Exploitation by asymmetry of information reference in coevolutionary learning in prisoner’s dilemma game

Yuma Fujimoto and Kunihiko Kaneko

1. Introduction

Cooperation, defection, and exploitation are important relationships that universally appear in biological and social systems. While cooperating, individuals are altruistic and achieve benefits for the entire group. In defection, they behave selfishly for their own benefit, which results in demerits for all. In exploitation, selfish individuals receive benefit at the expense of altruistic others. The choice of strategy, i.e., selfish or altruistic behavior, is important in establishing social relationships. Individuals, based on their abilities, sophisticate their strategies through their experiences. Generally, people’s ability to choose the best strategies differ. These differences in ability can affect how cooperation is established between people. Now the following question arises: do individuals with higher abilities exploit those with lower abilities or vice versa?

Game theory is a mathematical framework for analyzing such individual decision-making of strategies [1]. Everyone has a strategy for choosing given actions and receives a reward based on their chosen actions. In particular, the prisoner’s dilemma (PD) game (see figure 1(A)) has been used extensively to investigate how people act when competing for benefits. Each of two players chooses either cooperation (C) or defection (D), depending on their own strategies. Accordingly, the single game has four results, namely CC, CD, DC, and DD, where the left and right symbols (C or D) indicate the action taken by oneself and that of the opponent, respectively. The benefit of them is given by $R$, $S$, $T$, and $P$. The property of PD demands $T > R > P > S$: each
player receives a larger benefit for choosing D, which may lead to DD, but CC is more beneficial than DD. How to avoid falling into mutual defection, i.e., DD, has been a significant issue.

In this study, we assume the infinite iteration of games, and then each player can refer to the actions of the previous game and change the own choice depending on the observed actions. We consider that a player can change their next action based on the previous actions of the two players, i.e., CC, CD, DC, and DD at the maximum. Thus, a player with more detailed information about a previous round has a higher ability to choose their own optimal action. This ability to observe actions of their own and their opponents is seen in reality, such as intention recognition [2–8]. Furthermore, in social systems, from the bacterial community to international war, the reference to past actions might play an important role in establishing interpersonal relationships [9]. A representative example is tit-for-tat (TFT) strategy [10, 11], in which the player observes and mimics the other’s previous action. The monumental and the numerous following studies showed that the players with TFT strategies are selected in the optimization process and establish cooperation. This strategy is classified as a reactive strategy [12–17] as the player chooses their actions by referring only to their opponent’s previous action. Another type of strategy, the memory-one strategy [17–23], is introduced when a player refers to both their own previous action as well as their opponent’s. The memory-one strategy includes not only the TFT but the Win–Stay–Lose–Shift (WSLS) strategy [18], which generates cooperation even under the error in choice of action. Consequently, it is expected that a player who refers to more information will succeed in receiving a larger benefit.

Although such reference of past actions plays a crucial role to establish interpersonal relationships, how the difference in information for it between the players affects in the player’s benefit is unclear yet. Evolution of strategy in a single population [12–25] does not generate the difference in payoffs among the players in principle, where the strategy is selected within the same group. Evolution in multi-population [26–31] (i.e., intra-group selection of strategies by the inter-group game) and co-learning among multi-agent [32–39] can generate the exploitation, but so far most studies have focused only on the emergence of cooperation.

Recently, the exploitation has been studied as to ‘symmetry’ breaking of the payoffs of players [28, 36], where only the game between the same class of strategies is assumed. Here, we revised the coupled replicator model in the previous studies of multi-population evolution [40] and multi-agent learning [41–43] so that the player can refer to their own previous actions and those of their opponent and update their strategy accordingly within a class of strategies. In particular, we focused on the reactive and memory-one classes of strategies, as they are basic and have been studied extensively. We then investigated whether players using the memory-one strategy would win the game against opponents using the reactive strategy, by utilizing the extra information provided through the observation of their own previous actions.

The remainder of this paper is organized as follows. In section 2, we formulate the learning dynamics for various strategy classes. In addition, we confirm that in infinitely iterated games against an opponent’s fixed strategy, the strategy with the higher ability obtains a larger payoff in equilibrium. In section 3, we introduce an example of mutual learning between memory-one and reactive strategies. Then, we demonstrate that the memory-one class, i.e., the player with the higher ability, is, counterintuitively, one-sidedly exploited by the reactive one. In section 4, we analyze how this exploitation is achieved and elucidate that the ability to reference one’s own actions leads to generosity and leaves room for exploitation. Finally, in section 5, we show that high-ability players are generally exploited because of their generosity, independent of the strategy class and payoff matrix.
2. Formulation of learning dynamics of strategies

2.1. Formulation of class and strategy

Before formulating the learning process, we mathematically define the strategy and class. Recall that a single game can have one of four results: CC, CD, DC, and DD. First, a player using a memory-one strategy can refer to their own and the opponent’s previous actions and respond with a different action to each result of the previous game. Thus, the player has four independent stochastic variables \( x_1, x_2, x_3, \) and \( x_4 \) as the probabilities of choosing C regardless of the outcome of the previous game. Thus, the memory-one class is defined as the possible set of such memory-one strategies, which is denoted as \( \{x_1, x_2, x_3, x_4\} \in [0, 1]^4 \). The TFT and WSLS [18, 44–46] strategies are examples of the emergence of cooperation as a result of the long-term repeated game. Thus, the memory-one class is defined as the least complex of the three classes.

Second, a player using the reactive strategy can only refer to the opponent's action and therefore cannot distinguish between CC and DC (CD and DD). Thus, the strategy is given by two independent variables \( x_{13} \) and \( x_{24} \), where the former (latter) is the probability of choosing C where the previous result was either CC or DC (CD or DD). Therefore, the reactive class is defined as \( \{x_{13}, x_{24}\} \in [0, 1]^2 \). Here, the notation \( x_{13} \) (variable for CC and DC) clearly indicates the integration of \( x_1 \) (CC) and \( x_3 \) (DC) from the memory-one class. Indeed, all the strategies in the reactive class are included in the memory-one class, as one can set \( x_1 = x_3 = x_{13} \) and \( x_2 = x_4 = x_{24} \) for all \( x_{13} \) and \( x_{24} \). Thus, the above TFT strategy can be represented as \( x_{13} = 1 \) and \( x_{24} = 0 \), whereas the WSLS strategy cannot be represented. This makes it clear that the memory-one class is more complex than the reactive one, because the former can include all the strategies of the latter. The ordering of complexity can be defined as a class that includes all the strategies of the other is more complex.

