Efficient Convention Emergence through Decoupled Reinforcement Social Learning with Teacher-Student Mechanism

Yixi Wang, Wenhuan Lu, Jianye Hao*, Jianguo Wei
School of Computer Software, Tianjin University
Tianjin, China
yixiwang2017@outlook.com,{jianye.hao,jianguo, wenhuan}@tju.edu.cn

Ho-Fung Leung
The Chinese University of Hong Kong
Hong Kong, China
lhf@cuhk.edu.hk

ABSTRACT

In this paper, we design reinforcement learning based (RL-based) strategies to promote convention emergence in multiagent systems (MASs) with large convention space. We apply our approaches to a language coordination problem in which agents need to coordinate on a dominant lexicon for efficient communication. By modeling each lexicon which maps each concept to a single word as a Markov strategy representation, the original single-state convention learning problem can be transformed into a multi-state multiagent coordination problem. The dynamics of lexicon evolutions during an interaction episode can be modeled as a Markov game, which allows agents to improve the action values of each concept separately and incrementally. Specifically we propose two learning strategies, multiple-Q and multiple-R, and also propose incorporating teacher-student mechanism on top of the learning strategies to accelerate lexicon convergence speed. Extensive experiments verify that our approaches outperform the state-of-the-art approaches in terms of convergence efficiency, convention quality and scalability.

KEYWORDS

Multiagent social learning; Convention emergence

ACM Reference Format:
Yixi Wang, Wenhuan Lu, Jianye Hao*, Jianguo Wei and Ho-Fung Leung. 2018. Efficient Convention Emergence through Decoupled Reinforcement Social Learning with Teacher-Student Mechanism. In *Proc. of the 17th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2018), Stockholm, Sweden, July 10–15, 2018,* IFAAMAS, 9 pages.

1 INTRODUCTION

Conventions are effective mechanisms to facilitate coordinations among agents. Since top-down approaches require global knowledge to synthesize a convention beforehand, bottom-up approaches, which investigate how a convention emerges through repeated local interactions via learning among agents, are more suitable for distributed MASs [14, 18, 20, 24].

Until now, there exist two major classes of approaches for investigating the convention emergence problem: the spreading-based approach [4, 6, 10, 15, 16] and the reinforcement learning based (RL-based) approach [1, 12, 19, 22–25]. Traditional spreading-based approaches [4, 15] usually combine two decision mechanisms, local optimization and imitation, to establish conventions. However, these approaches can only deal with relatively simple convention spaces with two convention alternatives. With the increase of the convention space, there are a number of challenging issues to be addressed. First, there usually exist multiple convention seeds in a complex convention space. These spreading-based approaches fail to converge to conventions in this type of scenarios. Second, the existing convention seeds may not be good enough. Not all the convention seeds are equally preferred, because some of them can promote coordination more effectively. Later, some techniques [6, 10, 16] are proposed to study the language coordination problem with large convention space to overcome the above limitations of traditional spreading-based approaches. However, these approaches use simple transfer strategy to update lexicons based on the information of the current interaction episode, which causes lexicon qualities to oscillate frequently. They cannot converge efficiently into a dominant convention within a reasonable amount of time for large convention space.

Reinforcement learning based approaches enable convention emergence through reinforcement social learning [19]. Sen and Airiau [1] characterize the emergence of conventions through distributed adaption by agents from their online experiences. However, they only focus on relatively small-size games with two convention alternatives. Most of the existing approaches result in very slow convention emergence or even fail to converge for large convention space. Recently, some hierarchical learning strategies [12, 22–25] are proposed to improve the convention emergence rate for relatively large convention space problem of up to 6 conventions. However, the current RL-basd approaches learn conventions by exploring the convention space directly and usually fail when the convention space becomes significantly large.

*Corresponding author: Jianye Hao.
Proc. of the 17th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2018), M. Dastani, G. Sukthankar, E. André, S. Koenig (eds.), July 10–15, 2018, Stockholm, Sweden.

In this paper we explore the question of how RL-based strategies can be used for solving convention emergence problems for large convention spaces. Similar to previous works [6, 10, 16], we focus on investigating the challenging language convention emergence problem, whose convention space is significantly huge and exponential to the number of words. In MASs, communication is a key factor for agents to successfully interact with each other. When agents rely on explicit communication, a shared language (or lexicon) is required. Nevertheless, in open, heterogeneous MASs, such a lexicon does not exist since no central authority exists. Therefore, an approach that allows agents to reach a consistent language convention through local interactions is quite necessary.

In this work, we represent each lexicon as a Markov strategy to transform the original single-state convention learning problem into a multi-state multiagent coordination problem. Under the above formalization, we propose two RL-based strategies: multiple-Q and multiple-R, which learn to improve the action values of each concept separately and incrementally during agent interactions. We improve distributed value function [17] by incorporating the observation mechanism [8] and replacing the original maximum estimator with a weighted target value. Multiple-Q improves the estimation of Q-values by extending the idea of double estimator to the social learning framework. In contrast, multiple-R improves the estimation of the immediate reward under each state-action pair directly to improve the Q-values estimations. Simulation shows both multiple-Q and multiple-R outperform the state-of-the-art approaches. We also propose incorporating a teacher-student mechanism [3] on top of the above two learning strategies, which allows student agents to ask for advice from teacher agents. We make a series of improvements upon the original teacher-student framework to accelerate convention emergence speed in large convention spaces. Experimental results show that it can facilitate the acceleration of convention emergence speed for both strategies.

