# Modelling the Dynamics of Regret Minimization in Large Agent Populations: a Master Equation Approach

**Zhen Wang**[1,2] , **Chunjiang Mu**[1,2] , **Shuyue Hu**[3] , **Chen Chu**[2,4,*] , **Xuelong Li**[2]
[1]School of Cybersecurity, Northwestern Polytechnical University
[2]School of Artificial Intelligence, OPtics and ElectroNics (iOPEN), Northwestern Polytechnical University
[3]Shanghai Artificial Intelligence Laboratory
[4]School of Statistics and Mathematics, Yunnan University of Finance and Economics
chuchenynufe@hotmail.com

## Abstract

Understanding the learning dynamics in multi-agent systems is an important and challenging task. Past research on multi-agent learning mostly focuses on two-agent settings. In this paper, we consider the scenario in which a population of infinitely many agents apply regret minimization in repeated symmetric games. We propose a new formal model based on the master equation approach in statistical physics to describe the evolutionary dynamics in the agent population. Our model takes the form of a partial differential equation, which describes how the probability distribution of regret evolves over time. Through experiments, we show that our theoretical results are consistent with the agent-based simulation results.

## 1 Introduction

A multi-agent system (MAS) is a distributed system where independent actors called agents interact in a common environment. During the past decades, MASs have been proved to be useful frameworks to resolve real-world problems with inherently distributed nature, such as traffic control [Yang *et al.*, 2020], online trading [Barbosa and Belo, 2010], multi-robots coordination [Ota, 2006], and video games [Vinyals *et al.*, 2019; Jaderberg *et al.*, 2019]. Due to the non-stationarity of MASs, designing proper agent strategies beforehand is generally hard and thus the capability of learning is crucial for individual agents.

The most commonly studied learning technique is reinforcement learning (RL). A RL agent aims to maximize its cumulative payoffs and learns its strategy through repeated interactions with the environment [Sutton and Barto, 2018]. Although RL under single-agent settings has been well understood, there lacks a solid theoretical grounding for RL under multi-agent settings due to the non-stationary nature of MASs [Bloembergen *et al.*, 2015]. Hence, studying the dynamics of multi-agent learning is important because it can bring benefits to algorithm selection under a certain scenario, parameter tuning and inspiring the design of new learning algorithms [Tuyls *et al.*, 2003].

In this paper, we consider the regret minimization (RM) algorithm and study the learning dynamics under $n$-agent settings with $n \to \infty$. RM is an important class of learning algorithms in the multi-agent learning literature. The central idea of RM is that during repeated interactions, an agent may look back at the history of payoffs and actions taken so far, and regret not having played another action (i.e. the best action in hindsight). RM has been proved to be effective for solving Nash equilibrium in normal-form games [Blum and Monsour, 2007]. Recently, combined with deep learning, it is shown that RM can also solve complex games with imperfect information settings [Brown *et al.*, 2019; Brown and Sandholm, 2019]. In view of the importance, Blum and Monsour [2007] called for studies on the dynamics of RM algorithms to better understand its strengths and weaknesses.

Klos et al. [2010] derived a formal model for the dynamics of Polynomial Weights RM algorithm under the 2-agent settings; based on the model, they revealed an interesting connection between RM in 2-player normal form games and evolutionary game theory (EGT). However, just like other previous works that leverage EGT to study multi-agent learning [Börgers and Sarin, 1997; Tuyls *et al.*, 2003; Kaisers and Tuyls, 2010; Kaisers *et al.*, 2012; Bloembergen *et al.*, 2015], the approach of Klos et al. [2010] could not scale with a large number of agents.

More recently, utilizing the methods of statistical physics, formal models for the learning dynamics under $n$-agent settings with $n \to \infty$ have been developed. Lahkar et al. [2013] considered the Cross learning in population games and derived the continuity equation model. Hu et al. [2019; 2020] derived a Fokker-Planck equation to model the Q-learning dynamics and a general master-equation-based framework for the dynamics of independent learning in population games.

Despite these advances, the dynamics of RM in MASs with infinitely many agents is still up in the air. In this paper, we focus on the *Regret-Matching* algorithm proposed by [Hart and Mas-Colell, 2000], and consider that each agent has its own unique regret values and updates its regret values during repeated plays of 2-player symmetric games with random op-

[*] Corresponding Author

ponents. We propose a theoretical model based on the master equation approach to describe the learning dynamics of RM in such systems. To be more specific, we consider the evolution of regret value distribution on the regret value space. Starting from the master equation of such stochastic process, we derive a partial differential equation, which captures the time evolution of the learning system, depending on the initial regret value's distribution and game settings. Through comparing various learning dynamics obtained by our theoretical model and simulations, we show that our theoretical model not only accurately predicts the steady-state results, but also precisely captures the evolution of the expectation of policy and regret values in the MAS.

To summarize, our key contributions are as follows:

- we formalize the regret dynamics of individual agents that apply RM in 2-player symmetric games;
- we develop the first theoretical model (a partial differential equation) that can accurately describe the regret dynamics of an entire population based on the master equation approach;
- we provide experimental validation of the accuracy of our model through extensive comparisons with agent-based simulation results.

## 2 Preliminaries

In this paper, we consider a multi-agent system, where all agents play the symmetric bimatrix games repeatedly with randomly selected opponents and adapt their strategies by the Regret-Matching algorithm [Hart and Mas-Colell, 2000].

