Learning in Multi-Memory Games Triggers Complex Dynamics Diverging from Nash Equilibrium*†

Yuma Fujimoto[1,2,3], **Kaito Ariu**[3,4] and **Kenshi Abe**[3]
[1]Research Center for Integrative Evolutionary Science, SOKENDAI.
[2]Universal Biology Institute (UBI), the University of Tokyo.
[3]AI Lab, CyberAgent, Inc.
[4]KTH Royal Institute of Technology.
fujimoto_yuma@soken.ac.jp, kaito_ariu@cyberagent.co.jp, abe_kenshi@cyberagent.co.jp

Abstract

Repeated games consider a situation where multiple agents are motivated by their independent rewards throughout learning. In general, the dynamics of their learning become complex. Especially when their rewards compete with each other like zero-sum games, the dynamics often do not converge to their optimum, i.e., the Nash equilibrium. To tackle such complexity, many studies have understood various learning algorithms as dynamical systems and discovered qualitative insights among the algorithms. However, such studies have yet to handle multi-memory games (where agents can memorize actions they played in the past and choose their actions based on their memories), even though memorization plays a pivotal role in artificial intelligence and interpersonal relationship. This study extends two major learning algorithms in games, i.e., replicator dynamics and gradient ascent, into multi-memory games. Then, we prove their dynamics are identical. Furthermore, theoretically and experimentally, we clarify that the learning dynamics diverge from the Nash equilibrium in multi-memory zero-sum games and reach heteroclinic cycles (sojourn longer around the boundary of the strategy space), providing a fundamental advance in learning in games.

1 Introduction

Repeated games consider that multiple agents aim to optimize their objective functions based on a normal-form game [Fudenberg and Tirole, 1991]. It is known that in this game, the set of optimal strategies for all the agents always exists as Nash equilibria [Nash Jr, 1950]. Various algorithms with which each agent achieves its optimal strategy have been proposed, such as Cross learning [Cross, 1973], replicator dynamics [Börgers and Sarin, 1997; Hofbauer *et al.*, 1998], gradient ascent [Singh *et al.*, 2000; Zinkevich, 2003; Bowling and Veloso, 2002; Bowling, 2004], Q-learning [Watkins and Dayan, 1992; Kaisers and Tuyls, 2010; Abdallah and Kaisers, 2013], and so on. In zero-sum games where two agents have conflicts in their benefits, however, the above learning algorithms cannot converge to their equilibrium [Mertikopoulos and Sandholm, 2016; Mertikopoulos *et al.*, 2018]. Indeed, the dynamics of learning draw a loop around the equilibrium point, even though the shape of the trajectory differs more or less depending on the algorithm. Thus, solving the dynamics around the Nash equilibrium is a touchstone for discussing whether the learning works well.

Currently, several studies attempt to understand trajectories of multi-agent learning by integrating various cross-disciplinary algorithms [Tuyls and Nowé, 2005; Tuyls *et al.*, 2006; Bloembergen *et al.*, 2015; Barfuss, 2020b]. For example, if we take an infinitesimal step size of learning, Cross learning draws the same trajectory as a replicator dynamics. The replicator dynamics can be interpreted as the weighted version of infinitesimal gradient ascent. Furthermore, Q-learning differs only in the extra term of exploration with the replicator dynamics. Another study has shown a relationship between the replicator dynamics and Q-learning by introducing a generalized regularizer which pulls the strategy back to the probabilistic simplex at the shortest distance [Mertikopoulos and Sandholm, 2016]. Like these studies, it is important to understand the trajectory of multi-agent learning theoretically.

Repeated games potentially include memories of agents, i.e., a possibility that agents determine their actions depending on past actions they chose (see Fig. 1 for the illustration). Such memories can expand the choice of strategies and thus lead to the agents handling their gameplay better; for example, by reading how the other player chooses its action [Fujimoto and Kaneko, 2019b]. Indeed, agents with memories can use tit-for-tat [Axelrod and Hamilton, 1981] and win-stay-lose-shift [Nowak and Sigmund, 1993] strategies in prisoner's dilemma games, and these strategies achieve cooperation as a Nash equilibrium, explaining human behaviors. Furthermore, in the field of artificial intelligence, repeated games of agents with memory have long been of interest [Sandholm and Crites, 1996]. Learning in memorizing past actions has also been studied in extensive-form games [Zinkevich *et al.*, 2007; Lanctot *et al.*, 2012]. In economics, how a region of the Nash equilibrium is extended by multi-memory strategies is enthu-

*The full version is at https://arxiv.org/abs/2302.01073

†The codes that we used are available at https://github.com/CyberAgentAILab/with-memory_games

siastically studied as folk theorem [Fudenberg and Maskin, 2009]. In practice, Q-learning is frequently implemented in multi-memory games [Barfuss *et al.*, 2019; Barfuss, 2020a; Meylahn *et al.*, 2022]. Several studies [Fujimoto and Kaneko, 2019a; Fujimoto and Kaneko, 2021] partly discuss the relation between the replicator dynamics and the gradient ascent but consider only prisoner's dilemma games. In conclusion, this relation is still unclear in games with general numbers of memories and actions. Furthermore, the convergence of dynamics in such multi-memory games has been unexplored.

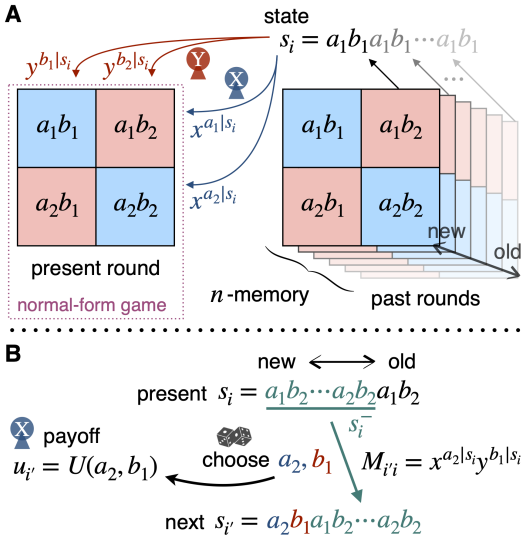

Figure 1: **A**. Illustration of a multi-memory repeated game. Focusing on the area surrounded by the purple dots, a normal-form game is illustrated. Player X (resp. Y) chooses its action a_1 or a_2 in the row (resp. b_1 or b_2 in the column). Then, each of them receives its payoff depending on their actions. The panel shows the matching-pennies game, where blue (resp. red) panels show that X (resp. Y) gains a payoff of 1 and Y (resp. X) loses it. Looking at the whole, each player memorizes their actions of the past n rounds. This memorized state is described as s_i given by $2n$-length bits of actions. **B**. Illustration for the detailed single round of repeated games, where present state s_i transitions to next state $s_{i'}$. In this transition, the oldest 2 bits are lost, and the other bits s_i^-, colored in green, are maintained. X's and Y's choices (a_2 (blue) and b_1 (red) in this figure) are appended as the newest 2 bits in $s_{i'}$. This transition occurs with the probability of $M_{i'i}$. Finally, X gains a payoff of $u_{i'}$ in the state transition.