Third, in the classical mixed strategy [47], a player stochastically chooses their action without referencing any actions from the previous game. Thus, the strategy controls only one variable \( x_{1234} \) \( \in [0, 1] \), which is the probability of choosing C. This class is the least complex of the three classes.

2.2. Analysis of infinitely repeated game

In this section, we analyze an infinitely repeated game under the condition that the strategy of both players are fixed. We define \( \mathbf{p} := (p_{CC}, p_{CD}, p_{DC}, p_{DD})^T \) as the probabilities that (CC, CD, DC, DD) are played in the present round, and the result of the next round is calculated by \( \mathbf{p}' = \mathbf{M} \mathbf{p} \) with

\[
\mathbf{M} := \begin{pmatrix}
    x_1 y_1 & x_2 y_1 & x_3 y_2 & x_4 y_2 \\
    x_1 y_1 & x_2 y_1 & x_3 y_2 & x_4 y_2 \\
    x_1 y_3 & x_2 y_3 & x_3 y_2 & x_4 y_2 \\
    x_1 y_3 & x_2 y_3 & x_3 y_2 & x_4 y_2
\end{pmatrix}.
\]

When none of the strategy variables, \( x_n \) and \( y_n \) for \( n \in \{1, \ldots, 4\} \) are 0 or 1, the infinitely repeated game has only one equilibrium state \( \mathbf{p}_e := (p_{CCe}, p_{CDe}, p_{DCe}, p_{DDe})^T \). Here, we can directly compute \( \mathbf{p}_e \) as

\[
p_{CCe} = k \{ (x_4 + (x_3 - x_4)y_2) (y_4 + (y_3 - y_4)x_2) - x_3 y_3 (x_2 - x_4)(y_2 - y_4) \},
\]
\[
p_{CDe} = k \{ (x_4 + (x_3 - x_4)y_4) (y_2 - (y_1 - y_2)x_1) - x_3 y_4 (x_1 - x_3)(y_2 - y_4) \},
\]
\[
p_{DDe} = k \{ (x_2 - (x_1 - x_2)y_1)(y_4 + (y_3 - y_4)x_4) - x_1 y_4 (x_4 - x_2)(y_1 - y_3) \}.
\]

Here the coefficient \( k \) is determined by the normalization of the probabilities \( p_{CCe} + p_{CDe} + p_{DCe} + p_{DDe} = 1 \).

2.3. Learning dynamics of memory-one class

We next consider adaptive learning from past experiences of repeated games. For instance, we assume that the probability of CC being the result of the previous round is \( p_{CCe} \). Then, the next action being C (D) would have the probability of \( x_1 (x_i) \). Here, we define \( u_{CC(G)} (u_{CC(D)}) \) as the benefit that the player gains by performing action C (D). First, the time evolution of \( x_1 \) is assumed to depend on the amount of experience: the previous game’s result and the action in the present one must be CC and C, respectively. Thus, \( x_1 \) is proportional to \( p_{CCx_1} \). Second, \( x_1 \) also depends on the benefit of action C and thus is proportional to \( u_{CC(C)} - u_{CC(D)} \). To summarize, we get

\[
x_1 = x_1 x_1 p_{CCx_1} (u_{CC(C)} - u_{CC(D)}).
\]

Next, we compute \( u_{CC(G)} \) and \( u_{CC(D)} \). When the previous game’s result and the present self-action are CC and C, respectively, the present state is given by \( \mathbf{p} = p_{CC(G)} := (y_1, y_1, 0, 0)^T \). If \( p_{CC(G)} \neq p_e \) holds, then the state gradually relaxes to equilibrium with the repetition of the game. Thus, \( u_{CC(G)} \) is the total payoff generated by
A pressed version of the memory-one model, as the constraints can derive the time evolution of the other strategy variables payoffs over time because the opponent’s strategy is fixed. However, there are two major differences between strategy’s payoff based on the learning dynamics. The payoffs of both classes monotonically increase their speed of our model depends on the amount of experience. Because the frequency of observing the opponent’s previous game’s result and the present self-action are CC and D, respectively, the present state is given by $p = p_{CC(D)} := (0, 0, y_1, y_1)^T$. Then, $u_{CC(D)}$ is computed in the same way using

$$u_{CC(D)} = \sum_{t=0}^{\infty} M'(p_{CC(D)} - p_e) \cdot u.$$ (5)

By substituting equations (4) and (5) into equation (3), we can write the learning dynamics of $x_1$ as

$$\dot{x}_1 = x_1 \bar{x}_1 p_{CC} \sum_{t=0}^{\infty} M'(p_{CC(C)} - p_{CC(D)}) \cdot u,$$ (6)

using only strategy variables, $x_n$ and $y_n$ for $n \in \{1, \ldots, 4\}$, and the payoff variables $(T, R, P, S)$. Similarly, we can derive the time evolution of the other strategy variables $x_2$, $x_3$, and $x_4$.

Equation (6) appears complicated at first glance, but it can be simplified as

$$\dot{x}_1 = x_1 \bar{x}_1 \frac{\partial p_{CC}}{\partial x_1} \cdot u,$$ (7)

(see the proof in appendix A). The same equations hold for the learning of $x_2$, $x_3$, and $x_4$, and for opponent player. Notably, this equation reproduces the original coupled replicator model [40, 41].

### 2.4. Learning dynamics of other strategies

In the previous sections, we formulated the learning dynamics of memory-one class strategies against another within the same class. In this section, we consider other cases in which both learned and learning players adopt the reactive class.

First, we consider a case in which a learned player uses the reactive class, and the learning player uses the memory-one class. In this case, the learning is easily given by

$$\dot{x}_n \left( \{ x_1, \ldots, x_4 \}, \{ y_{13}, y_{24} \} \right) = \dot{x}_1 \big|_{y_1 = y_3 = y_{13}, y_2 = y_{24}} = \bar{x}_1,$$ (8)

for $n \in \{1, \ldots, 4\}$ because the learned reactive player’s strategy is constrained by $y_1 = y_3 = y_{13}$ and $y_2 = y_4 = y_{24}$.

