Reinforcement Learning Explains Conditional Cooperation and Its Moody Cousin

FRANCOIS Nathan
000514241
ULB
nathan.francois@ulb.be

MANDEL Eliot
000493519
ULB
mandel.eliot@ulb.be

YILDIRIM Emirhan
000459800
ULB
emirhan.yildirim@ulb.be

Abstract

In two-player social dilemmas, cooperation is mostly supported by direct reciprocity, meaning repeated interactions. But when there are many players, people often behave differently. A lot of experiments show that participants do not follow one fixed strategy. Instead, they often use conditional cooperation: they are more likely to cooperate when others cooperated in the previous round. Another important pattern observed is moody conditional cooperation (MCC), where a player's response depends not only on what others did, but also on what the player did in the previous round. The goal of this project was to reproduce the main results of Ezaki et al. (2016)[1], which argues that these patterns can emerge from a simple reinforcement learning rule, without assuming conditional cooperation from the start.

1 Introduction

In situations involving many players, which better represent how a society works, observed behaviours become more complex than in two-player situations and cannot be explained by simple mechanisms.

Many experiments have shown that, in multi-player social dilemmas, individuals tend to adapt their behaviour to their environment. One of the most frequent patterns is conditional cooperation, where one tends to cooperate if others cooperated in the previous round. Another important observed pattern is the role of mood, called moody conditional cooperation, where the reaction of a player also depends on their own action in the previous round.

A lot of experiments report conditional cooperation and its moody version but the mechanisms behind these behaviors are still not fully understood[1]. Many behavioral models describe these patterns by assuming conditional decision rules, where players explicitly observe their neighbors and adjust their behavior. An important question is whether these behaviors need to be explicitly defined as strategies or decision rules, or whether they can emerge from simpler and more general learning mechanisms.

It is from this that the work of Ezaki et al. (2016)[1] propose another explanation based on reinforcement learning where each individual does not know what others have chosen (cooperate or defect). They introduce a simple model based on Bush-Mosteller rule, using satisfaction based on an asipration level. Players do not need to know what others did on previous round. Instead, they ajust their probability to cooperate based on the payoff obtained in the previous round and their own last action. When the gain is greater than the aspiration, the action is reinforced, otherwise it's weakened.

An important feature is that the influence of the player's previous action appears naturally through the learning dynamics. As a result, behaviors like conditional cooperation and moody conditional cooperation can emerge without being explicitly defined in the decision rule.

To test this, Ezaki et al. study several standard social dilemma settings. In this case, they use the Prisoner's Dilemma Game (PDG) on a spatial network and the Public Goods Game (PGG) in groups. The results are shown using time evolution curves, heatmaps, and simple linear summaries.

This work shows that a simple reinforcement learning model, with a fixed aspiration level, can already explain complex behaviors seen in human experiments. It suggests that we do not need complex network structures or advanced decision rules to explain conditional cooperation. Instead, these behaviors can appear naturally from simple learning dynamics.

In this report, we reproduce the main results of Ezaki et al. using numerical simulations. We explain how the models and environments are implemented, describe the main practical choices we made, and compare our results with the figures from the original paper to check whether the same conclusions are obtained.

2 Related Work

Horita et al. (2017)[2] study human behavior in repeated social dilemma games and compare several behavioral models. They test conditional cooperation (CC), moody conditional cooperation (MCC), and two reinforcement learning models: Bush-Mosteller and Roth-Erev. What they found is that those reinforcement learning models actually fit the human data pretty well. Sometimes even better than the straight MCC model. Different from our reproduced paper(Ezaki et al., 2016), Horita et al. focus on fitting these models right to the actual human data. But Ezaki et al.[1] uses simulations mostly. With aspiration-based reinforcement learning to show how patterns like CC and MCC can emerge.

Grujić et al. (2014)[3] analyze three spatial Prisoner's Dilemma experiments. They use the concept of moody conditional cooperation (MCC): players cooperate more when more neighbors cooperated in the previous round, and the probability to cooperate also depends on the player's own previous action (mood). They report that MCC is observed across the three experiments. Different from our reproduced paper (Ezaki et al., 2016), this work is mainly empirical and statistical: it describes and tests behavioral patterns in experiments. In contrast, Ezaki et al. propose a reinforcement learning mechanism (aspiration-based Bush-Mosteller) and show by simulation that the mechanism can generate CC and MCC without explicitly using neighbors' actions.

3 Practical aspects of the reproduction

In this section, we explain how we organized the implementation (the paper does not provide ready-to-run code), what choices we had to make when some details were not fully given.

3.1 General implementation plan

The original paper does not provide an official code repository, so we implemented the full pipeline ourselves: environments, learning agents, simulation loop, and analysis scripts. During development, we first built a minimal NumPy prototype to check the logic quickly. After that, we moved to a PettingZoo-based implementation to keep a clean multi-agent API and to make it easier to extend to the next game (PGG).

Our code is organized in small modules and we provide notebooks/scripts to reproduce each figure.

3.2 Learning rule and stimulus

File BMmodel.py contains the learning equations from the paper. We implemented: (i) the stimulus function (Eq. (2))

$$s = \tanh\left(\beta(r - A)\right), \tag{1}$$

and (ii) the Bush-Mosteller update rule for the cooperation probability p_t (Eq. (1)) [1].

After receiving reward r, the agent computes the stimulus where A is the aspiration level and $\beta > 0$ controls sensitivity [1]. Then the cooperation probability p_t is updated using the Bush-Mosteller rule (four cases depending on the previous action and the sign of s_{t-1}) [1].

We also implemented the *misimplementation* noise. The paper converts the intended probability p_t into an effective probability $\tilde{p}_t$ because the player may execute the opposite action with probability ϵ. In practice we use

$$\tilde{p}_t = p_t(1 - \epsilon) + (1 - p_t)\epsilon. \tag{2}$$

Unless stated otherwise, we use $\epsilon = 0.2$.

3.3 Agent wrapper around the BM model

File agents.py defines the behavior of one learner. Each agent stores its current internal probability p_t (and the last action). At each round, the agent: (1) samples an action (cooperate/defect) using $\tilde{p}_t$, (2) receives the reward from the environment, (3) updates p_t using the BM rule. In other words, agents increase the probability of repeating actions that led to satisfaction and decrease it after dissatisfaction.

3.4 PDG Reproduction

The pdg_env.py file implements the Prisoner's Dilemma Game on a 10×10 grid with periodic boundary conditions, exactly like in the original paper. Each agent always has four neighbors and, in a given round, plays the same action against all of them. We use the same payoff matrix ($T = 5$, $R = 3$, $P = 1$, $S = 0$). Since PettingZoo does not provide a ready-made environment for this kind of lattice-based interaction, we explicitly build the neighbor lists ourselves to reflect the grid structure.

