A Formal Model for Multiagent Q-Learning Dynamics on Regular Graphs

Chen Chu[1,2], **Yong Li**[3], **Jinzhuo Liu**[2,3], **Shuyue Hu**[4], **Xuelong Li**[2] and **Zhen Wang**[2,5*]
[1]School of Statistics and Mathematics, Yunnan University of Finance and Economics
[2]School of Artificial Intelligence, OPtics and ElectroNics (iOPEN), Northwestern Polytechnical University
[3]School of Software, Yunnan University
[4]Shanghai Artificial Intelligence Laboratory
[5]School of Cybersecurity, Northwestern Polytechnical University
w-zhen@nwpu.edu.cn

Abstract

Modeling the dynamics of multi-agent learning has long been an important research topic. The focus of previous research has been either on 2-agent settings or well-mixed infinitely large agent populations. In this paper, we consider the scenario where n Q-learning agents locate on regular graphs, such that agents can only interact with their neighbors. We examine the local interactions between individuals and their neighbors, and derive a formal model to capture the Q-value dynamics of the entire population. Through comparisons with agent-based simulations on different types of regular graphs, we show that our model describes the agent learning dynamics in an exact manner.

1 Introduction

Multi-agent learning has recently received much attention, as learning (in particular, reinforcement learning) has found wide applications in many real-world multi-agent systems (MASs), such as multiplayer online games [Brown and Sandholm, 2019], traffic control [Zhou *et al.*, 2019], motion coordination [Liu *et al.*, 2021a] and online auction [Jin *et al.*, 2018]. However, while learning under single-agent settings has gained a strong theoretical foundation, learning under multi-agent settings has still remained an open problem [Bailey and Piliouras, 2019].

One of the main foci of previous theory about multi-agent learning is to formalize and examine the learning dynamics [Bloembergen *et al.*, 2015]. As Tuyls and Parsons voiced [2007], this not only provides a better theoretical understanding of existing algorithms, but also facilitates many important tasks, such as parameter tuning and the design of new algorithms.

Early works in this line of research mostly consider that there are only few learning agents in a MAS [Panait *et al.*, 2008; Hennes *et al.*, 2009; Galstyan, 2013]. In the pioneering work, Tuyls et al. [2003] developed a system of replicator equations to model the policy (or strategy) dynamics of two Q-learning agents in repeated 2-player matrix games. Subsequent studies generalized this evolutionary game-theoretic approach to other learning algorithms, such as FAQ-learning, infinitesimal gradient ascent and regret minimization; as such, formal models for the dynamics of two agents that apply these learning algorithms have been proposed [Kaisers and Tuyls, 2010; Klos *et al.*, 2010].

In this paper, in contrast to the 2-agent settings considered in most previous works, we consider that there are a large number of Q-learning agents in a MAS and that the underlying topology is a *regular graph*. Recently, Hu et al. [2019] focused on a similar setting in which there are infinitely many Q-learning agents in a MAS, and they developed a Fokker-Planck equation-based model to characterize the population dynamics. However, we note that the underlying topology of their MASs are essentially well-mixed populations; put differently, for each game-play, every agent interact with another agent that is randomly drawn from the population.

The study of learning dynamics on graphs is important. Compared with all-to-all interactions in well-mixed populations, it is more realistic to assume some constraints in the interactions between agents, due to the high communication costs between remote agents in many practical scenarios, such as sensor networks, power grids, and unmanned vehicular networks [Zhang *et al.*, 2021; Delgado, 2002]. Moreover, multi-agent learning on graphs has recently gained growing interest [Zhang *et al.*, 2018; Wang *et al.*, 2018; Liu *et al.*, 2021b]. Previous findings, which rely on agent-based simulations, suggest that topological structure can significantly influence system evolution [Villatoro *et al.*, 2011; Hu and Leung, 2017; Sen and Sen, 2009].

The model of Hu et al. [2019] was proposed for learning in well-mixed populations; thus, it is by nature *inappropriate* for learning on regular graphs, which features constrained interactions among agents. To address this knowledge gap, we develop a new formal model for the learning dynamics on regular graphs. More specifically, we consider that given a regular graph with degree k, each vertex represents a Q-learning agent and agents learn from repeated symmetric games with their neighbours on the graph. To provide an accurate description for the learning dynamics, we first focus on the Q-values dynamics of individual agents. By capturing the ef-

*Corresponding Author

fects of different neighbour configurations of a regular graph, we derive a differential equation that describes the change of Q-values for individual agents. Then, following the approach of Hu et al. [2019], we derive a system of equations, one of which is a partial differential equation, to model the time evolution of the Q-values distribution of the system. To the best of our knowledge, our model is the first formal model for the dynamics of multiagent learning on regular graphs.

In the experiments, we consider two specific types of regular graphs: translational symmetric lattice and random regular graphs. We validate our model by showing that the Q-learning dynamics predicted by our model match the actual dynamics in agent-based simulations across different games and different regular graphs. In addition, we show that in prisoner's dilemma games, as the temptation to defect and the degree of the regular graph increases, Q-learning agents are less likely to cooperate. However, if the temptation to defect is low, the effects of the degree of the regular graph becomes insignificant.