This study provides a basic analysis of the multi-memory repeated game. First, we extend the two learning algorithms, i.e., replicator dynamics and gradient ascent, for multi-memory games. Then, we name them multi-memory replicator dynamics (MMRD) and gradient ascent (MMGA). As well as shown in the zero-memory games, the equivalence between MMRD and MMGA is proved in Theorems 1-3. Next, we tackle the convergence problem of such algorithms from both viewpoints of theory and experiment. Theorem 4 shows that under one-memory two-action zero-sum games, the Nash equilibrium is unique and essentially the same as that of zero-memory games. This theorem is non-trivial if taking into account the fact that diversification of strategies can expand the region of Nash equilibria in general games. Then, while utilizing these theorems, we see how multi-memory learning complicates the dynamics, leading to divergence from the Nash equilibrium with sensitivity to its initial condition like chaos.

2 Preliminary

2.1 Two-Player Normal-Form Game

Let us define two-player (of X and Y) $m(\in \mathbb{N})$-action games (see illustration of Fig. 1-A). Player X and Y choose their actions from $\mathcal{A} = \{a_1, \cdots, a_m\}$ and $\mathcal{B} = \{b_1, \cdots, b_m\}$ in a single round. After they finish choosing their actions $a \in \mathcal{A}$ and $b \in \mathcal{B}$, each of them gains a payoff $U(a,b) \in \mathbb{R}$ and $V(a,b) \in \mathbb{R}$, respectively.

2.2 Two-Player Multi-Memory Repeated Game

We further consider two-player $n(\in \mathbb{N})$-memory repeated games as an iteration of the two-player normal-form game (see illustration Fig. 1-A). The players are assumed to memorize their actions in the last n rounds. Since each player can take m actions, there are m^{2n} cases for possible memorized states, described as $\mathcal{S} = \prod_{k=1}^{n}(\mathcal{A} \times \mathcal{B})$. Under any memorized state, player X can choose any action stochastically. Such a stochastic choice of an action is described by a parameter $x^{a|s}$, which means the probability of choosing an action $a \in \mathcal{A}$ under memorized state $s \in \mathcal{S}$. Thus, X's strategy is represented by $|\mathcal{S}|(= m^{2n})$-numbers of $(m-1)$-dimension simplexes, $\mathbf{x} \in \prod_{s\in\mathcal{S}} \Delta^{m-1}$, while Y's is $\mathbf{y} \in \prod_{s\in\mathcal{S}} \Delta^{m-1}$.

2.3 Formulation as Markov Games

In order to handle this multi-memory repeated game as a Markov game [Shapley, 1953; Littman, 1994], we define a vector notation of memorized states;

$$\boldsymbol{s} = (\underbrace{a_1b_1\cdots a_1b_1}_{\times n}, \underbrace{a_1b_1\cdots a_1b_1}_{\times(n-1)} a_1b_2, \cdots, \underbrace{a_mb_m\cdots a_mb_m}_{\times n}),$$

which orders all the elements of $\mathcal{S}$ as a vector. We also define a vector notation of utility function as

$$\boldsymbol{u} = (\underbrace{U(a_1,b_1),\cdots,U(a_1,b_1)}_{\times m^{2n-2}}, \underbrace{U(a_1,b_2),\cdots,U(a_1,b_2)}_{\times m^{2n-2}}, \cdots, \underbrace{U(a_m,b_m),\cdots,U(a_m,b_m)}_{\times m^{2n-2}}),$$

which orders all the last-round payoffs for $\mathcal{S}$ as a vector. The utility function for Y, i.e., $\boldsymbol{v}$, is defined similarly. In addition, we denote an index for these vectors as $i \in \{1, \ldots, m^{2n}\}$. u_i is defined by the utility using the first 2 bits of actions in state s_i. For example, if $s_i = a_1b_2a_2b_1$, then $u_i = U(a_1, b_2)$.

Let $\boldsymbol{p} \in \Delta^{|\mathcal{S}|-1}$ be a probability distribution on $\boldsymbol{s}$ in a round. As the name Markov matrix implies, a distribution in the next round $\boldsymbol{p}'$ is given by $\boldsymbol{p}' = \boldsymbol{M}\boldsymbol{p}$, where $\boldsymbol{M}$ is a Markov transition matrix;

$$M_{i'i} = \begin{cases} x^{a|s_i}y^{b|s_i} & (s_{i'} = abs_i^-) \\ 0 & (\text{otherwise}) \end{cases}, \qquad (1)$$

which shows the transition probability from i-th state to i'-th one for $i, i' \in \{1, \dots, m^{2n}\}$. Here, note that s_i^- shows the state s_i except for the oldest two actions. See Fig. 1-B illustrating an example of Markov transition.

2.4 Nash Equilibrium

We now analyze the Nash equilibrium in multi-memory repeated games based on the formulation of Markov games. Let us assume that every agent uses a fixed strategy $\mathbf{x}$ and $\mathbf{y}$ or learns slowly enough for the timescale of the Markov transitions. We further assume that the strategies are located within the interiors of simplexes. Under this assumption, the Markov matrix becomes ergodic, and the stationary distribution is unique, denoted as $\boldsymbol{p}^{\mathrm{st}}(\mathbf{x}, \mathbf{y})$. This assumption is reasonable because all the actions should be learned in the replicator dynamics, and actions that are not played cannot be learned. This stationary distribution satisfies $\boldsymbol{p}^{\mathrm{st}} = \boldsymbol{M}\boldsymbol{p}^{\mathrm{st}}$. We also denote each player's expected payoff in the stationary distribution as $u^{\mathrm{st}}(\mathbf{x}, \mathbf{y}) = \boldsymbol{p}^{\mathrm{st}} \cdot \boldsymbol{u}$ and $v^{\mathrm{st}}(\mathbf{x}, \mathbf{y}) = \boldsymbol{p}^{\mathrm{st}} \cdot \boldsymbol{v}$. The goal of learning in the multi-memory game is to search for the Nash equilibrium, denoted by $(\mathbf{x}^*, \mathbf{y}^*)$, where their payoffs are maximized as

$$\begin{cases} \mathbf{x}^* \in \mathrm{argmax}_{\mathbf{x}} u^{\mathrm{st}}(\mathbf{x}, \mathbf{y}^*) \\ \mathbf{y}^* \in \mathrm{argmax}_{\mathbf{y}} v^{\mathrm{st}}(\mathbf{x}^*, \mathbf{y}) \end{cases}. \tag{2}$$

Here, u^{st} and v^{st} are complex non-linear functions for high-dimensional variables of $(\mathbf{x}, \mathbf{y})$. This Nash equilibrium is difficult to find in general.