2 RELATED WORK

Conventions play an important role in regulating agents' behaviours to ensure coordination among agents and functioning of agent societies. Spreading-based approaches are proved to establish conventions in agent populations [4, 6, 10, 15, 16]. Typically, a spreading mechanism encompasses some spreading (information transfer) strategy along with some selection strategy for incoming transfers. The common information transfer strategy [4, 15] is copy-transfer: each agent completely replicates an agent's convention seed to its neighbors. However, these traditional spreading-based models are only applicable to two possible convention alternatives. With the increasing of the action space, they usually fail to converge.

To address this issue, some works [6, 10, 16] extend the traditional spreading-based mechanisms to achieve a high-quality convention when multiple alternatives exist. Typically there will be some challenging problems in large and open MASs. Firstly, there is no guarantee that the good convention seeds are known by any of the agents. Secondly, if agents' communication becomes unreliable, convention emergence may fail. Salazar et al. [16] (SRA) first extend the existing spreading-based mechanism by incorporating evolutionary algorithm principles to solve aforementioned problems. However, they require extensive additional architecture (e.g., self protection) to be built into agents and not always converge to a high-quality convention. Franks et al. [6] (FGJ) propose inserting a number of influencer agents (IAs) with specific conventions, guides agents to emerge high quality convention efficiently. Unlike the above approaches, Hasan et al. [10] (TA) extend SRA [16] by leveraging the network reorganization mechanism to accelerate convention emergence. However, all these approaches use simple transfer strategy to update convention alternatives based on the information of the current interaction episode instead of all the information of the course of interactions, and thus they cannot maximize agents' accumulated rewards during interaction. This may also cause the qualities of the convention seeds to fluctuate.

Another class of techniques for convention emergence is reinforcement learning [1, 12, 19, 22–25]. Sen and Airiau [19] propose the social learning model to investigate the convention emergence problem over random networks. Later a number of papers [1, 18] extend this work by leveraging more realistic networks to evaluate the influence of agent systems on convention emergence. However, all these works focus on relatively small-size games, and do not address the issue of efficient convention emergence in large convention space problems. Recently, Yang et al. [22] propose a hierarchically heuristic learning strategy for relatively large convention space problem of up to 6 conventions. However, they explore the convention space directly and thus fail to emerge convention when the space becomes significantly large.

Some other distributed learning methods to coordinate the behaviour between agents are also remotely connected to our work. One such solution technique is based on the framework of coordination graphs (CGs) [7, 13]. Under this framework, a number of distributed optimization algorithms (e.g., max-sum) are proposed for learning the behavior of a group of agents for the single-state case in a collaborative multiagent setting. They use the locally optimized action-value function of the individual agents to approximate the maximal global action value. Kok et al. [13] propose Sparse Cooperative Q-learning to extend the single-state case to sequential decision making tasks. One may view the convention emergence problem under a single state as a distributed optimization problem and similar techniques can be used to generate conventions. However, the above techniques require thoughtful design of coordination and communication among all agents. This implicitly requires that all agents are inherently cooperative and social-oriented. In contrast, in convention emergence problems, agents are purely selfish and may have different preference towards different conventions, thus making these approaches unfeasible in convention emergence problems in open and heterogeneous MASs.

Table 1: A coordination game under concept *"belt"*

		Agent 2's actions	
		ribbon	*stripe*
Agent 1's actions	*ribbon*	$+r, +r$	$-r, -r$
	stripe	$-r, -r$	$+r, +r$

3 LANGUAGE COORDINATION GAME

First we define the components to illustrate the language coordination game: (a) the language coordination problem, (b) the interaction model that represents the topology of an MAS, and (c) how a language coordination problem among agents can be simulated as convention emergence problem.

3.1 Language coordination problem

Following the setting in [9], we consider a situation where agents self-organize their communication system from scratch. Initially, each agent randomly generates its lexicon representing a mapping from concepts (C) to words (W). Then agents reorganize their own lexicons via learning from repeated interaction. During each round of interaction, each pair of agents communicate over one concept. The sending agent will get a positive reward if both agents share the same word under the current concept and a negative reward otherwise. We model the single-round interaction between any pair of agents as a two-player n-action coordination game. The action space n consists of all the words that can be chosen under each concept. One simple 2-action coordination game is shown in Table 1, in which under concept *"belt"* each agent can select a word from *ribbon* and *stripe* to communicate. There exist two desirable outcomes *(ribbon, ribbon)* and *(stripe,stripe)* which are both Nash equilibria.

Moreover, communications between agents can be corrupted due to different reasons such as environmental disturb, unreliable transmission channel and so on. Therefore, we assume that rewards are stochastic following certain probability distribution. On the other hand, some concepts may be mapped to more than one word, which leads to synonym. The quality of a lexicon is measured by its specificity. Formally, assuming W_c is the set of words associated with concept c, if $|W_c| > 0$ the concept specificity is calculated as follows: $S_c = 1/|W_c|$. If no word is associated with concept c, $S_c = 0$. The lexicon specificity S is defined as the average of all concepts' specificities:

$$S = \sum_{c \in C} S_c/|C|, |C| > 0 \quad (1)$$

Thus, a highest-quality lexicon is the one with a one-to-one mapping. Assuming all agents are rational, each agent will update its lexicon towards a high-quality and consistent lexicon. The convention space is defined as the space of all lexicons, which is exponential to the number of words. We assume that the number of concepts and words are equal, and the convention space is $|W|^{|C|}$. We can see that the space becomes significantly large even for a moderate number of words and concepts.