At every round, the environment computes the pairwise payoffs between each agent and its four neighbors using the standard Prisoner's Dilemma payoff matrix. The reward assigned to an agent is simply the average of these four interactions. Using PettingZoo's parallel API was a natural fit because all agents act at the same time, and learning updates only happen once all rewards have been calculated. This avoids any ambiguity about action order and stays faithful to the paper [1].

3.5 Running simulations

File loop_pz.py contains the simulation loop. It runs multiple independent trials. In each trial, we iterate for $t_{\max}$ rounds. At each round, we collect actions from all agents, step the environment, store rewards, and call the agents' update.

A practical limitation is computation time. The paper reports large numbers of simulations (e.g., 10^3 simulations for some fitted quantities). Full runs are expensive. We therefore used a smaller number of trials for debugging and intermediate plots, and increased the number when generating the final figures.

3.6 Measures and figure reproduction

File analysis_PDG.py computes the measures and the quantities needed for Fig. 2-4[1].

Fig. 2A and Fig. 2B.. For Fig. 2A, we compute the cooperation fraction over time and plot mean and standard deviation across trials [1]. For Fig. 2B, we scan a grid of (A, β) values and compute mean cooperation over the first $t_{\max} = 25$ rounds, averaged over players and trials [1].

Fig. 3 (CC and MCC curves). For Fig. 3, we compute $f_C(t-1)$ for each player: it is the fraction of cooperating neighbors in the previous round, so it takes values $0, 0.25, 0.5, 0.75, 1$. We then estimate $P(C_t \mid f_C(t-1))$ for three cases: not conditioned, conditioned on $a_{t-1} = C$, and conditioned on $a_{t-1} = D$ [1].

Fig. 4 (least-squares fit and heatmaps). For Fig. 4, the paper summarizes the relationship between cooperation and $f_C(t-1)$ by a least-squares linear fit: $\tilde{p}_t \approx \alpha_1 f_C + \alpha_2$ [1].

3.7 PGG reproduction

For the Public Goods Game (PGG), we kept the same aspiration-based Bush–Mosteller learning model as in the PDG, but with some adjustments since actions are no longer binary. In this setting, each agent chooses a continuous contribution $a_t \in [0, 1]$, which meant rethinking the agent interface. Instead of selecting a simple cooperate or defect action, agents now sample a real-valued contribution from a truncated Gaussian distribution centered on their internal state variable p_t, exactly as described in the paper.

To apply the Bush–Mosteller update rule, these continuous contributions are then binarized using a threshold X. Contributions above X are interpreted as cooperation, while those below are treated as defection. This step may look a bit artificial at first, but it allowed us to reuse the same learning rule while staying consistent with the modeling choices made in the original study.

The PGG environment is implemented in pgg_env.py. It models a well-mixed group of four players, where each agent starts with an endowment of one unit and decides how much to contribute to a common pool. The total contribution is then multiplied by a fixed factor and redistributed equally among the players, which directly determines the rewards. Unlike in the PDG case, there is no underlying network structure which mean that agents do not have neighbors on a grid but are instead in a group of 4. As a result, the relevant reference quantity for conditional cooperation is simply the average contribution of the other members of the group.

Simulations for the PGG are run using loop_pgg.py. The overall structure of this loop is very similar to the one used for the PDG, but it has to deal with continuous actions instead of binary ones. For this reason, we store both the raw contributions and their binarized versions, which are needed for learning updates as well as for later analysis. The loop also computes group-level statistics specific to the PGG, such as the mean contribution of the other players in the group at each round.

Finally, analysis_PGG.py is used to reproduce the CC and MCC analyses for the Public Goods Game (also similar to analysis_PDG.py). Here, conditioning is no longer done on the fraction of cooperating neighbors, but on the average contribution of the other group members in the previous round, following the methodology of the paper. Linear fits are computed in exactly the same way as in the PDG case, which makes it easier to directly compare CC and MCC patterns across the two games.

4 Methods

This section summarizes the methods used to reproduce the main results of the target paper [1].

4.1 Game settings (PDG and PGG)

The paper studies repeated social dilemma games played by many agents interacting on a structured population [1]. In our reproduction, we implemented two environments:

PDG (Prisoner's Dilemma Game). Players are placed on a 10×10 square grid ($N = 100$) with periodic boundary conditions. Each player interacts with its four neighbors (up, down, left, right). At each round t, each player chooses one binary action: cooperate (C) or defect (D). The chosen action is applied to all neighbor interactions in that round. Pairwise payoffs follow the matrix used in the paper: $R = 3$ (C,C), $P = 1$ (D,D), $S = 0$ (C,D for the cooperator), and $T = 5$ (D,C for the defector) [1]. The reward of a player in one round is the average payoff over the four neighbor interactions.

PGG (Public Goods Game). Only 4 players interact with each other. Each round, calculate a probability p using the BM algorithm (instead of seeing if the previous action was C or D, it defines this action based on a threshold X). Then, the amount bet by the player is randomly selected according to a Gaussian distribution $N(p, \sigma^2)$. The initial bet is between 0 and 1 for each player. Instead of a payoff matrix, the total is multiplied by 1.6 and redistributed equally among each player. The action was labelised has a C or D depending of the treshold X.

4.2 Simulation protocol

Unless stated otherwise, we run repeated games for $t_{\max} = 25$ rounds, following the default setting in the paper [1]. For some results (e.g., the PDG time-course plot), we run longer horizons (up to 100 rounds) to match the corresponding figure in the paper [1]. We run multiple independent trials (different random seeds) for each parameter setting to obtain averages and variability estimates.

For each parameter configuration, we run multiple independent trials with different random seeds. This is mainly to account for the inherent stochasticity coming from action sampling, misimplementation noise, and the learning dynamics themselves. Reported quantities are then averaged over agents, rounds (when relevant), and trials. We also compute the standard deviation across trials.

To reproduce the parameter-dependent results, we scan grids of values for the aspiration level A and the sensitivity parameter β, following the same ranges as in the paper. In the PGG case, we also vary the contribution threshold X used to binarize continuous actions. Each parameter combination is simulated independently and processed through the same analysis pipeline, which makes it easier to directly compare different settings, as well as the PDG and PGG results.

4.3 Measures and reproduced figures

We compute summary statistics from the trajectories (actions and rewards) to reproduce the figures in the paper [1].