3 Algorithm

In the following, we define multi-memory versions of two major learning algorithms, i.e., replicator dynamics and gradient ascent. Although we consider the learning of player X, that of player Y can be formulated in the same manner.

Definition 1 (expected future payoff)**.** *We define the expected future payoff from the distribution $\boldsymbol{p}$ as*

$$\pi(\boldsymbol{p}, \mathbf{x}, \mathbf{y}) := \sum_{t=0}^{\infty} \boldsymbol{M}^t (\boldsymbol{p} - \boldsymbol{p}^{\mathrm{st}}) \cdot \boldsymbol{u}, \tag{3}$$

which is the total payoff player X obtains from the present round to the future.

In this definition, the stationary payoff $\boldsymbol{p}^{\mathrm{st}} \cdot \boldsymbol{u} = u^{\mathrm{st}}$ is the offset term every round, and thus $\pi(\boldsymbol{p}^{\mathrm{st}}, \mathbf{x}, \mathbf{y}) = 0$.

Definition 2 (normalization)**.** *We define the normalization function* $\mathrm{Norm} : \prod_{s \in \mathcal{S}} \mathbb{R}_+^m \mapsto \prod_{s \in \mathcal{S}} \mathrm{int}(\Delta^{m-1})$ *as*

$$\mathrm{Norm}(\mathbf{x}) = \left\{ \frac{x^{a|s}}{\sum_{a'} x^{a'|s}} \right\}_{a,s}, \tag{4}$$

In this definition, $\mathrm{Norm}(\mathbf{x})$ satisfies the condition of probability variables for all s.

Based on these definitions, we formulate discretized MMRD and MMGA as Algorithm 1 and 2.

Algorithm 1 Discretized MMRD

Input: η

1: **for** $t = 0, 1, 2, \cdots$ **do**
2: X chooses a with probability $x^{a|s_i}$
3: (Y chooses b with probability $y^{b|s_i}$)
4: $s_{i'} \leftarrow abs_i^-$
5: $x^{a|s_i} \leftarrow x^{a|s_i} + \eta\pi(\boldsymbol{e}_{i'}, \mathbf{x}, \mathbf{y})$
6: $\mathbf{x} \leftarrow \mathrm{Norm}(\mathbf{x})$
7: $s_i \leftarrow s_{i'}$
8: **end for**

Algorithm 1 (Discretized MMRD) takes its learning rate η as an input. In each time step, the players choose their actions following their strategies (lines 2 and 3), while the state is updated by their chosen actions (lines 4 and 7). In line 5, each player reinforces its strategy by how much payoff it receives in the future from state $s_{i'}$. Here, note that $\boldsymbol{e}_{i'}$ indicates the unit vector for the i'-th element, describing that state $s_{i'}$ occurs.

Algorithm 2 Discretized MMGA

Input: η, γ

1: **for** $t = 0, 1, 2, \cdots$ **do**
2: **for** $a \in \mathcal{A}$, $s \in \mathcal{S}$ **do**
3: $\mathbf{x}' \leftarrow \mathrm{Norm}(\mathbf{x} + \gamma \mathbf{e}^{a|s})$
4: $\Delta^{a|s} \leftarrow \dfrac{u^{\mathrm{st}}(\mathbf{x}', \mathbf{y}) - u^{\mathrm{st}}(\mathbf{x}, \mathbf{y})}{\gamma}$
5: **end for**
6: **for** $a \in \mathcal{A}$, $s \in \mathcal{S}$ **do**
7: $x^{a|s} \leftarrow x^{a|s}(1 + \eta\Delta^{a|s})$
8: **end for**
9: $\mathbf{x} \leftarrow \mathrm{Norm}(\mathbf{x})$
10: **end for**

Algorithm 2 (Discretized MMGA) takes not only its learning rate η but a small value γ in measuring an approximate gradient as inputs. In each time step, each player measures the gradients of its payoff for each variable of its strategy (lines 2-5). Here, $\mathbf{e}^{a|s}$ is an abused notation of unit vector for the element of action a for state s. Then, the player updates its strategy by the gradients (lines 6-9). Here, note that the strategy update is weighted by the probability $x^{a|s}$ (line 7) in order to correspond to Algorithm 1. Here, lines 3-4 can be parallelized for all a and s, and line 7 as well.

4 Theoretical Analysis

4.1 Continuous-Time Equivalence of Algorithms

The following theorems provide a unified understanding of different algorithms. Theorem 1 and 2 are concerned with continualization of the two discrete algorithms. Surprisingly, Theorem 3 proves the correspondence between these different continualized algorithms by Theorem 1 and 2.

Theorem 1 (Coutinualized MMRD)**.** *Let $\boldsymbol{p}^{a|s}$ be the ex-*

pected distribution when X chooses a under state s;

$$p_{i'}^{a|s} := \begin{cases} y^{b|s} & (s_{i'} = abs^{-}) \\ 0 & (\text{otherwise}) \end{cases}. \quad (5)$$

In the limit of $\eta \to 0$, Algorithm 1 is continualized as dynamics

$$\dot{x}^{a|s_i}(\mathbf{x},\mathbf{y}) = p_i^{\mathrm{st}} x^{a|s_i} \left(\pi(\boldsymbol{p}^{a|s_i},\mathbf{x},\mathbf{y}) - \bar{\pi}^{s_i}(\mathbf{x},\mathbf{y}) \right), \quad (6)$$

$$\bar{\pi}^{s_i}(\mathbf{x},\mathbf{y}) = \sum_a x^{a|s_i} \pi(\boldsymbol{p}^{a|s_i},\mathbf{x},\mathbf{y}), \quad (7)$$

for all $a \in \mathcal{A}$ and $s_i \in \mathcal{S}$. Here, $\bar{\pi}^{s_i}$ is the expected payoff under state s_i.

Theorem 2 (Continualized MMGA)**.** *In the limit of $\gamma \to 0$ and $\eta \to 0$, Algorithm 2 is continualized as dynamics*

$$\dot{x}^{a|s}(\mathbf{x},\mathbf{y}) = x^{a|s} \frac{\partial}{\partial x^{a|s}} u^{\mathrm{st}}(\mathrm{Norm}(\mathbf{x}),\mathbf{y}), \quad (8)$$

for all $a \in \mathcal{A}$ and $s \in \mathcal{S}$.

See Technical Appendix A.1 and A.2 for the proof of Theorems 1 and 2.