2 A Multiagent Q-Learning Framework on Regular Graphs

2.1 Agent Interaction on Graphs

The interaction topology of agents is generalized by an undirected connected graph $\mathcal{G}(\mathcal{N}, \mathcal{E}, \mathcal{B})$, where $\mathcal{N} = \{1, \dots, n\}$ is the vertex set (each agent occupies a vertex), $\mathcal{E} \subseteq \mathcal{N} \times \mathcal{N}$ is the edge set, and $\mathcal{B} = [b_{ij}]_{n\times n}$ is the adjacency matrix. An edge denoted by $e_{ij} = (i, j), i, j \in \mathcal{N}$ means that vertex i is connected to vertex j. Define the neighbor set of vertex i as $\mathcal{N}_i = \{j : e_{ij} \in \mathcal{E}, j \neq i\}$. In the adjacency matrix $\mathcal{B}$, all diagonal elements b_{ii} are zero, off-diagonal elements $b_{ij} = 1$ if and only if $e_{ij} \in \mathcal{E}$, otherwise, $b_{ij} = 0$. In this paper, we consider k-regular graphs such that each agent has exactly k neighbors.

Given a two-player symmetric game G with a set $\mathcal{A}$ of actions. The multiagent Q-learning on regular graphs proceeds as follows. For every time step t, an agent i chooses an action $a_j \in \mathcal{A}$ and plays the game G with each of its k neighbors. The immediate reward $r_t^i(a_j)$ of agent i is averaged over all the k games it played. Based on the immediate rewards, agents adapt their strategies using Q-learning. We summarize the learning framework in Algorithm 1.

2.2 Two-player Symmetric Game

A symmetric normal form game is defined as $G = \langle \mathcal{A}, \mathcal{R} \rangle$ where $\mathcal{A} = \{a_1, \dots, a_m\}$ is the available action set for each agent and $\mathcal{R}$ is the payoff matrix of the row player

$$\mathcal{R} = \begin{bmatrix} r_{a_1a_1} & \cdots & r_{a_1a_m} \\ \vdots & \ddots & \vdots \\ r_{a_ma_1} & \cdots & r_{a_ma_m} \end{bmatrix},$$

where $r_{a_ia_j}$ denotes the reward if the row player plays action a_i and the column player plays action a_j.

2.3 Q-Learning with Boltzmann Exploration

For a Q-learning agent i, it maintains a vector of Q-values $\mathbf{Q}^i = [Q^i(a_1), \dots, Q^i(a_m)]^\top$ which estimates the reward of using each action. The agent policy is defined by

$$\mathbf{x}_t^i = [x_t^i(a_1), \dots, x_t^i(a_m)]^\top,$$

where $x_t^i(a_j), \forall a_j \in \mathcal{A}$, denote the probability of agent i choosing action a_j at time t. Agents choose their actions based on Boltzmann probability

$$x_t^i(a_j) = \frac{e^{\beta Q_t^i(a_j)}}{\sum_{\forall a \in \mathcal{A}} e^{\beta Q_t^i(a)}}, \tag{1}$$

where β is the Boltzmann exploration temperature. When β is small, the agent tends to be more exploratory. If $\beta = 0$, agents will choose their actions randomly. If $\beta \to \infty$, agents will become greedy (that is, they choose the action with the highest Q-value).

For agent i, if it plays action a_j at time t, it will update the Q-value of action a_j as follows

$$Q_{t+1}^i(a_j) \leftarrow Q_t^i(a_j) + \alpha[r_t^i(a_j) - Q_t^i(a_j)], \tag{2}$$

where α is the learning rate and $r_t^i(a_j)$ is the immediate reward.

Algorithm 1 A Multiagent Q-Learning Framework on Regular Graphs

Require: a structured population $\mathcal{G}(\mathcal{N}, \mathcal{E}, \mathcal{B})$, a symmetric normal form game G, the maximum time step T

while $t < T$ **do**
  for each agent $i \in \mathcal{N}$ **do**
    Agent i selects an action $a \in \mathcal{A}$ according to its policy
  end for
  for each agent $i \in \mathcal{N}$ **do**
    for each neighbor $j \in \mathcal{N}_i$ **do**
      Agents i and j play the game G using their selected actions respectively
    end for
    Agent i receives a payoff $r_t^i(a)$ that is averaged over the k games it plays
  end for
  for each agent $i \in \mathcal{N}$ **do**
    Agent i updates its Q-value: $Q_{t+1}^i(a) \leftarrow Q_t^i(a) + \alpha[r_t^i(a) - Q_t^i(a)]$
  end for
  $t \leftarrow t + 1$
end while

3 Modelling Q-Learning Dynamics on Regular Graphs

In this section, we derive the theoretical model of multiagent Q-learning on regular graphs and use a set of equations to describe the system evolution.

3.1 Dynamics of Q-values for Individual Agents

First, let us focus on an arbitrary agent i. The reward of agent i depends on its own action and the actions of its neighbors.