Cooperation level over time (Fig. 2A)[1]. We compute the fraction of cooperators at each round, and report the mean and standard deviation across trials. As shown in Fig. 1

Mean cooperation heatmaps (Fig. 2B)[1]. For a grid of (A, β) values, we compute mean cooperation over the first $t_{\max} = 25$ rounds (averaged over players and trials) and display results as a heatmap, as shown in Fig. 2

Conditional cooperation and MCC (Fig. 3)[1]. For each agent i at round $t \geq 1$, we compute the fraction of cooperative neighbors in the previous round, $f_C^{(i)}(t-1)$. With four neighbors, $f_C^{(i)}(t-1) \in \{0, 0.25, 0.5, 0.75, 1\}$. We then estimate $P(C_t \mid f_C(t-1))$ and the moody versions conditioned on the agent's previous action ($a_{t-1} = C$ or $a_{t-1} = D$), as shown in Fig. 3

Linear fit summary (Fig. 4). Following the paper, we summarize the relationship between cooperation and neighbor cooperation by a least-squares linear fit,

$$P(C_t) \approx \alpha_1 f_C(t-1) + \alpha_2, \tag{3}$$

computed for each parameter pair (A, β) [1]. In practice, we build a dataset of pairs (x, y) where $x = f_C(t-1)$ (fraction of cooperative neighbors in the previous round) and $y \in \{0, 1\}$ is the agent's action at round t. We then estimate (α_1, α_2) by a standard least-squares line fit. At first, we tried a simple way to estimate (α_1, α_2) using aggregated statistics. The idea is that, for a linear relation

$y \approx \alpha_1 x + \alpha_2$, we can write:

$$\alpha_1 = \frac{\mathrm{Cov}(x, y)}{\mathrm{Var}(x)}, \qquad \alpha_2 = \bar{y} - \alpha_1 \bar{x}. \tag{4}$$

In practice, this first implementation was too sensitive and did not give stable or satisfying results/heatmaps. So we switched to $\mathrm{numpy.polyfit(x,y,1)}$, which directly computes the least-squares line and is very easy to use. We compute the fit separately for unconditioned samples, samples with $a_{t-1} = C$, and samples with $a_{t-1} = D$, and we report heatmaps of α_1 and $\Delta\alpha_2 = \alpha_2(a_{t-1} = C) - \alpha_2(a_{t-1} = D)$.

PGG: continuous $f_C(t-1)$ and discretization for plots. In the PGG, actions are continuous ($a_t \in [0, 1]$), so the conditioning variable $f_C(t-1)$ is also continuous. Unlike the PDG case (where $f_C(t-1)$ takes only five values because there are four neighbors), we cannot plot $P(C_t \mid f_C(t-1))$ using a small set of exact values. Therefore, for the CC/MCC curves, we discretize $f_C(t-1)$ into 12 reference points between 0 and 1 and assign each observed value to the closest reference point see Fig. 10. For the moody version, we split samples according to the player's previous action using a threshold X, following the idea in the paper [1]. Finally, for the linear summary, we estimate $a_t \approx \alpha_1 f_C + \alpha_2$ using a least-squares fit ($\mathrm{numpy.polyfit}$) for each conditioning case.

4.4 Use of LLM tools

LLM were used as help for plotting utilities (matplotlib) and to find existing functions in NumPy for linear fitting (e.g., $\mathrm{numpy.polyfit(x,y,1)}$). All modeling decisions, implementations, simulations, and interpretations were carried out by the authors.

5 Results

5.1 Reproduction of PDG results

5.1.1 *Time evolution of cooperation.*

We first reproduce the time evolution of cooperation in the Prisoner's Dilemma Game, corresponding to Fig. 2A of the original paper. To do so, we fixed the parameters to $\beta = 0.2$, $\epsilon = 0.2$, and gave two aspiration levels, $A = 0.5$ and $A = 1.5$, following the reference setup. We measure at each round the fraction of cooperating agents and report the mean and standard deviation across independent trials (Figure 1).

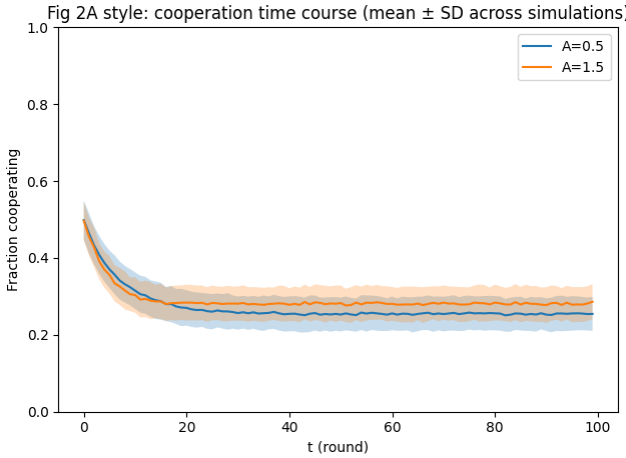


Figure 1: Time evolution of cooperation in the PDG. The figure shows the mean fraction of cooperating agents as a function of time, averaged over simulations. Shaded areas indicate one standard deviation. Results are shown for two aspiration levels, $A = 0.5$ and $A = 1.5$, with $\beta = 0.2$ and $\epsilon = 0.2$.

The cooperation for both aspiration levels drops quickly during the first few rounds and then settles into a stable regime. This initial drop in cooperation can be explained by the learning dynamics of the model since agents start with an initial cooperation probability of $p_1 = 0.5$ and quickly change their behavior when cooperation does not meet their aspiration level. After this short transient phase , the system appears to settle into a steady regime in which cooperation oscillates around a roughly constant level.

Our results are similar with those in the paper as we observe the same quick decrease in cooperation in the first rounds followed by a convergence toward a plateau. We note that the long-term cooperation level is slightly higher for $A = 1.5$ than for $A = 0.5$ which makes sense because higher aspiration levels tend to keep agents more reactive and less likely to settle into purely defective behavior.

We also have observed a slightly larger variance across trials during the early rounds compared to the original figure. This is likely due to our finite number of runs and to stochastic effects as we picked our own seed. But despite these differences, the overall temporal pattern and qualitative behavior are well reproduced.

5.1.2 *Mean cooperation as a function of A and β.*

We have reproduced the dependence of the mean cooperation level on the aspiration parameter A and the sensitivity parameter β, corresponding to the figure Fig. 2B of the original paper. We can see several patterns as the cooperation tends to increase with β, indicating that agents who react more strongly to satisfaction or dissatisfaction reinforce successful behaviors more efficiently, and cooperation is relatively high for low aspiration levels ($A \lesssim 0$), where agents are easily satisfied and therefore tend to maintain their current behavior.