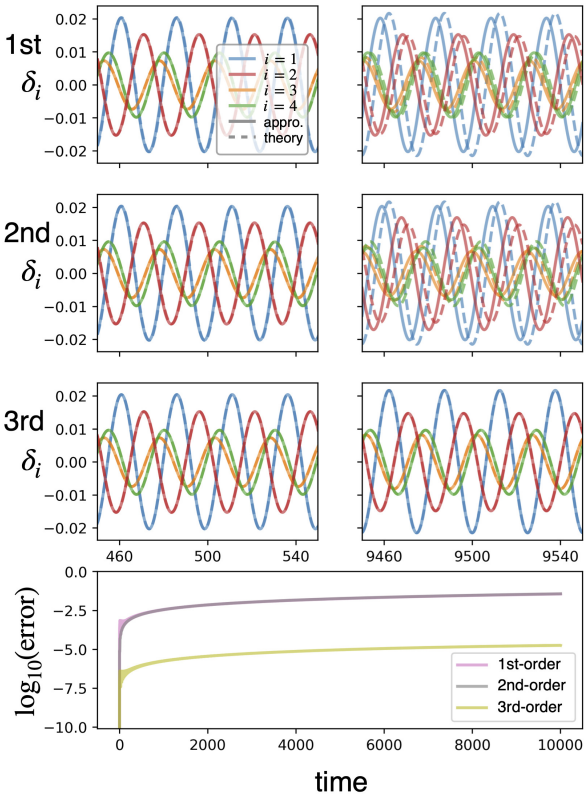

Figure 2: Multi-memory learning dynamics near the Nash equilibrium in the matching-pennies game. In the upper six panels, colored lines indicate the time series of δ_i (X's strategy). The solid (resp. broken) lines are approximated (resp. experimental) trajectories of learning dynamics. From the top, the trajectories are predicted by approximations up to the first, second, and third orders. The bottom panel shows the errors between the approximated and experimental trajectories.

Theorem 3 (Equivalence between the algorithms)**.** *The dynamics Eqs.* (7) *and* (8) *are equivalent.*

Proof Sketch. Let $\mathbf{x}'$ be the strategy given by $x^{a|s} \leftarrow x^{a|s} + \gamma$ in $\mathbf{x}$ for $a \in \mathcal{A}$ and $s \in \mathcal{S}$. Then, we consider the changes of the Markov transition matrix $\mathrm{d}\boldsymbol{M} := \boldsymbol{M}(\mathrm{Norm}(\mathbf{x}'),\mathbf{y}) - \boldsymbol{M}(\mathbf{x},\mathbf{y})$ and the stationary distribution $\mathrm{d}\boldsymbol{p}^{\mathrm{st}} := \boldsymbol{p}^{\mathrm{st}}(\mathrm{Norm}(\mathbf{x}'),\mathbf{y}) - \boldsymbol{p}^{\mathrm{st}}(\mathbf{x},\mathbf{y})$. By considering this changes in the stationary condition $\boldsymbol{p}^{\mathrm{st}} = \boldsymbol{M}\boldsymbol{p}^{\mathrm{st}}$, we get $\mathrm{d}\boldsymbol{p}^{\mathrm{st}} = (\boldsymbol{E} - \boldsymbol{M})^{-1}\mathrm{d}\boldsymbol{M}\boldsymbol{p}^{\mathrm{st}}$ in $O(\gamma)$. The right-hand (resp. left-hand) side of this equation corresponds to the continualized MMRD (resp. MMGA). □

For games with a general number of actions, the study [Zinkevich, 2003] has proposed a gradient ascent algorithm in relation to replicator dynamics. In light of this study, Theorem 3 extends the relation to the multi-memory games. This extension is neither simple nor trivial. The relation between replicator dynamics and gradient ascent has been proved by directly calculating $u^{\mathrm{st}} = \boldsymbol{p}^{\mathrm{st}} \cdot \boldsymbol{u}$ [Bloembergen *et al.*, 2015]. In multi-memory games, however, $u^{\mathrm{st}} = \boldsymbol{p}^{\mathrm{st}} \cdot \boldsymbol{u}$ is too hard to calculate. Thus, as seen in the proof sketch, we proved the relation by considering a slight change in the stationary condition $\boldsymbol{p}^{\mathrm{st}} = \boldsymbol{M}\boldsymbol{p}^{\mathrm{st}}$, technically avoiding such a hard direct calculation.

4.2 Learning Dynamics Near Nash Equilibrium

Below, let us discuss the learning dynamics in multi-memory games, especially divergence from the Nash equilibrium in zero-sum payoff matrices. In order to obtain a phenomenological insight into the learning dynamics simply, we assume one-memory two-action zero-sum games in Assumption 1.

Assumption 1 (One-memory two-action zero-sum game)**.** *We assume a two-action (i.e., $\mathcal{A} = \{a_1, a_2\}$ and $\mathcal{B} = \{b_1, b_2\}$), one-memory (i.e., $\boldsymbol{s} = (a_1b_1, a_1b_2, a_2b_1, a_2b_2)$), and zero-sum game (i.e., $\boldsymbol{v} = -\boldsymbol{u}$). In particular, we discuss zero-sum games where both u_1 and u_4 are smaller or larger than both u_2 and u_3.*

Under Assumption 1, we exclude uninteresting zero-sum payoff matrices that the Nash equilibrium exists as a set of pure strategies because the learning dynamics trivially converge to such pure strategies. The condition that both u_1 and u_4 are smaller or larger than both u_2 and u_3 is necessary and sufficient for the existence of no dominant pure strategy.

In the rest of this paper, we use a vector notation for strategies of X and Y; $\boldsymbol{x} := \{x_i\}_{i=1,\dots,4}$ and $\boldsymbol{y} := \{y_i\}_{i=1,\dots,4}$ as $x_i := x^{a_1|s_i}$ and $y_i := y^{b_1|s_i}$. Indeed, $x^{a_2|s_i} = 1 - x_i$ and $y^{b_2|s_i} = 1 - y_i$ hold.

Theorem 4 (Uniqueness of the Nash equilibrium)**.** *Under Assumption 1, the unique Nash equilibrium of this game is $(x_i, y_i) = (x^*, y^*)$ for all i as*

$$x^* = \frac{-u_3 + u_4}{u_1 - u_2 - u_3 + u_4},\ y^* = \frac{-u_2 + u_4}{u_1 - u_2 - u_3 + u_4}. \quad (9)$$

Proof Sketch. Let us prove that X's strategy in the Nash equilibrium is uniquely $\boldsymbol{x} = x^*\mathbf{1}$. First, we define u^* and v^* as X's and Y's payoffs in the Nash equilibrium in the zero-memory game. If $\boldsymbol{x} = x^*\mathbf{1}$, X's expected payoff is $u^{\mathrm{st}} = u^*$,