3.2 Interaction model

We consider a population N of agents where each agent is connected following a static network topology. In each round, each agent selects a neighbor randomly to interact with. The interaction model of the MAS is represented by the undirected graph, $G = (V, E)$, where G means the network structure, V is the set of nodes, and E is the set of edges between nodes. If $(v_i, v_j) \in E$, then v_i, v_j are neighbors. $N(i)$ is the set of the neighbors of agent i, i.e., $N(i) = \{v_j | (v_i, v_j) \in E\}$. Three representative network structures are considered: random network, small-world network [21], and scale-free network [2].

3.3 Convention emergence problem

Our goal is to engineer the emergence of the high-quality lexicon in an open and large MAS. We can model this language coordination problem among agents as a convention emergence problem. A desirable convention corresponds to all agents adopting the same high-quality lexicon with a one-to-one mapping. Similar to TA [10], each interaction episode for each agent consists of a sequence of interactions. We are interested in investigating how a population of agents can learn to coordinate on a consistent convention through repeated interactions. The original convention space is $|W|^{|C|}$, which is too large to learn directly. Thus, we decouple the lexicon convention into concept-word mappings. Then each lexicon which maps each concept to a single word can be defined as a Markov strategy. By transforming the original single-state convention learning problem into a muti-state multiagent coordination problem, we propose modeling the dynamics of lexicon evolutions during each episode of interaction as a two-player Markov Game, which is defined by a tuple $\langle S, \{A_i\}_{i \in N}, \{R_i\}_{i \in N}, T \rangle$, where

- S is the set of states and represents the set of concepts.
- N is the total amount of agents in the network.
- $\{A_i\}_{i \in N}$ is the collection of action sets, $A_1, A_2, \ldots, A_n$, one for each agent in the network. Each action set A_i contains all the words.
- $\{R_i\}_{i \in N}$ is the set of payoff functions, $R_i : S \times A_i \times A_j \to \mathcal{R}$ is the payoff function for agent i, where agents i and j are the interacting agents at the current time-step of interaction. It is positive if two agents select the same action and negative otherwise. R_i satisfies the Gaussian distribution.
- T is the state transition function: $S \times A_i \times A_j \to Prob(S)$, where agents i and j are the interacting agents. The probability distribution over next states for each state and joint action is represented as $P(s'|s, (a_i, a_j))$. The state transition function models the concepts usage frequencies.

Note that this Markov game is slightly different from the normal definition of a Markov game, in that the set of agents are not static since agent i randomly chooses a neighbor to interact during each round of interaction. And our approach

does not put any limitation on the concept usage frequency. Any reasonable concept frequency distribution specific to the lexicon can be used.

4 CONVENTION EMERGENCE FRAMEWORK

Algorithm 1 describes the overall interaction framework of agents playing language coordination games. During each round of interaction, each agent first randomly selects a neighbor to interact with (Line 4); second, each agent chooses a word for the current concept following its learning strategy and then communicates with the neighbor (Line 5); after that, it receives the corresponding reward to update strategy (Line 6); last, another concept is selected following the concept usage frequency and the next-round communication starts (Line 7). Each agent continues the interaction with others for λ rounds during each episode of interactions.

Since we model the language communication dynamics among each pair of agents during each episode of interaction as a Markov Game, it is natural to leverage reinforcement learning based (RL-based) approaches to design agents' strategies to promote convention emergence. There are mainly two benefits of RL-based approaches. First, RL-based learning strategies enable agents to compute the lexicon quality in a more accurate way. Each agent learns to maximize the accumulated quality during the course of interactions, which can make better use of the historical experiences. In contrast, the quality computation mechanism of spreading-based approaches are based on the information of the current interaction episode, which may cause the quality of each lexicon to fluctuate frequently.

Second, the other benefit of RL-based approaches is to allow updating lexicons in a more fine-grained way. RL-based approaches allow agents to calculate the quality for each concept-word mapping and improve concept-word mapping separately and incrementally until convention emerges. Each agent can update its value function during each round of interaction. In contrast, spreading-based approaches use transfer strategy to update lexicons. Each agent chooses an optimal lexicon from all its neighbors' lexicons to simply replicate some part of this lexicon with its own. This may cause that the optimal lexicon low-quality mappings replace its high-quality mappings to reduce its lexicon quality to slow down the convergence rate. Based on the above analysis, we propose two RL-based strategies: multiple-Q and multiple-R, which will be described in details next.

4.1 Multiple-Q learning strategy

4.1.1 Value function update. Q-learning may perform poorly in stochastic environments due to overestimation. Double Q-learning replaces the maximum estimator of Q-learning with double estimator to avoid the positive bias [5, 11]. Double estimator splits the sample set into two independent sets. Agents use one estimator to determine the action with the maximum value and use the other one to provide the target value estimation. We extend the distributed value function

Algorithm 1 The framework of convention emergence using RL-based approaches

```
1: for each episode of interaction do
2:   for each agent i ∈ N do
3:     while state transition times < λ do
4:       neighborSelection();
5:       actionSelection();
6:       UpdatingStrategy();
7:       stateTransfer();
8:     end while
9:   end for
10: end for
```

[17] using the idea of double estimator to the social learning framework. Specifically, we propose a new weighted multiple estimator to improve each agent's Q-value estimation by utilizing the estimators of agents in the neighborhood.