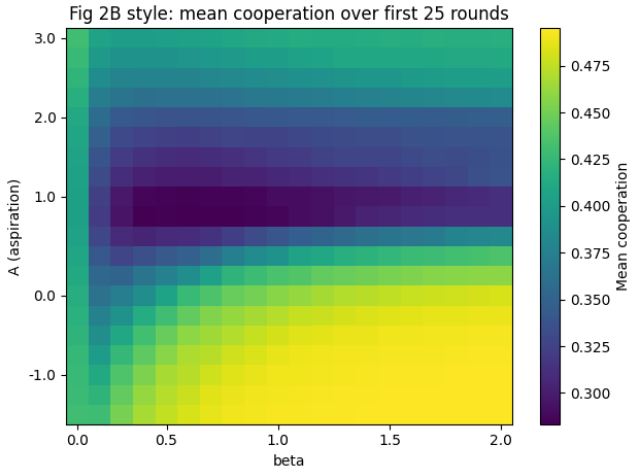


Figure 2: Mean cooperation level in the PDG as a function of the aspiration level A and the sensitivity parameter β. Cooperation is averaged over the first $t_{\max} = 25$ rounds, over all players, and over independent trials. The color scale indicates the mean fraction of cooperating agents.

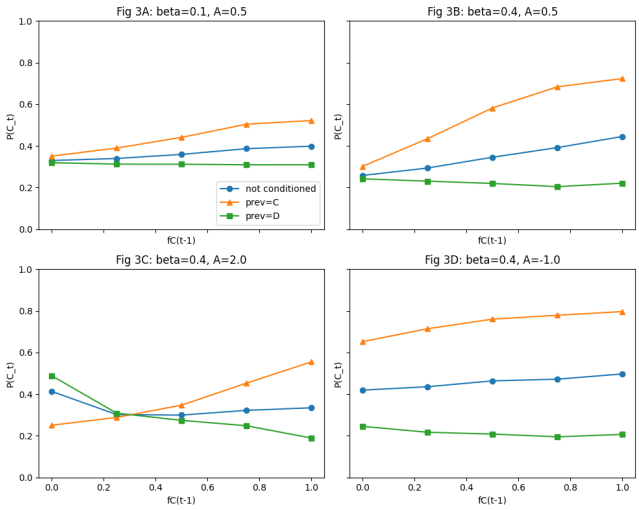


Figure 3: Conditional cooperation (CC) and moody conditional cooperation (MCC) in the PDG. The probability of cooperation at round t, $P(C_t)$, is shown as a function of the fraction of cooperating neighbors in the previous round, $f_C(t-1)$. Results are shown unconditioned (circles), conditioned on $a_{t-1} = C$ (triangles), and conditioned on $a_{t-1} = D$ (squares), for different values of A and β.

Figure 2 shows a heatmap of the mean cooperation level in the (A, β) parameter. Several clear patterns emerge. First, cooperation generally increases with β, meaning that agents who react more strongly to satisfaction or dissatisfaction reinforce successful behaviors more effectively. Second, cooperation remains relatively high for low aspiration levels ($A \lesssim 0$), where agents are easily satisfied and therefore tend to stick to their current behavior.

The medium range of aspiration levels $A \in [0.5, 1.5]$ is associated with lower cooperation, especially when β is small. In this range, agents are often dissatisfied even after cooperative outcomes, which leads them to reduce their tendency to cooperate. For larger aspiration levels ($A \gtrsim 2$), cooperation increases again reflecting frequent switching between actions when neither cooperation nor defection consistently meets the aspiration level.

Overall, these results are in close to those reported in the original paper. Small quantitative differences in absolute cooperation levels are likely due to stochastic effects and the finite number of trials, but the overall structure is the same as Fig. 2B.

5.1.3 Conditional cooperation (CC) and moody conditional cooperation (MCC).

We now turn to conditional cooperation (CC) and moody conditional cooperation (MCC) in the PDG, corresponding to Fig. 3 of the original paper. To do this, we compute the probability of cooperation at round t, denoted $P(C_t)$ as a function of the fraction of cooperating neighbors in the previous round, $f_C(t-1)$. We report results in three cases : without conditioning on the agent's previous action, conditioned on $a_{t-1} = C$, and conditioned on $a_{t-1} = D$.

Figure 3 shows these relationships for several parameter settings. Panels (A) and (B) focus on $A = 0.5$ with increasing sensitivity β. In both cases, the unconditioned probability of cooperation increases with $f_C(t-1)$, which is a clear signature of conditional cooperation. This dependence is rather weak for $\beta = 0.1$ (panel A), but becomes much more for $\beta = 0.4$ (panel B), where agents respond more strongly to satisfaction and dissatisfaction.

When conditioning on the agent's previous action, clearer patterns emerge. If the agent cooperated in the previous round ($a_{t-1} = C$), the probability of cooperating again increases sharply with $f_C(t-1)$. But after defection ($a_{t-1} = D$), $P(C_t)$ remains roughly constant or even slightly decreases as $f_C(t-1)$ increases. This asymmetry is precisely what characterizes moody conditional cooperation.

Panels (C) and (D) illustrate parameter regimes where CC and MCC no longer hold. For a high aspiration level (A = 2.0) on panel (C), the dependence on $f_C(t-1)$ becomes much weaker, and the relative ordering of the conditioned curves changes in a way that is inconsistent with experimental MCC observations. For a very low aspiration level (A = -1.0) on panel (D), agents are almost always satisfied, which leads to behavior that is largely insensitive to the actions of others. These cases closely mirror the breakdown regimes reported in the original paper.

Overall, our results are in agreement with Fig. 3 of Ezaki et al. [1]. We recover both CC and MCC in the same regions of parameter space and we observe their disappearance when aspiration levels

are set too high or too low.

5.1.4 *Linear fit analysis of CC and MCC.*

To give a more compact and quantitative summary of conditional cooperation (CC) and moody conditional cooperation (MCC), we reproduce the linear fit analysis introduced in Fig. 4 of the original paper. For each parameter pair (A, β), we fit a linear relationship between the probability of cooperation at round t and the fraction of cooperating neighbors in the previous round, as defined in (3), using least-squares regression. Fits are computed separately depending on whether we condition on the agent's previous action.

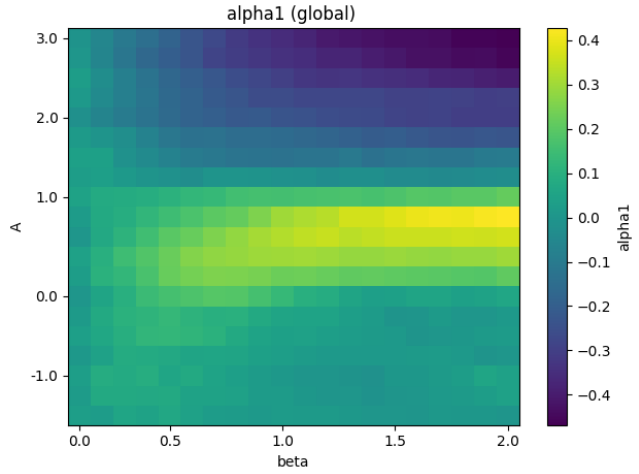


Figure 4: Slope α_1 of the linear fit $P(C_t) \approx \alpha_1 f_C(t-1) + \alpha_2$, unconditioned on the agent's previous action. Positive values indicate conditional cooperation.