With a slight abuse of notation, we denote the use of action $a_i \in \mathcal{A}$ by $a_i = [0, \dots, 1, \dots, 0]$, a one-hot vector with the

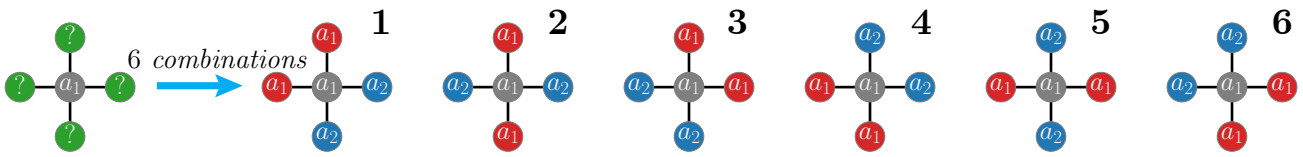

Figure 1: Illustration of agent local interactions with the degree $k = 4$. Each agent occupies an vertex on the graph and has two available actions a_1 and a_2. Suppose that an individual agent i (colored in grey) uses the first action (i.e. $a^i = a_1$) and its neighbor configuration is $\gamma = [k_{a_1}, k_{a_2}]^\top = [2, 2]^\top$. Then there are 6 combinations that lead to this neighbour configuration. Given that the mean policy of the population is $\bar{x}(a_1) = 0.4, \bar{x}(a_2) = 0.6$, the probability (or frequency) of this neighbor configuration is $f(\gamma) = C_k^{k_{a_1}} \bar{x}(a_1)^{k_{a_1}} C_{k-k_{a_1}}^{k_{a_2}} \bar{x}(a_2)^{k_{a_2}} = C_4^2 \times 0.4^2 \times C_2^2 \times 0.6^2 = 0.3456$.

i-th component equaling to 1 and others 0. Then the payoff of using action a_i against the opponent's using action a_j is

$$r(a_i|a_j) = a_i \mathcal{R} a_j^\top. \tag{3}$$

Define the action of agent i as a^i, and the joint action of its neighbors as $\mathbf{a}^{-i}$. Then in every round the agent i gets reward

$$r(a^i|\mathbf{a}^{-i}) = \frac{1}{k} \sum_{\forall j \in \mathcal{N}_i} r(a^i|a^j), \tag{4}$$

where $\mathcal{N}_i$ is the neighbor set of agent i. Let the number of neighbors using the action a_i be k_{a_i}. Then Equation (4) can be expressed as

$$\begin{aligned} r(a^i|\mathbf{a}^{-i}) &= \frac{1}{k} \sum_{\forall j \in \mathcal{N}_i} r(a^i|a^j) \\ &= \frac{1}{k}[k_{a_1} r(a^i|a_1) + \ldots + k_{a_m} r(a^i|a_m)] \\ &= \frac{1}{k}[k_{a_1} a^i \mathcal{R} a_1^\top + \ldots + k_{a_m} a^i \mathcal{R} a_1^\top] \\ &= \frac{1}{k} a^i \mathcal{R}(k_{a_1} a_1^\top + \ldots + k_{a_m} a_m^\top) \\ &= \frac{1}{k} a^i \mathcal{R}[k_{a_1}, \ldots, k_{a_m}]^\top, \end{aligned} \tag{5}$$

where $[k_{a_1}, \ldots, k_{a_m}]^\top = \sum_{\forall j \in \mathcal{N}_i} a^j$ describes the number of actions adopted by neighbors. We define $\gamma \triangleq [k_{a_1}, \ldots, k_{a_m}]^\top$ as the neighbor configuration.

Suppose that at time t, an agent i uses the j-th action and its neighbor configuration is $\gamma = [k_{a_1}, \ldots, k_{a_m}]^\top$. The change of its Q-values $\mathbf{Q}_t$ in the j-th dimension can be expressed as

$$\begin{aligned} v_j(\mathbf{Q}_t, \gamma) &= Q_{t+1}(a_j) - Q_t(a_j) \\ &= \alpha[\frac{1}{k} a^i \mathcal{R} \gamma - Q_t(a_j)], \end{aligned} \tag{6}$$

where the learning rate α and Boltzmann exploration temperature β are fixed for all agents. Moreover, for symmetric normal form games, the payoff is independent of the roles of individual agents. Therefore, at time step t, how fast agent i changes its Q-values should be attributed to its current Q-values $\mathbf{Q}_t$ and its neighbor configuration γ.

Next we consider the probability that the agent has a certain neighbor configuration γ. Using the mean-field theory, for an arbitrary agent in the population, its probability of using action a_j can be approximated as

$$p(a_j) \approx \bar{x}(a_j), \tag{7}$$

where

$$\bar{x}(a_j) = \frac{1}{n} \sum_{\forall i \in \mathcal{N}} x^i(a_j) = \frac{1}{n} \sum_{\forall i \in \mathcal{N}} \frac{e^{\beta Q^i(a_j)}}{\sum_{\forall a \in \mathcal{A}} e^{\beta Q^i(a)}} \tag{8}$$

is the average probability of using action a_j in the population.

For a given neighbour configuration $\gamma = [k_{a_1}, \ldots, k_{a_m}]^\top$, we calculate its probability as

$$\begin{aligned} f(\gamma) =& C_k^{k_{a_1}} p(a_1)^{k_{a_1}} C_{k-k_{a_1}}^{k_{a_2}} p(a_2)^{k_{a_2}} \ldots C_{k_{a_m}}^{k_{a_m}} p(a_m)^{k_{a_m}} \\ =& C_k^{k_{a_1}} C_{k-k_{a_1}}^{k_{a_2}} \ldots C_{k_{a_m}}^{k_{a_m}} \bar{x}(a_1)^{k_{a_1}} \ldots \bar{x}(a_m)^{k_{a_m}} \\ =& \frac{k!}{k_{a_1}!(k-k_{a_1})!} \frac{(k-k_{a_1})!}{k_{a_2}!(k-k_{a_1}-k_{a_2})!} \\ & \ldots \frac{(k-k_{a_1}-\ldots-k_{a_{m-1}})!}{k_{a_m}!0!} \prod_{i=1}^m \bar{x}(a_i)^{k_{a_i}} \\ =& \frac{k!}{\prod_{i=1}^m k_{a_i}!} \prod_{i=1}^m \bar{x}(a_i)^{k_{a_i}}. \end{aligned} \tag{9}$$

where C denotes a combination. We observe that the vector γ is subject to multinomial distribution.