First, agent i selects the action a^* with the maximum payoff under next state s' from its own Q-value function:

$$a^* = argmax_a Q_i(s', a) \tag{2}$$

Second, each agent observes its neighbors' Q-values of the state-action pair (s', a^*) using the observation mechanism [8], and computes a weighted target value to replace the original maximum estimator:

$$V(s') = \sum_{j \in \{N(i) \cup \{i\}\}} f(i,j) Q_j(s', a^*) \tag{3}$$

where $N(i)$ is the set of neighbors of agent i. The weight $f(i,j)$ reflects the relative importance of agent j in agent $i's$ neighborhood. Note that we assume the order of message (Q-values) passing is synchronous, i.e., each agent's V-values are updated after receiving the Q-values from all neighbors. In case the Q-values of certain neighbors cannot be received due to communication errors, the previous Q-value received from that neighbor can be used instead. One way of defining $f(i,j)$ is using each agent's connection degree as follows:

$$f(i,j) = degree(j)/totalDegree \tag{4}$$

where totalDegree is the sum of the degree of the agent i itself and its neighbors. Since the connection degree of each agent determines its interaction frequency with other agents and thus reflects its influence degree on other agents, we use it as the criterion to model the importance of each agent on convention emergence.

Lastly, agent i updates its Q-values of state-action pair (s, a) as follows:

$$Q_i(s,a) \leftarrow Q_i(s,a) + \alpha(r + \gamma V(s') - Q_i(s,a)) \tag{5}$$

where α is the learning rate, r is the immediate reward under the current state-action pair, and γ is the discount factor. $V(s')$ is the weighted average of Q-values under state-action pair (s', a^*) of agent i and its neighbors.

4.1.2 Action selection. During the communication period, each agent i records the states and the corresponding actions that have been selected. Every time agent i selects an action,

it excludes the actions recorded in the set RA_i and then uses the $\varepsilon - greedy$ strategy to choose the action with the highest value under the current state:

$$a_i \leftarrow \begin{cases} argmax_{a \notin RA_i} Q_i(s,a) & \text{with probability} \quad 1-\varepsilon \\ \text{a random action} \in \{A_i \setminus RA_i\} & \text{with probability} \quad \varepsilon \end{cases} \tag{6}$$

Note that if there exist multiple actions with the highest value, then one of them is selected randomly. Since an agent only selects from those words that have not been selected by other concepts, this avoids the situation that a word may correspond to multiple concepts. Therefore it significantly reduces synonym to improve the quality of dominant lexicons.

4.2 Multiple-R learning strategy

The above multiple-Q learning strategy focuses on improving the estimation of Q-values by leveraging the Q-value information from neighbors. However, the root cause of overestimating Q-values comes from the inaccurate estimation of immediate rewards, which propagates back to the corresponding Q-values and are accumulated continuously during Q-value updates. To alleviate this problem, here we propose an alternative way of updating Q-functions by improving the estimation accuracy of each immediate reward directly. We propose that each agent first computes an average of its own rewards and then a weighted average of its local average rewards and those of its neighbors during each update.

First, each time agent i receives its reward $r(s,a)$, it computes its average reward during the course of interaction:

$$R_i(s,a) = R_i(s,a) + 1/n_i(s,a)(r - R_i(s,a)) \tag{7}$$

where $n_i(s,a)$ is the number of times visiting (s,a).

After that, agent i collects all the average rewards under the same state-action pair from its neighborhood using the observation mechanism [8]. Formally, suppose that agents can observe the information of their neighbors' average rewards: $\{R_i(s,a), R_{j_1}(s,a), R_{j_2}(s,a), \ldots, R_{j_n}(s,a)\}$, where $R_i(s,a)$ is the average payoff of agent i and the rest are the average payoffs of its neighbors with the same state-action pair. We get a weighted average of rewards of agent i and its neighbors:

$$\bar{r} = \sum_{j \in \{N(i) \cup \{i\}\}} f(i,j) R_j(s,a) \tag{8}$$

where $N(i)$ is the set of agent $i's$ neighbors and $f(i,j)$ is the weight that agent j influences agent i.

Lastly, agent i updates its Q-table as follows:

$$Q_i(s,a) \leftarrow Q_i(s,a) + \alpha(\bar{r} - Q_i(s,a)) \tag{9}$$

where α is the learning rate and $\bar{r}$ is the weighted average.

Multiple-R uses $\varepsilon - greedy$ strategy for action selection. Each time with probability $1-\varepsilon$, it selects an action with maximum Q-value under current state from actions that have not been selected under other states, and a tie is broken randomly if multiple optimal actions exist. With the probability ε , one action is selected randomly.

4.3 Teacher-student mechanism

The teacher-student paradigm is very flexible because the teacher and student roles may be played by both humans and autonomous agents. It can be used concomitantly with other approaches to accelerate learning. We introduce an additional teacher-student mechanism on top of the previously two learning strategies to further accelerate the lexicon convergence speed among agents. We integrate the advising from teachers in the action selection to reduce the interaction times, in which an experienced teacher agent can advise a student to guide her action exploration. In this procedure each agent i is equipped with a tuple $\langle P^i_{ask}, P^i_{give}, b^i_{ask}, b^i_{give}, G^i \rangle$.

