Noisy Payoffs in an Infinitely Repeated Prisoner's Dilemma – Experimental Evidence

Alecia Evans and Juan Sesmero

Department of Agricultural Economics, Purdue University

Selected Paper prepared for presentation at the 2022 Agricultural & Applied Economics Association Annual Meeting, Anaheim, CA; July 31-August 2

Noisy Payoffs in Prisoner's Dilemma – Experimental Evidence

Alecia Evans and Juan Sesmero

Abstract

Noisy payoffs are empirically pervasive in repeated social dilemma settings. Noisy payoffs introduce imperfect monitoring amongst agents which can cause them to incorrectly infer the actions of others and change their own behavior, perhaps affecting cooperation. We conduct a laboratory experiment to examine the effect of noisy payoffs on cooperation in an infinitely repeated prisoner's dilemma. We find that noise inhibits cooperation, relative to a deterministic setting. Experimental evidence suggests that this is because noise raises inferential error, which hinders agents' ability to engage in conditional defection strategies. In other words, imperfect monitoring renders threats of punishment in repeated interactions, less effective.

JEL Codes: C73; C92; D81; D82

1 Introduction

In many settings, interactions amongst economic agents have the structure of a social dilemma – a situation in which agents fail to cooperate even when cooperation is mutually beneficial. When such interactions are repeated over time, cooperation is more likely. In repeated interactions, agents adjust their behavior to the past behavior of those they interact with. Sometimes the behavior of others is not directly observable, so agents use observed outcomes to infer others' behavior. When outcomes are affected by noise (that is, random shocks), an observed outcome may be consistent with multiple actions. This can introduce the possibility of inferential error – a situation where agents misperceive the past behavior of people they interact with. This may, in turn, impact their decisions and, consequently, cooperative outcomes. We study how noise affects inferential error, and by extension, behavior and cooperation in repeated social dilemmas.

The prisoner's dilemma (PD) is typically used to capture and study the social dilemma that is at the heart of many economic interactions. In social dilemmas, mutual cooperation achieves a first best. However, not cooperating is a dominant strategy for individual players, making no cooperation a Nash equilibrium of the stage game. In other words, individual actions, motivated by self-interest, often conflict with the collective interest of the entire group. Nevertheless, cooperation is still likely if interactions amongst players are infinitely repeated, and players are sufficiently patient. This is because players may engage in conditional defection, that is, players cooperate under the threat of future defection by others if they themselves defect. Ioannou (2014b) argues that cooperation is not a robust result, but it is rather driven by the assumption of an error-free environment, which facilitates conditional defection. However, most of the social dilemmas we observe in the real world are hardly error-free because outcomes are affected by random shocks.

When outcomes are affected by random shocks, conditional defection becomes harder because it is difficult for players to know when others have failed to cooperate. In other words, it is difficult for players to monitor others' actions. Given how ubiquitous random shocks are in real-world interactions, it is important to understand its impacts on monitoring and, consequently, cooperative behavior in social dilemmas.

For empirical context, consider an agricultural cooperative. Free-riding challenges the success of agricultural cooperatives. Members may free-ride on product quality (Bonroy et al. 2019) and even the very formation of cooperatives (Giannakas, Fulton, and Sesmero 2016). For instance, in marketing cooperatives, the ability to produce a certain amount of high-quality product depends on the coordinated actions and cooperation of all members. This is because members commit to investing in quality-enhancing inputs so that, as a group, they can achieve a certain volume at a minimum targeted quality.[1] However, actions by individual farmers are often unobserved and hard to infer from outcomes due to noise affecting them (e.g. weather)[2], thereby posing a challenge for monitoring. Bonroy et al. (2019) has offered some documentation of this phenomenon in winemaking cooperatives, among other situations. We examine how noise affecting output influences behavior and

1. In the case of the formation of cooperatives, farmers must commit to contribute a minimum amount for the cooperative to invest in projects (Giannakas, Fulton, and Sesmero 2016).

2. In other words, it is hard to determine whether a farmer that does not deliver a certain level of quality has shirked responsibilities, or has simply been unlucky and faced adverse growing conditions outside of her control.

cooperation in this type of social dilemma settings.

We examine this in a laboratory experiment. We implement an infinitely repeated PD with a continuation probability of $\delta = 0.9$. There are two treatments. In the first treatment, we implement a PD with no noise. In the second treatment, the stage game of the PD is affected by a random shock that is independent across players. This shock can be positive or negative and can be interpreted as an individual experiencing good luck (positive shock) or bad luck (negative shock). The shock introduces imperfect monitoring, resulting in agents making errors in inferring the actions of others. For our main result we find that cooperation is higher in the treatment without noise (perfect monitoring) than the treatment with noise (imperfect monitoring). Experimental data shed light on the mechanisms underlying our main result.

Under perfect monitoring, subjects can sustain cooperation by credibly engaging in conditional defection – that is, they can threaten others with defection if they defect. In the treatment with noise, subjects often made inferential errors. Imperfect monitoring resulting from inferential errors weakens the subjects' ability to credibly engage in conditional defection. It does so for two reasons. First, players are unsure about others' actions and think they may be failing to detect others' defection. In this case, they may preemptively defect to avoid the lowest payment (the "sucker" payment in a PD). In addition, they know it is also hard for others to detect defection, so they think they may be able to use imperfect monitoring to defect and avoid detection and punishment by others. In sum, random shocks affecting outcomes introduce imperfect monitoring, inhibiting the use of conditional defection strategies and, consequently, cooperation.

The strategies that the subjects played supported this belief. In the no noise treatment, subjects predominantly played Tit-for-Tat (TFT), a conditional defection strategy. This is not surprising given that defection is easily detected, and subjects can easily retaliate. However, in the treatment with noise, subjects leaned into more unconditional, uncooperative strategies such as Always Defect (AD). When they employed conditional strategies, these strategies were predominantly unforgiving, such as Grim Trigger (Grim). To summarize, imperfect monitoring pushed them to either unconditionally defect or defect when they first suspect the other player defected.

Our study contributes to the literature on cooperation in noisy in infinitely repeated PDs. Most of the literature have focused on the impact of implementation error (Fundenberg and Maskin 1990; Miller 1996; Fudenberg, Rand, and Dreber 2012; Imhof, Fudenberg, and Nowak 2007; Ioannou 2014a, 2014b; Zhang 2018). With implementation error, there is a probability that players' actions are implemented differently than they were intended. Implementation errors create an environment of imperfect information. Under these conditions, errors can alter strategies that dole out harsh punishment for defections. Overtime, such strategies may even be weeded out the environment (Fundenberg and Maskin 1990). However, the results on the impact of such errors are mixed. In some environments, agents do not tolerate defection, and in fact, are more likely to respond to defection with defection than to respond to cooperation with cooperation (Ioannou 2014b). In others, implementation errors can improve cooperation if the level of noise is low and the benefit to cooperation is high (Zhang 2018).

However, in an environment with noisy payoffs, actions are always implemented as intended. But subjects are often unsure about what action was actually implemented. Herein lies a critical difference of the two noisy environments. While implementation error creates imperfect information, the presence of inferential error creates an environment of imperfect monitoring.

We identify two ways in which an imperfect monitoring environment differs from an imperfect information environment. First, in an imperfect monitoring environment, an unconditional strategy such as AD will not accidentally cooperate. As matter of fact, Ioannou (2014b) identifies this as a reason for inferential errors being more devastating to cooperation than implementation errors. Second, conditional strategies are typically harder to implement, because of the likelihood of inferential error. Our results are consistent with both of these forces. In our treatment with noise, subjects do play more unconditional, uncooperative strategies.

Our setup is closest to Bendor, Kramer, and Stout (1991) and Bendor (1993). As a matter of fact, we experimentally test a framework very similar to theirs. Note that, while Ioannou (2014b) considers imperfect monitoring (perception errors) in their design, the inferential error rate was exogenously determined. However, our design has no such constraints. The error rate is endogenously determined and stems from the agents' ability to use the available information to update beliefs and make inference about others' actions. This environment mirrors many situations where the social dilemma is heightened by monitoring challenges.

The rest of the paper is organized as follows. In Section 2, we present the theoretical background. In Section 3, we give the details of the experimental design. In Sections 4, we outline the main results from our experiment. In Section 5, we conclude with a discussion of our main results.

2 Theoretical Background

We begin with a standard symmetric prisoner's dilemma where the stage game is given as $T > R > P > S$ and $2R > T + S$ (Table 1). Similar to Bendor (1993), we then introduce a random shock to the stage game payoffs. The players' realized payoffs become $\hat{T}$, $\hat{R}$, $\hat{P}$ and $\hat{S}$. For example, let $\hat{X} = X + V$, where $X = \{T, R, P, S\}$. That is, the payoff is composed of a deterministic stage game payoff plus a random shock. Unlike Bendor where the shock is normally distributed, we implement shocks that are uniformly distributed between $V_{LB} \leq V \leq V_{UB}$. A uniformly distributed payoff makes it easier for subjects in the laboratory experiment to understand the game and the mapping from outcomes to actions. Furthermore, the shock is also identically and independently distributed across players.