3.2 Evolution of the Distribution of Q-values

To capture the evolution of the Q-values in a population on regular graphs, we first derive a dynamic equation for a certain configuration of neighbours by adopting the approach proposed by Hu et al. [2019], and then obtain the dynamic equation considering all possible configurations of neighbours.

Consider a m-dimension Q-value space $\mathbb{R}^m$, in which m is the number of available actions. Following the approach of Hu et al. [2019], we denote each agent in the system with a particle in this space such that the change of Q-values can be viewed as the transport of particle (or agent) mass in the space. Denote the distribution of agent mass in this space by the joint density function $p(\mathbf{Q}_t, t)$, i.e., $p(\mathbf{Q}_t, t)$ intuitively means the fraction of agents having the Q-values $\mathbf{Q}_t$ in the system. Then, for an arbitrary infinitesimal box in the space, whose volume equals to dV, the number of agents in this

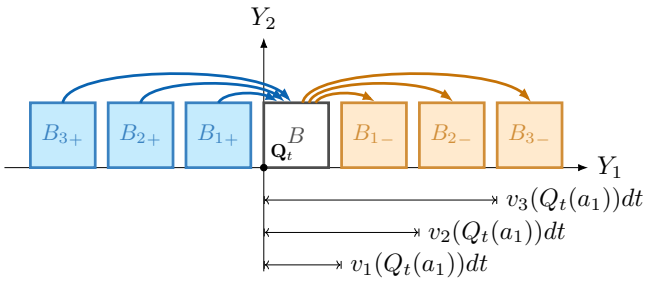

Figure 2: Each neighbor configuration $f(\gamma)$ produces a corresponding number of agents who leave and enter box B.

box at time t is $np(\mathbf{Q}_t, t)dV$, where n is the total number of agents.

The change of the number of agents in this box from time t to time $t + dt$ can be expressed by the difference between the number of agents entering the box and the number of agents leaving the box

$$np(\mathbf{Q}_t, t + dt)dV - np(\mathbf{Q}_t, t)dV. \tag{10}$$

Since agents adopting the action a_j are only allowed to update $Q_t(a_j)$, the change of the number of agents in this box equals to the sum of changes along k different directions. For a particular configuration of neighbours, Equation (10) can be rewritten as

$$\begin{aligned} &np(\mathbf{Q}_t, t + dt)dV - np(\mathbf{Q}_t, t)dV \\ &= -ndt \sum_{j=1}^{m} v_j(\mathbf{Q}_t, \gamma) \frac{\partial p(\mathbf{Q}_t, t) x_t(a_j, Q_t(a_j))}{\partial Q_t(a_j)} dV. \end{aligned} \tag{11}$$

A full derivation of the Equation (11) can be found in the Supplementary Material.

The right-hand side of Equation (11) is a function determined by the neighbor configuration γ, and we show the graphical demonstration for $k = 2$ and $m = 2$ in Figure 2. As the neighbor configuration is subject to multinomial distribution, the term $\frac{\partial p(\mathbf{Q}_t, t)}{\partial t}$ for total population can be written as the expectation of a function of γ as follows

$$\begin{aligned} \frac{\partial p\left(\mathbf{Q}_t, t\right)}{\partial t} = &\sum_{j=1}^{m} \sum_{\forall \gamma \in \Gamma} \Bigg[f(\gamma) v_j\left(\mathbf{Q}_t, \gamma\right) \\ &\times \frac{\partial p\left(\mathbf{Q}_t, t\right) x_t\left(a_j, \mathbf{Q}_t\right)}{\partial Q_t\left(a_j\right)} \Bigg]. \end{aligned} \tag{12}$$

where Γ is the set of all possible neighbour configurations γ given the average degree k and the number m of available actions. By this equation, the change of density function $p(\mathbf{Q}_t, t)$ is determined by the current density function $p(\mathbf{Q}_t, t)$, the velocity $v_j(\mathbf{Q}_t, \gamma)$ and the policy $x_t(a_j, Q_t(a_j))$. It is worth noting that the the policy of an agent depends only on its Q-values. Therefore, the mean policy $\bar{\mathbf{x}}_t$ follows

$$\bar{\mathbf{x}}_t = \int \dots \int \frac{e^{\beta Q_t(a_j)}}{\sum_{\forall a \in \mathcal{A}} e^{\beta Q_t(a)}} p(\mathbf{Q}_t, t) dQ_t(a_1) \dots dQ_t(a_m). \tag{13}$$