### 2.1 Symmetric Bimatrix Games

A 2-agent normal form game is a triple $\langle \{1,2\}, \{A_1, A_2\}, u \rangle$, where $\{1,2\}$ is the set of two agents, $A_i$ is the set of actions (or pure strategies) available for agent $i \in \{1,2\}$ and $u_i : A_1 \times A_2 \to \mathbb{R}$ means the payoff function of agent $i$. $\mathbf{x}_i \in \Delta(A_i)$ is the policy (or a mixed strategy) vector of agent $i$, which is the probability distribution over $A_i$. Correspondingly, $\mathbf{x}_i^t(j)$ is the probability of agent $i$ choosing action $j$ at time $t$. In this paper, we consider a typical 2-agent normal form game, called the symmetric bimatrix game, where each agent has the same action set $A = \{1, 2, \cdots, k\}$ and the payoff matrix $\mathbf{M}$ of a row agent is given by

$$\mathbf{M} = \begin{array}{c} \\ 1 \\ 2 \\ \vdots \\ k \end{array} \begin{array}{c} \begin{array}{cccc} 1 & 2 & \cdots & k \end{array} \\ \begin{pmatrix} p_{11} & p_{12} & \cdots & p_{1k} \\ p_{21} & p_{22} & \cdots & p_{2k} \\ \vdots & \vdots & \ddots & \vdots \\ p_{k1} & p_{k2} & \cdots & p_{kk} \end{pmatrix} \end{array}$$

where $p_{rc}$ is the row agent's payoff when the row agent chooses action $r$ and faces an action $c$ taken by the column agent. The payoff matrix of the column agent is $\mathbf{M}^{\mathrm{T}}$. The payoff of a row agent with action $j$ against its opponent with action $j'$ is

$$u(j, j') = \mathbf{e}_j^{\mathrm{T}} \mathbf{M} \mathbf{e}_{j'}, \tag{1}$$

where $\mathbf{e}_j$ is the action vector with size $k$ (the $j$th element equals to 1 and the other $k-1$ elements equal to 0).

### 2.2 Regret-Matching

The Regret-Matching algorithm is a classic regret minimization algorithm proposed by Hart and Mas-Colell [2000]. Suppose that at time $t$, agent $i$ uses action $a_i^t$ and its opponent uses action $a_{-i}^t$, we define $\mathbf{r}_i^t(a) = u(a, a_{-i}^t) - u(a_i^t, a_{-i}^t)$ as the regret of agent $i$ using action $a_i^t$ at time $t$ but not using action $a$. At time $t+1$, for agent $i$, the cumulative regret value of action $a$ is

$$\mathbf{R}_i^{t+1}(a) = \frac{1}{t} \sum_{\tau=1}^{t} \mathbf{r}_i^{\tau}(a). \tag{2}$$

Agents tend to choose the action with a high cumulative regret value. For agent $i$, the probability of choosing action $a$ at time $t$ is given by

$$\mathbf{x}_i^t(a) = \frac{\exp(\lambda \mathbf{R}_i^t(a))}{\sum_{a' \in A} \exp\left(\lambda \mathbf{R}_i^t(a')\right)}, \tag{3}$$

where $\lambda$ is a temperature and represents the level of rationality or the degree of exploitation of agents. If $\lambda = 0$, agents will choose each action with an equal probability. If $\lambda \to \infty$, agents will choose the action with the highest cumulative regret value.

### 2.3 Multi-agent Learning Framework

Consider a system $N = \{1, 2, \cdots, n\}$ containing $n$ agents, where each agent plays the symmetric bimatrix game repeatedly. The interactions in this MAS proceeds as follows. First, each agent $i$ obtains an initial $\mathbf{R}$-value vector $\mathbf{R}_i^1$ over its action set $A$. At each time step $t$, the agent chooses an action with the probability of exponential form of its cumulative regret value, according to Equation (3). Then, the agent plays the symmetric bimatrix game with $o$ randomly chosen agents and receives an immediate payoff which will be averaged over $o$ games. We define $O$ as the set of $o$ chosen agents. At the end of each time step $t$, each agent updates the cumulative regret values for all actions, according to Equation (2). To generalize the Regret-Matching algorithm to our learning framework, we consider the regret of action $j$ as $\mathbf{r}_i^t(a) = u(a, \mathbf{a}_{i' \in O}^t) - u(a_i^t, \mathbf{a}_{i' \in O}^t)$ where

$$u(a, \mathbf{a}_{i' \in O}^t) = \frac{1}{o} \sum_{i' \in O} u(a, a_{i'}^t). \tag{4}$$

We summarize our learning framework in Algorithm 1. We will analyze the dynamics of this system in the following section.

## 3 The Analysis of Regret Minimization Dynamics

In this section, we derive the continuous-time differential equations for describing the dynamics of policy and cumulative regret values in our MASs. Specifically, we introduce an analysis method based on the master equation. We first derive the dynamics of cumulative regret for individual agents, and then analyze the dynamics of the system on the regret value space, where the number of agents is infinite.

