indicates the proportion of NL agents, and the $y$-axis indicates the probability of converging to norm $Left$ over 100 independent runs. Results are averaged over 10 Monte Carlo experiments. As can be seen, when there are no NL agents in the population, the norm Right and the norm Left emerge with an almost equal frequency. When there are more agents driving on the left, the population converges more often to the norm Left. Two interesting phenomena can also be observed from the results. First, although a very small number of NL agents cannot boost the norm converging process, they can significantly bias the whole population to a particular norm (2% and 4% NL agents can increase the possibility of the norm Left from 50% to around 70% and 80%, respectively). Second, a small proportion (14%) of NL agents is sufficient to bias the population to a particular norm with 100% possibility. These results show that a small proportion of agents with fixed strategies can significantly manipulate the norm emergence in
Fig. 15. Norm emergence with different proportions of NL agents using the fixed observation strategy in network GR_{100} (averaged over 50 runs).

relatively large populations. These results are significant as they can help design efficient strategies for controlling large-scale distributed systems. For example, when an administrative authority wants to control the evolution and formation of group consensuses, conventions or trends in real human or agent-supported computational societies at a minimal administrative cost, the authority only needs to place a small proportion of people (agents) playing fixed actions according to the authority’s will to bias the whole society accordingly.

Scenario 2: Nonlearners with the same observation strategy: In this experiment, NL agents do not play the same fixed action but use the same observation strategy. The observation strategy means that an NL agent simply observes its neighbors and chooses the action which is played by most of its neighbors for next round play. We consider this kind of agents to be NL agents as they do not learn from local interactions with their neighbors. In other words, no learning information is kept by the agents during interaction.

Fig. 15 shows the learning dynamics with different proportions of NL agents using the same observation strategy in network GR_{100}. From the results, we can see that the norm emerges more quickly when the proportion of NL agents gets larger at the beginning (i.e., from 0% to 80%). However, increasing the proportion further from 90% to 100% significantly hinders the convergence process. Upon deeper examination, we found that the turning point of such a varying performance was around 83%. When the population consists of 100% NL agents, the system is fully in chaos because all the agents simply observe their neighbors, whose initial actions are random, in order to copy the most chosen action. Since no learning capability is assumed, the agents cannot remember the past learning experience to make a reasonable decision. That is why the learning curve for the system of 100% NL agents fluctuates at the beginning and then stabilizes at a reward of 0 afterward. However, incorporating a small number of learning agents, e.g., 10% (i.e., curve 90% in Fig. 15), can drastically boost the norm convergence. This is because the learning agents can take advantage of the interaction experience to exploit the other agents for a better outcome.

In real applications, equipping agents with a learning capability often means a cost to either the agents themselves or the whole system. For example, the agents might need physical space to store the learning information, thus imposing a managerial cost on the agents, or for some safety-critical environments, a fatal decision caused by the trial-and-error process during learning could bring about disastrous consequences to the whole system. It is therefore more efficient to deploy as few learning agents as possible in the whole system to decrease the side-effects caused by learning. The results of this experiment indicate that it is possible to achieve a maximal performance by incorporating only a small number of learning agents into a large group of nonlearners who use a fixed observation strategy. This principle can be helpful for the efficient mechanism design of large-scale norm-governed systems, in which the coexistence of millions of agents makes it inefficient or even unfeasible to achieve a global optimal performance through each agent’s individual learning.

Scenario 3: Nonlearners with different strategies: In practice, NL agents may be unrelated and even adopt conflicting actions. In this case, it is possible for these NL agents to decrease the speed of norm emergence, or even prevent the emergence of a norm in the entire population [14]. In the following experiment, we use different numbers of NL agents playing contradictory strategies and study their influence on norm emergence. We use the agent’s coordinate (x, y) to indicate its location in network GR_{100}, with (0, 0) indicating the upper left corner and (9, 9) indicating the lower right corner of the grid. Five different settings for the locations of NL agents are given in Table IV.

Fig. 16 shows the influence of NL agents with contradictory strategies on norm emergence in network GR_{100} (averaged over 50 runs).