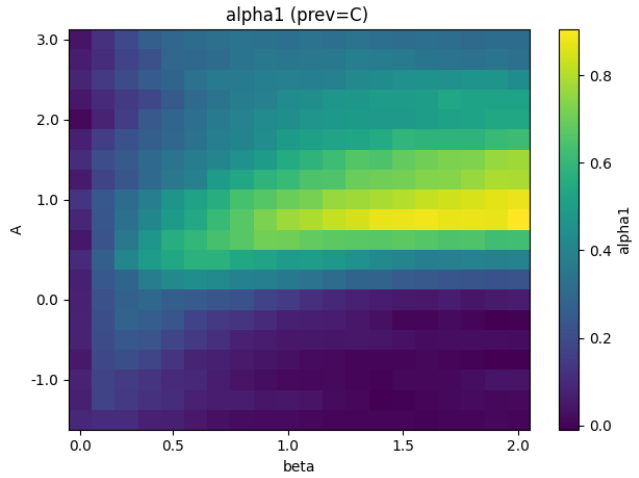


Figure 5: Slope α_1 of the linear fit conditioned on previous cooperation ($a_{t-1} = C$).

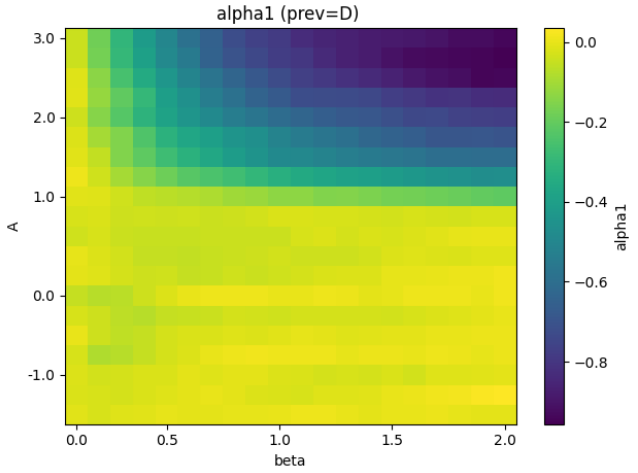


Figure 6: Slope α_1 of the linear fit conditioned on previous defection ($a_{t-1} = D$).

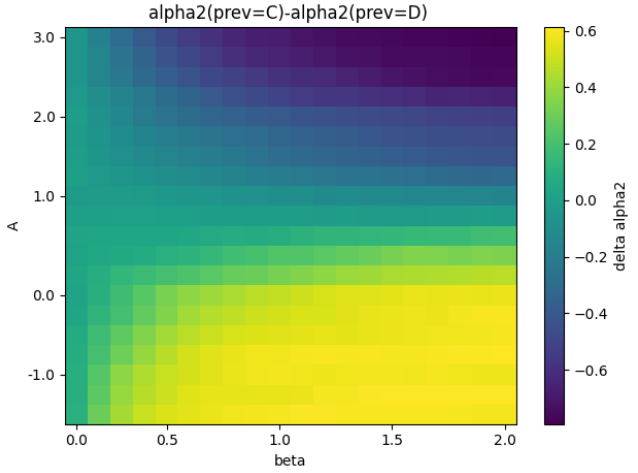


Figure 7: Difference between intercepts of the linear fit, $\Delta\alpha_2 = \alpha_2(a_{t-1} = C) - \alpha_2(a_{t-1} = D)$.

Figure 4 shows the slope α_1 obtained without conditioning on the previous action. Positive values of α_1 indicate conditional cooperation and since they correspond to a higher probability of cooperation when more neighbors cooperated in the previous round. We find that α_1 is mainly positive for aspiration levels below $A \approx 1$ and sufficiently large values of β, while it becomes close to zero or negative outside this region.

Figures 5 and 6 report the same slope when conditioning on the agent's previous action. After previous cooperation ($a_{t-1} = C$, Fig. 5), α_1 is positive over a broad region of the parameter space, indicating a strong dependence of cooperation on $f_C(t-1)$. In contrast, after previous defection ($a_{t-1} = D$, Fig. 6), the slope is close to zero or negative for most parameter values, meaning that

cooperation is only weakly influenced by neighbors' past actions.

Figure 7 shows the difference between the intercepts of the linear fits,

$$\Delta\alpha_2 = \alpha_2(a_{t-1} = C) - \alpha_2(a_{t-1} = D).$$

Positive values of $\Delta\alpha_2$ indicate that agents are more likely to cooperate after having cooperated in the previous round, even when no neighbor cooperated. We observe that this effect is mainly present for aspiration levels below $A \approx 1$.

Figures 4–7 recover the same patterns as those reported in Fig. 4 of the original paper. In particular, the same regions of parameter space associated with CC and MCC are reproduced, confirming that our implementation captures the linear-fit-based characterization of these behaviors.

5.1.5 *Dynamical aspiration.*

In the original paper, the aspiration level A is assumed to remain fixed throughout the game. As an additional experiment, we consider a variant of the model in which agents update their aspiration level over time based on the rewards they experience. The update rule is

$$A_{t+1} = (1 - \eta)A_t + \eta r_t, \tag{5}$$

where η controls the adaptation rate. Setting $\eta = 0$ naturally recovers the original fixed-aspiration model.

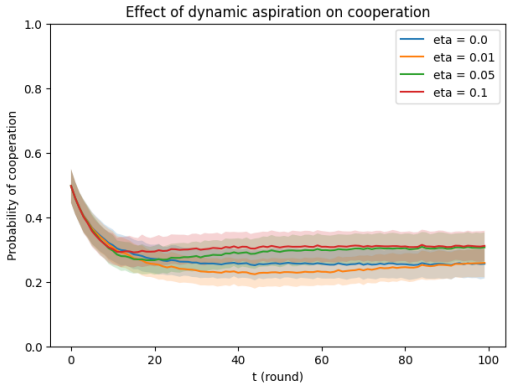


Figure 8: Effect of dynamic aspiration on the time evolution of cooperation in the PDG. The figure shows the mean probability of cooperation as a function of time for different values of the aspiration adaptation rate η. Shaded areas represent one standard deviation across trials.