Second, we consider a case in which the learning player uses the reactive class, and the learned player uses the memory-one class. In this case, the learning player’s strategy is given by $(x_{13}, x_{24})$. Recall that the learning speed of our model depends on the amount of experience. Because the frequency of observing the opponent’s previous action, C, is the total of both CC and DC, the time evolution of $x_{13}$ is the sum of $x_1$ and $x_3$. Thus, we get

$$\dot{x}_{13} \left( \{ x_{13}, x_{24} \}, \{ y_{11}, \ldots, y_{14} \} \right) = (\dot{x}_1 + \dot{x}_3) \big|_{x_1 = x_3, x_2 = x_4 = x_{24}} = \bar{x}_{13},$$

and

$$\dot{x}_{24} \left( \{ x_{13}, x_{24} \}, \{ y_{11}, \ldots, y_{14} \} \right) = (\dot{x}_2 + \dot{x}_4) \big|_{x_1 = x_3, x_2 = x_4 = x_{24}} = \bar{x}_{24}.$$ (9)

### 3. Numerical result for learning

#### 3.1. One-sided learning against a fixed strategy

Before investigating the game between the memory-one and reactive classes, we first study the learning of the memory-one and reactive classes against the other fixed strategy. Figure 2 shows the time series of each strategy’s payoff based on the learning dynamics. The payoffs of both classes monotonically increase their payoffs over time because the opponent’s strategy is fixed. However, there are two major differences between the two classes in the way the payoff increases.

First, the reactive class learns faster than the memory-one class. This is because the reactive class is a compressed version of the memory-one model, as the constraints $x_1 = x_3$ and $x_2 = x_4$ are postulated, that is, the learning in the cases of CC and DC (CD and DD) are integrated. Recall that the change in strategy is optimized based on the empirical data sampled through the played games. In the reactive class, the number of strategy variables is fewer; therefore, quick optimization can be achieved, as shown in equation (9).
Second, the memory-one strategy gains a larger payoff in equilibrium than the reactive one. This is simply because the memory-one strategy contains a reactive strategy. Accordingly,

$$\max_{x_1, x_2, x_3, x_4} p_e(\{x_1, x_2, x_3, x_4\}, y) \geq \max_{x_1, x_2, x_3, x_4} p_e(\{x_1, x_2\}, y) \cdot u,$$

is derived for all the opponent’s fixed strategies $y$.

### 3.2. Mutual learning between memory-one and reactive classes

In section 3.1, we considered one-sided learning, where a player dynamically optimizes the strategy against their opponent’s fixed strategy. In this section, we consider mutual learning, where both players optimize their strategies as the opponent’s strategy continues to change.

We excluded the mixed class in this study because the results of matches with the mixed class are trivial. A player with a mixed class must use the same action independently of the opponent’s previous actions. Thus, the opponent always receives a higher payoff by choosing D according to the payoff matrix of the PD. When the opponent’s choice is always D, the best choice of the mixed class is also D. Thus, only the pure DD will result in equilibrium. Therefore, the mixed class can establish neither cooperative nor exploitative relationships.

Therefore, we consider only the game between players using the memory-one and reactive classes. In our model, the dynamics of players’ strategies are deterministic. Thus, the equilibrium state is uniquely determined by the initial values of strategies $x$ and $y$. Here, we take sampling over the initial conditions. A match between each pair of classes was evaluated using a sufficiently large number of initial conditions of $x$ and $y$. In each sample, the initial values of the strategy were assumed to be given randomly. In other words, when the former player takes a memory-one (reactive) class, the strategy is randomly chosen from $\{x_1, x_2, x_3, x_4\} \in [0, 1]^4$ ($\{x_1, x_2\} \in [0, 1]^2$).

Figure 3 shows the final state of mutual learning for three matches: (A) between two memory-one classes, (B) between the memory-one and reactive classes, and (C) between two reactive classes. (Note that the last case represented in (C) was already studied in [36].) Here, recall that mutual cooperation satisfies $p_{CCe} = 1$, mutual defection satisfies $p_{DDe} = 1$, and exploitation satisfies $p_{CDe} \neq p_{DGe}$.

First, we studied the matches between the same classes, represented in (A) and (C). In these matches, exploitation with $p_{CDe} \neq p_{DGe}$ can be in equilibrium. In other words, asymmetry is permanently established between the players depending on their initial strategies, even though both deterministically improve their own strategies to receive a larger payoff. Notably, this asymmetry emerges symmetrically between the players when using the same class. In this case, the number of samples that satisfied $p_{CDe} > p_{DGe}$ was equal to those that satisfied $p_{CDe} < p_{DGe}$. In (C), i.e., the match between the reactive classes, the equilibrium exists as multiple fixed points with $p_{CDe} \neq p_{DGe}$ (see [36] for the detailed analysis). By contrast, each exploitative state in (A) permanently oscillated to form a limit cycle, whereas the temporal averages of $p_{CDe}$ and $p_{DGe}$ are not equal. There is an infinite number of limit cycles, one of which is achieved depending on the initial conditions. A detailed analysis of these limit cycles will be explained in the next section 4.

The heterogeneous match (B) between the memory-one and reactive classes has the same exploitative states as match (A). However, the most remarkable difference here is that in this exploitation, the reactive class can receive a larger payoff in the match with the memory-one class, and the reverse never occurs. In other words,
only the one-sided exploitation from the reactive class to the memory-one class emerges, regardless of the initial conditions. This result appears paradoxical, when one notes that the memory-one class has more information for the strategy choices and is indeed in a more advantageous position than the reactive one in equilibrium when the other player’s strategy is fixed, as already confirmed in section 3.1. We will discuss the origin of this unintuitive result in section 4.

