Learning about Initial Play Determines Average Cooperation in Repeated Games[*]

Drew Fudenberg[1] and Gustav Karreskog[2]

[1]Department of Economics, MIT
[2]Department of Economics, Stockholm School of Economics

March 2, 2021

Abstract

We propose a simple learning model to predict cooperation rates across treatments in the experimental play of the indefinitely repeated prisoner's dilemma. Using data from 28 treatments gathered from 16 papers, we find that our 6 parameter model performs at least as well as more complicated models and better than machine learning algorithms. We find that learning has the most effect on choices in the initial round of each supergame, and that whether cooperation rises or falls in the course of a session depends on the way the initial choices in a supergame determine play in subsequent rounds. Our results also explain and sharpen past findings on the role of strategic uncertainty.

Keywords: cooperation, prisoner's dilemma, risk dominance, predictive game theory

[*]We thank Anna Dreber Almenberg, Mathias Blonski, Yves Breitmoser, Ying Gao, Annie Liang, Indira Puri, Emanuel Vespa, Jörgen Weibull, David Rand, and seminar participants at UVA and Goethe University Frankfurt for helpful comments. NSF grants SES 1643517 and 1951056 provided financial support.

1 Introduction

Determining when and how people overcome short-run incentives to behave cooperatively is a key issue in the social sciences. The theory of repeated games has determined which factors allow cooperation as an equilibrium outcome, but since these games typically also have equilibria where people do not cooperate, equilibrium theory on its own is not a useful way of making predictions about cooperation rates. Moreover, the assumption that people play the most cooperative equilibrium possible, which is often used in applications, is a very poor fit for observed behavior in the laboratory. It is therefore important both for policy decisions and as a guide for the development of more useful theories to have a better understanding of how cooperation rates in experimental play of repeated games depend on their parameters.

To that end, we treat the relation between cooperation rates in the experimental play of the prisoner's dilemma and its exogenous parameters as a prediction problem. We formulate and evaluate a very simple model of reinforcement learning, where all that varies with treatment or personal experience is the probability of cooperating in the first round of a new match. After these initial rounds, play depends only on the outcome of the previous round: If both players cooperated they keep cooperating, and if they both defected they keep defecting. If they mismatch, i.e., one player cooperated and one defected, they both cooperate with roughly 1/3 probability (what is called a semi-grim strategy in Breitmoser (2015).)

With this model, the way that people play in the first round of their very first supergame depends on a composite parameter Δ^{RD} that proxies the effect of strategic uncertainty by the difference between the actual discount factor of the game and the discount factor that makes players indifferent between the strategies Grim and Always Defect in a hypothetical game that we explain below. The initial choices in a supergame and the fixed strategy in subsequent rounds of the supergame determine the payoffs in that supergame. Initial-round play in following supergames depends on both Δ^{RD} and on the initial actions and associated overall payoffs the player received in past supergames.

Using data from the 103 experimental sessions gathered in Dal Bó and Fréchette

(2018) as well as 40 sessions in papers published since then, we show that average cooperation within a given supergame is mostly determined by the actions chosen in its initial round, and that the distribution of the initial actions changes over the course of an experimental session as the participants learn from feedback. As we detail in Section 3, past work has already found evidence that most players use memory-1 strategies and that overall cooperation rates depend on Δ^{RD}. In our preliminary data analysis, we sharpen the latter conclusion by finding that cooperation tends to increase over the course of a session when $\Delta^{RD} > 0.15$, and to decrease when $\Delta^{RD} < 0$. Our learning model predicts this pattern, which suggests that the reason for the observed impact of the composite parameter Δ^{RD} is its effect on the reinforcement of cooperation in the initial round of each supergame. Moreover, according to our model, the direct effect of game parameters on cooperation rates is much smaller than their indirect effect through learning. As a consequence, participants in a session can behave differently even if they follow the same learning model.

The learning model predicts both average cooperation and the time-path of cooperation in a session better than any of the black-box methods we consider. Furthermore, we find that allowing for heterogeneous agents, or a more complex learning model with learning at all memory-1 histories, give no noticeable improvements.

The learning model also allows us to predict what would happen with longer experimental sessions. Here we find that even in the very long run, high rates of cooperation are predicted only when its benefit is high compared to its risk. For intermediate values, substantial shares of initial round cooperators and initial round defectors coexist in the population.

To further evaluate our learning model, we then consider the problem of predicting the next action that a participant will play, which is closely related to the commonly-studied task of identifying the strategies used by the participants. We find that the naive rule of simply predicting that a player's current action will be the same as their previous one fits quite well. Moreover, our six-parameter learning model does as well at predicting behavior as a pure strategy model that incorporates 12 different pure strategies estimated separately for each treatment. Adding a second type that always plays D (and an accompanying eighth parameter for its share) leads to better predictions. At the cost of using more parameters and possibly making the model less robust, we can improve the model yet again by allowing for multiple types with different behaviors at the non-initial histories.

This simple learning model does not use individual characteristics as data, so there are regularities it cannot capture. Indeed, Proto, Rustichini and Sofianos (2019) show that intelligence, and to a lesser extent, other personality traits, affect how people play infinitely repeated games. However, the learning model is parsimonious and portable, and does a good job at both overall predicting average cooperation and its time path.

2 Preliminaries

In the experiments we analyze, participants played a sequence of repeated prisoner's dilemma games with perfect monitoring.[1] The game parameters were held fixed within each session, so each participant only played one version of the repeated game. The treatments all had randomly chosen partners and a random stopping time, so the discount factor δ corresponds to the probability that the current repeated game ends at the end of the current round. (We will refer to the "rounds" of a given repeated game, and call each repeated game a new "supergame.")

We represent the prisoner's dilemma with the following strategic form, where $g, l > 0$ and $g < l + 1$. Here g measures the gain to defection when one's opponent cooperates, l measures the gain to defection when one's opponent defects, and $g < l+1$ implies that the efficient outcome is (C, C).