**Algorithm 1:** Multi-Agent Learning Model

```
Input: a symmetric bimatrix game $\langle \{1, 2\}, \{A\}, u \rangle$,
       a multi-agent system $M$, an initial regret
       vector $\mathbf{R}_i^1$ over $A_i$ for each $i \in M$, the
       maximum number of iteration time $\mathcal{T}$
for each $i \in M$ do
    initialize agent $i$'s policy $\mathbf{x}_i^1$ according to $\mathbf{R}_i^1$;
$t \leftarrow 1$;
while $t < \mathcal{T}$ do
    for each $i \in M$ do
        choose an action from $A$ according to $\mathbf{x}_i^t$;
    for each $i \in M$ do
        agent $i$ chooses $o$ opponents from $M$ and plays
          the matrix game with them;
    for each $i \in M$ do
        for each $a \in A$ do
            agent $i$ reviews games, and obtains $\mathbf{r}_i^t(a)$;
            obtains $\mathbf{R}_i^{t+1}(a)$ by $\mathbf{r}_i^t(a)$ and $\mathbf{R}_i^t(a)$;
    for each $i \in M$ do
        update $\mathbf{x}_i^{t+1}$ according to $\mathbf{R}_i^{t+1}$;
    $t \leftarrow t + 1$;
```

### 3.1 Dynamics of an Agent's Regrets

We start from the case of individual agent $i$, whose policy is $\mathbf{x}_i^t$ at time $t$. First, agent $i$ chooses an action $j$ with the probability $\mathbf{x}_i^t(j)$, and obtains the payoff $u(j, \mathbf{a}_{-i}^t)$. Then it reviews the games, and calculates the regrets for all actions:

$$\mathbf{r}_i^t(a) = u(a, \mathbf{a}_{-i}^t) - u(j, \mathbf{a}_{-i}^t), \forall a \in A.$$

According to Equation (2), if we consider the difference equation of the cumulative regret, we have

$$\begin{aligned}\mathbf{R}_i^{t+1}(a) - \mathbf{R}_i^t(a) &= \frac{1}{t}\sum_{\tau=1}^{t}\mathbf{r}_i^\tau(a) - \frac{1}{t-1}\sum_{\tau=1}^{t-1}\mathbf{r}_i^\tau(a) \\ &= \frac{1}{t}(\mathbf{r}_i^t(a) - \mathbf{R}_i^t(a)), \forall a \in A.\end{aligned} \tag{5}$$

Next, following the method of [Tuyls *et al.*, 2003], we derive a continuous-time differential equation for the update of the cumulative regret. Given the time interval between two adjacent updates of the cumulative regret being $\Delta t$, and let $t = n\Delta t$ (which means until time $t + \Delta t$, there will be $n$ updating processes of cumulative regrets),

$$\begin{aligned}\Delta\mathbf{R}_i^t(a) &= \mathbf{R}_i^{t+\Delta t}(a) - \mathbf{R}_i^t(a) \\ &= \frac{1}{n}\int_0^{n\Delta t}\mathbf{r}_i^\tau(a)\mathrm{d}\tau - \frac{1}{n-1}\int_0^{(n-1)\Delta t}\mathbf{r}_i^\tau(a)\,\mathrm{d}\tau \\ &= \frac{\Delta t}{t}\left(\mathbf{r}_i^t(a) - \mathbf{R}_i^t(a)\right), \forall a \in A.\end{aligned} \tag{6}$$

Naturally, we can define the velocity of cumulative regret's change as $\frac{\Delta R}{t}$. Due to the randomness of $\Delta R$, we consider its expectation. The velocity of cumulative regret's change for any action $a$ at time $t$, when agent $i$ chooses action $j$, can be described as a conditional expectation

$$\begin{aligned}\mathbf{V}_j(\mathbf{R}_i^t(a), t) &= \lim_{\Delta t \to 0}\mathbb{E}\left[\frac{\Delta\mathbf{R}_i^t(a)}{\Delta t} \mid a_i^t = j\right] \\ &= \mathbb{E}\left[\frac{1}{t}\left(\mathbf{r}_i^t(a) - \mathbf{R}_i^t(a)\right) \mid a_i^t = j\right] \\ &= \frac{1}{t}\left(\mathbb{E}\left[u(a, \mathbf{a}_{i'\in O}^t) - u(j, \mathbf{a}_{i'\in O}^t)\right] - \mathbf{R}_i^t(a)\right).\end{aligned} \tag{7}$$

Let us consider the expected payoff in above equation. The expected payoff of agent $i$ obtained from one agent $i'$ is equal to the payoff when agent $i$'s action facing with agent $i'$'s policy

$$\mathbb{E}_{\mathbf{a}_{i'}^t \sim \mathbf{x}_{i'}^t}\left[u(a, \mathbf{a}_{i'}^t)\right] = u(a, \mathbf{x}_{i'}^t).$$

Note that one agent interacts with $o$ other randomly chosen agents. The $o$ agents' average policy is a unbiased estimator of the system's expected (or mean) policy; as $o$ increases to infinity, the variance of that estimator will decrease to zero [Hu *et al.*, 2019]. Hence, we let the payoff expectation of agent $i$ facing with policy expectation of all agents in $N$ approximate her payoff expectation

$$\begin{aligned}\mathbb{E}_{\mathbf{a}_{i'\in O}^t \sim \mathbf{x}_{i'\in O}^t}\left[u(a, \mathbf{a}_{i'\in O}^t)\right] &= \mathbb{E}_{\mathbf{x}_{i'\in O}^t \sim \mathbf{x}^t}\left[u(a, \mathbf{x}_{i'\in O}^t)\right] \\ &\approx u(a, \mathbb{E}[\mathbf{x}^t])\end{aligned}$$

where $\mathbb{E}[\mathbf{x}^t]$ is the system's policy expectation vector of all agents such that its element $\mathbb{E}[\mathbf{x}^t(j)]$ is the expected probability of taking action $j$ in the system. Therefore, the Equation (7) can be rewritten as

$$\mathbf{V}_j(\mathbf{R}_i^t(a), t) = \frac{1}{t}\left(u(a, \mathbb{E}[\mathbf{x}^t]) - u(j, \mathbb{E}[\mathbf{x}^t]) - \mathbf{R}_i^t(a)\right). \tag{8}$$