4. Emergence of oscillatory exploitation

4.1. Analysis of exploitation

We first analyzed the exploitation between the memory-one classes, but the analysis is also applicable to the case between memory-one and reactive classes. An example of the trajectory of strategies $x_i$ and $y_j$ during exploitation is shown in figure 4. For all cases, the exploiting player’s strategy satisfies

$$x_1 : \text{fixed point, } x_2 \to 0,$$

$$x_3 : \text{oscillates, } x_4 \to 0.$$  \hfill (11)

On the other hand, the exploited opponent’s strategy satisfies

$$y_1 : \text{fixed point, } y_2 \to 0,$$

$$y_3 \to 1, \quad y_4 : \text{oscillates.}$$  \hfill (12)

Here, note that $x_1$ and $y_1$ are neutrally stable, and the asymptotic value continuously varies with the initial condition. Assuming that $x_2 = x_4 = y_2 = 0$ and $y_3 = 1$ and inserting them into equation (2), the possibility vector $p$ satisfies

$$p \propto (0, y_4 x_3, y_4, x_3)^T.$$  \hfill (13)

This equation leads to $p_{DCe} > p_{CDe}$, which proves that player 1 always receives a larger payoff than player 2. By inserting this into the learning dynamics in equation (7), we obtain

$$\dot{x}_1 = \dot{y}_1 = 0,$$

$$\dot{x}_3 = x_1((T - 2P + S) - (T - S)y_4)\frac{y_4 x_3}{(1 + y_4 - x_3 + y_4 x_3)^2},$$

$$\dot{y}_4 = x_1((T - P)x_3 - (P - S))\frac{y_4 x_3}{(1 + y_4 - x_3 + y_4 x_3)^2}.$$  \hfill (14)

These equations indicate that $x_1$ and $y_1$ are neutral, as expected. This is because the players do not experience CC in this oscillatory equilibrium of exploitation and do not have the chance to change $x_1$ or $y_1$ by learning. Now, the two-variable dynamics $x_3$ and $y_4$ are obtained, which leads to oscillation.
The oscillatory dynamics for \((x_3, y_4)\) follow the Lotka–Volterra type equation. Equation (14) has an infinite number of periodic solutions (cycles), and the cycle that is reached depends on the initial strategy. An example of a trajectory is presented in figure 4(B). In equation (14), there is only one fixed point, \(x^*_3\) and \(y^*_4\), which is given by

\[
(x^*_3, y^*_4) = \left( \frac{P - S}{T - P}, \frac{T - 2P + S}{T - S} \right). 
\]

Then, the players’ expected payoffs, \(u^*_e\) and \(v^*_e\), are given by

\[
(u^*_e, v^*_e) = \left( \frac{T + S}{2}, P \right). 
\]

This linear stability analysis shows that this fixed point is neutrally stable, that is, the fixed point is not a focus but a center, as in the original Lotka–Volterra equation. Indeed, the time evolution of \((x_3(t), y_4(t))\) has a conserved quantity, given by

\[
H(x_3, y_4) = -(P - S)(\log x_3 + 2 \log y_4) + (T - P)x_3 - (T - S)y_4, 
\]

which is determined by the initial condition and preserved.

Furthermore, this Lotka–Volterra type oscillation provides an explanation of the exploitation we observed. The original Lotka–Volterra equation shows the prey–predator relationship, where the predator increases its own population by sacrificing the prey population. Here \(x_3\) (\(y_4\)) represents the exploiter’s defection (cooperation on the exploited side), which is a selfish (altruistic) action in the PD. Figure 4(B) shows that \(x_3\) is larger when \(x_4\) is larger. In other words, the exploiting side learns to use the selfish action with the altruistic action of the exploited. This result means that the exploiting one increases its own payoff at the expense of the exploited side. Thus, the oscillation of the exploitative relationship is interpreted as a prey–predator relationship.

### 4.2. Mechanism of one-sided exploitation: self-reference leads to generosity

In the previous sections, we mathematically showed how the exploitative relationship is maintained. Next, we intuitively interpret the strategies in equations (11) and (12), which implies that exploitation emerges between the narrow-minded and generous players. Here, we also focus on why the memory-one class is exploited one-sidedly by the reactive one.

Before analyzing equations (11) and (12), we present the well-known TFT and two related strategies in table 1. In the TFT strategy, the player deterministically responds with C to the opponent’s previous C, and with D to the opponent’s D. A more generous strategy [14] is that the player accepts the opponent’s D and probabilistically responds with C. In contrast, in a more narrow-minded strategy [36], the player betrays the opponent’s C probabilistically and responds with D. In contrast to the TFT strategy that was adopted to represent the emergence of symmetric cooperation, the generous and narrow-minded TFT strategies represent asymmetric exploitation. However, these strategies do not refer to previous self-actions. Here, we analyze the exploiting and exploited strategies with the generous and narrow-minded TFT, under the constraint that the previous actions of the self are C and D.

As seen in equation (11), the exploiting player uses the narrow-minded TFT strategy in both cases where their previous action was C and D. In other words, the strategy is characterized by \(x_1 > 0\) and \(x_2 = 0\) for a self C,
Table 1. Summary of TFT and related strategies. The TFT strategy deterministically responds with C (D) to the opponent’s C (D). Thus, if the player does not refer to their own action, the strategy is represented by a reactive class with \( x_{13} = 1 \) and \( x_{24} = 0 \). The generous TFT’s response to the opponent’s C is the same as the original TFT but probabilistically cooperates with the opponent’s D. This strategy is represented by \( x_{13} = 1 \) and \( x_{24} > 0 \). Finally, a narrow-minded TFT probabilistically defects the opponent’s C, which is different from the original TFT. This strategy is represented by \( x_{13} < 1 \) and \( x_{24} = 0 \).

| Strategy                   | Action to opponent’s C | Action to opponent’s D |
|----------------------------|-------------------------|-------------------------|
| TFT                        | Deterministic C         | Deterministic D         |
| Generous TFT               | Deterministic C         | Probabilistic C         |
| Narrow-minded TFT          | Probabilistic D         | Deterministic D         |

Figure 5. In the left panel, payoffs of all the equilibrium states between reactive classes are plotted excluding symmetry. The color map indicates the degree of exploitation. The black star indicates the average value. The right panel shows the payoffs when the exploiting player changes the class into a memory-one class and learns the strategy one-sidedly. The black stars indicate the change in the average payoff. Thus, the learning of the memory-one side leads to the benefit of the reactive opponent rather than for the memory-one itself.

and \( x_3 > 0 \) and \( x_4 = 0 \). Therefore, CC never occurs in exploitative equilibrium, where \( x_1 > 0 \) can be arbitrarily chosen. Thus, a player has the potential to use the exploiting strategy without referring to their own action, as \( x_{13} = x_1 = x_5 > 0 \) and \( x_{24} = x_2 = x_4 = 0 \). Similarly, the exploited player also uses the narrow-minded TFT for a previous self C, that is, \( y_1 > 0 \) and \( y_2 = 0 \). However, for a self D, the player uses the generous TFT, that is, \( y_3 = 1 \) and \( y_4 > 0 \). Thus, the exploited player refers to their own action: the player cannot take \( x_{13} = x_1 = x_3 \) and \( x_{24} = x_2 = x_4 \). Although this additional reference to self-action enriches the player’s choice of strategy, the player instead tends to be more generous to the opponent’s defection and accepts the opponent’s exploitation.