	C	D
C	$1, 1$	$-l, 1+g$
D	$1+g, -l$	$0, 0$

Figure 1: The Prisoner's Dilemma

Standard arguments show that "Cooperate every round" is the outcome of a

[1] There are many more experiments on this case than on the prisoner's dilemma with implementation errors or imperfect monitoring.

subgame-perfect equilibrium if and only if it is a subgame-perfect equilibrium (SPE) for both players to use the strategy “Grim”: Play C in the first round and then play C iff no one has ever played D in the past. This profile is a SPE iff

$$1 \geq (1-\delta)(1+g) \iff \delta \geq g/(1+g) \iff \delta \geq \delta^{\text{SPE}}.$$

Note that the loss l incurred to (C, D) does not enter in to this equation, because the incentive constraints for equilibrium assume that each player is certain their opponent uses their conjectured equilibrium strategy.

Applied theoretical work on repeated games often assumes that players will cooperate whenever cooperation can be supported by an equilibrium,[2] but this hypothesis has little experimental support. Instead, the level of cooperation in repeated game experiments can be better predicted by measures that reflect uncertainty about the opponents' play. In particular, Grim is risk dominant in a 2x2 matrix game with the strategies Grim and Always Defect iff

$$\delta \geq (g+l)/(1+g+l) \equiv \delta^{\text{RD}}.$$

The composite parameter referred to above is the difference between the actual discount factor and this threshold:

$$\Delta^{RD} = \delta - \delta^{\text{RD}} = \delta - (g+l)/(1+g+l).$$

Inspired by previous work and descriptive evidence we present later, we develop a very simple model that assumes all individuals use memory-1 strategies, and moreover that these strategies differ across treatments and supergames only with respect to play in the initial round of each supergame. We assume that this initial behavior is driven by two different components: a direct effect of game parameters captured by the linear function $\alpha + \beta \cdot \Delta^{RD}$ and the effect of reinforcement learning captured by individual experience $e_i(s)$. We assume that the cooperation on the initial of each supergame, $p_i^{initial}(s)$, depends on the game parameters and the effect of individual

[2] See e.g. Rotemberg and Saloner (1986), Athey and Bagwell (2001), and Harrington (2017).

experience $e_i(s)$:

$$p_i^{initial}(s) = \frac{1}{1 + \exp\left(-(\alpha + \beta \cdot \Delta^{RD} + e_i(s))\right)}. \tag{1}$$

To model learning, we suppose that after each supergame s, $e_i(s)$ is updated according to

$$e_i(s) = \lambda \cdot a_i(s-1) \cdot V_i(s-1) + e_i(s-1), \tag{2}$$

where $a_i(s) \in \{-1, 1\}$ is the action taken, $V_i(s)$ is the total payoff received in supergame[3] s, λ determines the strength of the learning , and $e_i(1) = 0$. Cooperation or defection in the initial round is thus reinforced depending on the resulting supergame payoffs, while the direct influence of Δ^{RD} is constant across supergames.

We assume that behavior at non-initial rounds follows a memory-1 mixed strategy that is constant across individuals, treatments, and time. Let $h \in \{CC, DC, CD, DD\}$ denote a memory-1 history, and let σ_h be the probability of cooperation at one of these histories. Following Breitmoser (2015) we assume these follow a semi-grim behavior where $\sigma_{CC} > \sigma_{DC} = \sigma_{CD} > \sigma_{DD}$. In section 5.6 we relax this assumption and consider multiple extensions and relaxations, but see that this does not improve predictions. In total, our model thus has 6 parameters, $(\alpha, \beta, \lambda, \sigma_{CC}, \sigma_{CD/DC}, \sigma_{DD})$.

Importantly, in our main analysis we do not make predictions based on the actual payoffs that participants received, but rather on simulations that suppose all participants used learning rules of the form (1) and (2). (We do use the actual payoffs when we turn to the problem of predicting the next action a given participant will play.)

3 Prior Work

Blonski, Ockenfels and Spagnolo (2011), Rand and Nowak (2013), and Blonski and Spagnolo (2015) show that the average cooperation rates in a session are increasing in

[3] We also tried an alternative specification where learning responds to the average payoff in a supergame instead of the total, but it performed less well. This suggests that learning between supergames is stronger when the supergames are longer.

Δ^{RD}. Dal Bó and Fréchette (2018) show that the sign of Δ^{RD} is much more correlated with high cooperation rates than the sign of $(\delta - \delta^{\text{SPE}})$.[4]

Several papers have attempted to estimate the strategies used by participants on the assumption that each participant uses a fixed strategy either in the entire session or in the latter part of it. A consistent finding in the papers that assume the use of pure strategies is that most of the behavior can be captured by the strategies AllD (Always Defect), TFT (Play C in the initial round of a supergame, and thereafter play the action your partner played in the previous round), Grim (Play C in the initial round and thereafter play D if either partner has ever defected), and for lower values of Δ^{RD}, D-TFT (play D in the initial round and thereafter play what your partner played in the previous round.) See for example Dal Bó and Fréchette (2011); Fudenberg, Rand and Dreber (2012); Dal Bó and Fréchette (2018). In Romero and Rosokha (2018*a*) and Dal Bó and Fréchette (2019), the pure strategies used are elicited from the participants instead of being estimated. Those studies confirm the finding that a small set of memory-1 strategies are enough to capture most of the strategies used.[5]

Recently, studies have found evidence for the use of constant memory-1 mixed strategies. Breitmoser (2015) finds that strategies of the form "semi-grim" better fit play after the initial round than pure strategies do. These strategies are defined by $\sigma_{CC} > \sigma_{CD} = \sigma_{DC} > \sigma_{DD}$. Backhaus and Breitmoser (2018) follows up on this analysis by more carefully considering alternative models and behavior in the initial round. The authors argue that a combination of AllD and semi-grim, with the mixture estimated treatment by treatment, best fits behavior, and that only initial round behavior responds to incentives. We will use strategies similar to this semi-grim form in our main learning model.[6]

[4] Dal Bó and Fréchette (2011) use the alternative measure $\frac{(1-\delta)l}{1-(1-\delta)(1+g-l)}$ as a regressor; it is very correlated with Δ^{RD}.