Table 1: Deterministic payoff for the prisoner's dilemma

	C	**D**
C	R, R	S, T
D	T, S	P, P

This random shock introduces an environment of imperfect monitoring. Given the distribution of realized payoffs, there is an area in which a player cannot know for sure the actions of the other player. Imagine a situation where player 1 is cooperating. As long as $S + V_{UB} > R + V_{LB}$ there is region of uncertainty where she is unsure about the action of her opponent. If she is defecting, this is true for $P + V_{UB} > T + V_{LB}$. In this region of uncertainty, there is a positive probability that players will make inferential errors, in that, they can either incorrectly infer that the other player defected (Type 1 error) or incorrectly infer that they had cooperated (Type 2 error). To see how this happens, again, assume that player 1 is cooperating. If her realized payoff is within, but close to the lower bound of the region of uncertainty, one of two things could have occurred. The other player could have cooperated as well, and she received a large negative shock. But alternatively, she could have received the sucker payoff along with a large positive shock.

Given this region of uncertainty, players will face a trade-off between Type 1 and Type 2 errors. Each player selects a benchmark value and is informed if the other player's realized payoff lies above, below, or at this value. The benchmark values that a subject selects, serve as a private noisy signal about the other player's payoff.[3] In an environment of completely uncorrelated shocks, the value of the benchmark indicates the weight a player places on committing Type 1 or Type 2 error.

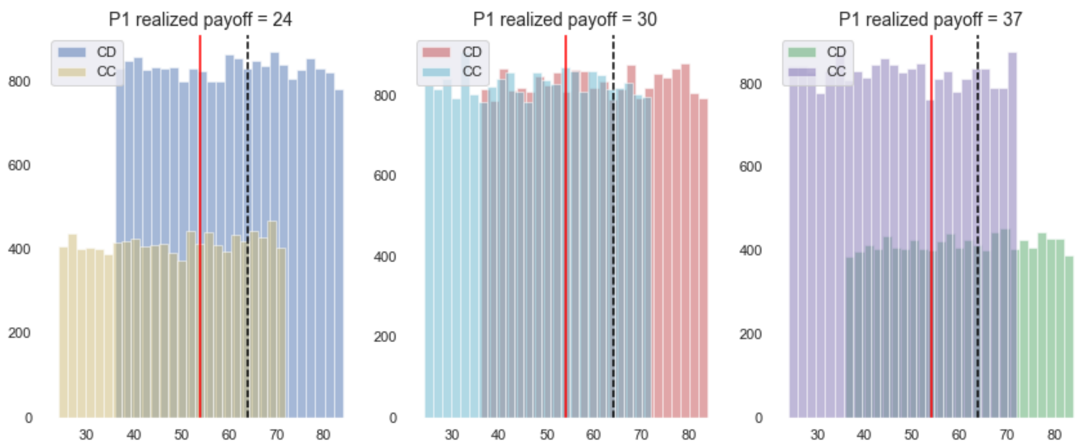
To see this, consider first a setting with the following deterministic payoffs: $R = 48$, $T = 60$, $S = 13$, and $P = 25$. Also, let us assume that shocks are uniformly distributed in the range $[-24, 24]$. In this setting, when a player cooperates, the region of uncertainty ranges from $[24, 37]$. This means that the player's payoff could, with a positive probability, fall within this range if the other players defects or cooperates. If the player's payoff is above 37, then the other player could not have defected. If the player's payoff is below 24, the other player could not have cooperated. When a player defects, the region of uncertainty ranges from $[36, 49]$.

But how does the player infer the actions of her opponent within the region of uncertainty? In the absence of more information, the player simply makes a guess. And each time she guesses, there is a chance of inferential error. But if the player has some information about her opponent's payoff, they can use this information to make a more informed guess and reduce inferential error. Generally speaking, if the other player obtains a high (low) payoff, then they are, all else constant, more likely to have defected (cooperated). In practice, players often have some information, if noisy, about their opponent's well-being; whether they buy machinery, pay debt, purchase durables, make renovations on the house, and other similar expenses/investments.

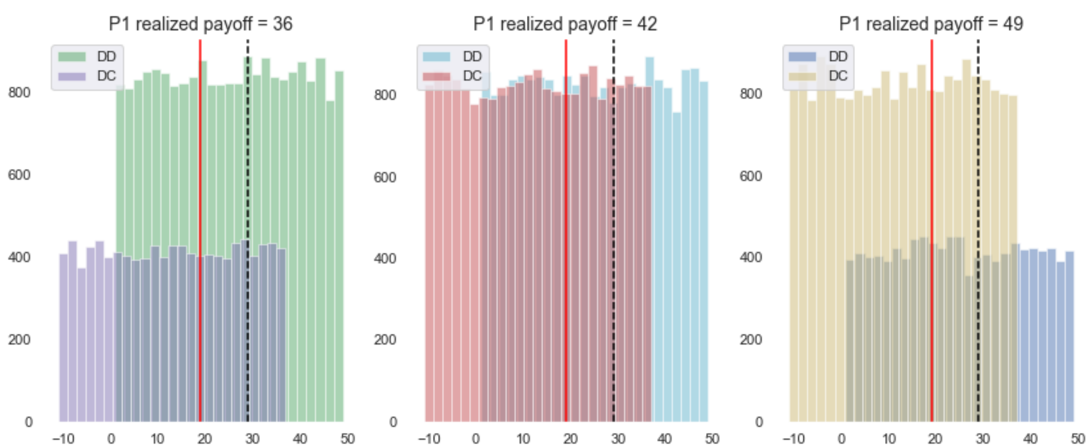
To mimic this feature of many empirical settings, we introduce a noisy signal of the opponent's payoffs. Notice that, due to the symmetry across subjects, a player knows the probability distribution of their opponent's payoffs conditional on her own actions. And she can use this information and a noisy signal about her opponent's payoff to infer their actions. To illustrate this point, consider the payoff distributions portrayed in Figure 2. In this figure, we show the results of simulating player 2's payoff one million times, conditional on player 1's action and payoff. In each scenario, player 1 is cooperating and we show the distributions of player 2 cooperating and defecting given various realized payoffs of player 1.

Suppose that player 1's payoff is 24 as in the left figure of Figure 1a. If player 1 knew with accuracy that player 2's payoff is 30, then she would know that player 2 cooperated – no negative shock is large enough to lower player 2's payoff to 30 when player 2 defects. By the same logic, player

3. Given that the realized payoffs are independent across players, this signal is essentially uninformative. However, in other work, we used the same experimental framework but allowed for correlation in payoffs across payoffs. With greater degrees of correlation, the signal becomes more informative.



(a) The distribution of realized payoffs for player 2 when player 1 cooperates



(b) The distribution of realized payoffs for player2 when player 1 defects

Figure 1: In Panel (a), player 1 is cooperating. We show the distributions for player 2, when player 2 cooperates as well (CC) and when player 2 defects (CD). The distribution of player 2's realized payoff for a realized payoff of 24, 30 and 37 for player 1. The possible realized payoffs for player 2 are in the range [24, 85]. As the benchmark increases from 54 (red solid line) to 64 (black dashed line), a greater importance is placed on not committing Type 1 errors (incorrectly inferring defection). In Panel (b), player 1 is defecting. We show the distributions for player 2, when player 2 cooperates (DC) and when player 2 defects as well (DD). We show the distribution of player 2's realized payoff, when player 1's realized payoff is 36, 42 and 49. The possible realized payoffs for player 2 are in the range [-11, 49]. As the benchmark increases from 19 (red solid line) to 29 (black dashed line), a greater importance is placed on not committing Type 1 errors (incorrectly inferring defection).

1 would know that player 2 defected if her payoff was 80. If player 2's payoff fell on the part of the domain where both distributions overlap (henceforth, the "overlapping region"), then she would be unsure about player 2's actions and could incorrectly infer it. Moreover, in reality, players seldom have access to accurate information. They typically have noisy information that, at best, tells a player whether their opponent's payoff falls within some "region" that the player considers informative.

We formalize this situation by letting the player establish a benchmark value, on the opponent's payoff domain, above which the player assumes the opponent defected and below which the player assumes the opponent cooperated. The noisy signal indicates whether the opponent is above or below the benchmark, but not her exact payoff.[4] The level of this benchmark not only affects inferential error but also the type of error incurred. Suppose player 1 sets a very high benchmark. This means that she will infer defection from player 2 only if player 2's payoff is very high. In other words, player 1 will mostly infer cooperation. This will probably lead to a high frequency of inferential error, mostly consisting of Type 1 errors. If the benchmark is very low, the frequency of inferential error will also tend to be high, but errors will mostly consist of Type 2. Finally, a benchmark located towards the middle of the "overlapping region" will result in lower frequency of inferential error and a balanced prevalence of Type 1 and Type 2 errors.

From the simulations in Figure 1, we can numerically show that Type 1 and Type 2 errors are equalized at a benchmark value of 54. If we increase the benchmark value to, say, 64, Type 1 error (incorrectly inferring defection) increases, while Type 2 error (incorrectly inferring cooperation) decreases. If we decrease the benchmark value below 54, Type 1 error increases and Type 2 error decreases.