- P^i_{ask} is the probability of agent i asking the neighbors for action advice.
- P^i_{give} represents the probability of agent i giving advice when requested. It encodes the confidence of agent i in its own policy.
- b^i_{ask} is the budget constrain of asking for advice for agent i.
- b^i_{give} is the budget constrain of giving opinions for agent i. This models the intrinsic willingness of agent i giving advice.
- G^i is the set of all reachable agents for agent i. It is equal to its neighborhood.

The two probabilities P^i_{ask} and P^i_{give} change over time. Intuitively P^i_{ask} should be decreased as agents' behaviors converge. In contrast P^i_{give} increases over time since the confidence of teacher agents increases as more experience is gained.

The probability P^i_{ask} only specifies the probability of an agent asking for advices. Another related question is which neighbor a student agent should resort to ask for advice. One straightforward way of defining the probability of an agent asking for advice from a neighbor agent is based on their relative closeness, which can be defined as their shortest distance. Formally agent i asks for advice from neighbor j according to the following probability distribution:

$$P^j_i = \frac{\frac{1}{d^j_i}}{\sum_{j=1}^{N_i} \frac{1}{d^j_i}} \tag{10}$$

where P^j_i represents the probability of agent i choosing neighbor j, N_i is the neighborhood size and d^j_i is the closeness between agent i and j. The neighborhood size is set as 1 by default, which means agents interact with the immediate neighbors.

Algorithm 2 describes the action selection strategy for advisees, in which we use the connection degree to evaluate the expertise of an advisor. An agent with higher degree can interact with more agents to gain more experience and improve its expertise in giving advices. At each time step, agent i observes the advice asking budget b^i_{ask}. If the value of b^i_{ask} is greater than zero, agent i asks for advice with probability P^i_{ask} (Line 4). For each reachable agent j, agent i asks for its advice with probability P^j_i (Lines 7 - 9). If agent i receives action advice, its budget b^i_{ask} is decremented by 1. Then it calculates the relative weighted frequency f_a of

each action a in the receiving set Π using the advisor degree as the weight w_a and selects the action with the maximum frequency as the advice (Lines 10 - 18). Finally, if no advice is provided, agent i selects its action following the same action selection strategy as multiple-Q (Lines 16 - 18).

Algorithm 3 describes how an advisor gives opinions. When agent i receives a call for advice, it first measures whether there is still any budget b^i_{give} available. If yes, agent i provides advice using any learning strategy (e.g., multiple-Q) with probability P^i_{give} (Line 3). Each time an advice is given, the advice giving budget b^i_{give} is decremented by 1 (Lines 4 - 9).

The remaining question is how to define P^i_{ask} and P^i_{give}. Intuitively, the lower confidence agent i has under state s, the higher the probability of agent i should ask for advice. Agent i should be more likely to give advice when it is more confident about currently learned strategy. Formally, we have:

$$P^i_{ask}(s, \Upsilon) = (1 + V_i(s))^{-\Upsilon^i(s)} \tag{11}$$

$$P^i_{give}(s, \Psi) = 1 - (1 + V_i(s))^{-\Psi^i(s)} \tag{12}$$

where Υ and Ψ are confidence functions about the number of times that state s is encountered and $V_i(s)$ is the maximum action value under state s. The confidence functions Υ and Ψ are defined as follows:

$$\Upsilon^i(s) = \sqrt{n^i_{visit}(s)} \tag{13}$$

$$\Psi^i(s) = \log_2 n^i_{visit}(s) \tag{14}$$

where $n^i_{visit}(s)$ is the number of times that agent i visites state s.

Algorithm 2 action selection strategy for an advisee

```
1: for during the interaction period do
2:   observe current state s;
3:   if b_ask > 0 then
4:     P ← P_ask(s, Υ);
5:     p ← getRandomNumber(0, 1);
6:     if p < P then
7:       for each agent j ∈ G do
8:         ask agent j for advice with probability P_i^j;
9:       end for
10:      define Π as the set of receiving advices;
11:      if Π ≠ ∅ then
12:        b_ask ← b_ask − 1;
13:        set f_a for each action a ∈ Π as 0;
14:        for each action a ∈ Π do
15:          f_a = f_a + w_a/|Π|
16:        end for
17:        a ← argmax_a f_a;
18:      end if
19:    end if
20:  end if
21:  if no action is executed then
22:    perform the action selection strategy in any learning
       strategy(e.g., multiple-Q);
23:  end if
24: end for
```

Algorithm 3 action selection strategy for an advisor

```
1: observe the advisee's state s;
2: if b_give > 0 then
3:   P ← P_give(s, Ψ);
4:   p ← getRandomNumber(0, 1);
5:   if p < P then
6:     choose action using any learning strategy (e.g.,
       multiple-Q);
7:     response to advice requirement;
8:     b_give ← b_give − 1;
9:     return π(s)
10:  end if
11: end if
12: return ∅
```