[5] Fudenberg, Rand and Dreber (2012)show that longer memories are used when the intended actions are implemented with noise and only the realized actions are observed.

[6] Romero and Rosokha (2018*b*) elicits memory-1 mixed strategies and finds that a finite mixture of elicited mixed and pure strategies matches behavior better than pure strategies. In their data, $\sigma_{CD} = .45, \sigma_{DC} = .35$ and they reject semi-grim's restriction that $\sigma_{CD} = \sigma_{DC}$; in our larger data set $\sigma_{CD} = .31, \sigma_{DC} = .34$.

Dal Bó (2005) and subsequent work has established that behavior changes between the first and last supergame in a session. Moreover, Dal Bó and Fréchette (2011) argue that δ has no apparent effect on behavior in the first supergame, but a substantial impact on later supergames. Similarly, the difference between treatments increases over time, with average cooperation going down in games where no cooperative SPE exist, and going up in games where Δ^{RD} is high.

A common explanation for the observed time trends is that participants learn from feedback over the course of a session, and choose their supergame strategies based on outcomes in the previous supergames. Dal Bó and Fréchette (2011) considers a simple belief learning model involving only TFT and AllD. [7]

Two other established empirical regularities related to learning are that cooperation in the first round is increased if the realized length of the previous supergame is longer than expected, and if the partner cooperated in the first round of the previous supergame (Engle-Warnick and Slonim, 2006; Dal Bó and Fréchette, 2011, 2018). These two effects also point to a model of behavior where some form of reinforcement or learning drives cooperation in the initial round.

The larger literature on learning in games has been focused on one-shot games, for example in (Cheung and Friedman, 1997; Erev and Roth, 1998; Camerer and Ho, 1999), and has not emphasized the issue of out-of-sample prediction. Fudenberg and Liang (2019) and Wright and Leyton-Brown (2017) study ways to predict initial play in matrix games, but don't consider learning.

4 Summary of the data

We analyze the meta-data from Dal Bó and Fréchette (2018), who included experiments on the repeated prisoner's dilemma with perfect monitoring, deterministic payoffs, and constant parameters within a session that were published before 2014, all of which . We consider only their treatments with $\delta > 0$. We extend this data with data from sessions that match these criteria from four papers published since

[7] Erev and Roth (2001), Hanaki et al. (2005) and Ioannou and Romero (2014) study learning *within* supergames.

then, (Aoyagi, Bhaskar and Fréchette (2019); Dal Bó and Fréchette (2019); Proto, Rustichini and Sofianos (2019); Honhon and Hyndman (2020)) increasing the number of observations with approximately 40%. Our resulting data set contains observations from 16 different papers, 28 different treatments[8] and 143 incentivized experimental laboratory sessions, with 2,432 distinct participants and 205,468 individual choices. Here we highlight some aspects of the data that are of particular relevance to our work.

The discount factors ranged from 0.5 to 0.95. In 17 of the sessions, $\delta < \delta^{\mathrm{SPE}}$, so no cooperation can occur in a subgame perfect equilibrium. In 22, cooperation can be supported by a SPE, i.e. $\delta > \delta^{\mathrm{SPE}}$, but $\delta < \delta^{\mathrm{RD}}$, so it is not risk dominant in the sense of Blonski, Ockenfels and Spagnolo (2011). In the remaining 104 sessions, $\delta > \delta^{\mathrm{RD}}$.

The average rate of cooperation over all sessions was 44.2%. It was 11.4% for games where $\delta < \delta^{\mathrm{SPE}}$, 18.8d% for $\delta^{\mathrm{SPE}} < \delta < \delta^{\mathrm{RD}}$, and and 52.4% for $\delta > \delta^{\mathrm{RD}}$.

The average play after the different memory-1 histories, and their frequencies, are shown in table 1. We see that the CD and DC histories are only a small subset of observations, roughly 15% together. Furthermore, we see that the average behavior is closed to the semi-grim memory-1 mixed strategy from Breitmoser (2015), with the difference that the probability of cooperation is slightly higher after DC, than CD.

Table 1: Average cooperation rate after different memory-1 histories.

History	Avg C	N
CC	0.965	52 769
CD	0.312	15 254
DC	0.339	15 256
DD	0.054	66 359
Initial	0.472	55 830
Total	0.442	205 468

To visualize how behavior differs depending on Δ^{RD} we group the sessions in the

[8] Following Dal Bó and Fréchette (2018),we consider experiments in different labs to be the same treatment if they had the same normalized parameters. The total number of unique paper parameter combinations is 41.

five groups: $\delta < \delta^{SPE}$, $\delta^{SPE} < \delta < \delta^{RD}$, $0 < \Delta^{RD} < 0.15$, $0.15 < \Delta^{RD} < 0.3$, and $0.3 < \Delta^{RD}$. Here the first 2 groups were motivated by theory, while the subdivision of the treatments with $\Delta^{RD} > 0$ was based on a look at the data. The thresholds and relative frequencies of Δ^{RD} can be seen in figure 2.