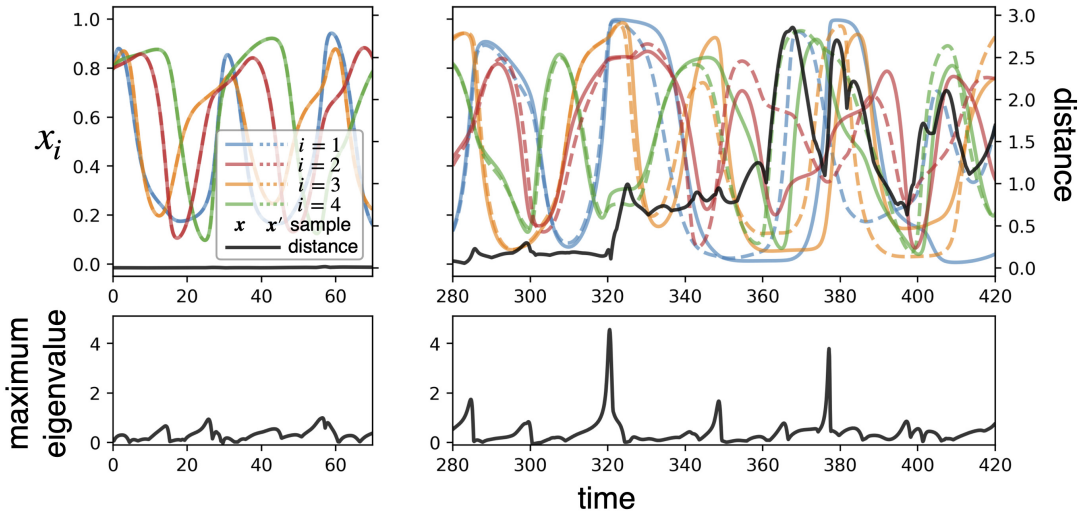

Figure 3: Initial state sensitivity in learning dynamics in multi-memory games. In the top panels, colored lines are time series of x_i (X's strategy). The black line is the distance between the solid (sample of $\boldsymbol{x}$) and broken ($\boldsymbol{x}'$) lines. In the bottom panels, the black lines indicate the maximum eigenvalue in the learning dynamics of the solid line.

regardless of Y's strategy $\boldsymbol{y}$. Second, we consider that X uses another strategy $\boldsymbol{x} \neq x^*\mathbf{1}$. Then, there is Y's strategy such that $v^{\mathrm{st}} > v^* \Leftrightarrow u^{\mathrm{st}} < u^*$. Thus, X's minimax strategy is uniquely $\boldsymbol{x} = x^*\mathbf{1}$, completing the proof. □

Regarding Theorem 4, X (Y) chooses each action in the same probability independent of the last state. Here, they do not utilize their memory. Thus, note that in this sense, the Nash equilibrium is the same as that in the zero-memory version of the game. This theorem means that in zero-sum games, the region of the Nash equilibrium does not expand even if players have memories. Taking into account that having multiple memories expands the region of Nash equilibria, such as a cooperative equilibrium in prisoner's dilemma games [Axelrod and Hamilton, 1981], this theorem is nontrivial.

In order to discuss whether our algorithms converge to this unique Nash equilibrium under Assumption 1, we consider the neighbor of the Nash equilibrium and define sufficient small deviation from the Nash equilibrium, i.e., $\boldsymbol{\delta} := \boldsymbol{x} - x^*\mathbf{1}$ and $\boldsymbol{\epsilon} := \boldsymbol{y} - y^*\mathbf{1}$. Here, we assume that these deviations have the same scale $O(\delta) := O(\delta_i) = O(\epsilon_i)$ for all i. Then, defining that the superscript (k) shows $O(\delta^k)$ terms, the dynamics are approximated by $\dot{\boldsymbol{x}} \simeq \dot{\boldsymbol{x}}^{(1)} + \dot{\boldsymbol{x}}^{(2)}$ and $\dot{\boldsymbol{y}} \simeq \dot{\boldsymbol{y}}^{(1)} + \dot{\boldsymbol{y}}^{(2)}$;

$$\dot{\boldsymbol{x}}^{(1)} = +x^*(1-x^*)(\boldsymbol{u}\cdot\mathbf{1}_{\mathsf{z}})\boldsymbol{p}^* \circ \boldsymbol{\epsilon}, \tag{10}$$

$$\dot{\boldsymbol{y}}^{(1)} = -y^*(1-y^*)(\boldsymbol{u}\cdot\mathbf{1}_{\mathsf{z}})\boldsymbol{p}^* \circ \boldsymbol{\delta}, \tag{11}$$

$$\begin{aligned}\dot{\boldsymbol{x}}^{(2)} = &-(x^*-\tilde{x}^*)(\boldsymbol{u}\cdot\mathbf{1}_{\mathsf{z}})\boldsymbol{\delta}\circ\boldsymbol{\epsilon}\circ\boldsymbol{p}^* \\ &+ x^*\tilde{x}^*(\boldsymbol{u}\cdot\mathbf{1}_{\mathsf{z}})\{(\boldsymbol{\delta}\cdot\boldsymbol{p}^*)\boldsymbol{\epsilon}\circ\boldsymbol{y}^*\circ\mathbf{1}_{\mathsf{x}} \\ &+ (\boldsymbol{\epsilon}\cdot\boldsymbol{p}^*)\boldsymbol{\epsilon}\circ\boldsymbol{x}^*\circ\mathbf{1}_{\mathsf{y}} + (\boldsymbol{\delta}\circ\boldsymbol{\epsilon}\circ\boldsymbol{y}^*\cdot\mathbf{1}_{\mathsf{x}})\boldsymbol{p}^*\},\end{aligned} \tag{12}$$

$$\begin{aligned}\dot{\boldsymbol{y}}^{(2)} = &+(y^*-\tilde{y}^*)(\boldsymbol{u}\cdot\mathbf{1}_{\mathsf{z}})\boldsymbol{\delta}\circ\boldsymbol{\epsilon}\circ\boldsymbol{p}^* \\ &- y^*\tilde{y}^*(\boldsymbol{u}\cdot\mathbf{1}_{\mathsf{z}})\{(\boldsymbol{\delta}\cdot\boldsymbol{p}^*)\boldsymbol{\delta}\circ\boldsymbol{y}^*\circ\mathbf{1}_{\mathsf{x}} \\ &+ (\boldsymbol{\epsilon}\cdot\boldsymbol{p}^*)\boldsymbol{\delta}\circ\boldsymbol{x}^*\circ\mathbf{1}_{\mathsf{y}} + (\boldsymbol{\delta}\circ\boldsymbol{\epsilon}\circ\boldsymbol{x}^*\cdot\mathbf{1}_{\mathsf{y}})\boldsymbol{p}^*\},\end{aligned} \tag{13}$$

with $\boldsymbol{x}^* := (x^*, x^*, \tilde{x}^*, \tilde{x}^*)$, $\boldsymbol{y}^* := (y^*, \tilde{y}^*, y^*, \tilde{y}^*)$, $\boldsymbol{p}^* := \boldsymbol{x}^* \circ \boldsymbol{y}^*$, $\mathbf{1}_{\mathsf{x}} := (+1, +1, -1, -1)$, $\mathbf{1}_{\mathsf{y}} := (+1, -1, +1, -1)$, and $\mathbf{1}_{\mathsf{z}} := \mathbf{1}_{\mathsf{x}} \circ \mathbf{1}_{\mathsf{y}}$. Eqs. (10)-(13) are derived by considering small changes in the stationary condition $\boldsymbol{p}^{\mathrm{st}} = \boldsymbol{M}\boldsymbol{p}^{\mathrm{st}}$ for deviations of $\boldsymbol{\delta}$ and $\boldsymbol{\epsilon}$ (see Technical Appendix B.1 and B.2 for the detailed calculation). By that, we can avoid a direct calculation of $\boldsymbol{p}^{\mathrm{st}}$, which is hard to be obtained.