By Equation (8), we can obtain the velocity of agent $i$'s cumulative regret's change for all actions if agent $i$ chooses action $j$ at time $t$, which is composed of the velocity vector $\mathbf{V}_j(\mathbf{R}_i^t, t) = [\mathbf{V}_j(\mathbf{R}_i^t(1), t), \mathbf{V}_j(\mathbf{R}_i^t(2), t), ..., \mathbf{V}_j(\mathbf{R}_i^t(k), t)]$. For simplicity, we call cumulative regret value as regret value in the rest of this paper. Theoretically, if we can write out the velocity vector for each agent, the dynamics of the regret values is straightforward. However, this becomes impossible when the number of agents in that system tends to infinity. In the next subsection, we introduce a method based on the master equation, which allows us to obtain the dynamics of the system based on the regret value distribution in the regret value space.

### 3.2 Dynamics on the Regret Space

Now we consider the regret value space, a $k$-dimensional euclidean space, where $k$ is the size of action set. A specific $\mathbf{R}$ value vector $\mathbf{R} = [\mathbf{R}(1), \mathbf{R}(2), ..., \mathbf{R}(k)]$ is a coordinate or a position in this space. We denote by $p^t(\mathbf{R})$ the probability distribution of the regret values in the system at time $t$ such that $p^t(\mathbf{R})$ can be regarded as how many agents in the system with their regret value equaling to $\mathbf{R}$. Next, we focus on the evolution of this distribution in this regret value space.

Note that all agents in the system have the same payoff function and apply the same learning parameters. As we have

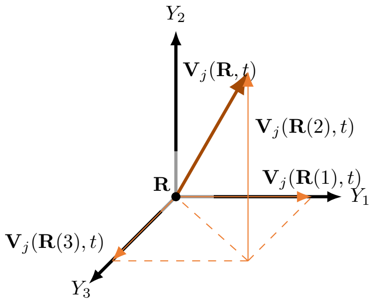
$Y_2$
$\mathbf{V}_j(\mathbf{R},t)$
$\mathbf{V}_j(\mathbf{R}(2),t)$
$\mathbf{R}$
$\mathbf{V}_j(\mathbf{R}(1),t)$
$Y_1$
$\mathbf{V}_j(\mathbf{R}(3),t)$
$Y_3$

Figure 1: An illustration of an agent moving away from $\mathbf{R}$ in 3-dimensional space.

used the policy expectation to ignore the randomness of the agent's opponent, this fact enables us to remove the agent index directly and rewrite Equation (8) as the function of any coordinate $\mathbf{R}$ in the regret value space and time $t$ as follows:

$$\mathbf{V}_j(\mathbf{R}(a),t) = \frac{1}{t}\left(u(a,\mathbb{E}[\mathbf{x}^t]) - u(j,\mathbb{E}[\mathbf{x}^t]) - \mathbf{R}(a)\right). \quad (9)$$

In the regret value space, $\mathbf{V}_j(\mathbf{R}(a),t)$ is the change of regret's velocity component in direction $a$, due to agent in position $\mathbf{R}$ selecting action $j$. And $\mathbf{V}_j(\mathbf{R},t)$ is the vector sum of the velocities in all directions, as shown in Figure 1, which is a 3-dimensional situation. Since we have defined the probability distribution $p^t(\mathbf{R})$, any element $\mathbb{E}\mathbf{x}^t(a)$ of the vector $\mathbb{E}\mathbf{x}^t$ in Equation (9) can be calculated by formula of expectation of random variable $\mathbf{R}$'s function:

$$\mathbb{E}[\mathbf{x}^t(a)] = \int \ldots \int \frac{\exp(\lambda\mathbf{R}\,(a))}{\sum_{a'\in A}\exp\left(\lambda\mathbf{R}(a')\right)}\, p^t(\mathbf{R})\mathrm{d}\mathbf{R}(1)\ldots\mathrm{d}\mathbf{R}(k). \quad (10)$$

According to Equation (9) and (10), the key challenge is to capture the evolution dynamics of probability distribution $p^t(\mathbf{R})$. To address this challenge, we introduce our method based on the master equation, which is one of the main contributions of our paper.

From a spatial perspective, the evolution of $p^t(\mathbf{R})$ can be described as the transport of agents among different positions in regret value space. If we focus on a given $\mathbf{R}$, we can find that the change of its probability density from $t$ to $t+\Delta t$ is the process of existing agents leaving the position $\mathbf{R}$, to other positions and incoming agents entering $\mathbf{R}$ from other positions. We denote the set of positions, including the target positions towards which the agents in $\mathbf{R}$ moving and the source positions from which other agents entering $\mathbf{R}$ by $\{\mathbf{R}'\}$. In other words, $\{\mathbf{R}'\}$ is all the positions where agents may exchange with position $\mathbf{R}$. Here we set $\mathbf{R} \notin \{\mathbf{R}'\}$. Consider that the time interval $\Delta t$ is small enough so that the agents can not move once more, we can represent the change of $p^t(\mathbf{R})$ by the master equation

$$\begin{aligned}\frac{\partial p^t(\mathbf{R})}{\partial t} &= \lim_{\Delta t\to 0}(p^{t+\Delta t}(\mathbf{R}) - p^t(\mathbf{R})) \\ &= \int \left(W(\mathbf{R},\mathbf{R}',t)p^t(\mathbf{R}') - W(\mathbf{R}',\mathbf{R},t)p^t(\mathbf{R})\right)\mathrm{d}\mathbf{R}',\end{aligned} \quad (11)$$