Therefore, by changing the benchmark relative to which the player gets a noisy signal, she is changing not only the overall frequency of inferential error but also the implicit weight that is placed on committing Type 1 or Type 2 errors. If the player is particularly concerned about being the "sucker" (cooperating when the opponent is defecting), then she will set a low benchmark. But this can impact overall cooperation. If the player infers defection, she may want to retaliate and defect. In this case, the other player may eventually infer defection and defect themselves (if they were not already defecting). This situation may prompt cooperation to unravel. The opposite may happen if players set a low benchmark.

Another important insight from Figure 1 is that, the frequency of inferential error is also determined by the player's own payoff. For instance, if player 1 obtains a payoff of 30, at any point within the "overlapping region" defection and cooperation by player 2 are equally likely. But if the player obtains a high payoff of 37 then, within the "ovelapping region" it is more likely that player 2 cooperated – which is partly why player 1 obtained a high payoff to begin with. Therefore, if player 1's payoff is high, it is slightly easier to correctly infer the actions of player 2, thereby reducing inferential error. This also true of player 1 obtains a low payoff of 24. In this case, it is slightly easier for player 1 to correctly infer defection.

Ultimately, the frequency and composition of inferential error depends upon how players use the information available to them, regarding their own payoff and the payoff of their opponent. And, in

4. In an empirical situation this is similar to player 1 obtaining information regarding expenses/investments made by player 2 or other information of similar nature, that indicates the overall region in which player 2's payoff fell.

turn, the frequency and composition of inferential error is likely to affect their willingness to cooperate and, overall cooperation in the repeated game with noise.

A key mechanism underlying cooperation in repeated PDs is the type of strategies followed by players. Players may engage in unconditional strategies (that is, always cooperate or always defect) or conditional ones. Conditional strategies are easy to follow (and, therefore, more credible and effective) under perfect monitoring. Both players can observe their opponent's behavior and condition their actions to them. And both players know the other players can observe their behavior, and so on.

In contrast, imperfect monitoring makes conditional strategies harder to implement. First, if the player's payoff falls in her region of uncertainty, then she cannot know with certainty her opponent's past behavior. It is, naturally, harder to condition a strategy on an uncertain event. Second, the threat of a conditional defection from another player may be perceived as weaker if the player knows that her opponent may not detect her defection with certainty. It is unclear, therefore, how imperfect monitoring may influence players' strategies. Will they use more or less conditional strategies? If they use conditional strategies, what are the conditions under which players will infer defection? And conditional on inferring defection, how quickly would they punish their opponent, and for how long?

Our discussion reveals a direct effect of noise on inferential error, and then two channels through which inferential error may affect cooperation. These mechanisms are portrayed in Figure 2. Noisy payoffs introduce uncertainty regarding the opponent's past behavior, which may induce subjects to incorrectly infer their opponents' past actions. This has a direct and an indirect effect on cooperation.

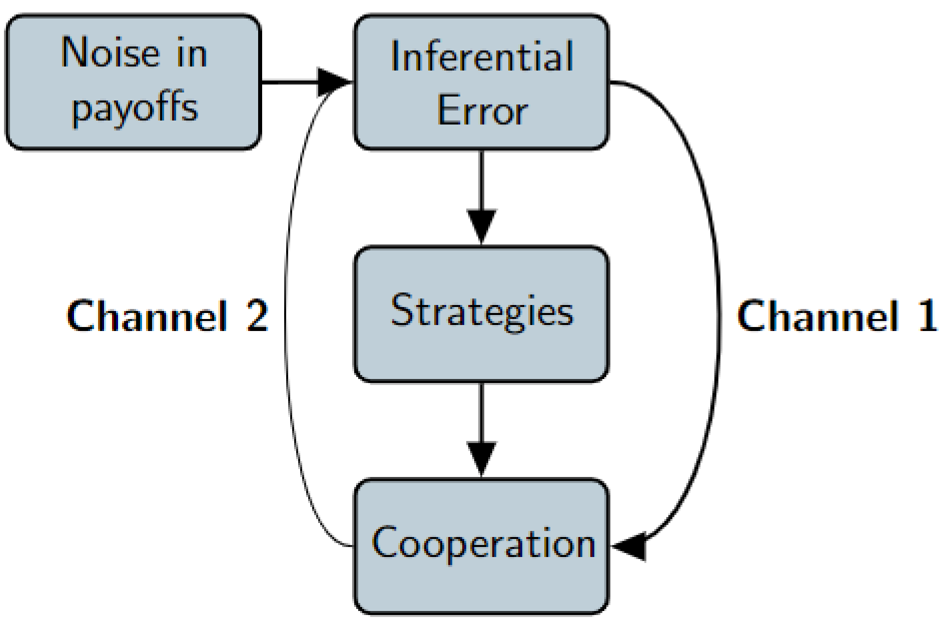


Figure 2: Causal channels and the effect of noisy payoffs on cooperation

We call the direct effect of inferential error on cooperation, channel 1 (Figure 2). A player that defects after she detects a defection by her opponent, may incorrectly infer defection, and punish the opponent unnecessarily. She may also be unnecessarily punished by her opponent, prompting the player to defect herself, even if she was not defecting before. A similar logic applies when a player incorrectly infers cooperation. In this case, players may fail to punish defection by their opponent, or may be able to hide defection from their opponents due to uncertainty.

We call the indirect effect of inferential error on cooperation, channel 2 (Figure 2). In the absence of uncertainty regarding the opponent's actions, a player may commit more credibly to conditional strategies. But when inferential errors emerge due to noisy payoffs, players may be prone to rely less on conditional strategies; or at the very least simplify conditional strategies to avoid switching every time the player infers (perhaps incorrectly) a change in her opponent's actions. In this channel, inferential error affects cooperation, but through a change in the type of strategies used by players.With these considerations in mind, our experimental design is aimed at answering the following question:

Question 1: *Is cooperation lower in the presence of noise?*

We predict that the presence of noise in outcomes will inhibit cooperation. We believe this will be mostly driven by two distinct mechanisms. First, the presence of noise may weaken monitoring, that is, increase inferential error. This prediction is not only motivated by our numerical simulations (illustrated in Figure 1), but also in line with theoretical discussions in Aoyagi, Bhaskar, and Fréchette (2019) and Ioannou (2014b). But, as previously discussed, overall frequency and nature of inferential errors will depend on players' ability to process information and their choice of benchmark. Therefore, we raise the following question about this mechanism:

Question 2: *How frequent are inferential errors with noisy outcomes?*

We predict that, in the presence of noisy outcomes, players will make inferential errors.

A second mechanism relates to the type of strategies used by players, conditional on their inference. A player may decide to play unconditional strategies, that is, defect or cooperate regardless of the opponent's actions. But often, players engage in conditional strategies (Dal Bó and Fréchette 2018). If a player infers that their opponent defected in the previous round, she can be tolerant or punish them. If she punishes them, she may do it for one round or multiple rounds. If noise affects the ability of players to infer their opponent's past actions, and/or the confidence with which players infer such actions, then noise may change the type of strategies used by the players. This motivates our next question:

Question 3: *How do the strategies that players use vary with noise?*

We predict that subjects will use less forgiving strategies in our treatment with noise, in comparison to the treatment with no noise. Our prediction is motivated by observations in Ioannou (2014b). His automata are more likely to respond to defection with defection than to respond to cooperation with cooperation. We expect similar results with our subjects, where they are less forgiving and lenient with noise.

To obtain an answer to these questions we conduct a laboratory experiment where subjects play the game we discussed and simulated in our theoretical background. We now turn to the design of the laboratory experiment.

3 Experimental Design

The experiment was conducted at Purdue University's Vernon Smith Experimental Economics Laboratory (VSEEL). Two treatments were implemented, a treatment with noise and one without. A total of 80 subjects participated in seven sessions, with each session consisting of 10 or 12 subjects. Table 2 shows the treatment details. Subjects accumulated points throughout the session. At the end of the session, these were converted at an exchange rate of $1 = 300 points. We implemented a between-subject design, where each subject participated in only one session. There was only one treatment per session. The sessions for the noise treatment lasted approximately 90 minutes, while the no noise sessions lasted on average 60 minutes.

Table 2: Treatment, sessions, and subjects in the experiment

Treatment	No. of Sessions	Total Subjects
Noise	3	36
No noise	4	44

In each session, players are matched in pairs. The pair then plays the prisoner's dilemma game repeatedly. In other words, after each round of the game there is a probability that the game continues onto another round (and, conversely, a probability that the game ends). We use a continuation probability of $\delta = 0.9$. The game in which the same pair plays multiple rounds until termination is called a supergame. For both treatments, subjects are randomly rematched before each supergame. We pre-drew these game lengths using a geometric distribution (Romero and Rosokha 2019).

Table 3 shows the payoff matrix of the stage game in the no noise treatment. The payoffs for subjects are denoted in points. In each round, subjects choose between cooperation and defection (in the experiment we used neutral language of A or B). At the end of each round, subjects received feedback on only their payoff. On each decision page, subjects had a history of all their previous actions.