Then, the evolutionary dynamics of an infinite population in which agents interact with local neighbours can be modelled by the following system of equations

$$\begin{cases} f(\gamma) = \frac{k!}{\prod_{i=1}^{m} k_{a_i}!} \prod_{i=1}^{m} \bar{x}(a_i)^{k_{a_i}} \\ v_j(\mathbf{Q}_t, \gamma) = \alpha[\frac{1}{k} a^i \mathcal{R} \gamma - Q_t(a_j)] \\ \frac{\partial p\left(\mathbf{Q}_t, t\right)}{\partial t} = \sum_{j=1}^{m} \sum_{\forall \gamma \in \Gamma} \Bigg[f(\gamma) v_j\left(\mathbf{Q}_t, \gamma\right) \\ \qquad \times \frac{\partial p\left(\mathbf{Q}_t, t\right) x_t\left(a_j, \mathbf{Q}_t\right)}{\partial Q_t\left(a_j\right)} \Bigg] \\ \bar{\mathbf{x}}_t = \int \dots \int \frac{e^{\beta Q_t(a_j)}}{\sum_{\forall a \in \mathcal{A}} e^{\beta Q_t(a)}} p(\mathbf{Q}_t, t) dQ_t(a_1) \dots dQ_t(a_m). \end{cases} \tag{14}$$

3.3 Connection to the Previous Model

For a large number of neighbours (i.e. a large average degree k), the limit of multinomial distribution with k random variables can be approximated by $(k-1)$-dimension normal distribution,

$$\begin{aligned} &\sum_{\forall \gamma \in \Gamma} f(\gamma) g(a_j, r_t(a_j)) \\ &\approx \int_{-\infty}^{+\infty} \cdots \int_{-\infty}^{+\infty} \varphi(k_{a_j}) g(a_j, r_t(a_j)) dk_{a_1} \cdots dk_{a_{m-1}} \\ &= \mathbb{E}[g(a_j, a^i \mathcal{R} \gamma / k)] \\ &= \mathbb{E}[g(a_j, a^i \mathcal{R} \bar{\mathbf{x}}_t)] + \mathbb{E}[\sum_{h=1}^{\infty} \frac{\Delta^h}{h!} g^{(h)}(a_j, a^i \mathcal{R} \bar{\mathbf{x}}_t)] \\ &= \mathbb{E}[g(a_j, a^i \mathcal{R} \bar{\mathbf{x}}_t)] + \mathbb{E}[\sum_{l=1}^{\infty} \frac{\Delta^{2l}}{(2l)!} g^{(2l)}(a_j, a^i \mathcal{R} \bar{\mathbf{x}}_t)] \\ &= \mathbb{E}[g(a_j, a^i \mathcal{R} \bar{\mathbf{x}}_t)] + \sum_{l=1}^{\infty} [\frac{g^{(2l)}(a_j, a^i \mathcal{R} \bar{\mathbf{x}}_t)}{(2l)!} (2l-1)!! \\ &\qquad \times \frac{\sum_{j=1}^{m} [\bar{x}_t(a_j)(1 - \bar{x}_t(a_j))]^l}{k^l}] \\ &\approx g(a_j, \bar{r}_t(a_j)), \end{aligned} \tag{15}$$

where $\varphi(k_{a_j}) = \frac{1}{\sqrt{2\pi n x(a_j)(1-x(a_j))}} \exp(-\frac{(k_{a_j} - n x(a_j))^2}{2n x(a_j)(1-x(a_j))})$, $g(a_j, r_t(a_j)) = \frac{\partial p(\mathbf{Q}_t, t) x_t(a_j, Q_t(a_j))}{\partial Q_t(a_j)} v_j(\mathbf{Q}_t, f(\gamma))$, and $\Delta = \gamma/k - \bar{\mathbf{x}}_t$.

Thus, as the number of neighbours is sufficiently large, our model coincides with the model reported by Hu et al. [2019].

4 Experiments

4.1 Experimental Settings

Game Configurations. We consider three widely used symmetric normal form games including prisoner's dilemma (PD), hawk dove (HD) and common interest (CI) games to verify the effectiveness of our model with different settings.

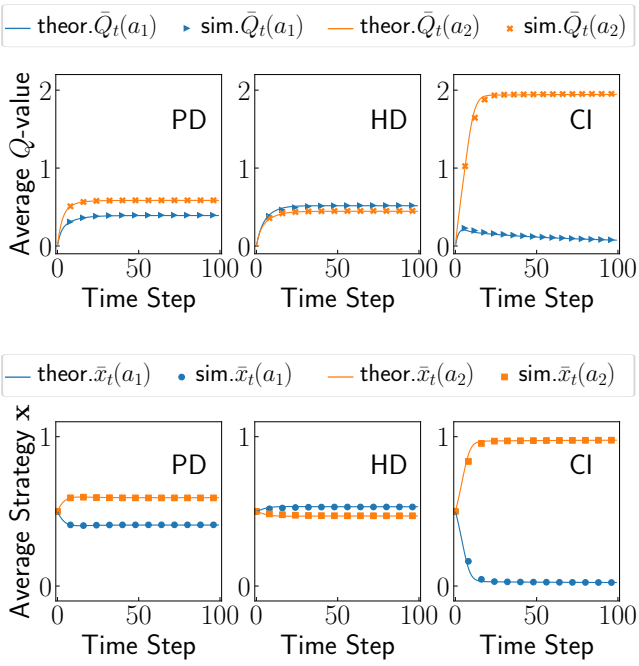

Figure 3: The time evolution of the mean Q-values and the mean policy in PD, HD and CI games.