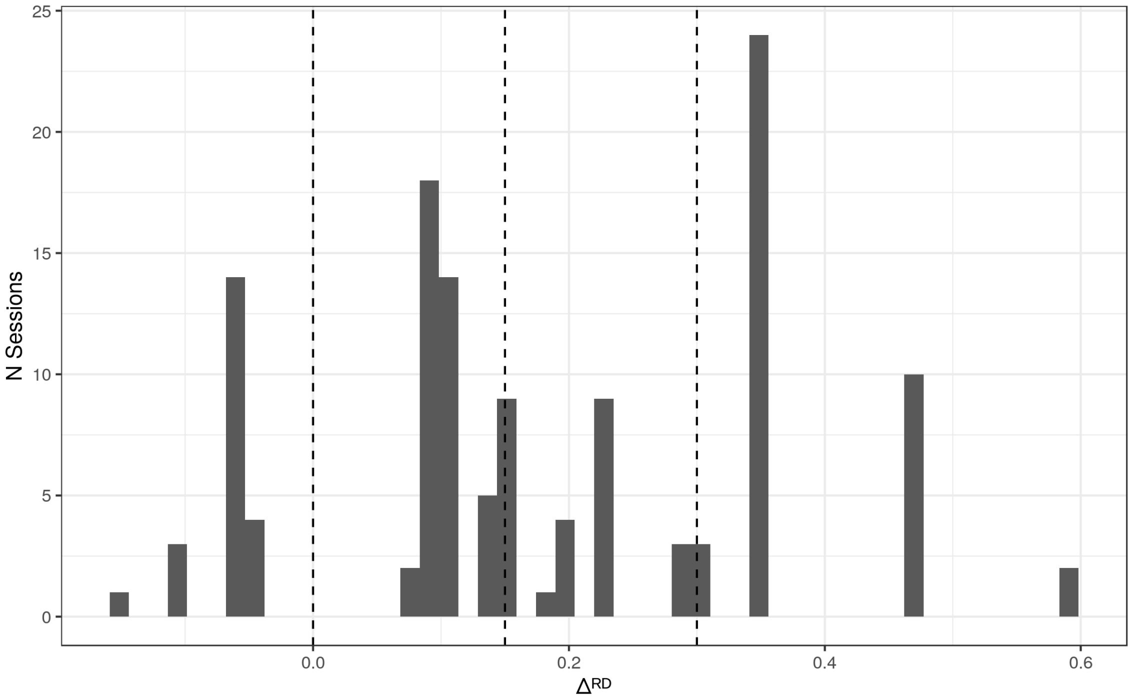

Figure 2: Distribution of Δ^{RD} in experimental sessions with $\delta > \delta^{SPE}$. The dashed vertical lines show to the thresholds for the groups.

Figure 3 shows the evolution of cooperation during the first 10 supergames, restricted to sessions of at least 10 supergames (116 of 143), and in figure 4 the first 20 supergames restricted to the sessions that included at least 20 supergames (75 of 143).

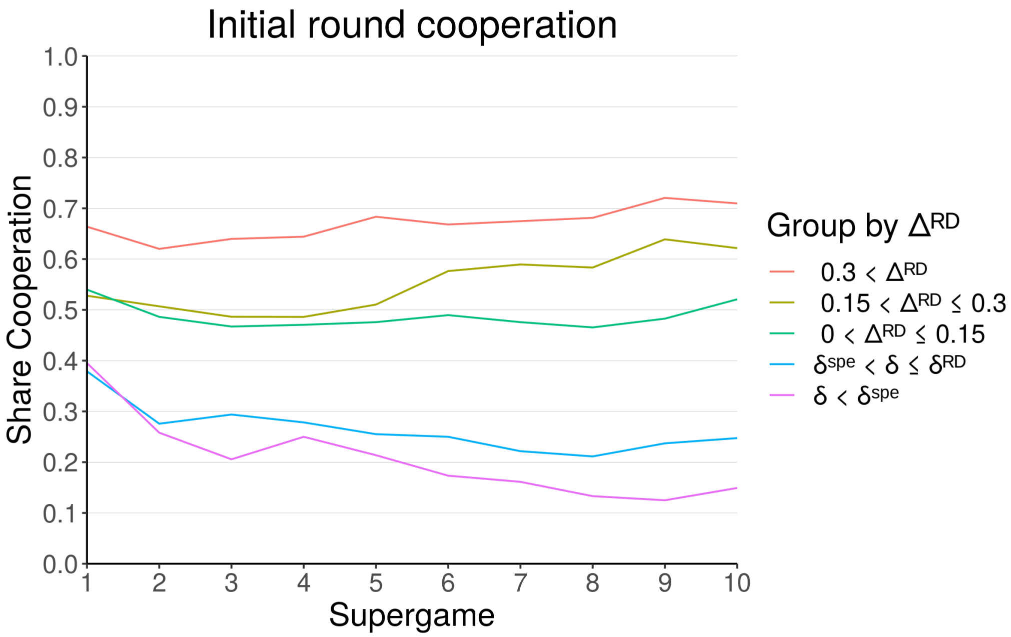

Figure 3: Cooperation in the initial round over the 10 first supergames.