In the above, we consider only the equilibrium states given by equations (11) and (12). However, the question remains whether a memory-one class acquires generosity and accepts exploitation during the transient learning process. We attempt to answer this question in the following three steps: first, we classify three equilibrium states in the game between reactive classes: mutual defection, cooperation, and exploitation. Second, by assuming that one of the players adopt a memory-one class instead of the reactive one under these three equilibrium states, we discuss whether the player changes the strategy for each of the three equilibrium states. Third, we consider the one-sided learning process by the above memory-one class under the equilibrium states and examine if the memory-one side’s learning increases the opponent’s payoff by acquiring generosity.

Step 1. As shown in figure 3(C), there are three cases of equilibria in a match between reactive classes: mutual defection (yellow dots), mutual cooperation (blue dots), exploitation (purple and orange dots). Here, all possible equilibria for exploitation are shown in figure 5.

Step 2. Next, we assume that in each of the above states, one player adopts a memory-one class instead of a reactive one. Here, the player can refer to their own actions, after which the state can be unstable. First, for cases of mutual defection and cooperation, the equilibrium states are stable even if one player alternatively adopts a memory-one class. This is easily explained by the analytical result that all the equilibrium states between reactive classes are completely included in those among the memory-one and reactive classes. By contrast,
case of exploitation, the memory-one class receives a larger payoff by releasing constraints $x_1 = x_3 = x_{13}$ and $x_2 = x_4 = x_{24}$.

**Step 3.** Finally, we consider one-sided learning by a memory-one class under the exploitation case. Since we assume that the strategy of the reactive opponent is fixed, the memory-one side receives a larger payoff. For all states, the memory-one class releases the constraints $x_1 = x_3$ and $x_2 = x_4$ and learns to be $x_1 = x_2 = 0$ and $x_3 = x_4 = 1$, which represents an asymptotic relationship to the exploited strategy of equation (12). Then, the learning of the memory-one side increases the opponent’s benefit much more than the increase in one’s own benefit (see figure 5). This result shows that the memory-one class becomes generous toward the opponent’s defection by also referring to their own actions.

Thus, we have shown that memory-one class becomes generous for the opponent’s defection in equilibrium. If the learning feedback from the opponent reactive class is considered, the one-sided exploitation is generated as seen in figure 3(B).

### 5. Generality of the result

#### 5.1. Generality over different classes

We have demonstrated that the reference to own actions lead to generosity toward the opponent’s defection when comparing memory-one and reactive classes. Recall that the reactive class identifies both CC and DC, and CD and DD as the same. Therefore, we can consider two intermediate classes between the memory-one and reactive classes: the player only compresses either CC with DC or CD with DD. These strategies refer to the opponent’s action completely but refer to the self-action only when the opponent’s action is C or D. In this section, we study the learning dynamics of such extended classes.

Before analyzing the matches, we labeled these new classes according to figure 6. Here, we renamed the memory-one class as the ‘1234’ class because the class distinguishes all of CC (1), CD (2), DC (3), and DD (4), and has four strategy variables $x_1$, $x_2$, $x_3$, and $x_4$. Further, the reactive class uses two variables $x_{13}$ and $x_{24}$, as DC (3) and CC (1) represent one class, and DD (4) and CD (2) represent another. Thus, we renamed it as the ‘1212’ class. In the same way, the newly defined classes are renamed as ‘1214’ and ‘1232’; the former (latter) combines CC and DC (CD and DD). Among these four classes, complexity can be introduced as the degree to which the self-action is referred to. Figure 6(A) shows the ordering of this complexity. 1234 and 1212 are the most complex and simple of the four classes, respectively, whereas 1214 and 1232 lie between the two.
The outcomes of all possible 10 matches between any pairs of the 1234, 1232, 1214, and 1212 classes are shown in figure 6(B). As shown in these figures, 1212, which is the simplest class, can exploit 1212, 1232, and 1234. Class 1214 exploits 1232 and 1234. Class 1232 can exploit 1234. These exploitative relationships are summarized in the bottom right panel. Interestingly, the results show that the simpler classes generally exploit the more complex classes, but the reverse never occurs. Thus, the reference to self-action generally leads to generosity toward the opponent’s defection and accepts exploitation.

Theoretically, there are 15 types of classes for choosing strategies that use the information of the self and the opponent’s actions up to the previous turn (see appendix B for the formulation and appendix C for the results of matches). The ordering of complexity can be defined over these 15 classes. Although, the complexity among these 15 classes does not always indicate the degree of reference to self-action. Interestingly, the one-sided exploitation by a complex class of a simple one is not observed, except for a single case, i.e., the match between the 1232 and 1131 classes.

5.2. Generality over different payoff matrices
So far, we have studied the PD game with the standard score \((T, R, P, S) = (5, 3, 1, 0)\). The above results shows that one-sided exploitation by simple classes of complex classes is valid if the payoff satisfies \(T - R - P + S > 0\). This condition is called the ‘submodular’ PD \([48, 49]\); the summation of asymmetric payoffs, \(T + S\), is larger than the summation of symmetric payoffs, \(R + P\).

When the payoff matrix does not satisfy the submodularity, mutual learning does not generate an exploitative relationship. In all the matches among 1234, 1214, 1232, and 1212 classes, the players achieve either mutual cooperation or mutual defection, depending on the initial condition.

6. Summary and discussion
In this study, we investigated how players’ payoffs after learning depend on the complexity of their strategies, that is, the degree of reference to previous actions. By extending the coupled replicator model for learning, we formulated an adaptive optimization strategy by learning previous actions. By focusing on a reactive strategy in which the player refers only to the last action of the opponent and a memory-one class in which the player refers to both their own last action and that of the opponent, we uncovered that the latter, which has more information and includes the former, will be exploited by the former, independent of the initial state. Here, both the strategies of the latter (exploited) and former (exploiting) players permanently oscillate as in the prey and predator dynamics, whereas the exploit relationship is maintained. The exploiting (exploited) side uses the narrow-minded (generous) TFT when the previous self-action was defection.