Figure 8 shows the time evolution of cooperation for different values of η. For all adaptation rates, cooperation drops rapidly during the first rounds and then converges to a stationary level, much like in the fixed-aspiration case. Increasing η leads to slightly higher long-term cooperation levels, while the overall shape of the curves remains comparable across conditions. Shaded areas indicate variability across independent trials.

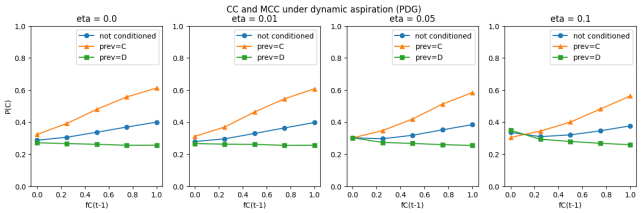


Figure 9: Conditional cooperation (CC) and moody conditional cooperation (MCC) under dynamic aspiration in the PDG. The probability of cooperation is shown as a function of the fraction of cooperating neighbors in the previous round, $f_C(t-1)$, unconditioned and conditioned on the agent's previous action, for different values of η.

Figure 9 reports conditional cooperation (CC) and moody conditional cooperation (MCC) patterns under dynamic aspiration. For each value of η, we plot the probability of cooperation at round t as a function of the fraction of cooperating neighbors in the previous round $f_C(t-1)$ conditioned only on the agent's previous action. The same qualitative structure as in the fixed-aspiration case is recovered: $P(C_t)$ increases with $f_C(t-1)$ when the agent cooperated previously, while it remains roughly constant when the agent defected previously.

5.2 Reproduction of PGG results

5.2.1 *Conditional cooperation (CC) and Moody Conditional cooperation (MCC).*

We reproduce conditional cooperation (CC) and moody conditional cooperation (MCC) in the Public Goods Game (PGG), following Fig. 5 of the original paper. In the PGG, agents choose continuous contributions $a_t \in [0, 1]$. As in the paper, we analyze CC and MCC by relating an agent's contribution at round t to the average contribution of the other group members in the previous round, denoted $f_C(t-1)$.

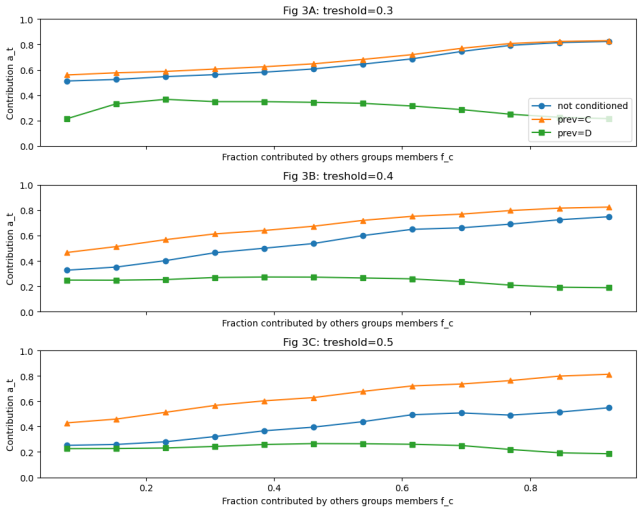


Figure 10: Graph of last contribution by fraction contributed by other groups member depending of the Treshold

Figure 10 shows the results for three different contribution thresholds ($X = 0.3$, 0.4, and 0.5), displayed as three panels stacked vertically. For all thresholds, the unconditioned contribution increases with $f_C(t-1)$, indicating conditional cooperation: agents tend to contribute more when others contributed more in the previous round.

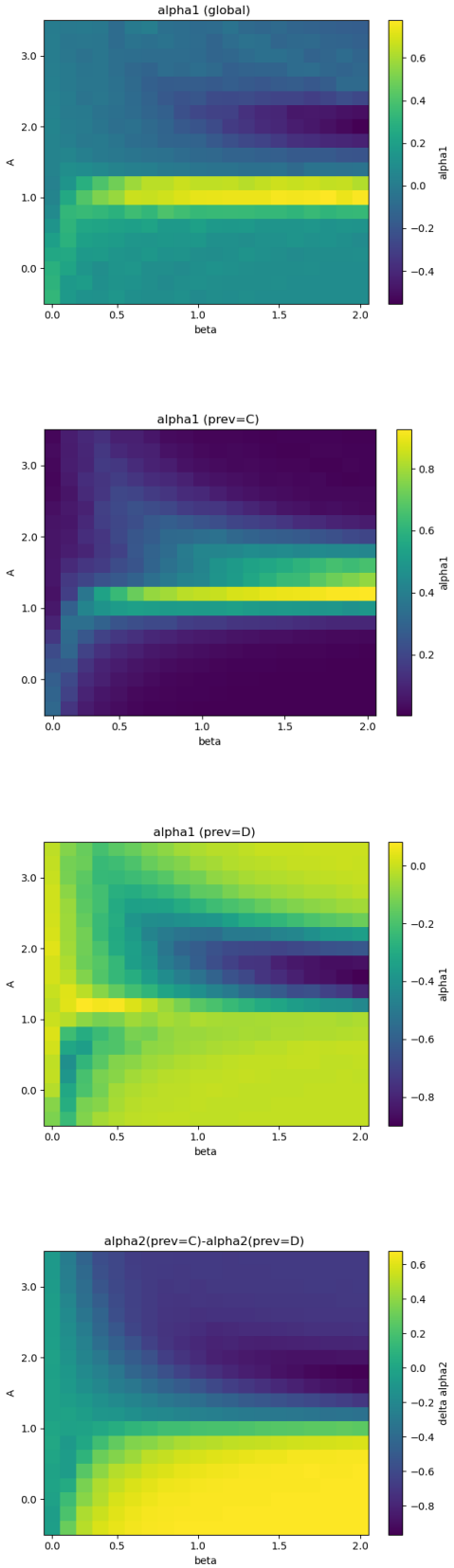
As reported in the paper, we see that the CC trend is present for graphs a and b (the general cooperation trend increases as the average contribution increases). The same is true for graph C, although this is less noticeable with a slightly flatter curve. Similarly, for the MCC, we observe the same results as those in the paper, namely that cooperation depends on the cooperation of the rest of the group AND on the previous action (which can be observed in all three graphs).