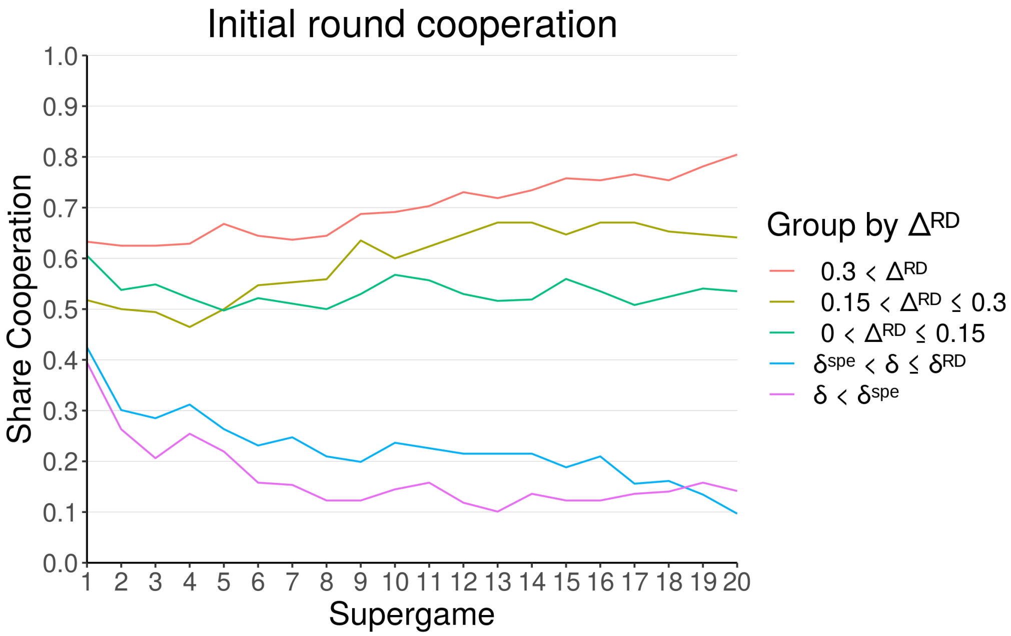

Figure 4: Cooperation in the initial round over the 20 first supergames.

The average rate of cooperation in the initial round of the supergames differs across the treatment groups. For $\Delta^{RD} > 0.15$ cooperation rates increase over the course of a session while for $\Delta^{RD} < 0$ they decrease. For sessions where cooperation is only marginally risk dominant, $0 < \Delta^{RD} < 0.15$, cooperation rates remain roughly constant at around 50%. As we see in online appendix this pattern is less sharp after the other memory-1 histories. This suggests that the differences in average cooperation across different treatments is primarily driven by differences in the behavior at the initial round.

As a further illustration of this, we look at the average cooperation in the non-initial rounds of the supergames. For each participant and supergame that was played at least two rounds, we let the outcome variable be the average cooperation by that participant in the non-initial rounds. As reported in the online appendix, we then consider three different regressions. In the first we only condition on the outcome of the initial round, in the second we add the game parameters (g, l, δ, and Δ^{RD}), and in the last we remove the outcome of the initial round. The difference in R^2 between the first and second regression is less than 0.01, while the third regression has a much lower R^2. Doing the same regressions but with the second round average cooperation as the outcome does not change the picture.

Summing up, the outcome of the initial round is highly predictive of the cooperation in the rest of the supergame, and taking into account the game parameters does not improve those predictions. Similarly, we will find that the fit of our simple learning model is not improved by allowing the cooperation probabilities at non-initial histories to depend on game parameters. We find this somewhat surprising, but do not have a good explanation.[9] It is possible that the reason the game parameters seem to only matter in the initial rounds is somehow driven by selection or interaction effects that we fail to correct for, or that adding an additional non-linear function of game parameters to Δ^{RD} would improve predictions, but none of these possibilities seem likely to eliminate the striking impact of initial play.

[9] Backhaus and Breitmoser (2018) also find that play after the initial round is relatively insensitive to the game parameters.

5 Predicting Cooperation

5.1 Making and Evaluating Predictions

Our goal in this paper is to develop models that can successfully predict cooperation levels in repeated game experiments. We consider versions of this prediction task: Predicting average cooperate in a given session, and predicting the time-path of cooperation over the course of a session. The first task is in some sense more fundamental, but addressing the second one will let us make predictions about what cooperation levels would be if sessions were longer.

We evaluate models based on their estimated out of sample predictive performance as measured by cross-validated mean squared error (MSE). Using cross-validation helps prevent overfitting, and makes sure that the regularities we find actually improve predictions. In general, out of sample predictions will favor models that rely on stable predictors and do not overfit the data, and such models are typically simpler than those that give the best in-sample fit.

In addition, using out of sample prediction error as the benchmark makes it easy for us to compare models of different complexities, because the out of sample prediction criterion endogenously penalizes models that are too complex. We also report the relative improvement of the models compared to a constant prediction benchmark, in order to get a better sense of how big the differences are. [10]

When we try to predict the average cooperation level in a session, each session is a single data point. Here the feature set consists of Δ^{RD}, the game parameters (g, l, δ), the total number of rounds played in the session, the number of supergames played in the session, the sequence of supergame lengths, an indicator variable for whether $\Delta^{RD} > 0$, and some interaction terms. When we predict the time path of cooperation, a data point is the average (across participants) cooperation level on each round of each supergame in a session; this gives a total of 12,915 data points, though the data within each session is highly correlated. Here we start with the same features as for

[10] This use of a simple prediction rule as a benchmark is inspired by the completeness measure of (Fudenberg et al., 2020) but we do not have enough data to estimate the problem's irreducible error as the completeness measure requires.

predicting average cooperation, and then add an indicator for the initial round, the round number, and the supergame number, along with some interaction terms.

As we will see, Δ^{RD} is a strong predictor of per-session average cooperation. While a larger set of features and more complicated predictive algorithms do improve predictions slightly, a linear function of Δ^{RD} captures most of the difference between a constant prediction benchmark and the best performing black-box predictions.

5.2 Predicting Cooperation with Learning

To make predictions with the learning model, we simulate populations playing the different sessions assuming they behave according to the learning model. We make predictions using only the game parameters and the sequence of supergame lengths. In particular,we use the simulations to generate the experience levels e_i, and do not use data on the payoffs that people actually received in the sessions.

Our main learning model, presented earlier in equations (2) and (1), assumes all agents use the same learning rule, which is an over-simplification. In particular, past work has shown that in most experiments there is a non-negligible share of people who defect all or almost all of the time. As we show in section 7, adding a share of such individuals improves the prediction of the next action played, but as it does not improve predictions of the overall average cooperation rates we do not include them in the estimates we report here.

The restriction to memory-1 strategies is motivated by past work, and also by our machine learning analysis in Section 7. The assumption that play across treatments is the same except in the initial rounds is motivated by the descriptive statistics. Specifically, we assume that play at each non-initial memory-1 history follows the same semi-grim strategy.