Table 3: Payoff of the stage game

	C	**D**
C	48, 48	13, 60
D	60, 13	25, 25

The interface for the noise treatment had a few notable differences. For this treatment, we introduce a random shock to the stage game payoff in Table 3. This shock is independent across rounds and across players and uniformly distributed within the range [-24, 24]. Also, in the noise treatment, before each supergame, subjects select two benchmark values. One benchmark value is used if the subject opts to cooperate and the other if they defect. The allowable range of values for each corresponds with the range of possible realized payoff the opponent could receive. That is, subjects were restricted to the range [24, 84] if they choose cooperation and a range of [-11, 49] if they choose defection. After each round, subjects receive information about the other player's realized payoff relative to these benchmark values (in addition to exact information on their own payoffs, of course). Specifically, the signal indicates if the opponent's realized payoff is above, below, or equal to the benchmark value selected.

To understand decisions by the subjects, we elicit their beliefs about the likely action of their opponent by using a Binarized Scoring Rule (BSR). The BSR is incentive compatible in that, as long as subjects prefer getting a reward as opposed to no reward, to maximize the probability of getting the reward, they find it optimal to truthfully report their beliefs about the other subject's action (Hossain and Okui 2013). We incentivize this truth telling with 2 points per decision. That is, for the 84 decisions that each subject made, they could earn an additional 168 points for truthfully reporting their beliefs. In Appendix A.1, we describe the belief elicitation process. In the instructions, we did not give subjects the full details of the belief elicitation process. They were informed that the details were available after the session.[5] This design feature follows Danz, Vesterlund, and Wilson (2020), who, in an experiment using BSR, found that giving subjects very detailed information on the incentive structure of the BSR results in them making errors in excess of 40%, than if detailed information was not given.

For both treatments, all the details of the experiment were explained to the subjects in the instructions (a copy is presented in Appendix A.2). Subjects read the detailed instructions onscreen before each session. They also had a written copy throughout the entire session. The experiment was programmed in oTree (Chen, Schonger, and Wickens 2016).

4 Results

To recap, the no noise treatment is an environment of perfect monitoring. In the absence of noise, subjects are fully aware of the action of their opponent given their observed payoff. This is not true in the presence of noise. While subjects can correctly infer the action of their opponent outside of the region of uncertainty, within this region, it is possible to make incorrect inferences about the actions of others. This creates an environment of imperfect monitoring. We are interested in understanding how imperfect monitoring affects cooperation in an infinitely repeated PD. We examine this in three parts. First, we examine the impact of noise on cooperation (Question 1 in Section 2). Then we examine the underlying mechanisms: the impact of noise on inferential errors (Question 2 in Section 2), and the impact of noise on the strategies subjects play, conditional on inference (Question 3 in Section 2).

4.1 Cooperation Rates

Figure 3 shows the evolution of cooperation under the baseline of no noise and the treatment with noise. Average cooperation measures the proportion of rounds in which subjects cooperated in a supergame. Across all supergames, for both treatments, cooperation rate is statistically greater than zero. However, overall cooperation is significantly higher in the no noise treatment (0.68 versus 0.40, *p-value* 0.00). Statistical significance is established using a probit regression clustered at the session

5. No subject asked for this information after the session.

level. No statistical significance refers to a *p-value* above 0.1. While cooperation increased over subsequent supergames played by subjects under the no noise treatment, it did not increase with noise. This suggests players learned to play conditional strategies and sustain higher cooperation in an environment of perfect monitoring, but that imperfect monitoring likely inhibited this learning process.

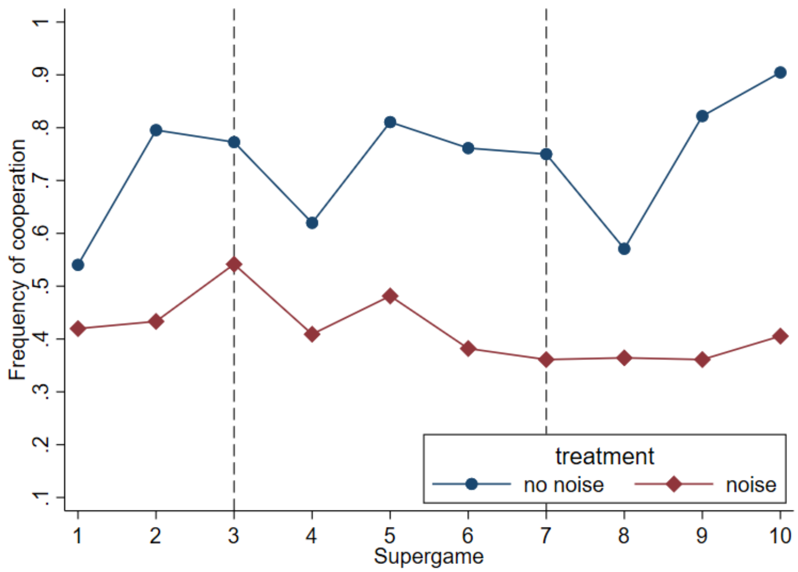


Figure 3: Frequency of cooperation across supergames for noise (red line) and no noise (blue line)

In Table 4, we further disaggregate these results by first round of the first supergame (column 1), early supergames (column 2), late supergames (column 3) and by all rounds of all supergames (column 4). We find that average cooperation in the very first round is not statistically different between the two treatments (*p-value* 0.53). For supergame 1-3, the difference between the two treatments is 0.21 (*p-value* 0.000), and this increases to 0.31 (*p-value* 0.001) in late supergames (that is, supergames 8-10).

A comparison of cooperation rates in the first round of the first supergame with later supergames suggests learning takes place under perfect and imperfect monitoring. But this learning process have opposite effects across environments. In a perfect monitoring environment, players learn to cooperate as revealed by an increase in cooperation rates from 0.57 to 0.68. In an imperfect monitoring environment, as players learn to play, cooperation unravels. As a result, cooperation rates decrease drastically from 0.61 to 0.37.

We use values reported in Figure 3 and Table 4 to answer Question 1 in Section 2:

Result 1: *Cooperation is higher with perfect monitoring than imperfect monitoring.*

Result 1 supports our prediction that cooperation is higher under perfect monitoring. Our two treatments are parallel to the perfect monitoring and imperfect (noisy) private monitoring treatments of Aoyagi, Bhaskar, and Fréchette (2019). Our approach differs in an important way from the imperfect

Table 4: Average Cooperation

Treatment	First round	Supergame 1-3	Supergame 8-10	All
Noise	0.61**	0.45***	0.37**	0.40***
	(0.07)	(0.04)	(0.06)	(0.04)
No noise	0.57***	0.66***	0.68***	0.68**
	(0.02)	(0.03)	(0.08)	(0.06)
p-value	0.53	0.00	0.00	0.00

Notes: Robust standard errors (in parenthesis) are clustered at the session level. * Indicates statistical significance at the 10% level (0.05 < p-value < 0.1). ** Indicates statistical significance at the 5% level (0.01 < p-value < 0.05). *** Indicates statistical significance at the 1% level (p-value < 0.01)

private monitoring of Aoyagi, Bhaskar, and Fréchette (2019). Inferential error is endogenous in our setting, and it depends on the agents' ability to use available information. In their study, inferential error is exogenous. Players do not directly observe their opponent's past action, but instead receive a private signal with a set known accuracy about such action. The signal is either good or bad, and a good signal is more likely to occur when their opponent is cooperative. Unlike our observations, they found no statistical difference in cooperation rates between imperfect private monitoring and perfect monitoring.

We now turn our attention to the mechanisms underlying the link between noise and cooperation.

4.2 Inferential Error and Cooperation

To measure inferential error, we use the subjects' beliefs regarding their opponent's past action elicited from subjects using the BSR and contrast those beliefs with actual actions taken by their opponents. In our framework, after each round, subjects selected the probability with which they believed that the other player had cooperated. If they indicated a probability greater than 0.5, we assigned inference to cooperation. For probabilities less than 0.5, we assigned inference to defection. When subjects assigned equal probability to cooperation and defection, we randomly assigned inference to cooperation or defection with a 0.5 probability.[6]

Subjects did commit inferential error outside of the region of uncertainty. However, this only occurred about 8% of the time. In contrast, inside of the region of uncertainty, subjects committed inferential error 33% of the time. For this reason, in our analysis, we examine inferential error both within the region of uncertainty (ROU) and across all decisions (All). Table 5 outlines both.

Regardless of the disaggregation across supergames, inferential error rate is statistically significantly greater than zero for both early supergames and later supergames. If we focus on errors within

6. Of the 3024 observations for the noise treatment, there are 69 observations where subjects reported 0.5 probability of the other subject cooperating.