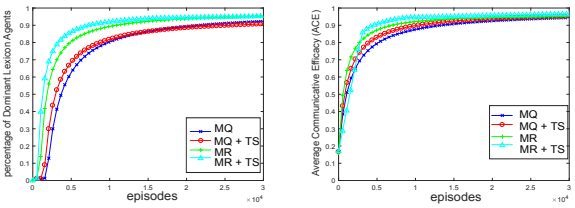
5 SIMULATION AND RESULTS ANALYSIS

We conduct experiments on the language game with 10^{10} convention space to compare the performance of multiple-Q (MQ), multiple-Q with teacher-student mechanism (MQ+TS), multiple-R (MR), and multiple-R with teacher-student mechanism (MR+TS) with the state-of-the-art approaches SRA [16], FGJ [6], and TA [10], (detailed in Section 2), on different networks including random, small-world and scale-free networks. The existing RL-based strategies [1, 19, 25], are not compared here since they are only designed for small-size convention emergence problems. We define the dominant lexicon convention as the one adopted by the largest number of agents. We use the following metrics for comparison:

- Effectiveness: If a mechanism allows agents to converge into a lexicon convention within an acceptable amount of time, then we say it is effective.
- Efficiency: It measures how fast agents converge into a dominant convention. It is defined as the number of interaction episodes required before converging into a dominant lexicon.
- Average Communicative Efficacy(ACE): It is defined as the proportion of successful communications in total. It reflects the coordination level of the overall system.
- Dominant Lexicon Specificity(DLS): It is defined as the specificity of the dominant lexicon, which is the sum of the reciprocal of the number of words mapping to each concept. Any lexicon with a one-to-one mapping has a specificity of 100%.

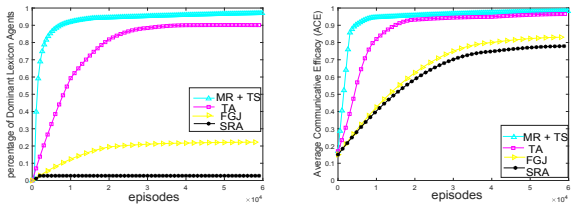
5.1 Simulation Setup

We use Watts and Strogatz small-world [21] and Barabasi-Albert model [2] to create small-world and scale-free network respectively. Each network consists of 1000 agents and the average node degree is set to 20. Following the settings in previous work TA, each lexicon contains 10 concepts and 10 words and thus the total convention space is 10^{10}. We adopt the random frequency distribution as the concepts usage frequencies, while other types of distributions can be used as well. For the teacher-student mechanism, budgets



(a) ACC (b) ACE

Figure 1: Comparison over small-world networks.



(a) ACC (b) ACE

Figure 2: Comparison over small-world networks.

b_{ask} and b_{give} for each agent are set to 4000. We set the closeness between teachers and students to be 3. For the FGJ mechanism, 50 influencer agents are randomly deployed into the network initially following the setting in FGJ. Those influencer agents start with a lexicon with 100% specificity. Each simulation executes 60000 time steps where a time step refers to a single run of the program and all the results are averages over 50 realizations for each network.

5.2 Simulation Results

5.2.1 Comparison among our RL-based strategies. Figure 1 (a) shows the dynamics of the percentage of agents converged into a convention (ACC) over time for MQ, MQ+TS, MR, and MR+TS in small-world networks. First, we observe that MR strategy converges faster than MQ strategy. Second, the TS mechanism can improve the convergence speed for both MR and MQ strategies. Figure 1 (b) shows how average communicative efficacy (ACE) evolves over time for MQ, MQ+TS, MR, and MR+TS. We can also observe the similar phenomenon of ACC and ACE in random and scale-free networks, which are omitted here due to the limited space.

Table 2 summarizes the simulation results in terms of the convergence speed and the dominant lexicon specificity when 80% and 90% of agents converged into a convention respectively for all three networks. For all these networks, MR outperforms MQ: MR strategy improves the convergence rate by 20% comparing with MQ strategy in random and small-world networks and MR performs slightly better than MQ in scale-free networks. The reason is that MQ focuses on the optimization of Q-value estimation by leveraging the Q-value information from neighbors while MR optimizes the immediate reward estimation which is the root cause of overestimating Q-values. We observe that TS mechanism boosts the performance for both learning strategies for all three networks. There is an increase of approximately 15% in convergence speed using TS mechanism for each network. This is due to the advising procedure in TS. It enables experienced agents to guide the action exploration of student agents to reduce the interaction times between agents. This explains the accelerated convergence rates of MQ and MR. We also investigate the influence of the budgets in TS on the learning performance and find that the convergence is accelerated as the budgets increase until 3000. After that, budgets have negligible effect on the performance. On the other hand, all our approaches achieve the highest DLS value of 1.0.

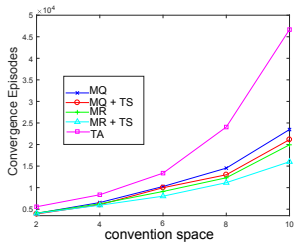


Figure 3: Comparison of scalability as the convention space increases over random networks.