5 Experimental Findings

5.1 Simulation and Low-Order Approximation

From the obtained dynamics, i.e., Eqs. (10)-(13), we interpret the learning dynamics in detail. In the first-order dynamics, multi-memory learning is no more than a simple extension of the zero-memory one. Indeed, the zero-memory learning draws an elliptical orbit given by Hamiltonian as the conserved quantity [Hofbauer, 1996; Mertikopoulos *et al.*, 2018]. Eqs. (10) and (11) mean that the multi-memory dynamics also draw similar elliptical orbits for each pair of x_i and y_i. In other words, the dynamics are given by a linear flow on a four-dimensional torus. Because no interaction occurs between the pair of i and i' such that $i \neq i'$, the dynamics of the multi-memory learning for each state are qualitatively the same as learning without memories. Fig. 2 shows the time series of the multi-memory learning dynamics near the Nash equilibrium in an example of a two-action zero-sum game, the matching-pennies game ($u_1 = u_4 = 1$, $u_2 = u_3 = -1$). The experimental trajectories are generated by the Runge-Kutta fourth-order method of Eq. (8) (see Technical Appendix B.3 for details), while the approximated trajectories are by the Runge-Kutta fourth-order method for the first- (Eqs. (10) and (11)), the second- (Eqs. (12) and (13)), and the third-order approximations (in Technical Appendix B.2). The step-size is 10^{-2} in common. The top-left panel in the figure shows that the dynamics roughly draw a circular orbit for each state and are well approximated by the first-order dynamics of Eqs. (10) and (11). However, the top-right panel, where a sufficiently long time has passed, shows that the dynamics

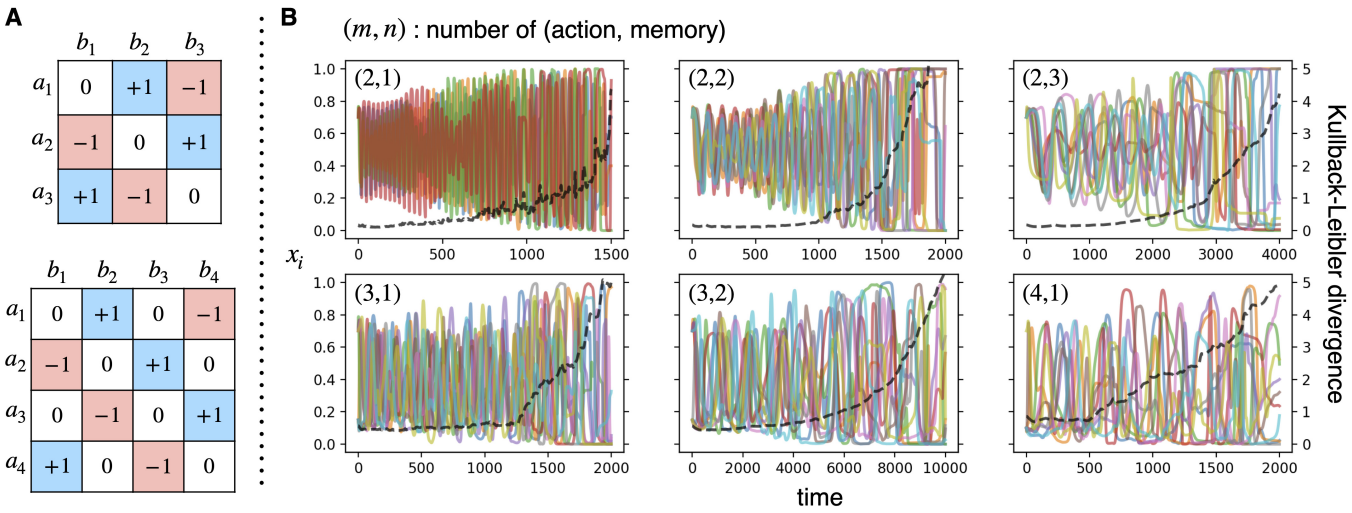

Figure 4: **A**. Payoff matrices of three-action (rock-paper-scissors) and four-action (extended rock-paper-scissors) games. **B**. In each panel, colored lines indicate time series of $x^{a|s}$ for random $a \in \mathcal{A}$ and $s \in \mathcal{S}$. The black broken line indicates the Kullback-Leibler divergence averaged over all the states $s \in \mathcal{S}$, intuitively meaning a distance from the Nash equilibrium.

deviate from the circular orbits (see Technical Appendix C in detail).

Such deviation from the circular orbits is given by higher-order dynamics than Eqs. (10) and (11). In the second-order dynamics given by Eqs. (12) and (13), the multi-memory learning is qualitatively different from the zero-memory one. Indeed, Eqs. (12) and (13) obviously mean that interactions occur between the pair of i and i' such that $i \neq i'$. Thereby, the dynamics of multi-memory learning become much more complex than that of zero-memory learning. In practice, no Hamiltonian function, denoted by $H^{(2)}$, exists in the second-order dynamics, as different from the first-order one. One can check this by calculating $\partial \dot{x}_i^{(2)}/\partial \epsilon_{i'} + \partial \dot{y}_{i'}^{(2)}/\partial \delta_i \neq 0$ for i and $i' \neq i$, if assuming that Hamiltonian should satisfy $\dot{\boldsymbol{x}}^{(2)} = +\partial H^{(2)}/\partial \boldsymbol{\epsilon}$ and $\dot{\boldsymbol{y}}^{(2)} = -\partial H^{(2)}/\partial \boldsymbol{\delta}$. Thus, the multi-memory dynamics might not have any conserved quantities and not draw any closed trajectory. Indeed, the right panels in Fig. 2 show that the dynamics tend to diverge from the Nash equilibrium. This divergence from the Nash equilibrium is surprising because zero-memory learning in zero-sum games always has a closed trajectory and keeps the Kullback-Leibler divergence from the Nash equilibrium constant [Piliouras *et al.*, 2014; Mertikopoulos *et al.*, 2018]. Here, note that we need the third-order dynamics to fit the experimental dynamics well, as seen by comparing the middle-right and lower-right panels in Fig. 2. The error between the experiment ($\boldsymbol{\delta}$ and $\boldsymbol{\epsilon}$) and approximation ($\boldsymbol{\delta}'$ and $\boldsymbol{\epsilon}'$) is evaluated by

$$\text{error} := \frac{1}{4}\sum_{i=1}^{4}\sqrt{|\delta_i - \delta_i'|^2 + |\epsilon_i - \epsilon_i'|^2}. \tag{14}$$