Table 5: Inferential Error Rates with Noise

	Supergame 1-3	Supergame 8-10	*p-value*
All	0.17**	0.15**	0.12
	(0.04)	(0.03)	
ROU	0.38**	0.31**	0.10
	(0.06)	(0.06)	

Notes: Robust standard errors (in parenthesis) are clustered at the session level. * Indicates statistical significance at the 10% level (0.05 < p-value < 0.1). ** Indicates statistical significance at the 5% level (0.01 < p-value < 0.05). *** Indicates statistical significance at the 1% level (p-value < 0.01)

the region of uncertainty, the error rate slightly decreases from 0.38 (38%) in supergames 1-3 to 0.31 (31%) in supergames 8-10. This difference is at the margin of statistical significance *(p-value* 0.10). Likewise, the difference between earlier rounds (0.17) and later rounds (0.15) across all decisions is subjected to considerable noise (*p-value* 0.11).

Since, in the absence of noise, subjects know the opponent's past action with certainty, no inferential error takes place. In the presence of noise, however, inferential error emerges. The results in Table 5 deliver an answer to Question 2 in Section 2:

Result 2: *In the presence of noise, when the past action of the other player is uncertain, subjects commit inferential errors about a third of the time.*

Result 2 supports our prediction that inferential error raises with noise. We now turn our attention to the indirect channel through which noise affects cooperation; a behavioral channel, whereby noise prompts subjects to change the nature of the strategies they play.

4.3 Discussion of Strategies

We start this section by conducting an initial, crude analysis of how subjects respond to inferred past actions of their opponents. We then move on to a more elaborate scheme to elicit strategies (a set of contingent actions) played by subjects with and without noise in the infinitely repeated PD.

In Table 6 we report the frequency with which subjects defect in response to (inferred) defection and the frequency with which they cooperate in response to (inferred) cooperation. Like Ioannou (2014b), we see evidence that subjects are more likely to respond to defection with defection than to reciprocate with cooperation after cooperation when noise is present. As shown in Table 6, in the noise treatment, subjects respond to a perceived defection with defection roughly 47% of the time, while they defected in response to a perceived cooperation only 16% of the time. In contrast, in the absence of noise, players are more likely to cooperate in response to inferred cooperation (62% of the time) than they are likely to defect in response to defection (28% of the time).

Interestingly, the incidence of cooperation after an inferred defection is much higher with noise

Table 6: Player's Response to Other Player's Action

	CC	DC	CD	DD
No Noise	61.6%	6.23%	4.4%	27.8%
Noise	28.8%	16%	8.3%	46.9%

Notes: CC: a player cooperates after perceiving cooperation; DC: a player defects after perceiving cooperation; CD: a player cooperates after perceiving defection; DD: a player defects after perceiving defection

than without. This perhaps indicates that subjects realize that their opponent is susceptible to making errors as well. Therefore, in these instances, they are giving their opponent the benefit of the doubt. Nevertheless, the incidence of defection after an inferred cooperation is also higher with noise than without. This seems to point that some subjects are simply going for the temptation payoff, perhaps conjecturing that their opponent will give them the benefit of the doubt. Therefore, after our crude analysis of actions, it remains unclear whether or not noise prompts subjects to use more conditional strategies and, if so, what these strategies are.

We use the Strategy Frequency Estimation Method (SFEM) from Dal Bó and Fréchette (2011) to elicit the frequency of strategies across treatments. SFEM uses a Maximum Likelihood Estimation (MLE) to estimate the frequency with which each strategy–from a set of pre-determined set of strategies–is found in the experimental data. This methodology has since been employed in Fudenberg, Rand, and Dreber (2012), Rand, Fudenberg, and Dreber (2015), Dal Bó and Fréchette (2018), Aoyagi, Bhaskar, and Fréchette (2019), Dal Bó and Fréchette (2019) and Romero and Rosokha (2019), for example. This method assumes that each subject uses the same strategy across supergames. However, they can make mistakes. These mistakes are not the errors that are generated from the experimental design, but rather, it is assumed that subjects can make mistakes when choosing their intended actions for the particular strategy they are following.

Using the notations of Dal Bó and Fréchette (2019), assume that the probability with which subject i makes mistakes is $1-\beta$ and the probability that her chosen actions correspond with a strategy k is β. The likelihood that her observed choices were actually generated by strategy k is $Pr_i(s^k) = \Pi_{M_i}\Pi_{R_{im}}(\beta)^{I^k_{imr}}(1-\beta)^{1-I^k_{imr}}$. In this expression, I^k_{imr} is an indicator function that takes the value 1 when the choice that was actually made in round r and supergame m is the same as what the subject would have made if she were following strategy k. It is coded 0 otherwise. M and R are the sets of supergames and rounds. The parameter β is estimated within the model. It can also be interpreted as the probability that an action is taken given that it is prescribed by a strategy k. Therefore β is the basis for evaluation of model fit, that is, as the model fit improves β approaches 1.

Therefore, the MLE process entails choosing both the probability of mistakes and the frequency of strategies that maximizes the likelihood of the sequences of choices. That is, the log-likelihood is $\Sigma_I ln(\Sigma_K \phi^k Pr_i(s^k))$, where K is the subset of strategies being considered and ϕ^k is a vector of parameter estimates that represent the frequency of strategies. We bootstrapped the standard errors in a way that respects the data generating process of our experimental data. We randomly draw the appropriate number of sessions, then for each session the appropriate number of subjects, then supergames. All with replacement. The bootstrapping process was done 1000 times. The standard deviation of the bootstrapped MLE estimates provide the standard errors.

We considered a subset of the ten strategies described in Appendix A.3. We first estimated the MLE using the twenty strategies described in Fudenberg, Rand, and Dreber (2012). Then we reduced this to ten strategies. All strategies that were statistically significantly identified from the larger set are included in this subset. We employed the additional step, given that Dal Bó and Fréchette (2019) found that the estimates of the MLE are robust to including additional strategies. However, excluding essential strategies could lead to misleading results. We will disaggregate results on the early stages of the game (supergames 1-3) and the late stages (supergames 8-10).

Results on strategy frequency are reported in Table 7. Subjects predominantly used memory-1 strategies. These are simple strategies in which subjects condition their actions only on the immediate past round. As shown in Table 7, for both treatments, the fraction of memory-1 strategies increased between supergames 1 – 3 and supergames 8 – 10. For the noise treatment this increased from 0.48 to 0.68, while for the no noise treatment this increased from 0.59 to 0.79. This indicates that players employ increasingly simple, though still conditional, strategies are they learn to play the game. Overall, the most employed strategies are AD, GRIM and TFT – all memory-1 strategies.

-4

As expected, the strategies played are consistent with the cooperation rates observed across treatments. In the noise treatment, the fraction of more cooperative strategies decreased from 0.39 (17% TFT and 22% TF2T) in supergames 1 – 3 to 0.15 in supergames 8 – 10 (15% TFT). However, under perfect monitoring where noise is absent, the fraction of cooperative strategies increased from roughly 50% (35% TFT and 14% GRIM3) to about 54% (TFT). For the noise treatment, the most dominant strategy is AD, whereas the most dominant strategy without noise is TFT.

Our analysis indicates that players are more prone to use unconditional strategies under noise, while they rely more on conditional strategies without noise. Conditional strategies are played 40% of the time under noise and 80% of the time without noise. Also, the unconditional strategies used in a noisy environment, are predominantly non-cooperative. The results in Table 7 deliver an answer to Question 3 in Section 2:

Result 3: *Subjects used less cooperative strategies in the presence of noise.*

Result 3 supports our prediction that noise prompts subjects to play less forgiving and lenient strategies. The strategies played are consistent with the pattern of lower cooperation under the noise treatment relative to the treatment with perfect public monitoring.

5 Conclusion

The "Folk Theorem" for repeated games suggests that with repetition and sufficiently patient players, cooperation can arise as a sub-game perfect Nash equilibrium. However, when noise is introduced to social dilemma type settings, cooperation tends to be weaker than is expected. Previous studies have

Table 7: Estimation of Strategies Used

		Unfriendly			Provocable			Lenient				
Supergames	Treatment	AD	DTFT	DGRIM2	GRIM	2TFT	TFT	GRIM2	GRIM3	TF2T	AC	Beta
1-3	Noise	0.31***	0.01	0.03	0.02	0.13*	0.17**		0.08	0.22**	0.02	0.83***
		(0.09)	(0.05)	(0.03)	(0.10)	(0.08)	(0.09)		(0.10)	(0.10)	(0.05)	(0.03)
	No Noise	0.10**	0.02	0.05	0.14*	0.12	0.35***	0.06	0.14*	0.01		0.90***
		(0.05)	(0.04)	(0.04)	(0.10)	(0.15)	(0.14)	(0.10)	(0.10)	(0.08)		(0.02)
8-10	Noise	0.38***	0.05*		0.25***		0.15**	0.06		0.07	0.03	0.91***
		(0.12)	(0.04)		(0.10)		(0.09)	(0.7)		(0.07)	(0.05)	(0.02)
	No Noise	0.04	0.05	0.04	0.25**		0.54***		0.07			0.94***
		(0.04)	(0.04)	(0.04)	(0.12)		(0.12)		(0.07)			(0.02)

Notes: Bootstrapped standard errors are in parenthesis. Strategies that are 0.0 are dropped *** $p<0.01$, ** $p<0.05$, * $p<0.1$

mostly focused on noise in the form of implementation error, where an action is accidentally change from what was intended. We examined the less explored inferential error, arising from noisy payoff structure. This noise creates imperfect monitoring, and region of uncertainty exists where players are prone to incorrectly infer the likely action of their opponent.