where $W(\mathbf{R},\mathbf{R}',t)$ is the transition rate from position $\mathbf{R}'$ to $\mathbf{R}$ at time $t$, and likewise for $W\left(\mathbf{R}',\mathbf{R},t\right)$. The condition of using this equation is that the change of probability $p^t(\mathbf{R})$ is a Markov process, that is, $p^{t+\Delta t}(\mathbf{R})$ only depends on $p^t(\mathbf{R})$. In our setting, this condition is satisfied. The physical meaning of the master equation is that all the change velocity of probability at $\mathbf{R}$ is the increase velocity of probability minus decrease velocity of probability. Next, we further expand and deduce the master equation.

We consider the first term of Equation (11), $\int W(\mathbf{R},\mathbf{R}',t)p^t(\mathbf{R}')\mathrm{d}\mathbf{R}'$, which is the increase velocity of probability from all possible position $\{\mathbf{R}'\}$. Therefore, we rewrite $\mathbf{R}'$ in the first term as $\mathbf{R}^{from}$, which makes the term become $\int W(\mathbf{R},\mathbf{R}^{from},t)p^t(\mathbf{R}^{from})\mathrm{d}\mathbf{R}^{from}$. According to the definition, $W(\mathbf{R},\mathbf{R}^{from},t)$ is defined as follows [Mandel and Wolf, 1995]:

$$\begin{aligned}W(\mathbf{R},\mathbf{R}^{from},t) = \lim_{\Delta t\to 0}\frac{1}{\Delta t}&(\Pr(\mathbf{R},t+\Delta t \mid \mathbf{R}^{from},t) \\ &- \Pr(\mathbf{R},t \mid \mathbf{R}^{from},t)).\end{aligned} \quad (12)$$

where $\Pr(\mathbf{R},t+\Delta t \mid \mathbf{R}^{from},t)$ is the conditional probability of the agents in $\mathbf{R}^{from}$ at time $t$ moving to $\mathbf{R}$ in $\Delta t$. Here $\Pr(\mathbf{R},t \mid \mathbf{R}^{from},t) = 0$ always holds according to our definition of $\mathbf{R}^{from}$.

Note that the agent in one position can move to different positions because agents can choose different actions randomly. Therefore, to further distinguish the different positions $\{\mathbf{R}^{from}\}$, where agents may choose different actions to arrive to $\mathbf{R}$, we define the set $\left\{\mathbf{R}^{from}\right\} = \left\{\mathbf{R}_1^{from}\right\} \cup \left\{\mathbf{R}_2^{from}\right\} \cup ... \cup \left\{\mathbf{R}_k^{from}\right\}$. $\mathbf{R}_j^{from}$ means if an agent chooses action $j$ to reach $\mathbf{R}$ in $\Delta t$, her position at time $t$ must be in set $\left\{\mathbf{R}_j^{from}\right\}$. Now we can further rewrite the first term of the master equation:

$$\int W(\mathbf{R},\mathbf{R}',t)p^t(\mathbf{R}')\mathrm{d}\mathbf{R}' = \sum_{j\in A}\int W(\mathbf{R},\mathbf{R}_j^{from},t)p^t(\mathbf{R}_j^{from})\mathrm{d}\mathbf{R}_j^{from}. \quad (13)$$

Let us consider the size of set $\{\mathbf{R}_a^{from}\}$. According to the definition of element $\mathbf{R}_j^{from}$ and the velocity $\mathbf{V}_j(\mathbf{R},t)$, we have $\mathbf{R}_j^{from}(a) + \mathbf{V}_j(\mathbf{R}_j^{from}(a),t)\Delta t = \mathbf{R}_j^{from}(a) + \frac{1}{t}\left(u(a,\mathbb{E}[\mathbf{x}^t]) - u(j,\mathbb{E}[\mathbf{x}^t]) - \mathbf{R}_j^{from}(a)\right)\Delta t = \mathbf{R}(a)$. For each $a \in A$, there is only one $\mathbf{R}_j^{from}(a)$ to satisfy the requirements of the above equation, which means $\left|\left\{\mathbf{R}_j^{from}\right\}\right| = 1$. Therefore, we can remove the integral sign

$$\begin{aligned}&\int W(\mathbf{R},\mathbf{R}',t)p^t(\mathbf{R}')\mathrm{d}\mathbf{R}' \\ &= \sum_{j\in A} W(\mathbf{R},\mathbf{R}_j^{from},t)p^t(\mathbf{R}_j^{from}) \\ &= \sum_{j\in A}\lim_{\Delta t\to 0}\frac{1}{\Delta t}(\Pr(\mathbf{R},t+\Delta t \mid \mathbf{R}_j^{from},t)p^t(\mathbf{R}_j^{from})\end{aligned} \quad (14)$$