Our formulation is applicable to other games like snowdrift \((R > T > S > P)\) and stag hunt \((R > T > P > S)\) games (see [50, 51] for the recent studies). In the one-shot play, these games have different Nash equilibria from the PD game. Indeed, the Nash equilibria are CD and DC in the snowdrift game, whereas CC and DD are in the stag hunt game. Then, it is an interesting future work to study how players’ possible choices of actions in equilibrium change depending on the players’ reference to information from the previous games. Furthermore, the application will be extended to other games which concern human morality [52].

Using this definition, a player using a memory-one class has a larger number of choices of strategy than that of the reactive class. The memory-one class includes extortion strategies [53–57], where the player has an advantage over their opponent using a fixed strategy, that is, the player receives a larger payoff than the opponent, independent of the opponent’s strategy. Given this, it is quite surprising that the reactive class unilaterally exploits the memory-one class after mutual learning, regardless of the initial strategy. The results show that even if there are possible advantages to choices of strategy, the player may not realize them through learning if the opponent’s strategy continues to change. Here, we demonstrated that learning with reference to self-action makes the player generous toward an opponent’s defection, as the unknown way to acquire the generosity [55, 58–60]. In this way, learning to obtain a higher payoff with more information counterintuitively results in a poorer payoff than the opponent, who learns with limited information.

It is common for a player to change their next choice depending on past choices. As already seen in reactive and memory-one classes, it is common for a player to change the next choice of behavior depending on their own or opponent’s choice. As briefly mentioned in section 5.1, our formulation can be extended to reference arbitrary information. For instance, we can assume a memory-\(n\) strategy, which refers to actions in more than one previous round. Even the two-memory strategy is quite different from the memory-one strategy: the player can use a greater variety of strategies, such as tit-for-2-tat [61]. It has been discussed that a reference to multi-memory generates cooperation more efficiently [23]. Our model could be extended to study whether this is true under mutual learning, or whether the player with more information would exploit their opponent or be exploited.
Game theory is often relevant in explaining characteristic human behaviors. The advantage of the TFT strategy indicates poetic justice in human nature. However, humans also reflect on their past behavior. For instance, they could be motivated to perform beneficial actions toward others after betraying others. This study supports how such behavior emerges or is preserved through learning. This, however, provides room for exploitation. Indeed, equation (12) shows that the player with reference to their own previous actions becomes generous toward the opponent’s defection after the player defected in the previous round. Ironically, this can be exploited.

Acknowledgments

The authors would like to thank E Akiyama, and H Ohtsuki for useful discussions. This research was partially supported by JPSJ KAKENHI Grant Numbers JP18J13333, JP21J01393, and Grant-in-Aid for Scientific Research on Innovative Areas (17H06386) from the Ministry of Education, Culture, Sports, Science, and Technology of Japan.

Data availability statement

All data that support the findings of this study are included within the article (and any supplementary files).

Appendix A. Proof of coupled replicator in learning dynamics

In this section, we prove that equations (6) and (7) are equivalent. First, because $p_e$ is the equilibrium state for the Markov transition matrix $M$, we obtain

$$\begin{align*}
(E - M)p_e &= 0, 
\end{align*}$$

Here, by a perturbation of strategy $\delta x_1$, the equilibrium state $\delta p_e$ changes accordingly. By substituting $x_1 \rightarrow x_1 + \delta x_1$ and $p_e \rightarrow p_e + \delta p_e$ into equation (1), we obtain:

$$\begin{align*}
(E - M)\delta p_e - p_{CC}(p_{CC(C)} - p_{CC(D)})\delta x_1 &= 0, \\
\Rightarrow \sum_{t=0}^{\infty} M'(E - M)\frac{\partial p_e}{\partial x_1} &= p_{CC}\sum_{t=0}^{\infty} M'(p_{CC(C)} - p_{CC(D)}), \\
\Rightarrow \left( E - \lim_{t \rightarrow \infty} M't \right) \frac{\partial p_e}{\partial x_1} &= p_{CC}\sum_{t=0}^{\infty} M'(p_{CC(C)} - p_{CC(D)}),
\end{align*}$$

Here, we use $\lim_{t \rightarrow \infty} M't \frac{\partial p_e}{\partial x_1} = 0$ because $M$ has only one eigenvector for which the eigenvalue is 1 as the preservation of probability (i.e., $p_{CC} + p_{CD} + p_{DC} + p_{DD} = 1$).

Equation (19) not only provides a simple representation of the time evolution but is also useful for the numerical simulation of equation (6). The right-hand side of equation (19) requires an approximate numerical calculation for all eigenvectors of $M$. By contrast, the left-hand side demands only the information on the equilibrium state $p_e$, which is analytically given by equation (2).

Appendix B. General formulation of classes

In section 1 of our main manuscript, we defined the previous memory-one class and reactive class. Moreover, in section 4, we renamed these classes as 1234 and 1212 ones, and additionally defined 1214 and 1232 ones. In general, however, there are 15 classes in total when the player can refer to only the previous one game at the maximum. This section provides the general extension of such classes.

Recall that there are four kinds of results in a single game, CC, CD, DC, and DD, as the left and right indices represents the actions of self and other. Then, we assigned numbers to these pairs of actions from 1 to 4, respectively. Then, we assign a number 1234 to the memory-one class, because it distinguishes all of 1 (CC), 2 (CD), 3 (DC), and 4 (DD) and can cooperate in different probabilities depending on the observed action. Next, we consider the reactive class. This reactive class only refers to the other’s action, so the player cannot
Figure 7. Schematic diagram of possible 15 classes and the complexity among them. Each colored bar represents one class of strategies, which consists of four sets of actions, i.e., CC, CD, DC, and DD. When the player reflects the same action to different sets of actions, the sets are painted in the same color. Red, yellow, green and purple indicate numbers 1, 2, 3, and 4. Solid black lines represent the order of complexity between the connected bars. The upper bar is more complex than the connected lower bar.

distinguish 3 (DC) with 1 (CC) and 4 (DD) with 2 (CD). When we compress multiple pieces of information, we replace ones with the larger numbers (i.e., 3 and 4) as ones with the smaller numbers (i.e., 1 and 2). Here, 3 and 4 are replaced by 1 and 2 respectively, so that the reactive class is coded as 1212. In the same way, we can define 1134, 1214, 1231, 1224, 1232, 1233, 1114, 1131, 1133, 1211, 1222, and 1221 classes in all. Figure 7 shows the schematic diagram of these 15 classes.