The primary focus of this paper is on the difference in cooperation rate between an environment with no noise and one without. Findings from Ioannou (2014b) suggest that inferential errors will erode cooperation, and to a greater extent than implementation error. We do find evidence of lower cooperation rate, but with respect to a baseline of no noise. We present evidence of this lower cooperation rate resulting from the prevalence of inferential error. With noise, subjects frequently incorrectly infer the previous action of their opponent. The noisy environment may have made them more comfortable to attempt to gain the temptation payoff. We saw a greater prevalence of subjects responding to cooperation with defection under the noise treatment, than under the no noise treatment.

This may also explain the prevalence of AD in the population of strategies employed. Subjects could have pre-emptively defected to avoid being a sucker. Dawes and Thaler (1988) observed that people tend to cooperate until they have evidence to show that they are being taken advantage of by who they are interacting with. Also, Dal Bó and Fréchette (2018) notes that cooperation is more likely to emerge as an equilibrium when the environment allows cooperation to be robust to strategic uncertainty. Subjects may have anticipated that the noisy environment would inhibit their ability to decipher their opponent's action. It could also be a case that players felt comfortable defecting under a veil of ignorance. Future work could examine if removing this veil, through improved monitoring, could recover cooperation.

References

Aoyagi, Masaki, V Bhaskar, and Guillaume R Fréchette. 2019. "The impact of monitoring in infinitely repeated games: Perfect, public, and private." *American Economic Journal: Microeconomics* 11 (1): 1–43.

Bendor, Jonathan. 1993. "Uncertainty and the evolution of cooperation." *Journal of Conflict resolution* 37 (4): 709–734.

Bendor, Jonathan, Roderick M Kramer, and Suzanne Stout. 1991. "When in doubt... Cooperation in a noisy prisoner's dilemma." *Journal of conflict resolution* 35 (4): 691–719.

Bonroy, Olivier, Alexis Garapin, Stephen F Hamilton, and Diogo M Souza Monteiro. 2019. "Free-riding on Product Quality in Cooperatives: Lessons from an Experiment." *American Journal of Agricultural Economics* 101 (1): 89–108.

Chen, Daniel L, Martin Schonger, and Chris Wickens. 2016. "oTree—An open-source platform for laboratory, online, and field experiments." *Journal of Behavioral and Experimental Finance* 9:88–97.

Dal Bó, Pedro, and Guillaume R Fréchette. 2011. "The evolution of cooperation in infinitely repeated games: Experimental evidence." *American Economic Review* 101 (1): 411–29.

Dal Bó, Pedro, and Guillaume R Fréchette. 2018. "On the determinants of cooperation in infinitely repeated games: A survey." *Journal of Economic Literature* 56 (1): 60–114.

———. 2019. "Strategy choice in the infinitely repeated Prisoner's Dilemma." *American Economic Review* 109 (11): 3929–52.

Danz, David, Lise Vesterlund, and Alistair J Wilson. 2020. *Belief elicitation: Limiting truth telling with information on incentives.* Technical report. National Bureau of Economic Research.

Dawes, Robyn M, and Richard H Thaler. 1988. "Anomalies: cooperation." *Journal of economic perspectives* 2 (3): 187–197.

Fudenberg, Drew, David G Rand, and Anna Dreber. 2012. "Slow to anger and fast to forgive: Cooperation in an uncertain world." *American Economic Review* 102 (2): 720–49.

Fundenberg, Drew, and Eric Maskin. 1990. "Evolution and cooperation in noisy repeated games." *The American Economic Review* 80 (2): 274–279.

Giannakas, Konstantinos, Murray Fulton, and Juan Sesmero. 2016. "Horizon and free-rider problems in cooperative organizations." *Journal of Agricultural and Resource Economics,* 372–392.

Hossain, Tanjim, and Ryo Okui. 2013. "The binarized scoring rule." *Review of Economic Studies* 80 (3): 984–1001.

Imhof, Lorens A, Drew Fudenberg, and Martin A Nowak. 2007. "Tit-for-tat or win-stay, lose-shift?" *Journal of theoretical biology* 247 (3): 574–580.

Ioannou, Christos A. 2014a. "Asymptotic behavior of strategies in the repeated prisoner's dilemma game in the presence of errors." *Artif. Intell. Res.* 3 (4): 28–37.

———. 2014b. "Coevolution of finite automata with errors." *Journal of Evolutionary Economics* 24 (3): 541–571.

Miller, John H. 1996. "The coevolution of automata in the repeated prisoner's dilemma." *Journal of Economic Behavior & Organization* 29 (1): 87–112.

Rand, David G, Drew Fudenberg, and Anna Dreber. 2015. "It's the thought that counts: The role of intentions in noisy repeated games." *Journal of Economic Behavior & Organization* 116:481–499.

Romero, Julian, and Yaroslav Rosokha. 2019. "The evolution of cooperation: The role of costly strategy adjustments." *American Economic Journal: Microeconomics* 11 (1): 299–328.

Zhang, Huanren. 2018. "Errors can increase cooperation in finite populations." *Games and Economic Behavior* 107:203–219.

Appendix A Description of the Belief Elicitation strategy

For this, you will receive either 0 points or 2 points. Your chance to win 2 points depends on both your guess and if the other participant invested in Project A. Specifically, your chance of receiving 2 points is determined in the following way:

1. First, you will guess the probability that the other participant invested in Project A. You will guess a number from 0 to 100, that we convert to a decimal.
2. If the other participant invested in Project A, your chance-to-win 2 points will be: $2z - z^2$ where z is the probability you selected, that the other participant selected Project A
3. If the other participant invested in Project B, your chance-to-win 2 points will be: $1 - z^2$
4. To determine whether you receive 2 points, the computer will randomly draw a number between 0 and 100. Each number between 0 and 100 is equally likely to be picked
5. If the number drawn by the computer is less than or equal to your chance-to-win, then you will receive 2 points. Otherwise, you receive 0 points

Appendix B Experimental Instructions

Instructions for the Noise Treatment

Welcome!

Today's experiment will last about 60 minutes. You will be paid a show-up fee of $5 together with any money you accumulate during this experiment. The money you accumulate will **depend partly on your actions, partly on the actions of others, and partly on chance. Therefore, please read the instructions carefully**. This money will be paid at the end of the experiment in private and in cash.

Your returns will be recorded in points. At the end of the session, the total number of points in your account will be converted into cash at an exchange rate of 300 unit = $1. It is possible for you to get negative points in a round. If at the end of the session you have negative units in your account, you will be paid the show-up fee.

It is important that during the experiment you remain silent. If you have any questions or need assistance of any kind please raise your hand, but do not speak, and an experiment administrator will come to you and you may then whisper your question.

In addition, please turn off your cell phones and put them way now. Please do not look into anyone's booth at any time.

Please read the following instructions carefully. You will be given a quiz at the end to test your understanding and you earn $0.50 for each correct answer.

Agenda:

- Experimental instructions
- Quiz
- Experiment

How a match works

This session is made up of 10 matches between you and other participants in the room. In each match, you play a random number of rounds with another participant. The length of a match is randomly determined in that, after each round, there is a 90% chance that the match will continue for at least another round. Once the match ends, you will be randomly re-grouped with another participant to play another match. Whenever a match ends, you will be informed of this before you are re-grouped.

Decisions and Payoffs (Before Random Draw)

In each match, you will make a series of investments in a project with the same participant. For each round of a match, you can invest in either Project A or Project B. The participant you are playing with has the same two options. You each choose your project at the same time. The returns on investment depends on the project you choose, the project the other participant chooses and a random draw. That is, the returns you get depend on:

- the investment you made (Project A or Project B)
- the investment made by the other participant
- a random draw

The following table summarizes the return you get based on your decision and the other participant's decision:

		Other participant's choice	
		A	*B*
2*Your choice	*A*	**48**,**48**	**13**,**60**
	B	**60**,**13**	**25**,**25**

The first red bolded entry in each cell represents your returns before accounting for the random draw, while the second entry in blue represents the returns of the participant you are grouped with (how the random draw affects you and the other participant's returns will be explained below).Ignoring the random draw, if:

- You invest in Project A and the other participant invests in Project A, your both earn **48 points**
- You invest in Project A and the other participant invests in Project B, you earn **13 points** and the other participant earns **60 points**

- You invest in Project B and the other participant invests in Project B, you both earn **25 points**
- You invest in Project B and the other participant invests in Project A, you earn **60 points** and the other participant earns **13 points**

Your project returns may change depending on a random draw

In each round, after you have invested in a project, your return may change by a random draw. Let's call this random draw v_1. This means that, your return may increase, decrease, or stay the same by an amount v_1. The computer will randomly select this number in each round. This random draw does not depend on the project that you or the other participant choose or the random draw in previous rounds. This draw is completely random.