5.2 Chaos-Like and Heteroclinic Dynamics

Interestingly, learning dynamics in multi-memory games are complex. Fig. 3 shows two learning dynamics between which there is a slight difference in their initial strategies ($\boldsymbol{x} = \boldsymbol{y} = 0.8 \times \mathbf{1}$ in the solid line, but in the broken line ($\boldsymbol{x}'$ and $\boldsymbol{y}'$), $x_1' = 0.801$ and others are the same as the solid line). We use Algorithm 2 with $\eta = 10^{-3}$ and $\gamma = 10^{-6}$. These dynamics are similar in the beginning ($0 \leq t \leq 320$). However, the difference between these dynamics is gradually amplified ($320 \leq t \leq 360$), leading to the crucial difference eventually ($360 \leq t \leq 420$). We here introduce the distance between $\boldsymbol{x}'$ and $\boldsymbol{x}$ as

$$D(\boldsymbol{x}', \boldsymbol{x}) := \frac{1}{4}\sum_{i=1}^{4}|L(x_i') - L(x_i)|, \tag{15}$$

with $L(x) := \log x - \log(1-x)$; $L(x)$ is the measure taking into account the weight in replicator dynamics. Furthermore, in order to analyze how the difference is amplified, Fig. 3 also shows the maximum eigenvalue in learning dynamics. We can see that the larger the maximum eigenvalue is, the more the difference between the two trajectories is amplified. We observe that such an amplification typically occurs when strategies are close to the boundary of the simplex. In conclusion, the learning dynamics provide chaos-like sensitivity to the initial condition.

5.3 Divergence in General Memories and Actions

Although we have focused on the one-memory two-action zero-sum games so far, numerical simulations demonstrate that similar phenomena are seen in games of other numbers of memories and actions. Fig. 4 shows the trajectories of learning dynamics in various multi-memory and multi-action games, where we use Algorithm 2 with $\eta = 10^{-2}$ and $\gamma = 10^{-6}$. Note that we consider zero-sum games in all the panels (see Fig. 4-A for the payoff matrices). In Fig. 4-B, each panel shows that strategy variables $x^{a|s}$ roughly diverge from the Nash equilibrium and sojourn longer at the edges of the simplex, i.e., $x^{a|s} = 0$ or 1. Furthermore, Kullback-Leibler divergence from the Nash equilibrium averaged over

the whole states, i.e.,

$$D_{\mathrm{KL}}(\mathbf{x}^*\|\mathbf{x}) := \frac{1}{|\mathcal{S}|}\sum_{s\in\mathcal{S}}\sum_{a\in\mathcal{A}} x^{*a|s}\log\frac{x^{*a|s}}{x^{a|s}}. \tag{16}$$

also increases with time in each panel of the figure. Thus, we confirm that learning reaches heteroclinic cycles under various (action, memory) pairs.

6 Conclusion

This study contributes to an understanding of a cutting-edge model of learning in games in Sections 3 and 4. In practice, several famous algorithms, i.e., replicator dynamics and gradient ascent, were newly extended to multi-memory games (Algorithms 1 and 2). We proved the correspondence between these algorithms (Theorems 1-3) in general. Under the assumptions of one-memory two-action zero-sum games, we further proved the uniqueness of the Nash equilibrium in two-action zero-sum games (Theorem 4). As a background, even if agents do not have their memories, multi-agent learning dynamics are generally complicated. Thus, many theoretical approaches usually have been taken to grasp such complicated dynamics. Learning dynamics in multi-memory games are much more complicated and the dimension of strategy space of an agent explodes as $(m-1)m^{2n}$ with memory number n and action number m. Despite these challenges, our theorems succeeded in capturing chaos-like and diverging behaviors of the dynamics. Potential future studies may focus on considering how to avoid the curse of dimension in the strategy space and proving whether the Nash equilibrium is unique in general numbers of action and memory.

This study also experimentally discovered a novel and non-trivial phenomenon that simple learning algorithms such as replicator dynamics and gradient ascent asymptotically reaches a heteroclinic cycle in multi-memory zero-sum games. In other words, the players choose actions in highly skewed proportions throughout learning. Such a phenomenon is specific to multi-memory games: Perhaps this is because the gameplay becomes extreme in learning between those who can use equally sophisticated (i.e., multi-memory) strategies. We also found a novel problem that the Nash equilibrium is difficult to reach in multi-memory zero-sum games. Here, note that convergence to the Nash equilibrium, either as a last-iterate [Daskalakis *et al.*, 2018; Daskalakis and Panageas, 2019; Mertikopoulos *et al.*, 2019; Golowich *et al.*, 2020; Wei *et al.*, 2021; Lei *et al.*, 2021; Abe *et al.*, 2022] or as an average of trajectories [Banerjee and Peng, 2005; Zinkevich *et al.*, 2007; Daskalakis *et al.*, 2011], is a frequently discussed topic. In general, heteroclinic cycles fail to converge even on average. What algorithm can converge to the Nash equilibrium in multi-memory zero-sum games would be interesting future work.

Acknowledgments

We thank Tetsuro Morimura and Kunihiko Kaneko for fruitful discussions. Y.F. acknowledges the support by JSPS KAKENHI Grant No. JP21J01393.

References

[Abdallah and Kaisers, 2013] Sherief Abdallah and Michael Kaisers. Addressing the policy-bias of q-learning by repeating updates. In *AAMAS*, pages 1045–1052, 2013.

[Abe *et al.*, 2022] Kenshi Abe, Mitsuki Sakamoto, and Atsushi Iwasaki. Mutation-driven follow the regularized leader for last-iterate convergence in zero-sum games. In *UAI*, pages 1–10, 2022.

[Axelrod and Hamilton, 1981] Robert Axelrod and William D Hamilton. The evolution of cooperation. *Science*, 211(4489):1390–1396, 1981.

[Banerjee and Peng, 2005] Bikramjit Banerjee and Jing Peng. Efficient no-regret multiagent learning. In *AAAI*, pages 41–46, 2005.

[Barfuss *et al.*, 2019] Wolfram Barfuss, Jonathan F Donges, and Jürgen Kurths. Deterministic limit of temporal difference reinforcement learning for stochastic games. *Physical Review E*, 99(4):043305, 2019.

[Barfuss, 2020a] Wolfram Barfuss. Reinforcement learning dynamics in the infinite memory limit. In *AAMAS*, pages 1768–1770, 2020.

[Barfuss, 2020b] Wolfram Barfuss. Towards a unified treatment of the dynamics of collective learning. In *Challenges and Opportunities for Multi-Agent Reinforcement Learning, AAAI Spring Symposium*, 2020.

[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.

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

[Bowling and Veloso, 2002] Michael Bowling and Manuela Veloso. Multiagent learning using a variable learning rate. *Artificial Intelligence*, 136(2):215–250, 2002.

[Bowling, 2004] Michael Bowling. Convergence and no-regret in multiagent learning. In *NeurIPS*, pages 209–216, 2004.