In addition, recall the definition of complexity of class. When a class includes all the strategies of another class, the former one is defined to be more complex than the latter one. Figure 7 also shows all the relationships of complexity among the classes.

Appendix C. Mutual learning in general classes of strategies

Before analyzing the games of these 15 classes, we omit 1133 and 1111 classes from the analysis. These two classes do not refer to the other’s action. In other words, they do not distinguish X1C with X2D for all X1, X2 ∈ {C, D}^2. From the rule of PD, the other player learns to choose D deterministically against 1133 or 1111 classes. Thus, they have no other equilibrium than DD.

Figure 8 shows the equilibrium of mutual learning for still existing 13 classes. From this figure, we see a variety of equilibria which include various degrees of cooperation and exploitation. Here, note that there are several kinds of oscillation in the payoffs of both the players, as similar to the case of 1234 v. 1234 in the main manuscript. Figure 8 shows that the same oscillation is frequently seen in the game of 1234, 1134, 1214, 1231, 1232, 1131, and 1212. Another oscillatory state is seen in the case of 1232 v.s. 1131 on the upper right triangle. All the other states than these two types of oscillation exist as the fixed point in equilibrium of learning.

Then, we give several remarks on computational methods on mutual learning. First, we gives a constant bound on the stochastic strategies \( \epsilon \leq x_i \leq 1 - \epsilon \) with \( \epsilon = 10^{-4} \) in the computation of learning. This is to avoid false convergences into CC (i.e., \( u_e = v_e = 3 \)) if CC is saddle point. Second, we remove equilibria on \( u_e = 3, 1 < v_e < 3 \) or vise versa on several panels. This is because these equilibria only exist on the condition of \( R = (T + P)/2 \).

Figure 9 gives a statistical analysis corresponding to figure8. The figure (A) shows the payoff of each class obtained statistically from the numerous ensembles. In principle, a learning player receives at least \( P(=1) \) amount of payoff in equilibrium. Thus, the difference from such minimal payoff 1 represents the class’s surplus benefit. Interestingly, we see that besides previous 1234 (i.e., memory-one) and 1212 (i.e., reactive) classes, 1232 and 1131 classes achieve high scores in statistics. The figure (B) shows the difference of payoffs between the two players. Interestingly, no exploitation from the simpler class to the more complex one is seen with an exception of 1232 vs 1131.
Figure 8. The possible equilibrium states in all the pairs of 13 classes of strategies. As the combination of 13 classes, there are $13 + 13 \times 12/2 = 91$ panels of equilibrium states. In each panel, X (Y) axis indicates the equilibrium payoff of vertical (horizontal) class. Each panel has 10,000 samples of equilibrium states in mutual learning with payoff $\left(T, R, P, S\right) = (5, 3, 1, 0)$, even though some of equilibria are not plotted.

Figure 9. Statistical results of figure 8. Panel (A) shows the payoff of class of horizontal axis number in the ensemble of games against the class of vertical axis. The darker color represents that the class of vertical axis gains a larger payoff. Panel (B) shows the difference in payoff from the horizontal class to the vertical one. The blue (red) color means that the class of horizontal axis exploits (is exploited from) that of vertical axis.