This random draw will always be a number from -24 to 24. Each number, of the 49 integer values between -24 and 24, is equally likely to occur.

The other participant's return, after they have invested, will also change by a random draw. Let's call this amount v_2. This means that, the returns from their project may increase, decrease, or stay the same by an amount v_2. The computer will generate these integers for you both. **These integers will be completely independent. This means that your random draw is completely unrelated to the random draw of the other participant.**

In the diagram below, there are 500 examples of random draws for you and the other participant, where each dot represents a random draw. If you hover your cursor over one of these dots, you will see a pair of numbers where the first integer (labelled 'yours') represents your random draw and the second integer (labelled 'theirs') is the random draw for the other participant.

Now, the total return you receive is dependent on your random draw AND the choices made by the other participant. The other participant's random draw does not affect your return. With the random draw, the rule for your investment returns now becomes:

Pay close attention to the following information.

For both you and the other participant, taking into account the return and the random draw:

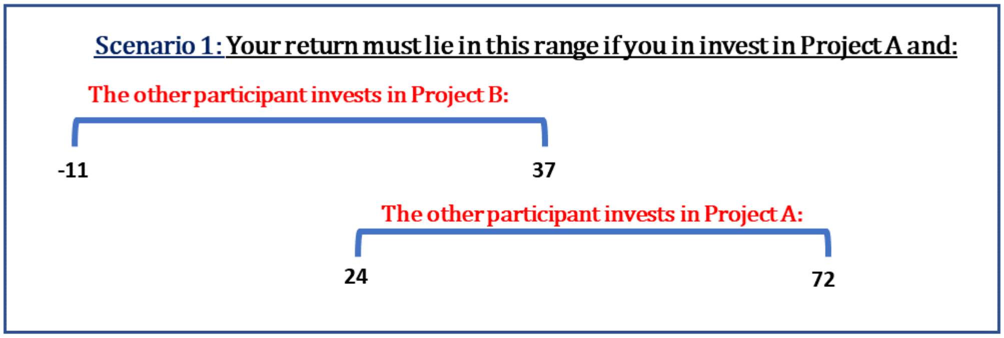
		Other participant's choice	
		A	B
2*Your choice	A	**48** + v_1,**48** + v_2	**13** + v_1,**60** + v_2
	B	**60** + v_1,**13** +v_2	**25** + v_1,**25** +v_2

- the minimum return that can be received is **-11** (the lowest possible return 13 plus minimum possible random draw of -24)
- and the maximum return that can be received is **84** (the highest possible return 60 plus maximum possible random draw of +24).

After you have made an investment choice in A or B, the range of the returns after accounting for the random draw is:

1. **If you invest in Project A and the other participant invests in Project A:**
   - Your return will range from **24** (48 plus worst random draw -24) to **72** (48 plus best random draw +24)
   - The other participant's return will range from **24** (48 plus worst random draw -24) to **72** (48 plus best random draw +24)
2. **If you invest in Project A and the other participant invests in Project B:**
   - Your return will range from **-11** (13 plus worst random draw -24) to **37** (13 plus best random draw 24)
   - The other participant's return will range from **36** (60 plus worst random draw -24) to **84** (60 plus best random draw +24)
3. **Note that if YOUR return falls between 24 and 37, you CANNOT know for sure whether the other participant invested in Project A or B. If YOUR return is less than 24 or greater than 37 you CAN know for sure what project the other participant invested in. (see Scenario 1 in the diagram below).**

1. **If you invest in Project B and the other participant invests in Project B:**
   - Your return will range **1** to **49**
   - The other participant's return will range from **1** to **49**
2. **If you invest in Project B and the other participant invests in Project A:**
   - Your return will range **36** to **84**
   - The other participant's return will range from **-11** to **37**
3. **Note that if YOUR return falls between 36 and 49, you CANNOT know for sure whether the other participant invested in Project A or B. If YOUR return is less than 36 or greater than 49 you CAN know for sure what project the other participant invested in. (see Scenario 2 in the diagram below).**

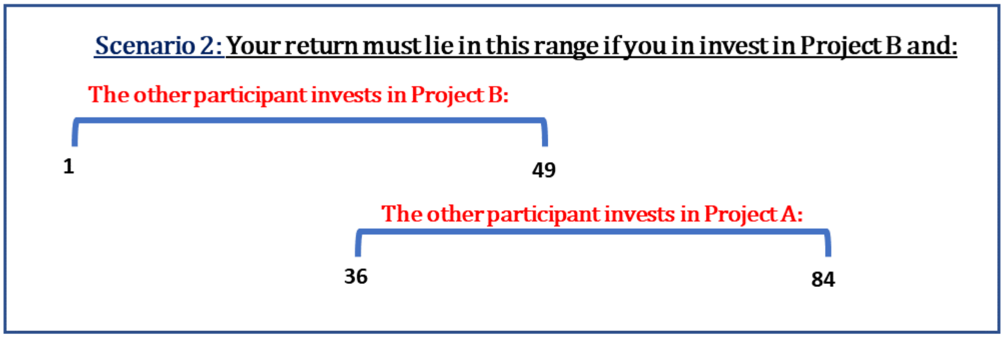


You will get a signal

You will not be told the return of the other participant, but you will always get a signal about their return. This signal will tell if the other participant's return is above, below, or equal to a benchmark value. This benchmark value can help you rule out ranges of values of the other participant's return. You will set one benchmark value that will be used if you select Project A and another if you select Project B. These benchmark values have to be within the range of possible return for both projects for the other participant. That is, 24 to 84 for Project A and -11 to 49 for Project B.

At the beginning of each match, you will be prompted to select these benchmark values. For example, if you set a benchmark value of 50 for Project A, you will be signaled that the other participant's return is above, below, and equal to 50. You only set this benchmark value at the beginning of each match. The same benchmark will be used for all the rounds in a match. When a new match begins, you will be prompted to set the benchmark value again.

Here you can simulate how the benchmark can be used. First select a project, then the computer will randomly select a project as well. Move the slider to see what information you receive about the other participant based on the benchmark you select.



After Each Round

On the result page, **we will ask you what you think the chances are that the other participant chose Project A.**

Depending on your guess, you can earn 2 points or 0 points. We are interested in learning about your best and honest guesses. **You will be paid according to a formula which is specifically designed to maximize the chances that you will win the 2 points if you submit your best guess.**

Your guess will be converted into a chance-to-win. This is calculated by the computer according to a formula that is explained on separate page that you can request after the experiment. On the computer interface, you will be able to see the chance-to-win for each outcome directly below your guess.

You will not be paid for your answer until the end of the experiment. Your answer will not be shown to any other participant. Your answer will not affect the experiment in any way.

The Interface of the Experiment

Before each match, your computer screen will look like this:

After you select these benchmark values, the round will begin. In each round, the screen to select a project for you and the other participant looks like this:

Practice for using the Benchmark [Please take a few minutes to try this!]

Click on a project then use the slider below to see how the information you get about the other participant changes with all possible benchmark values. You can click as many times as you want.

In each cell, the amount to the left and ***bolded in red*** *is the return for you, and the one to the right in* ***blue*** *is for the other participant.*

		The Other Participant	
		A	B
You	Project A	**48** , 48	**13** , 60
	Project B	**60** , 13	**25** , 25

You selected Project A and as a result your return is 23

This slider represents the different benchmark values you can choose

Click to select a benchmark value. Remember to select a project first!!

If you select a benchmark value of 76, you will be told that: The other participant's return is less than your benchmark value

Please select your benchmark values.

You will select two values:

- The benchmark value you want to use if you select Project A (an integer between 24 and 84)
- The benchmark value you want to use if you select Project B (an integer between -11 and 49)

What is your benchmark value when you select Project A:

What is your benchmark value when you select Project B:

Figure 4: Set Your Benchmark Values

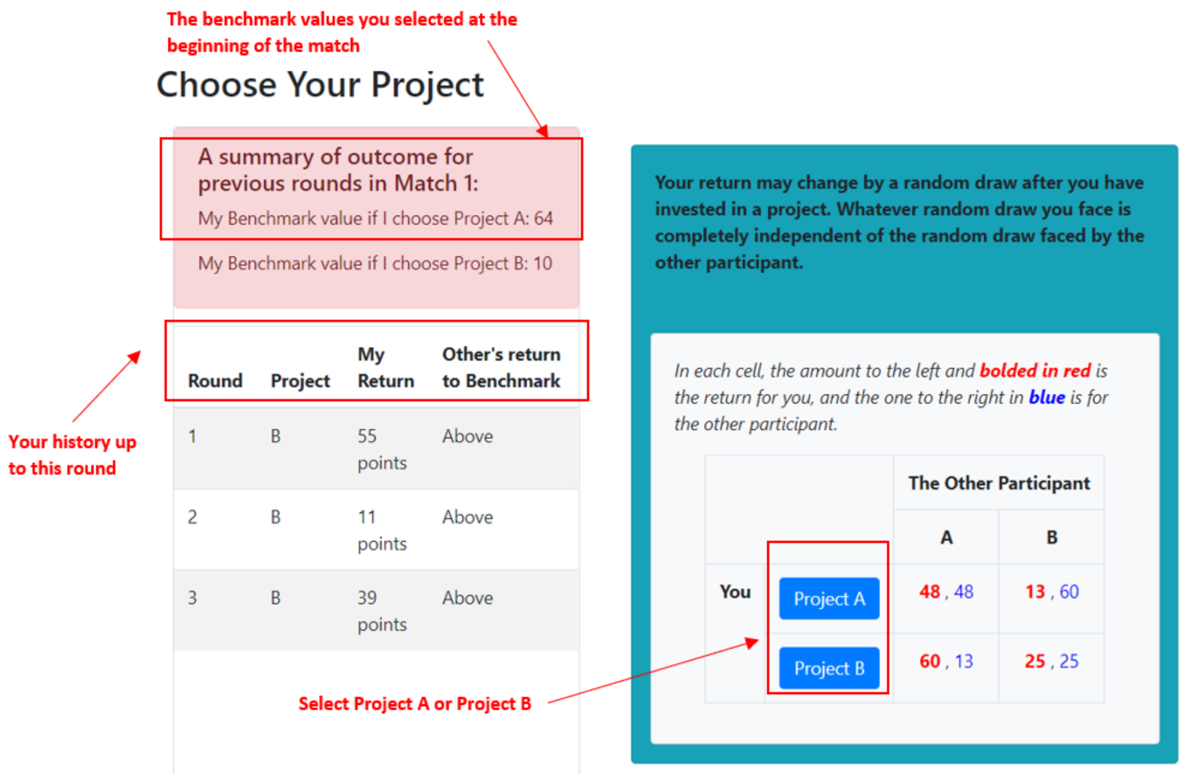


Figure 5: SYour Selection Screen

This screen also displays a summary of the outcome of all previous rounds including: the project you chose; your return (inclusive of the random draw you faced); and if the other participant's return is above, below or equal to the benchmark value you set at the beginning of the match.

After you and the other participant have made a decision, your result screen will display:

- The decision made BY YOU
- YOUR realized returns (inclusive of your random draw).
- If the other participant is above, below or equal to the benchmark
- A slider for you to guess the probability that the other participant selected Project A

This is an example of what the computer screen may look like after you have made your choice:

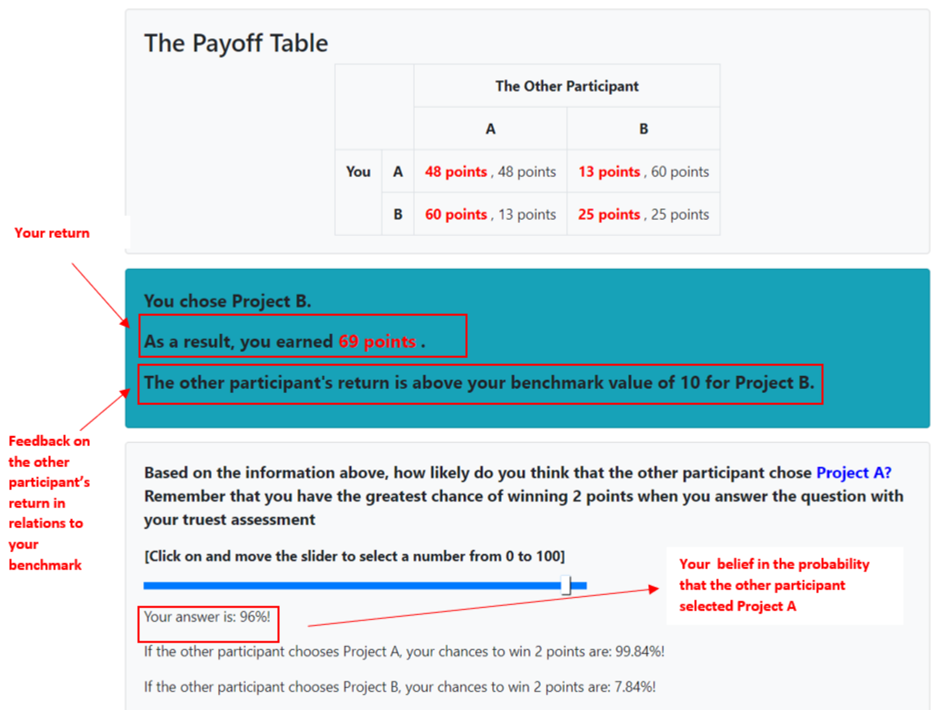


Figure 6: Your Result Screen

Once a match ends, you will be randomly re-grouped with a different participant in the room for another match. Each match has the same setup. You will play a number of such matches with different people in the room.

Reminders

To summarize, the number of rounds in a match is randomly determined. After each round, there is a 90% chance that the match will continue for at least another round. You and the other participant will get a random draw. Whatever random draw you get is completely independent of the random draw faced by the other participant. This means that your random draw is completely unrelated to the random draw of the other participant.

You will not know the return of the other participant, but you will be able to select a benchmark value to signal to you the possible range of returns for the other participant. After you both have invested in a project, you will be told if the other participant's return was above, below or equal to this benchmark value. At the end of this session, you will receive $1 for every 300 point in your account. You will now take a very short quiz to make sure you understand the setup. You will earn $0.50 for each correct answer.

After the quiz, you will play 4 practice rounds to get you familiarized with the game. For the practice rounds, you will play against the computer and NOT the other participants. The computer will randomly select responses. Also, you will be able to select benchmark values for each round. This ONLY happens for the practice rounds. After the practice rounds, you will begin playing with the other participants in the room.

Appendix C Description of Strategies

Strategy	Description
AD	Always defect
DTFT	Defect in the first round, then play TFT
DGRIM2	Defect in the first round, then play GRIM2
GRIM	Cooperate until the other player defects, then defect forever
TFT	Cooperate unless other player played defection in the last round
2TFT	Cooperate unless other player defected in either of the last two rounds
GRIM2	Cooperate until the other player defects in 2 consecutive rounds, then defect forever
GRIM3	Cooperate until the other player defects in 3 consecutive rounds, then defect forever
TF2T	Cooperate unless other player defected in both of the last two rounds
AC	Always cooperate

Appendix D Feedback After Supergame

Match 1 has ended. Your cumulative decision and payoff for this match is:

All periods:

Rounds	Your Decision	Your Return	Other Decision	Other Return
1	A	-8 points	B	39 points
2	B	38 points	A	-10 points
3	A	35 points	B	82 points
4	B	66 points	A	20 points
5	B	51 points	A	-5 points
6	A	50 points	A	52 points
7	A	41 points	A	39 points
8	B	53 points	A	17 points
9	A	65 points	A	60 points

Appendix E The Maximum Likelihood Estimation Method

We use the Strategy Frequency Estimation Method (SFEM) from Dal Bó and Fréchette (2011) to estimate the fraction of strategies employed in each treatment. This methodology uses a Maximum Likelihood Estimation (MLE) to estimate the frequency with which each strategy from a set of pre-determined set of strategies is found experimental data. This methodology has since been employed Fudenberg, Rand, and Dreber (2012), Rand, Fudenberg, and Dreber (2015), Dal Bó and Fréchette (2018), Aoyagi, Bhaskar, and Fréchette (2019), Dal Bó and Fréchette (2019) and Romero and Rosokha (2019), for example. This method assumes that each subject uses the same strategy across supergames. However, they can make mistakes. These mistakes are not the errors that are generated from the experimental design, but rather, it is assumed that subjects can make mistakes when choosing their intended actions for the particular strategy they are following.

Using the notations of Dal Bó and Fréchette (2011), assume that the probability with which subject i makes mistakes is $1-\beta$ and the probability that her chosen actions correspond with a strategy k is β. The likelihood that her observed choices were actually generated by strategy k is $Pr_i(s^k) = \Pi_{M_i}\Pi_{R_{im}}(\beta)^{I^k_{imr}}(1-\beta)^{1-I^k_{imr}}$. I^k_{imr} is an indicator function that takes the value 1 when the choice that was actually made in round r and supergame m is the same as what the subject would have made if she were following strategy k. It is coded 0 otherwise. M and R are the sets of supergames and rounds. β is estimated within the model. It can also be interpreted as the probability that an action is taken given that it is prescribed by a strategy k. Therefore β is the basis for evaluation of model fit, that is, as the model fit improves β approaches 1.

Therefore, the MLE process entails choosing both the probability of mistakes and the frequency of strategies that maximizes the likelihood of the sequences of choices. That is, the log-likelihood is $\Sigma_I ln(\Sigma_K \phi^k Pr_i(s^k))$, where K is the subset of strategies being considered and ϕ^k is a vector of parameter estimates that represent the frequency of strategies.

We bootstrapped the standard errors in a way that respects the data generating process of our experimental data. We randomly draw the appropriate number of sessions, then for each session the appropriate number of subjects, then supergames. All with replacement. The bootstrapping process was done 1000 times. The standard deviation of the bootstrapped MLE estimates provide the standard errors.