5.2.2 Comparison with the state-of-the-art approaches. Based on the above simulation results, in the following, we choose MR+TS in random and small-world networks and MQ+TS in scale-free networks to compare with the state-of-the-art approaches respectively. Figure 2 (a) shows that how the percentage of agents converged into a convention (ACC) evolves over time for MR+TS, TA, FGJ and SRA in small-world networks. We observe that MR+TS clearly outperforms the other approaches. Agents through reinforcement social learning with decoupled convention space converge into a dominant lexicon much faster than the existing spreading-based approaches. Figure 2 (b) shows how average communicative efficacy (ACE) evolves over time for MR+TS, TA, FGJ and SRA. The performance of MR+TS is better than the state-of-the-art approaches. By comparing the results in Figure 2 (a) and (b), we observe that the average communication efficacy is high even if there exists no dominant lexicon. We hypothesize that it is because the lexicons agents adopt share a large percentage of common concept-word mappings. Thus, the communication efficiency can still be quite high even if no consistent lexicon emerges. The similar results of ACC and ACE can be observed in random and scale-free networks and we omit them here due to the limited space.

Table 2 shows the simulation results in terms of the convergence speed and the dominant lexicon specificity when 80% and 90% of agents converged into a convention respectively over all three topologies. We can see that agents using our strategy require shorter convergence time for each network while TA performs the best among all existing approaches. As we discussed above, MR+TS is chosen for comparison in random and small-world networks. It only requires approximately 33% of TA's execution time steps to reach a convention for these two networks. MQ+TS is chosen for comparison

Table 2: Performance Comparison: %ACC refers to the percentage of agents converging into a convention at time step t. DLS refers to dominant lexicon specificity at time-step t.

	Random			SmallWorld			ScaleFree		
	%ACC	t	DLS	%ACC	t	DLS	%ACC	t	DLS
MQ	80	8993	1.0	80	8224	1.0	80	9159	1.0
	90	20542	1.0	90	17564	1.0	90	18624	1.0
MQ+TS	80	7654	1.0	80	8355	1.0	80	8071	1.0
	90	18475	1.0	90	16843	1.0	90	15839	1.0
MR	80	7201	1.0	80	5831	1.0	80	8311	1.0
	90	14224	1.0	90	13735	1.0	90	18688	1.0
MR+TS	80	5396	1.0	80	4615	1.0	80	7649	1.0
	90	12968	1.0	90	12534	1.0	90	16162	1.0
TA	80	27563	0.86	80	11834	0.85	80	25461	0.91
	90	48712	0.88	90	31422	0.88	90	X	N/A
FGJ	80	45121	1.0	80	X	N/A	80	35635	1.0
	90	X	N/A	90	X	N/A	90	X	N/A
SRA	80	X	N/A	80	X	N/A	80	X	N/A
	90	X	N/A	90	X	N/A	90	X	N/A

in scale-free networks. It only requires around 30% of TA's execution time steps before 80% of agents converging into a dominant lexicon. In the worse case, TA fails to have 90% of agents to converge into a dominant lexicon in scale-free networks. Moreover, we observe relatively poor performance for FGJ mechanism. It requires more than 45000 rounds for 80% agents to use the dominant lexicon in random and scale-free networks while it fails to converge in small-world networks. SRA performs the worst as it fails to converge to a convention in all three networks. The results confirm our theoretical analysis in Section 4: decoupling the convention into correlated subconventions and concurrent learning over each subconventions can significantly improve the convergence efficiency than the state-of-the-art approaches.

On the other hand, we observe the dominant lexicon specificities of our approaches are better than those of TA and SRA for all three networks. The reason is that we assume each agent only selects from words that have not been selected by other concepts, which reduces synonym to improve the lexicon quality. FGJ approach achieves the same performance as ours in terms of DLS. We hypothesize that this is because FGJ has the advantage of initializing a fraction of the agents with the high-quality lexicons. Next we extensively evaluate the coordination level of different strategies under each network and and the results are summarized in Table 3. The differences of the average communicative efficacy between our strategies and TA are statistically significant. It indicates that our strategies are robust and can achieve higher coordination level than the state-of-the-art approaches across different networks. Finally we evaluate the scalability of different approaches shown in Figure 3. We can see that though convergence episodes of all approaches increase nearly in an exponential way as the convention space increases, the increases of convergence episodes of our approaches are slower than that of TA. Thus the scalabilities of our approaches are a little better than the state-of-art approaches.

Table 3: ACE Performance Comparison: ACE refers to average communicative efficacy after convergence.

	ACE		
	Random	Small-World	Scale-Free
MQ	0.956	0.963	0.971
MQ+TS	0.969	0.952	0.983
MR	0.975	0.981	0.977
MR+TS	0.980	0.994	0.982
TA	0.924	0.935	0.951
FGJ	N/A	N/A	N/A
SRA	N/A	N/A	N/A

6 CONCLUSION AND FUTURE WORK

In this paper, our goal is to design RL-based strategies to evolve a consistent convention among a large convention space. We model the dynamics of lexicon evolutions during an interaction episode as a Markov game. Under this formalization, we propose multiple-Q and multiple-R with teacher-student mechanism. Extensive experiments show that our approaches outperform the state-of-the-art approaches in terms of convention emergence efficiency and quality. As future work, one worthwhile direction is to investigate how to incorporate our decoupled RL-based strategies into a hierarchical learning framework to further accelerate the convergence speed. We also intend to evaluate the efficacy on dynamic topologies and investigate how to use rewiring strategy to accelerate learning in dynamic MASs.

ACKNOWLEDGMENTS

The work is supported by the National Natural Science Foundation of China under Grant No. 61702362, No. 61471259, and Special Program of Artificial Intelligence of Tianjin Municipal Science and Technology Commission (No.:569 17ZXRGGX00150).