5.2.2 *Linear fit analysis of CC and MCC.*

To further analyze conditional cooperation (CC) and moody conditional cooperation (MCC) in the Public Goods Game, we use the same linear fit approach as in the PDG case. For each parameter pair (A, β) and a fixed contribution threshold $X = 0.5$, we fit a linear relation between an agent's contribution at round t and the average contribution of the other group members in the previous round,

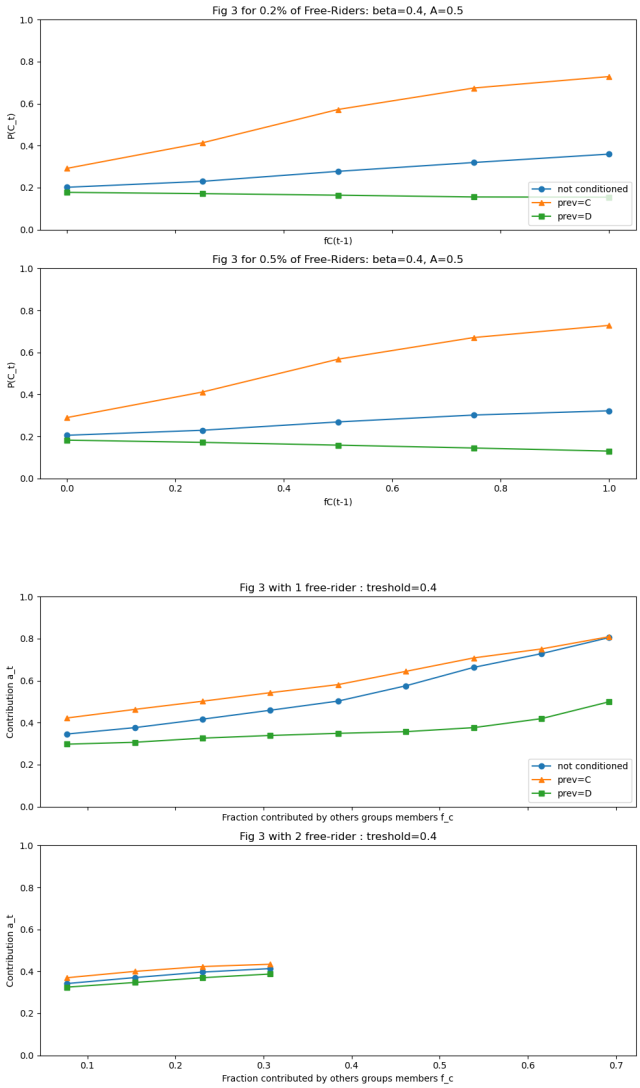
$$a_t \approx \alpha_1 f_C(t-1) + \alpha_2.$$

Here, the slope α_1 captures the strength of conditional cooperation, while differences in intercepts between conditioning cases reflect MCC effects.



By also comparing the values of A and Beta, for a threshold initialized at 0.5, the paper indicates that CC and MCC behaviors are observable for values of Beta > 0.2 and A <= 1. Our results appear to be similar to these. The only difference lies in their accuracy, given that we only calculated for 20 values of each parameter (400 configurations in total). Our values are therefore no more accurate than to the nearest 10th, which is more than sufficient for the results we wanted to obtain.

5.2.3 Presence of Free Riders. The presence of Free Riders can have an impact on the behavior of other players. A Free Rider is a player who does not cooperate, regardless of the actions of those around them. In the case of the CEO, this is trivial; however, for the PGG, no information is given regarding the specific action of a Free Rider, so for simplicity's sake, we have chosen to define their action as 0.0.



Compared to the paper, the PGG curves are less significant (alpha1 and alpha 2 values are smaller than in the paper), but the PDG curves remain in the same order of magnitude. These differences can be explained by the lack of information on the parameters for this variant: we do not have the precise actions of the PGG free riders, nor do we have the number of episodes or rounds per episode.

The impact is therefore not too significant in the PDG framework, and CC and MCC reactions remain clearly observable. For the PGG with 1 free rider, the impact does not seem to be noticeable, but when half of the players are free riders, these principles are no longer observable to the same extent.

5.2.4 Dynamical aspiration.

As with the PDG, we have also added an extension of the original model with dynamically updated aspiration level in the Public Goods Game.

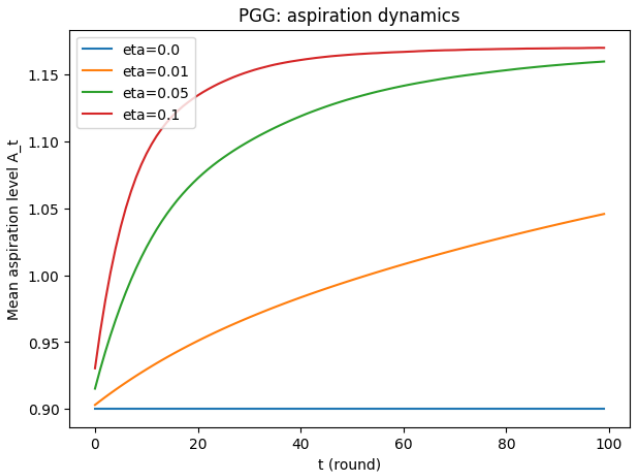


Figure 11: Time evolution of the mean contribution in the PGG for different aspiration adaptation rates η. Shaded areas indicate one standard deviation across trials.

Figure 11 shows the time evolution of the mean aspiration level A_t in the PGG for different values of the aspiration adaptation rate η. As expected, when $\eta = 0$ the aspiration level remains constant over time. For $\eta > 0$, the aspiration level increases during the first rounds and then converges to a stationary value, with larger values of η leading to faster convergence.

Figure 12 presents conditional cooperation (CC) and moody conditional cooperation (MCC) patterns under dynamic aspiration for a fixed threshold $X = 0.4$. For all values of η, the unconditioned contribution increases with the average contribution of the other group members in the previous round, indicating CC. When conditioning on the agent's previous action, we can see that contributions are higher after previous cooperation and lower after previous defection. These patterns can be seen across different adaptation rates.

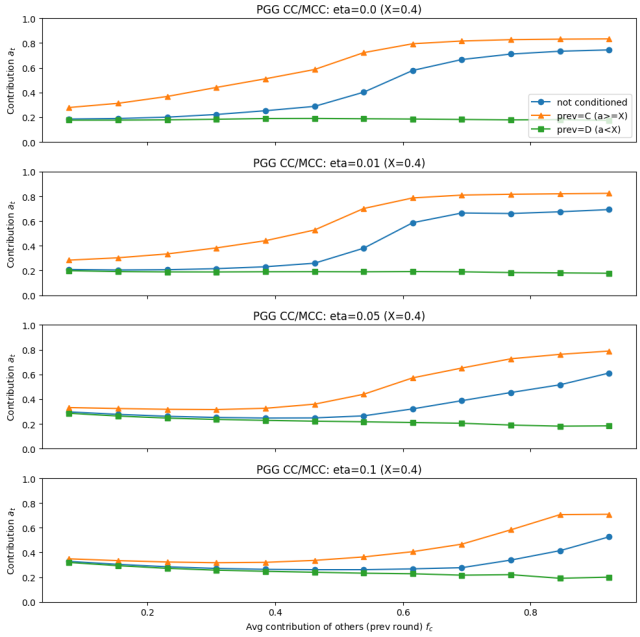


Figure 12: Conditional cooperation (CC) and moody conditional cooperation (MCC) in the PGG under dynamic aspiration, shown as a function of the average contribution of the other group members in the previous round. Results are shown for different values of η with threshold $X = 0.4$.