[Cross, 1973] John G Cross. A stochastic learning model of economic behavior. *The Quarterly Journal of Economics*, 87(2):239–266, 1973.

[Daskalakis and Panageas, 2019] Constantinos Daskalakis and Ioannis Panageas. Last-iterate convergence: Zero-sum games and constrained min-max optimization. In *ITCS*, pages 27:1–27:18, 2019.

[Daskalakis *et al.*, 2011] Constantinos Daskalakis, Alan Deckelbaum, and Anthony Kim. Near-optimal no-regret algorithms for zero-sum games. In *SODA*, pages 235–254, 2011.

[Daskalakis *et al.*, 2018] Constantinos Daskalakis, Andrew Ilyas, Vasilis Syrgkanis, and Haoyang Zeng. Training GANs with optimism. In *ICLR*, 2018.

[Fudenberg and Maskin, 2009] Drew Fudenberg and Eric Maskin. The folk theorem in repeated games with discounting or with incomplete information. In *A long-run collaboration on long-run games*, pages 209–230. World Scientific, 2009.

[Fudenberg and Tirole, 1991] Drew Fudenberg and Jean Tirole. *Game theory*. MIT press, 1991.

[Fujimoto and Kaneko, 2019a] Yuma Fujimoto and Kunihiko Kaneko. Emergence of exploitation as symmetry breaking in iterated prisoner's dilemma. *Physical Review Research*, 1(3):033077, 2019.

[Fujimoto and Kaneko, 2019b] Yuma Fujimoto and Kunihiko Kaneko. Functional dynamic by intention recognition in iterated games. *New Journal of Physics*, 21(2):023025, 2019.

[Fujimoto and Kaneko, 2021] Yuma Fujimoto and Kunihiko Kaneko. Exploitation by asymmetry of information reference in coevolutionary learning in prisoner's dilemma game. *Journal of Physics: Complexity*, 2(4):045007, 2021.

[Golowich *et al.*, 2020] Noah Golowich, Sarath Pattathil, and Constantinos Daskalakis. Tight last-iterate convergence rates for no-regret learning in multi-player games. In *NeurIPS*, pages 20766–20778, 2020.

[Hofbauer *et al.*, 1998] Josef Hofbauer, Karl Sigmund, et al. *Evolutionary games and population dynamics*. Cambridge university press, 1998.

[Hofbauer, 1996] Josef Hofbauer. Evolutionary dynamics for bimatrix games: A hamiltonian system? *Journal of Mathematical Biology*, 34(5):675–688, 1996.

[Kaisers and Tuyls, 2010] Michael Kaisers and Karl Tuyls. Frequency adjusted multi-agent q-learning. In *AAMAS*, pages 309–316, 2010.

[Lanctot *et al.*, 2012] Marc Lanctot, Richard Gibson, Neil Burch, Martin Zinkevich, and Michael Bowling. No-regret learning in extensive-form games with imperfect recall. In *ICML*, pages 1035–1042, 2012.

[Lei *et al.*, 2021] Qi Lei, Sai Ganesh Nagarajan, Ioannis Panageas, et al. Last iterate convergence in no-regret learning: constrained min-max optimization for convex-concave landscapes. In *AISTATS*, pages 1441–1449, 2021.

[Littman, 1994] Michael L Littman. Markov games as a framework for multi-agent reinforcement learning. In *ICML*, pages 157–163, 1994.

[Mertikopoulos and Sandholm, 2016] Panayotis Mertikopoulos and William H Sandholm. Learning in games via reinforcement and regularization. *Mathematics of Operations Research*, 41(4):1297–1324, 2016.

[Mertikopoulos *et al.*, 2018] Panayotis Mertikopoulos, Christos Papadimitriou, and Georgios Piliouras. Cycles in adversarial regularized learning. In *SODA*, pages 2703–2717, 2018.

[Mertikopoulos *et al.*, 2019] Panayotis Mertikopoulos, Bruno Lecouat, Houssam Zenati, Chuan-Sheng Foo, Vijay Chandrasekhar, and Georgios Piliouras. Optimistic mirror descent in saddle-point problems: Going the extra(-gradient) mile. In *ICLR*, 2019.

[Meylahn *et al.*, 2022] Janusz M Meylahn, Lars Janssen, et al. Limiting dynamics for q-learning with memory one in symmetric two-player, two-action games. *Complexity*, 2022, 2022.

[Nash Jr, 1950] John F Nash Jr. Equilibrium points in n-person games. *Proceedings of the National Academy of Sciences*, 36(1):48–49, 1950.

[Nowak and Sigmund, 1993] Martin Nowak and Karl Sigmund. A strategy of win-stay, lose-shift that outperforms tit-for-tat in the prisoner's dilemma game. *Nature*, 364(6432):56–58, 1993.

[Piliouras *et al.*, 2014] Georgios Piliouras, Carlos Nieto-Granda, Henrik I Christensen, and Jeff S Shamma. Persistent patterns: Multi-agent learning beyond equilibrium and utility. In *AAMAS*, pages 181–188, 2014.

[Sandholm and Crites, 1996] Tuomas W Sandholm and Robert H Crites. Multiagent reinforcement learning in the iterated prisoner's dilemma. *Biosystems*, 37(1-2):147–166, 1996.

[Shapley, 1953] Lloyd S Shapley. Stochastic games. *Proceedings of the National Academy of Sciences*, 39(10):1095–1100, 1953.

[Singh *et al.*, 2000] Satinder Singh, Michael J Kearns, and Yishay Mansour. Nash convergence of gradient dynamics in general-sum games. In *UAI*, pages 541–548, 2000.

[Tuyls and Nowé, 2005] Karl Tuyls and Ann Nowé. Evolutionary game theory and multi-agent reinforcement learning. *The Knowledge Engineering Review*, 20(1):63–90, 2005.

[Tuyls *et al.*, 2006] Karl Tuyls, Pieter Jan'T Hoen, and Bram Vanschoenwinkel. An evolutionary dynamical analysis of multi-agent learning in iterated games. *Autonomous Agents and Multi-Agent Systems*, 12(1):115–153, 2006.

[Watkins and Dayan, 1992] Christopher JCH Watkins and Peter Dayan. Q-learning. *Machine learning*, 8(3):279–292, 1992.

[Wei *et al.*, 2021] Chen-Yu Wei, Chung-Wei Lee, Mengxiao Zhang, and Haipeng Luo. Linear last-iterate convergence in constrained saddle-point optimization. In *ICLR*, 2021.

[Zinkevich *et al.*, 2007] Martin Zinkevich, Michael Johanson, Michael Bowling, and Carmelo Piccione. Regret minimization in games with incomplete information. In *NeurIPS*, pages 1729–1736, 2007.

[Zinkevich, 2003] Martin Zinkevich. Online convex programming and generalized infinitesimal gradient ascent. In *ICML*, pages 928–936, 2003.