REFERENCES

[1] Stéphane Airiau, Sandip Sen, and Daniel Villatoro. 2014. Emergence of conventions through social learning. *Autonomous Agents and Multi-Agent Systems* (2014), 779–804.

[2] Albert-László Barabási and Réka Albert. 1999. Emergence of scaling in random networks. *Science* (1999), 509–512.

[3] Felipe Leno da Silva, Ruben Glatt, and Anna Helena Reali Costa. 2017. Simultaneously learning and advising in multiagent reinforcement learning. In *Proceedings of the 16th International Conference on Autonomous Agents and MultiAgent Systems*. 1100–1108.

[4] Jordi Delgado. 2002. Emergence of social conventions in complex networks. *Artificial Intelligence* (2002), 171–185.

[5] Carlo DąŕEramo, Marcello Restelli, and Alessandro Nuara. 2016. Estimating maximum expected value through gaussian approximation. In *Proceedings of the 33rd International Conference on Machine Learning*. 1032–1040.

[6] Henry Franks, Nathan Griffiths, and Arshad Jhumka. 2013. Manipulating convention emergence using influencer agents. *Autonomous Agents and Multi-Agent Systems* (2013), 1–39.

[7] Carlos Guestrin, Daphne Koller, and Ronald Parr. 2002. Multiagent planning with factored MDPs. In *Proceedings of the 16th International Conderence on Neural Information Processing Systems*. 1523–1530.

[8] Jianye Hao and Ho-fung Leung. 2013. The Dynamics of Reinforcement Social Learning in Cooperative Multiagent Systems.. In *Proceedings of the 23rd International Joint Conference on Artificial intelligence*. 184–190.

[9] Mohammad Rashedul Hasan. 2014. Communication convention formation in large multiagent systems. In *Proceedings of the 13th International Conference on Autonomous agents and Multiagent Systems*. 1747–1748.

[10] Mohammad Rashedul Hasan, Anita Raja, and Ana LC Bazzan. 2015. Fast Convention Formation in Dynamic Networks Using Topological Knowledge.. In *Proceedings of the 29th AAAI Conference on Artificial Intelligence*. 2067–2073.

[11] Hado V Hasselt. 2010. Double Q-learning. In *Proceedings of the 24th International Conference on Neural Information Processing Systems*. 2613–2621.

[12] Shuyue Hu and Ho-fung Leung. 2017. Achieving Coordination in Multi-Agent Systems by Stable Local Conventions under Community Networks. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence*. 4731–4737.

[13] Jelle R Kok and Nikos Vlassis. 2006. Collaborative multiagent reinforcement learning by payoff propagation. *Journal of Machine Learning Research* (2006), 1789–1828.

[14] Mihail Mihaylov, Karl Tuyls, and Ann Nowé. 2014. A decentralized approach for convention emergence in multi-agent systems. *Autonomous Agents and Multi-Agent Systems* (2014), 749–778.

[15] Josep M Pujol, Jordi Delgado, Ramon Sangüesa, and Andreas Flache. 2005. The role of clustering on the emergence of efficient social conventions. In *Proceedings of the 19th International Joint Conference on Artificial intelligence*. 965–970.

[16] Norman Salazar, Juan A Rodriguez-Aguilar, and Josep L Arcos. 2010. Robust coordination in large convention spaces. *AI Communications* (2010), 357–372.

[17] Jeff Schneider, Weng-Keen Wong, Andrew Moore, and Martin Riedmiller. 1999. Distributed value functions. *Robotics Institute* (1999), 264.

[18] Onkur Sen and Sandip Sen. 2010. Effects of social network topology and options on norm emergence. In *Coordination, Organizations, Institutions and Norms in Agent Systems V*. Springer, 211–222.

[19] Sandip Sen and Stéphane Airiau. 2007. Emergence of norms through social learning.. In *Proceedings of the 20th International Joint Conference on Artificial intelligence*. 1512.

[20] Daniel Villatoro, Jordi Sabater-Mir, and Sandip Sen. 2013. Robust convention emergence in social networks through self-reinforcing structures dissolution. *ACM Transactions on Autonomous and Adaptive Systems* (2013), 2.

[21] Duncan J Watts and Steven H Strogatz. 1998. Collective dynamics of'small-world'networks. *Nature* (1998), 440.

[22] Tianpei Yang, Zhaopeng Meng, Jianye Hao, Sandip Sen, and Chao Yu. 2016. Accelerating Norm Emergence Through Hierarchical Heuristic Learning. In *Proceedings of the 25th European Conference on Artificial Intelligenc*. 1344–1352.

[23] Chao Yu, Hongtao Lv, Fenghui Ren, Honglin Bao, and Jianye Hao. 2015. Hierarchical learning for emergence of social norms in networked multiagent systems. In *Proceedings of the 28th Australasian Joint Conference on Artificial Intelligence*. 630–643.

[24] Chao Yu, Hongtao Lv, Sandip Sen, Jianye Hao, Fenghui Ren, and Rui Liu. 2016. An adaptive learning framework for efficient emergence of social norms. In *Proceedings of the 15th International Conference on Autonomous Agents and Multiagent Systems*. 1307–1308.

[25] Chao Yu, Guozhen Tan, Hongtao Lv, Zhen Wang, Jun Meng, Jianye Hao, and Fenghui Ren. 2016. Modelling Adaptive Learning Behaviours for Consensus Formation in Human Societies. *Scientific Reports* (2016), 27626.