The action set and payoff matrices for these three games are summarized as follows.

$$\mathcal{A}^{\mathrm{PD}} = \{\text{cooperate C}, \text{defect D}\},$$
$$\mathcal{R}^{\mathrm{PD}} = \begin{bmatrix} r_{\mathrm{CC}} & r_{\mathrm{CD}} \\ r_{\mathrm{DC}} & r_{\mathrm{DD}} \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 1.5 & 0 \end{bmatrix},$$
$$\mathcal{A}^{\mathrm{HD}} = \{\text{hawk H}, \text{dove D}\},$$
$$\mathcal{R}^{\mathrm{HD}} = \begin{bmatrix} r_{\mathrm{DD}} & r_{\mathrm{DH}} \\ r_{\mathrm{HD}} & r_{\mathrm{HH}} \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 2 & -1 \end{bmatrix},$$
$$\mathcal{A}^{\mathrm{CI}} = \{\text{interest X}, \text{interest Y}\},$$
$$\mathcal{R}^{\mathrm{CI}} = \begin{bmatrix} r_{\mathrm{XX}} & r_{\mathrm{XY}} \\ r_{\mathrm{YX}} & r_{\mathrm{YY}} \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 2 \end{bmatrix}.$$

In PD game, although mutual cooperation is Pareto dominant, mutual defection is the unique Nash equilibrium. In HD game, the Nash equilibrium is that one agent adopts the "hawk" policy and the other adopts the "dove" policy. In CI game, there are two Nash equilibria, i.e.,(X,X) and (Y,Y). Choosing the optimal joint action (Y, Y) will lead to the highest payoff, while choosing the joint action (X,X) will lead to the suboptimal payoff.

Regular Graphs. We consider two typical types of regular graphs with degree k:

- Translational Symmetry Lattice. A regular graph is translational symmetric if the graph is constructed by translational splicing of the smallest repeating unit (a node and its edges). We provide examples of this type of regular graphs in Figure 5 (a). The graph is a ring if $k = 2$, a honeycomb if $k = 3$, a square lattice if $k = 4$, and a triangular if $k = 6$.
- Random Regular Graph. A random regular graph is a random graph where all nodes have the same degree. Here, we consider random regular graphs that have no self-loops and allow at most one edge between any two nodes. As shown in Figure 5 (b), a random regular graph is in general not translational symmetric.

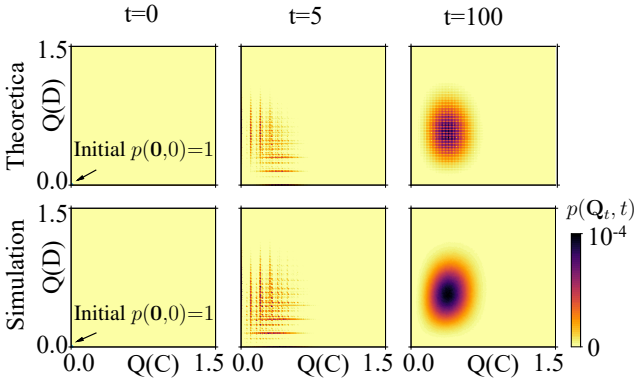

Figure 4: The snapshots of the Q-values distribution $p(\mathbf{Q}_t, t)$ at three different time steps ($t = 0, 5, 100$) in PD games.

Learning Parameters. The exploration temperature β is set to 2 and the learning rate α is set to 0.4. Unless otherwise specified, we set the initial Q-values to zero for all agents which means that agents take their actions randomly.

Agent-based Simulations. In the simulations, there are 100 agents, and agents interact and adapt their strategies according to Algorithm 1. To avoid the finite-size effects and randomness, the presented simulation results are averaged over 100 independent runs.

4.2 The Evolution of Agent Behaviors

We first focus the case that degree $k = 4$. Unless stated otherwise, we consider the translational symmetric lattice in the simulations for comparison. Figure 3 shows the time evolution of the mean Q-value and the mean policy of the population, as predicted by our model and observed in the simulations. It is clear that our model precisely predicts the time evolution across three different games.

Our model also reflects the differences in Q-learning in different games. In PD games, defection becomes the dominant policy which is consist with the prediction of classic game theory. However, some cooperation still exists. For HD games and CI games, it is shown that the population eventually converges to the evolutionary stable strategy.

In Figure 4, we present the Q-value density distribution for PD games. The first row is the theoretical result, and the second row is the simulation result. For ease of presentation, we focus on three time steps: $t = 0$, 5 and 100. It is shown that given the same initialization ($t = 0$), the theoretical predictions on the Q-value distributions are consistent with the counterparts observed in the simulations, no matter whether the system has stabilized ($t = 100$) or not ($t = 5$).

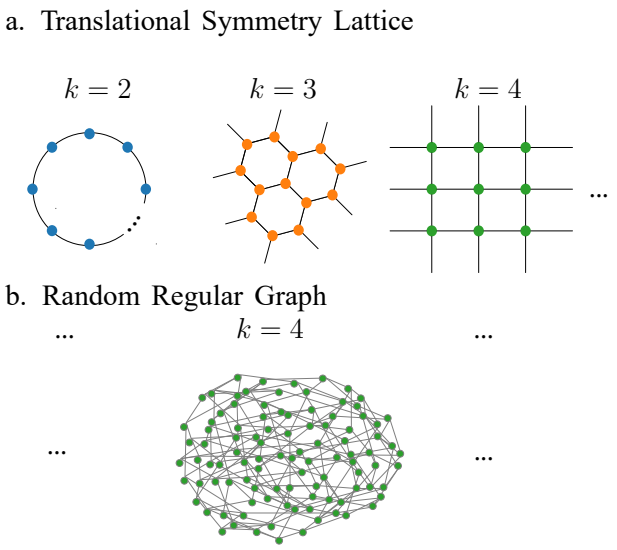

Figure 5: Examples of Translational Symmetry Lattice and Random Regular Graph.