We relax the assumption of fixed behavior at non-initial histories in section 5.6, which considers a more richly-parameterized model that lets play at these histories depend on Δ^{RD}. We also consider a model that extends learning to those non-initial histories. Neither of these extensions improve predictions.

5.3 Estimating the Learning Model

To generate the learning model's predictions for an experimental session ζ, we take as input the game parameters $\Gamma(\zeta)$ and the realized sequence of supergame lengths $S(\zeta)$. We initialize a large population of individuals, all with $e_i(1) = 0$. For a given specification of parameters of the learning model, we randomly match these simulated individuals to play a sequence of supergames. After the first supergame, the individual experiences $e_i(2)$ are updated according to (2), using the simulated values. The learning thus takes place between supergames. The individuals are then randomly re-matched and play the second supergame for the number of rounds it was played in the experimental session. So it continues until we have simulated a population playing exactly the same sequence of supergames as in the experimental session, updating the experience $e_i(s)$ after each supergame. Thus the sequence of supergame lengths, and the total number of rounds in the session, enter the predictions through this simulation procedure.

Once we have simulated a population like this, we can calculate either average cooperation or the time-path of cooperation, that is the percentage of participants who cooperate in each round $1, 2, ...$ of any supergame in a given treatment. We use the simulations as predictions and compute the approximate prediction losses and associated standard errors as we did for the black-box models.

We estimate the learning model based on the time-path of cooperation, even when predicting average cooperation. That is, we find the parameters that best predict the *time-path of cooperation* in the training set, and use those parameters to predict both the average cooperation and the time-path of cooperation in the test sets. This way, the variation in parameter estimates between folds is smaller, and as a result we get better out of sample predictions.

Appendix B gives a detailed description of the numerical process. In Appendix E we evaluate this estimation procedure on data simulated under different assumptions about how people actually behave.

5.4 Results

In table 2 the out of sample prediction errors for average cooperation are shown for the best- performing atheoretical prediction method (Lasso) and our learning model. We see that our learning model is in fact better than the atheoretical prediction algorithms given the features we let the algorithms use. We also see that a linear function of Δ^{RD} is a strong predictor of behavior. We will return to question of the relationship of Δ^{RD} and average cooperation in subsection 5.5.

Model	Avg C MSE	S.E.	Relative Improvement
Constant	0.0484	(0.0014)	0.0%
OLS on Δ^{RD}	0.0183	(0.0006)	62.19%
Lasso	0.0154	(0.0006)	68.18%
Learning with semi-grim	0.0147	(0.0005)	69.82%

Table 2: Out of sample prediction MSE for average cooperation in an experimental session.

To estimate the out of sample prediction errors, we use 10-fold cross validation. This means we divide the sessions into 10 different folds. We split the data on the level of the session, so each observation is predicted using only data from other sessions. For each fold, we use the other nine folds as a training set to estimate the parameters, and make predictions on the test fold using those parameters. [11] To estimate the standard errors of the estimated mean squared error (MSE), we do 10 different such cross-validations, leading to a total of 100 MSEs estimated on different folds. Using these 100 different values we estimate the standard errors of the out of sample MSE prediction error. By using the same folds for all models we can perform pairwise comparisons. Pairwise tests are presented in D. According to those pairwise tests, our learning model is indeed significantly better than the atheoretical prediction algorithms. [12]

[11] See e.g. Hastie, Tibshirani and Friedman (2009) for an explanation of cross-validation.

[12] The same data is used to estimate the model multiple times, so there are no asymptotic guarantees that these standard errors will match the true standard errors. We also consider non-parametric

At first glance, it might seem surprising that our learning model outperforms the ML algorithms. Part of the explanation is that our data set is relatively small by machine learning standards. In addition, our learning model is able to better incorporate the effect of the realized supergame lengths, as can be seen by what happens when we redo the estimations without using the realized supergame lengths– we randomly generate supergame lengths when simulating the learning model, and simply remove that information about from the ML feature set. This increases the out of sample MSE for both the learning model and the best performing ML-algorithm to 0.0163.

Not only is our proposed learning model better than alternatives at predicting average cooperation in a session, it is also better at predicting the time-path of cooperation: We better predict not only how much the participants cooperate on average, but how behavior evolves across and within supergames.

In table 3 we see similar results for predicting the time-path of cooperation.

Model	Time-path MSE	S.E.	Relative Improvement
Constant	0.0705	(0.0015)	0.0%
OLS on Δ^{RD}	0.038	(0.0008)	46.10%
GBT	0.0333	(0.0007)	52.77%
Learning with semi-grim	0.0319	(0.0007)	54.75%

Table 3: Out of sample prediction loss for predicting the time-path of cooperation.

Figure 5 shows the out of sample predictions and actual values of cooperation in the initial round of the first 20 supergames. (We plot the initial round to reduce the noise introduced by changing supergame lengths.) To get the out of sample predictions, we use a cross-validation split and then predict each session's time path with the parameters estimated without data from that session. The learning model predicts the general pattern well, but it slightly underestimates the level of cooperation for the intermediate values of Δ^{RD}.

pairwise tests.