<table>
<thead>
<tr>
<th>Nf</th>
<th>Agents (fixed on the left)</th>
<th>Agents (fixed on the right)</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>(1, 1.2, 2, 2)</td>
<td>(1, 1.2, 2, 2)</td>
</tr>
<tr>
<td>4</td>
<td>(1, 1.2, 2, 2, 0.5)</td>
<td>(1, 1.2, 2, 2, 0.5)</td>
</tr>
<tr>
<td>10</td>
<td>(1, 1.2, 2, 2, 0.5, 0.2, 2, 2)</td>
<td>(1, 1.2, 2, 2, 0.5, 0.2, 2, 2)</td>
</tr>
<tr>
<td>14</td>
<td>(1, 1.2, 2, 2, 0.5, 0.2, 2, 2, 0.2, 0.2)</td>
<td>(1, 1.2, 2, 2, 0.5, 0.2, 2, 2, 0.2)</td>
</tr>
<tr>
<td>18</td>
<td>(1, 1.2, 2, 2, 0.5, 0.2, 2, 2, 0.2, 0.2, 0.2)</td>
<td>(1, 1.2, 2, 2, 0.5, 0.2, 2, 2, 0.2, 0.2)</td>
</tr>
</tbody>
</table>

Table IV: Settings for locations of NL agents in network GR_{100}.
in the upper left area of the grid are more inclined to converge to the norm Left and agents in the lower right are more likely to converge to the norm Right. This results in two subnorms in the population during learning. We were expecting that these two subnorms could co-exist in the network. As can be seen from Fig. 16, however, the learning curves are still in the process of converging after 500 episodes. This means that the two subnorms are competing with each other after they emerged in each area and ultimately there will be only one norm for the whole population. However, the existence of subnorms indicates that more interactions are needed to resolve the conflicts for a uniform norm to emerge, which accordingly delays the emergence of a social norm.

VI. CONCLUSION

A systematic study and development of robust mechanisms that can facilitate the emergence of stable and efficient norms via learning is a promising research paradigm for coordinated control of distributed MASs. This paper studied the emergence of social norms through agent collective learning from local interactions in networked MASs. The proposed collective learning framework requires each agent to learn repeatedly with all the neighbors at the same time. We showed that the collective learning framework was more efficient and robust than the paired-learning framework which has been adopted in most previous studies. We investigated the effects of different ensemble methods, population and neighborhood sizes, numbers of actions, learning strategies, and the cognitive abilities of agents, etc., on norm emergence. Experimental studies confirmed that collective learning was indeed a robust mechanism for evolving stable norms in networked MASs.

The long-term goal of this research is to design robust mechanisms that are capable of sustaining global systematic order, increasing the predictability and controllability of macroscopic social behavior, and facilitating coordination and cooperation in large distributed MASs. Such mechanisms can not only provide us with a better understanding of the formation and evolution process of opinions, conventions, and rules in human societies, but also enable us to build and control large virtual MASs. Although this paper makes a move toward this goal, much work still remains to be done. For example, relationships between agents can be added to the network structure. The collective learning process of an agent should thus not only consider the neighbors’ position or reputation, but also consider such relationships between the agent and its neighbors.

ACKNOWLEDGMENT

The authors would like to thank the reviewers for their valuable comments and Dr. M. Cincotta for proof reading the paper.

REFERENCES


Minjie Zhang (SM’13) received the B.Sc. and M.Sc. degrees from Fudan University, Shanghai, China and the Ph.D. degree in computer science from the University of New England, Armidale, Australia. She is currently an Associate Professor of computer science with the University of Wollongong, Wollongong, Australia. She is an active Researcher and has published over 100 papers in the past 10 years. Her current research interests include multiagent systems and agent-based modeling in complex domains.

Fenghui Ren received the B.Sc. degree from the Xidian University, Shaanxi, China, in 2003, and the M.Sc. and Ph.D. degrees from the University of Wollongong, Wollongong, Australia, in 2006 and 2010, respectively. He is currently an ARC DECRA Fellow with the School of Computer Science and Software Engineering, University of Wollongong. He is an active Researcher and has published over 40 research papers. His current research interests include agent-based concept modeling of complex systems, data mining and pattern discovery in complex domains, agent-based learning, smart grid systems, and self-organizations in distributed and complex systems.

Chao Yu received the B.Sc. degree from the Huazhong University of Science and Technology, Wuhan, China, in 2007, the M.Sc. degree from Huazhong Normal University, Wuhan, China, in 2010, and the Ph.D. degree in computer science from the University of Wollongong, Wollongong, Australia, in 2013. He joined the Department of Computer Science and Engineering as a Lecturer with the Dalian University of Technology, Dalian, China, in 2014. His current research interests include multiagent systems and multiagent learning, with their wide applications in modeling and solving various real-world problems.