To further verify the accuracy of our model, in the following, we contrast the theoretical predictions made by our model and the simulation results obtained from translational symmetry lattice and random regular graph. In Figure 6, we focus on PD games. However, different than the PD game shown in Section 4.1, we consider that there is a parameter b in the payoff matrix such that

$$\mathcal{R}^{\text{PD}} = \begin{bmatrix} r_{\text{CC}} & r_{\text{CD}} \\ r_{\text{DC}} & r_{\text{DD}} \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ b & 0 \end{bmatrix},$$

where the payoff obtained by unilateral defection is controlled by b representing the temptation to defect. We vary the value of b from 1.0 to 2.0. Obviously, if b is equal to 1, players will get the same payoff by choosing to cooperate or defect. As the value of b increases, players will get higher payoffs by choosing to defect. As shown in Figure 6, with the increase of b, Q-learning agents are significantly less likely to cooperate.

Moreover, we observe from Figure 6 that how likely Q-learning agents choose cooperate are also under the effect of the degree k. Given the same temptation to defect (denoted by b), agents are more likely to cooperate in regular graphs where agents have few neighbours (i.e. a small value of k). However, in general, this effect is marginal when the temptation to defect is low (i.e. a small value of b).

Note that when $k = 99$, our theoretical model approximates the model of Hu et al. [2019], as analyzed in Section 3.3. Therefore, Figure 6 also shows that the previous model fails to capture how regular graphs with different degrees affect Q-learning dynamics.

5 Conclusions

In the paper, we propose the first formal model for Q-learning dynamics on regular graphs. By capturing the influence of different neighbour configurations on the learning of agents,

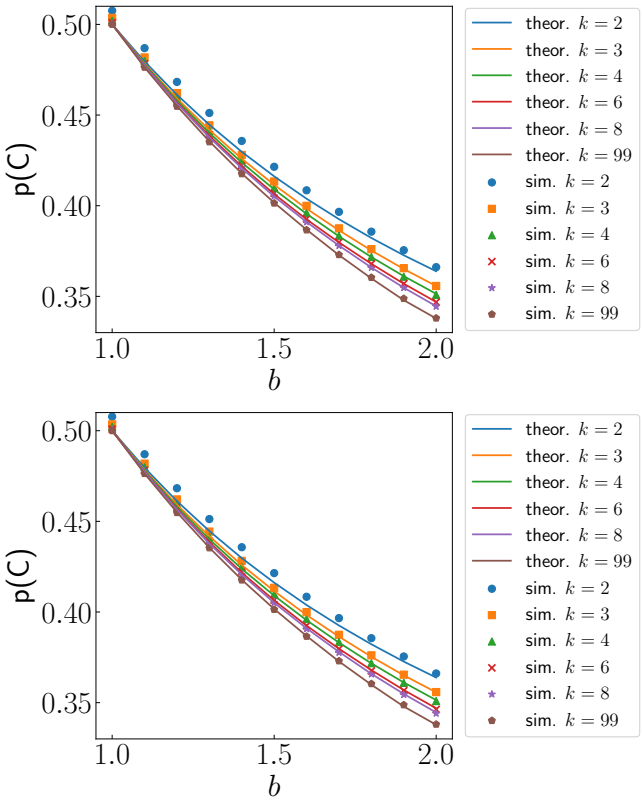

Figure 6: The time evolution of the average probability of choosing C in PD games, under the joint effects of the temptation b to defect and the degree k. The simulation results of the first row and the second row correspond to translational symmetric lattice and random regular graphs, respectively.

we derive a system of differential equations to describe the evolution of the Q-value distribution of the system. To verify the accuracy of our model, we contrast the theoretical predictions of our model with agent-based simulation results. Under three typical types of symmetric normal form games, we show that our model accurately describes the dynamics of the mean Q-values and the mean policy. Moreover, our model reveals that in PD games, as the temptation to defect and the degree of the regular graph increases, Q-learning agents are less likely to cooperate. However, in general, when the temptation to defect is low, the effect of the degree of the regular graph becomes insignificant.

Acknowledgments

This research was supported by the National Science Fund for Distinguished Young Scholarship of China (No. 62025602), the National Key RD Program of China (No. 2018AAA0100900), the National Natural Science Foundation of China (Nos. U1803263, 11931015, 62066045 and 62866039), Key Technology Research and Development Program of Science and Technology-Scientific and Technological Innovation Team of Shaanxi Province (Grant No. 2020TD-013), the Natural Science Foundation of Yunnan Province (No. 20190FB083) and the XPLORER PRIZE.

References

[Bailey and Piliouras, 2019] James P Bailey and Georgios Piliouras. Multi-agent learning in network zero-sum games is a hamiltonian system. In *Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems*, pages 233–241. International Foundation for Autonomous Agents and Multiagent Systems, 2019.