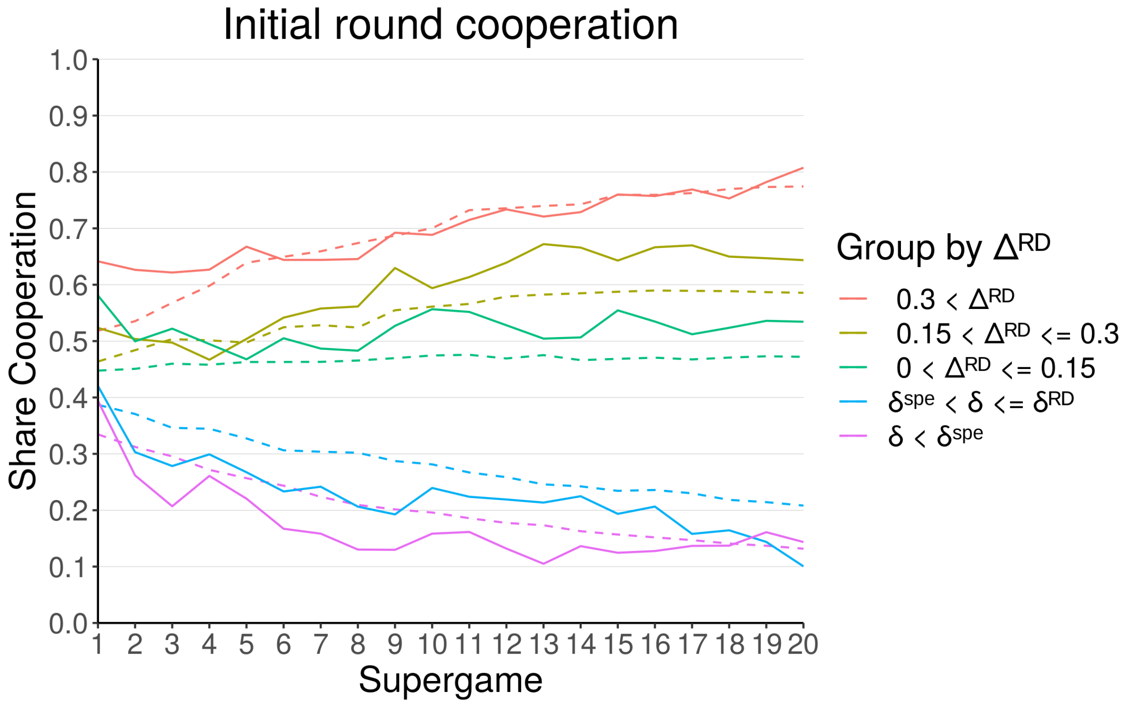

Figure 5: Actual (solid line) and out of sample predicted (dashed line) initial-round cooperation by supergame for sessions of at least 20 supergames .

In the table we show the average of the estimated parameters with standard deviations.

Parameter	α	β	λ	p_{CC}	$p_{CD/DC}$	p_{DD}
Average	-0.313	1.294	0.196	0.994	0.373	0.016
Standard Deviation	(0.063)	(0.156)	(0.043)	(0.002)	(0.026)	(0.006)

Table 4: Parameter estimates

From here on, when we analyze the behavior of the model and discuss the values we will be using these average estimates.

Remember that the experience is updated according to

$$e_i(s) = \lambda \cdot a_i(s-1) \cdot V_i(s-1) + \rho_i \cdot e_i(s-1).$$

which then enters into the probability of initial round cooperation by

$$p_i^{initial}(s) = \frac{1}{1 + \exp\left(-(\alpha + \beta \cdot \Delta^{RD} + e_i(s))\right)}.$$

The estimated $\alpha = -0.313$ means that for $\Delta^{RD} = 0$, about 42.2% of participants would cooperate in the first round of their first supergame. With $\Delta^{RD} = 0.1$, the probability of cooperation in the first supergame increases to 45.4%.

In contrast, $\lambda = 0.196$ implies a strong learning effect. As an example, consider the case where $g = l = 2$ and $\delta = 0.8$, so $\Delta^{RD} = 0$. If the first supergame an individual i plays goes the expected 5 rounds, and both partners cooperate all 5 rounds, i's probability of cooperation $p_i^{initial}(2)$ goes from 42.2% to 66%. An individual j experiencing DC in the first round and DD in the remaining 4 rounds gets payoff of 3, which implies that $p_j^{initial}(2)$ would go down to 28.8%.

To get a sense of the relative importance the model assigns to Δ^{RD} and learning, we compute the Shapley values of these terms in a decomposition of the variance of predicted initial play in the last supergame. As we show in Appendix C, this decomposition suggests that in the last supergame of each session, approximately 90% of the variation between treatments is driven by learning and not the direct influence of the game parameters.

5.5 Understanding the Model

Our simple learning model is able to accurately predict average cooperation and the time-path of cooperation while holding fixed the strategies used in the initial round. The model's assumption that all individuals use the semi-grim strategy implies that higher rates of initial cooperation lead to more cooperation in that supergame, and the reinforcement-learning component of the model implies that this will lead to more cooperation in subsequent supergames.

To better understand the success of our learning model, and why Δ^{RD} is such a strong predictor, we relate the supergame payoffs participants receive with their initial actions. For each session ζ, let $\pi(C)$ be the average supergame payoff received by participants who cooperated in the initial round, and define $\pi(D)$ analogously.

Figure 6 demonstrates the correlation $\pi(C) - \pi(D)$ and Δ^{RD} in the data.