ORCID iDs

Yuma Fujimoto  
https://orcid.org/0000-0002-5436-3363

References

[1] von Neumann J and Morgenstern O 2007 Theory of Games and Economic Behavior commemorative edn (Princeton, NJ: Princeton University Press)
[2] Premack D and Woodruff G 1978 Does the chimpanzee have a theory of mind? Behav. Brain Sci. 1 515–26
[3] Saxe R and Baron-Cohen S (ed) 2007 Theory of Mind: A Special Issue of Social Neuroscience (London: Psychology Press)
[4] Lurz R W 2011 Mindreading Animals: The Debate over what Animals Know about Other Minds (Cambridge, MA: MIT Press)
[5] Han The Anh T A, Moniz Pereira L and Santos F C 2011 Intention recognition promotes the emergence of cooperation Adapt. Behav. 19 264–79
[6] Han T A et al 2015 Synergy between intention recognition and commitments in cooperation dilemmas Sci. Rep. 5 1–7
[7] Moniz P L, Santos F C and Lenaerts T 2015 Emergence of cooperation via intention recognition, commitment and apology—a research summary AI Commun. 28 709–15
[8] Fujimoto Y and Kaneko K 2019 Functional dynamic by intention recognition in iterated games New J. Phys. 21 023025
[9] Axelrod R 1984 The Evolution of Cooperation (US: Basic Books)
[10] Axelrod R and Hamilton W D 1981 The evolution of cooperation science 211 1390–6
[11] Axelrod R and Dion D 1988 The further evolution of cooperation Science 242 1385–90
[12] Nowak M 1990 Stochastic strategies in the prisoner’s dilemma Theor. Population Biol. 38 93–112
[13] Nowak M and Sigmund K 1990 The evolution of stochastic strategies in the prisoner’s dilemma Acta Appl. Math. 20 247–65
[14] Nowak M A and Sigmund K 1992 Tit for tat in heterogeneous populations Nature 355 250–3
[15] Imhof L A and Nowak M A 2010 Stochastic evolutionary dynamics of direct reciprocity Proc. R. Soc. B. 277 463–8
[16] Zhang J, Zhang G, Chu T and Perc M 2011 Resolution of the stochastic strategy spatial prisoner’s dilemma by means of particle swarm optimization PLoS One 6 e12178
[17] Baek S K et al 2016 Comparing reactive and memory–one strategies of direct reciprocity Sci. Rep. 6 1–13
[18] Nowak M and Sigmund K 1993 A strategy of win–stay, lose–shift that outperforms tit-for-tat in the prisoner’s dilemma game Nature 364 56–8
[19] Brauchli K, Killingsback T and Doebeli M 1999 Evolution of cooperation in spatially structured populations J. Theor. Biol. 200 405–17
[20] Kraines D P and Kraines V Y 2000 Natural selection of memory–one strategies for the iterated prisoner’s dilemma J. Theor. Biol. 203 335–55
[21] Iliopoulos D, Hintze A and Adami C 2010 Critical dynamics in the evolution of stochastic strategies for the iterated prisoner’s dilemma PLoS Comput. Biol. 6 e1000948
[22] Stewart A J and Plotkin J B 2014 Collapse of cooperation in evolving games Proc. Natl Acad. Sci. USA 111 17558–63
[23] Hölle C, Martínez-Vaquero I L, Chatterjee K and Nowak M A 2017 Memory–n strategies of direct reciprocity Proc. Natl Acad. Sci. USA 114 4715–20
[24] Szolnoki A and Perc M 2009 Promoting cooperation in social dilemmas via simple coevolutionary rules Eur. Phys. J. B 67 337–44
[25] Perc M and Szolnoki A 2010 Coevolutionary games–A mini review BioSystems 99 109–25
[26] Traulsen A and Nowak M A 2006 Evolution of cooperation by multilevel selection Proc. Natl Acad. Sci. 103 10952–5
[27] Wang Z, Szolnoki A and Perc M 2013 Interdependent network reciprocity in evolutionary games Sci. Rep. 3 1–7
[28] Jin Q, Wang L, Xia C-Y and Wang Z 2014 Spontaneous symmetry breaking in interdependent networked game Sci. Rep. 4 4095
[29] Xia C, Li X, Wang Z and Perc M 2018 Doubly effects of information sharing on interdependent network reciprocity New J. Phys. 20 075005
[30] Liu Y, Yang C, Huang K and Wang Z 2019 Swarm intelligence inspired cooperation promotion and symmetry breaking in interdependent networked game Chaos 29 043101
[31] Takesue H 2021 Symmetry breaking in the prisoner’s dilemma on two-layer dynamic multiplex networks Appl. Math. Comput. 388 125545
[32] Macy M W 1991 Learning to cooperate: stochastic and tacit collusion in social exchange J. Econ. Theory 54 489–512
[33] Macy M W and Skvoretz J 1998 The evolution of trust and cooperation between strangers: a computational model Acta Sociol. Rev. 38 63 638–60
[34] Macy M W and Flache A 2002 Learning dynamics in social dilemmas Proc. Natl Acad. Sci. 99 7214–20
[35] Fujimoto Y and Kaneko K 2019 Emergence of exploitation as symmetry breaking in iterated prisoner’s dilemma Phys. Rev. Res. 1 033077
[36] Sandholm T W and Crites R H 1996 Multistage reinforcement learning in the iterated prisoner’s dilemma Biosystems 37 147–66
[37] Tajti M and Ikegami T 1999 Dynamics of internal models in game players Physica D 134 253–66
[38] Masuda N and Ohtsuki H 2009 A theoretical analysis of temporal difference learning in the iterated prisoner’s dilemma game Bull. Math. Biol. 71 1818–50
[39] Hofbauer J and Sigmund K 1998 Evolutionary Games and Population Dynamics (Cambridge: Cambridge University Press)
[40] Borgers T and Sarin R 1997 Learning through reinforcement and replicator dynamics J. Econ. Theory 77 1–41
[41] Sato Y, Akiyama E and Farmer J D 2002 Chaos in learning a simple two-person game Proc. Natl Acad. Sci. 99 4748–51
[42] Sato Y and Crucifix T P 2003 Coupled replicator equations for the dynamics of learning in multistage games Phys. Rev. E 67 015206
[43] Posch M 1999 Win–Stay, lose–shift strategies for repeated games-memory length, aspiration levels and noise J. Theor. Biol. 198 183–95
[44] Imhof L A, Fudenberg D and Nowak M A 2007 Tit-for-tat or win–stay, lose–shift? J. Theor. Biol. 247 574–80
[45] Amaral M A et al 2016 Stochastic win–stay, lose–shift strategy with dynamic aspirations in evolutionary social dilemmas Phys. Rev. E 94 032317
[46] Nash J F 1950 Equilibrium points in n-person games Proc. Natl Acad. Sci. 36 48–9
[47] Dixit A 2003 On modes of economic governance Econometrica 71 449–81
[48] Takahashi S 2010 Community enforcement when players observe partners’ past play J. Econ. Theory 145 42–62
[49] Iyer S and Killingback T 2016 Evolution of cooperation in social dilemmas on complex networks PLoS Comput. Biol. 12 e1004779
[50] Iyer S and Killingback T 2020 Evolution of cooperation in social dilemmas with assortative interactions Games 11 41
[51] Capraro V and Perc M 2021 Mathematical foundations of moral preferences J. R. Soc. Interface 18 20200880
[52] Press W H and Dyson F J 2012 Iterated prisoner’s dilemma contains strategies that dominate any evolutionary opponent Proc. Natl Acad. Sci. 109 10409–13
[53] Hölle C, Nowak M A and Sigmund K 2013 Evolution of extortion in iterated prisoner’s dilemma games Proc. Natl Acad. Sci. 110 6913–8
[54] Stewart A J and Plotkin J B 2013 From extortion to generosity, evolution in the iterated prisoner’s dilemma Proc. Natl Acad. Sci. 110 15348–53
[55] Adami C and Hintze A 2013 Evolutionary instability of zero-determinant strategies demonstrates that winning is not everything Nat. Commun. 4 1–8
[57] Szolnoki A and Perc M 2014 Evolution of extortion in structured populations Phys. Rev. E 89 022804
[58] Delton A W, Krasnow M M, Cosmides L and Tooby J 2011 Evolution of direct reciprocity under uncertainty can explain human generosity in one-shot encounters Proc. Natl Acad. Sci. USA 108 13335–40
[59] Feng X, Zhang Y and Wang L 2017 Evolution of stinginess and generosity in finite populations J. Theor. Biol. 421 71–80
[60] Kurokawa S, Wakano J Y and Ihara Y 2018 Evolution of groupwise cooperation: generosity, paradoxical behavior, and non-linear payoff functions Games 9 100
[61] Boyd R and Lorberbaum J P 1987 No pure strategy is evolutionarily stable in the repeated prisoner’s dilemma game Nature 327 58–9