6 Discussion

The aim of this project was to understand and reproduce the results of Ezaki et al., who tell us that conditional cooperation (CC) and moody conditional cooperation (MCC) can emerge from a simple aspiration-based learning rule. Using our own implementation, we were able to reproduce the same main qualitative patterns in both the Prisoner's Dilemma Game (PDG) and the Public Goods Game (PGG) supporting the claim.

Indeed, CC and MCC do not appear to be explicit strategies that agents deliberately choose but instead arise naturally from a simple learning process. Agents tend to repeat actions that satisfy them and reduce actions that do not. Over time, this makes their behavior depend on both what others did before and on their own past actions. From this perspective, CC and MCC are outcomes of the learning dynamics rather than assumptions built into the model.

Our reproduction is largely consistent with the original paper, but it also provides some practical insight. In particular, the parameter regions where CC and MCC appear are fairly wide. Even with a coarse scan of the (A, β) space and a limited number of simulations, the same qualitative behaviors clearly emerge. This suggests that the model does not rely on finely tuned parameters. The small differences we observe compared to the original figures are most likely due to randomness and computational limits rather than to conceptual differences.

The linear fit analysis is very helpful for clarifying behaviors. In both the PDG and the PGG, positive slopes in the unconditioned fits indicate that cooperation increases when others cooperated more in the previous round. This give the confirmation of conditional cooperation. When conditioning on the agent's previous action, we see the MCC asymmetry: after cooperation, the slope remains positive, while after defection it is close to zero or negative in most relevant parameter regions.

The aspiration level A plays a central role in both games. CC and MCC mainly appear for aspiration levels below $A \approx 1$. When aspiration is too high, agents are often dissatisfied, even after cooperative outcomes, which weakens the link between past cooperation and future behavior. When aspiration is very low, agents are almost always satisfied, and their behavior becomes largely insensitive to rewards and to others' actions. These two extreme cases explain why CC and MCC disappear and show that aspiration acts as a balance between sensitivity and stability.

The sensitivity parameter β has a similarly clear effect. For small values of β, learning is weak and agents react only slightly to satisfaction or dissatisfaction. This results in flatter CC curves and weak MCC effects. As β increases, reinforcement becomes stronger and conditional cooperation becomes much more visible. This pattern is observed in both the PDG and the PGG, suggesting that the same learning mechanism drives behavior in the two games.

The comparaison of the PDG and the PGG tell us thatespite the difference between binary actions in the PDG and continuous contributions in the PGG, the same qualitative patterns emerge. In the PGG, conditional cooperation appears as gradual changes in contribution levels rather than changes in cooperation probability, but the underlying mechanism is the same. This suggests that aspiration-based learning gave a general behavioral principle that is not specific to a particular game.

The implementation of Free Rider (players being uncooperative every time) does not seem to impact the rest of the group in the context of the PDG. However, due to the small number of players in the PGG, this seems to reduce the level of cooperation among players in such a way that their mses, which are always cooperative, are less significan.

A key extension of our work is the introduction of dynamic aspiration levels in both games to give a more humain behavior. In the original paper, aspiration is fixed. Allowing aspiration to evolve over time does not strongly change the overall behavior of the system. Cooperation still drops quickly at the beginning and then stabilizes, and CC and MCC patterns remain clearly visible. In the PGG, agents continue to adjust their contributions based on both others' past behavior and their own previous actions, even when aspiration evolves.

This extension slightly changes how the model can be understood. Rather than viewing aspiration as a fixed personal trait, it can be seen as a reference level that agents gradually adjust through experience as humain does in real life. The persistence of CC and

MCC under dynamic aspiration makes the model more realistic and strengthens its behavioral interpretation.

Finally, several directions could be explored in future work. One option would be to study aspiration dynamics in more detail, for example by analyzing the effect of different adaptation rates. Another would be to introduce heterogeneity between agents, such as different aspiration levels or learning sensitivities. Applying the same learning rule to other social dilemmas or to settings with changing group composition would also help test how general this mechanism is. We could also vary the actions of Free Riders within the PGG framework to make them non-zero but still defective.

7 Conclusions

In this report, we reproduced the main results of Ezaki et al. on conditional cooperation (CC) and moody conditional cooperation (MCC). We also showed that these behaviors can emerge on their own using a simple aspiration-based reinforcement learning rule and without being explicitly built into the model.

Our simulations reproduce the main qualitative patterns in both the Prisoner's Dilemma Game (PDG) and the Public Goods Game (PGG) and in both games the agents adjust their behavior based on what others did in the previous round and on their own past actions. Confirming the idea that CC and MCC are natural outcomes of the learning process rather than fixed strategies that agents follow.

The reproduction also gave some practical insight as the CC and MCC have a large parameter space telling us that the model does not depend on precise parameter tuning. The small differences we observe compared to the original paper are likely due to randomness and to limited computational resources (fewer round, ect).

Furthermore, allowing aspiration levels to change over time does not alter the main conclusions. Cooperation still stabilizes, and CC and MCC remain clearly visible. Overall, this work shows that a very simple learning mechanism is enough to explain key patterns of cooperative behavior.

Finally, we also found that replicating this work was an excellent way to see the concepts discussed in class applied to a concrete model. Implementing the simulations and analyzing the results helped us better understand how simple learning rules can lead to complex collective behavior.

References

[1] T. Ezaki, Y. Horita, M. Takezawa, and N. Masuda, "Reinforcement learning explains conditional cooperation and its moody cousin," *PLoS computational biology*, vol. 12, no. 7, p. e1005034, 2016.

[2] Y. Horita, M. Takezawa, K. Inukai, T. Kita, and N. Masuda, "Reinforcement learning accounts for moody conditional cooperation behavior: experimental results," *Scientific reports*, vol. 7, no. 1, p. 39275, 2017.

[3] J. Grujić, C. Gracia-Lázaro, M. Milinski, D. Semmann, A. Traulsen, J. A. Cuesta, Y. Moreno, and A. Sánchez, "A comparative analysis of spatial prisoner's dilemma experiments: Conditional cooperation and payoff irrelevance," *Scientific reports*, vol. 4, no. 1, p. 4615, 2014.