Next, we consider the conditional probability $\Pr(\mathbf{R}, t + \Delta t \mid \mathbf{R}_j^{from}, t)$. An agent in position $\mathbf{R}_j^{from}$ at time $t$ is able to reach $\mathbf{R}$ at $t + \Delta t$ if and only if it chooses action $j$. Therefore, we have $\Pr(\mathbf{R}, t+\Delta t \mid \mathbf{R}_j^{from}, t) = \Pr_j(\mathbf{R}_j^{from})$, where $\Pr_j(\mathbf{R}_j^{from}) = \frac{\exp(\lambda \mathbf{R}_j^{from}))}{\sum_{j' \in A} \exp(\lambda \mathbf{R}_{j'}^{from})}$ is the probability of the agent at position $\mathbf{R}_j^{from}$ choosing action $j$ at time $t$. Note that $\mathbf{V}_j(\mathbf{R}_j^{from}(a), t) \approx \mathbf{V}_j(\mathbf{R}(a), t)$ for each $a \in A$ as $\Delta \to 0$, so we have

$$\mathbf{R}_j^{from} = \mathbf{R} - \mathbf{V}_j(\mathbf{R}, t)\Delta t. \tag{15}$$

Finally, after our derivation, the first term of the right hand side of the master equation can be rewritten as

$$\begin{aligned}
&\int W(\mathbf{R}, \mathbf{R}', t) p^t(\mathbf{R}') \mathrm{d}\mathbf{R}' \\
&= \sum_{j \in A} \lim_{\Delta t \to 0} \frac{1}{\Delta t} (\Pr_j(\mathbf{R}_j^{from}) p^t(\mathbf{R}_j^{from}) \\
&= \lim_{\Delta t \to 0} \frac{1}{\Delta t} \sum_{j \in A} (\Pr_j(\mathbf{R} - \mathbf{V}_j(\mathbf{R}, t)\Delta t) p^t(\mathbf{R} - \mathbf{V}_j(\mathbf{R}, t)\Delta t).
\end{aligned} \tag{16}$$

Using almost the same derivation, we can rewrite the second term of the right hand side of the master equation $\int W(\mathbf{R}', \mathbf{R}, t) p^t(\mathbf{R}') \mathrm{d}\mathbf{R}'$ as

$$\int W(\mathbf{R}', \mathbf{R}, t) p^t(\mathbf{R}') \mathrm{d}\mathbf{R}' = \lim_{\Delta t \to 0} \frac{1}{\Delta t} \sum_{j \in A} \Pr_j(\mathbf{R}) p^t(\mathbf{R}). \tag{17}$$

The above equation is simpler than Equation (16) because we do not need to know the exact target position of the agent at $\mathbf{R}$. What we only need to know is the probability of leaving from each direction, due to choosing different actions.

Based on Equations (16), (17) and Taylor expansion at $\mathbf{R}$, we obtain a Partial Differential equation, which describes the velocity of change of probability density $p^t(\mathbf{R})$ at time $t$

$$\begin{aligned}
\frac{\partial p^t(\mathbf{R})}{\partial t} &= \lim_{\Delta t \to 0} \frac{1}{\Delta t} \sum_{j \in A} (\Pr_j(\mathbf{R} - \mathbf{V}_j(\mathbf{R})\Delta t, t) \\
&\quad p^t(\mathbf{R} - \mathbf{V}_j(\mathbf{R})\Delta t, t) - \Pr_j(\mathbf{R}) p^t(\mathbf{R})) \\
&= -\sum_{j \in A} \frac{\partial \Pr_j(\mathbf{R}) p^t(\mathbf{R})^{\mathrm{T}}}{\partial \mathbf{R}} \mathbf{V}_j(\mathbf{R}, t).
\end{aligned} \tag{18}$$

By Equations (9), (10) and (18), we obtain a system of differential equations, which can completely describe the dynamics of the system

$$\begin{cases}
\dfrac{\partial p^t(\mathbf{R})}{\partial t} = -\displaystyle\sum_{j \in A} \frac{\partial \Pr_j(\mathbf{R}) p^t(\mathbf{R})^{\mathrm{T}}}{\partial \mathbf{R}} \mathbf{V}_j(\mathbf{R}, t), \\
\Pr_j(\mathbf{R}) = \dfrac{\exp(\lambda \mathbf{R}(j))}{\sum_{j' \in A} \exp(\lambda \mathbf{R}(j'))} \\
\mathbf{V}_j(\mathbf{R}(a), t) = \dfrac{1}{t} \left( u(a, \mathbb{E}\mathbf{x}^t) - u(j, \mathbb{E}\mathbf{x}^t) - \mathbf{R}(a) \right). \\
\mathbb{E}[\mathbf{x}^t(a)] = \displaystyle\int \cdots \int \frac{\exp(\lambda \mathbf{R}(a))}{\sum_{a' \in A} \exp(\lambda \mathbf{R}(a'))} p^t(\mathbf{R}) \mathrm{d}\mathbf{R}(1) \ldots \mathrm{d}\mathbf{R}(k)..
\end{cases} \tag{19}$$

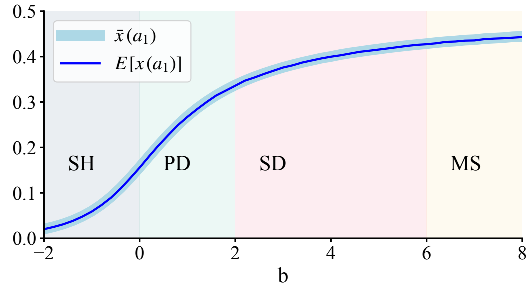

Figure 2: The fraction of agents that cooperate, when agents' policies become stable. We use $E[x]$ and $\bar{x}$ to denote our theoretical predictions and the simulation results, respectively. In the simulations, there are $1,000$ agents. The number of opponents per time step is $o = 50$. The temperature $\lambda$ in Equation (3) is 1.