[Bloembergen *et al.*, 2015] Daan Bloembergen, Karl Tuyls, Daniel Hennes, and Michael Kaisers. Evolutionary dynamics of multi-agent learning: A survey. *Journal of Artificial Intelligence Research*, 53:659–697, 2015.

[Brown and Sandholm, 2019] Noam Brown and Tuomas Sandholm. Superhuman ai for multiplayer poker. *Science*, 365(6456):885–890, 2019.

[Delgado, 2002] Jordi Delgado. Emergence of social conventions in complex networks. *Artificial intelligence*, 141(1-2):171–185, 2002.

[Galstyan, 2013] Aram Galstyan. Continuous strategy replicator dynamics for multi-agent q-learning. *Autonomous agents and multi-agent systems*, 26(1):37–53, 2013.

[Hennes *et al.*, 2009] Daniel Hennes, Karl Tuyls, and Matthias Rauterberg. State-coupled replicator dynamics. In *Proceedings of The 8th International Conference on Autonomous Agents and Multiagent Systems-Volume 2*, pages 789–796, 2009.

[Hu and Leung, 2017] Shuyue Hu and Ho-fung Leung. Achieving coordination in multi-agent systems by stable local conventions under community networks. In *Ijcai*, pages 4731–4737, 2017.

[Hu *et al.*, 2019] Shuyue Hu, Chin-wing Leung, and Ho-fung Leung. Modelling the dynamics of multiagent q-learning in repeated symmetric games: a mean field theoretic approach. *Advances in Neural Information Processing Systems*, 32:12125–12135, 2019.

[Jin *et al.*, 2018] Junqi Jin, Chengru Song, Han Li, Kun Gai, Jun Wang, and Weinan Zhang. Real-time bidding with multi-agent reinforcement learning in display advertising. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, pages 2193–2201, 2018.

[Kaisers and Tuyls, 2010] Michael Kaisers and Karl Tuyls. Frequency adjusted multi-agent q-learning. In *Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems: volume 1-Volume 1*, pages 309–316, 2010.

[Klos *et al.*, 2010] Tomas Klos, Gerrit Jan Van Ahee, and Karl Tuyls. Evolutionary dynamics of regret minimization. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 82–96. Springer, 2010.

[Liu *et al.*, 2021a] Siqi Liu, Guy Lever, Zhe Wang, Josh Merel, SM Eslami, Daniel Hennes, Wojciech M Czarnecki, Yuval Tassa, Shayegan Omidshafiei, Abbas Abdolmaleki, et al. From motor control to team play in simulated humanoid football. *arXiv preprint arXiv:2105.12196*, 2021.

[Liu *et al.*, 2021b] Yiwei Liu, Jiamou Liu, Kaibin Wan, Zhan Qin, Zijian Zhang, Bakhadyr Khoussainov, and Liehuang Zhu. From local to global norm emergence: Dissolving self-reinforcing substructures with incremental social instruments. In *International Conference on Machine Learning*, pages 6871–6881. PMLR, 2021.

[Panait *et al.*, 2008] Liviu Panait, Karl Tuyls, and Sean Luke. Theoretical advantages of lenient learners: An evolutionary game theoretic perspective. *The Journal of Machine Learning Research*, 9:423–457, 2008.

[Sen and Sen, 2009] Onkur Sen and Sandip Sen. Effects of social network topology and options on norm emergence. In *International Workshop on Coordination, Organizations, Institutions, and Norms in Agent Systems*, pages 211–222. Springer, 2009.

[Tuyls and Parsons, 2007] Karl Tuyls and Simon Parsons. What evolutionary game theory tells us about multiagent learning. *Artificial Intelligence*, 171(7):406–416, 2007.

[Tuyls *et al.*, 2003] Karl Tuyls, Katja Verbeeck, and Tom Lenaerts. A selection-mutation model for q-learning in multi-agent systems. In *Proceedings of the second international joint conference on Autonomous agents and multiagent systems*, pages 693–700, 2003.

[Villatoro *et al.*, 2011] Daniel Villatoro, Jordi Sabater-Mir, and Sandip Sen. Social instruments for robust convention emergence. In *Twenty-Second International Joint Conference on Artificial Intelligence*, 2011.

[Wang *et al.*, 2018] Yixi Wang, Wenhuan Lu, Jianye Hao, Jianguo Wei, and Ho-fung Leung. Efficient convention emergence through decoupled reinforcement social learning with teacher-student mechanism. In *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems*, pages 795–803, 2018.

[Zhang *et al.*, 2018] Kaiqing Zhang, Zhuoran Yang, Han Liu, Tong Zhang, and Tamer Basar. Fully decentralized multi-agent reinforcement learning with networked agents. In *International Conference on Machine Learning*, pages 5872–5881. PMLR, 2018.

[Zhang *et al.*, 2021] Kaiqing Zhang, Zhuoran Yang, and Tamer Başar. Multi-agent reinforcement learning: A selective overview of theories and algorithms. *Handbook of Reinforcement Learning and Control*, pages 321–384, 2021.

[Zhou *et al.*, 2019] Ming Zhou, Jiarui Jin, Weinan Zhang, Zhiwei Qin, Yan Jiao, Chenxi Wang, Guobin Wu, Yong Yu, and Jieping Ye. Multi-agent reinforcement learning for order-dispatching via order-vehicle distribution matching. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, pages 2645–2653, 2019.