In fact, the above Equation (18) can be further expanded according to the law of vector derivation and multiplication:

$$\frac{\partial p^t(\mathbf{R})}{\partial t} = -\sum_{j \in A} \sum_{a \in A} \frac{\mathrm{d}\left(\Pr_j(\mathbf{R})\ p^t(\mathbf{R})\right)}{\mathrm{d}\mathbf{R}(a)} \mathbf{V}_j(\mathbf{R}(a), t), \tag{20}$$

where the term $d_{ja} = \frac{\mathrm{d}\left(\Pr_j(\mathbf{R})\ p^t(\mathbf{R})\right)}{\mathrm{d}\mathbf{R}(a)} \mathbf{V}_j(\mathbf{R}(a), t)$ is the change of $p^t(\mathbf{R})$ in direction $a$ caused by choosing action $j$. Note the size of action set is $k$. We introduce a change square matrix $\mathbf{D}$ consisting of $k \times k$ terms:

$$\mathbf{D} = \{d_{ja}\}_{j, a \in A}.$$

Here the sum of $a$th column of this matrix $\sum_{j'=1}^{k} d_{j'a}$ is the change of action $a$'s regret.

## 4 Experiments

In this section, we will verify the effectiveness of our dynamics model by different experiments. First, we introduce our method of calculating Equation (19). Then, we show our experimental settings including the game models and parameter settings. Under these settings, we compare the expected policies predicted by our theoretical model and the mean policies obtained from agent-based simulations.

All the results shown below are divided into two types: theoretical analysis results and simulation results. The theoretical analysis results are obtained by Equation (19); we solve the equation by the finite difference method. The simulations are carried out according to Algorithm 1. Simulation results are averaged over up to 100 independent runs in order to ensure accuracy.

We consider the following payoff matrix with one parameter $b$:

$$\mathbf{M} = \begin{array}{cc} & \begin{array}{cc} C & D \end{array} \\ \begin{array}{c} C \\ D \end{array} & \begin{pmatrix} 6 & b \\ 6+b & 2 \end{pmatrix} \end{array}.$$

Different ranges of $b$ correspond to different game types: $b \in [0, 2]$ for Prisoner's Dilemma (PD) games; $b \in [2, 6]$ for Snowdrift (SD) games; $b \in (-\infty, 0)$ for Stag Hunt (SH)

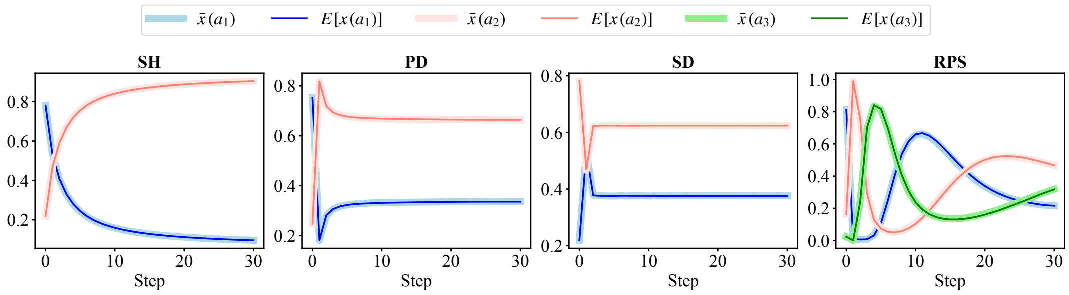

Figure 3: Evolution of the expected policy $E[x]$ predicted by our theoretical model and the mean policy $\bar{x}$ observed in agent-based simulations. In the simulations, there are $1,000$ agents. The number of opponents per time step is $o = 50$. The temperature $\lambda$ in Equation (3) is 1 for SH, PD, SD games and is 5 for RPS games.

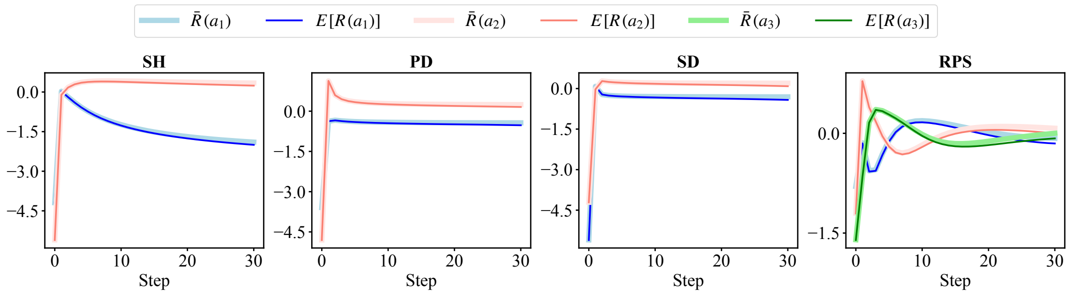

Figure 4: Evolution of the expected cumulative regret value $E[R]$ predicted by our theoretical model and the mean $\bar{R}$ cumulative regret value observed in agent-based simulations. The experimental settings are the same as in Figure 3.

games; and $b \in (6, \infty)$ for the mixed stable (MS) games. Without loss of generality, we let the initial regret values of the first action and the second action follow Beta distributions Beta (20, 80, $r_{\min}$ - $r_{\max}$, $r_{\max}$ - $r_{\min}$) and Beta (10, 90, $r_{\min}$ - $r_{\max}$, $r_{\max}$ - $r_{\min}$), where $r_{\min}$ and $r_{\max}$ are the minimum and maximum payoff of games, respectively. Figure 2 shows the fraction of agents that play cooperation, when agents' policies become stable. We can see that agents are more likely to cooperate as $b$ increases, and our theoretical predictions are consistent with simulation results.

Next, we show that our model is able to precisely capture the evolution of policy and cumulative regret values in the MAS. In Figure 3, we show how the expected policy of the system evolves in SH ($b = -1$), PD ($b = 2$), SD ($b = 3$) game and Rock-Paper-Scissors (RPS) games whose payoff bimatrix is as follows:

$$\mathbf{M}_{RPS} = \begin{array}{c c} & \begin{array}{ccc} R & P & S \end{array} \\ \begin{array}{c} R \\ P \\ S \end{array} & \begin{pmatrix} 0,0 & 1,-1 & -1,1 \\ -1,1 & 0,0 & 1,-1 \\ 1,-1 & -1,1 & 0,0 \end{pmatrix} \end{array} .$$

In SH,PD,and SD games, due to the temperature constant $\lambda$, the mean policy of this system does not exactly converge to the Nash equilibria. In RPS game, interestingly, our model successfully captures the oscillation phenomenon of evolution. Figure 4 shows the evolution of the cumulative regret values of the theoretic prediction and simulation results. It is clear that our model well captures the evolution of policy and cumulative regret values across different kinds of games.

## 5 Conclusion

In this paper, we model the dynamics of regret minimization in MASs with infinitely many agents, where the regret values of each agent are heterogeneous. In detail, we consider the evolution of the cumulative regret values in the regret value space. Started from the master equation, we derive a partial differential equation that describes the evolution of the distribution of cumulative regret values in the population. By carrying out experiments on typical types of symmetric bimatrix games with different parameter settings, we verify that the evolutionary dynamics of multi-agent learning system can be well predicted by our theoretical model. However, our method works for the MAS where agents are allowed to apply the same learning algorithm to play symmetric bimatrix games. In future works, we can extend our model to asymmetric games and stochastic games.

## Acknowledgments

This research was supported by the National Science Fund for Distinguished Young Scholarship of China (No. 62025602), the National Key R&D Program of China (No. 2018AAA0100900), the National Natural Science Foundation of China ( Nos. U1803263, 11931015 and 62066045), 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

[Barbosa and Belo, 2010] Rui Pedro Barbosa and Orlando Belo. Multi-agent forex trading system. In *Agent and multi-agent technology for internet and enterprise systems*, pages 91–118. Springer, 2010.

[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.

[Blum and Monsour, 2007] Avrim Blum and Yishay Monsour. Learning, regret minimization, and equilibria. *Algorithmic Game Theory*, pages 79–100, 2007.

[Börgers and Sarin, 1997] Tilman Börgers and Rajiv Sarin. Learning through reinforcement and replicator dynamics. *Journal of economic theory*, 77(1):1–14, 1997.

[Brown and Sandholm, 2019] Noam Brown and Tuomas Sandholm. Solving imperfect-information games via discounted regret minimization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 1829–1836, 2019.

[Brown *et al.*, 2019] Noam Brown, Adam Lerer, Sam Gross, and Tuomas Sandholm. Deep counterfactual regret minimization. In *International conference on machine learning*, pages 793–802. PMLR, 2019.

[Hart and Mas-Colell, 2000] Sergiu Hart and Andreu Mas-Colell. A simple adaptive procedure leading to correlated equilibrium. *Econometrica*, 68(5):1127–1150, 2000.

[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.

[Hu *et al.*, 2020] Shuyue Hu, Chin-Wing Leung, Ho-fung Leung, and Harold Soh. The evolutionary dynamics of independent learning agents in population games. *arXiv preprint arXiv:2006.16068*, 2020.

[Jaderberg *et al.*, 2019] Max Jaderberg, Wojciech M Czarnecki, Iain Dunning, Luke Marris, Guy Lever, Antonio Garcia Castaneda, Charles Beattie, Neil C Rabinowitz, Ari S Morcos, Avraham Ruderman, et al. Human-level performance in 3d multiplayer games with population-based reinforcement learning. *Science*, 364(6443):859–865, 2019.

[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.

[Kaisers *et al.*, 2012] Michael Kaisers, Daan Bloembergen, and Karl Tuyls. A common gradient in multi-agent reinforcement learning. In *AAMAS*, pages 1393–1394, 2012.

[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.

[Lahkar and Seymour, 2013] Ratul Lahkar and Robert M Seymour. Reinforcement learning in population games. *Games and Economic Behavior*, 80:10–38, 2013.

[Mandel and Wolf, 1995] Leonard Mandel and Emil Wolf. *Optical coherence and quantum optics*. Cambridge university press, 1995.

[Ota, 2006] Jun Ota. Multi-agent robot systems as distributed autonomous systems. *Advanced engineering informatics*, 20(1):59–70, 2006.

[Sutton and Barto, 2018] Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 2018.

[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.

[Vinyals *et al.*, 2019] Oriol Vinyals, Igor Babuschkin, Wojciech M Czarnecki, Michaël Mathieu, Andrew Dudzik, Junyoung Chung, David H Choi, Richard Powell, Timo Ewalds, Petko Georgiev, et al. Grandmaster level in starcraft ii using multi-agent reinforcement learning. *Nature*, 575(7782):350–354, 2019.

[Yang *et al.*, 2020] Jiachen Yang, Jipeng Zhang, and Huihui Wang. Urban traffic control in software defined internet of things via a multi-agent deep reinforcement learning approach. *IEEE Transactions on Intelligent Transportation Systems*, 22(6):3742–3754, 2020.