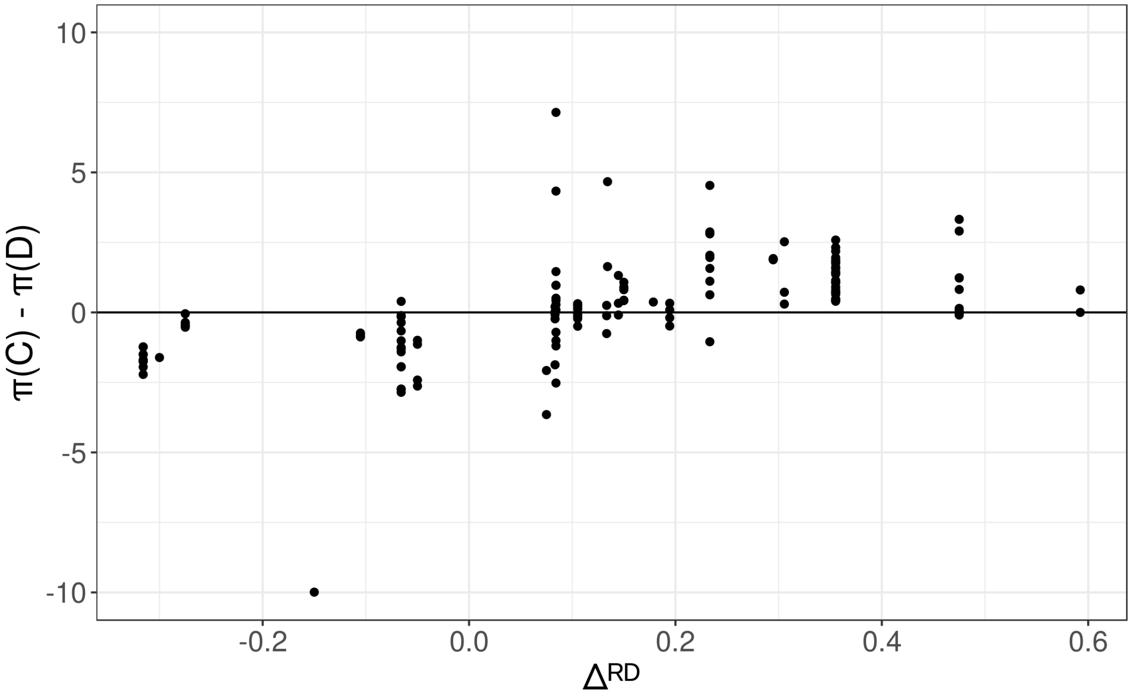

Figure 6: Average empirical difference between total payoff in supergames where the participant cooperated and defected. Each dot corresponds to one experimental session.

For $\Delta^{RD} < 0$, defection is reinforced more strongly than cooperation in all but 1 session. For positive but low values of Δ^{RD}, the difference in reinforcement $\pi(C) - \pi(D)$ is centered around 0, so cooperating and defecting are on average equally reinforced. This helps explain why we see no clear time trends in the sessions where $0 < \Delta^{RD} < 0.15$.

In figure 7 we do the same analysis on simulated data. We simulate 100 participants for each session, playing the same sequence of supergame lengths as in the actual data, and calculate the corresponding value for $\pi(C) - \pi(D)$. The payoff difference has less variation due to the larger number of simulated than actual participants, but it follows the same pattern as in the actual data.

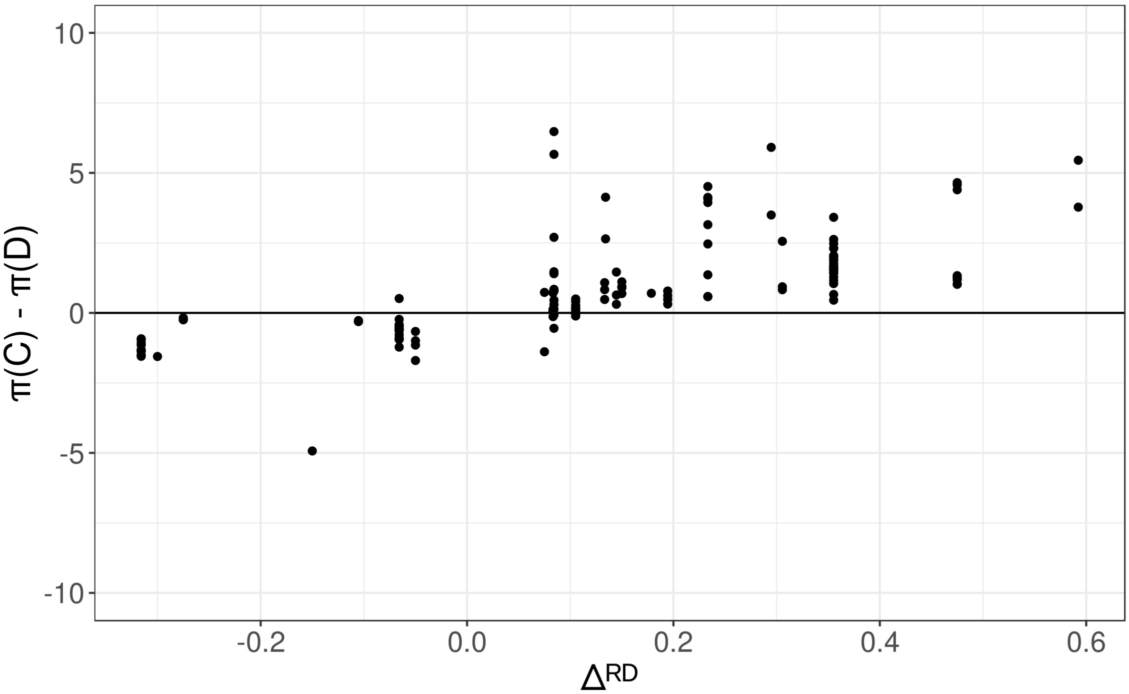

Figure 7: Average simulated difference between total payoff in supergames where the participant cooperated and defected. Each dot corresponds to one simulated session.

Even though $\pi(C) - \pi(D)$ is correlated with Δ^{RD}, we predict average cooperation better with our learning model than by using Δ^{RD} directly. This suggests that the dynamics of our learning model help it capture some additional forces that determine cooperation, such as how many supergames were played and their realized lengths. It is also the case that Δ^{RD} is derived from the assumption of Grim or AllD play without noise, which does not match either the data or our model perfectly.

5.6 Comparison with Alternative Models

We have now seen that our simple learning model can improve out of sample predictions of both average cooperation and the time-path of cooperation. Here we want to answer two questions: Can we improve our model by adding more parameters or by introducing heterogeneity? Would a different learning model perform as well?

In particular, we also want to see if a model with learning to play different pure strategies can at least match our model's performance.

Learning with a recency effect It seems plausible that more recent supergame could have a larger impact on behavior than earlier supergames, so we consider a model with experience weighted by recency. In that model, experience is updated according to

$$e_i(s) = \lambda \cdot a_i(s-1) \cdot V_i(s-1) + \rho \cdot e_i(s-1),$$

where $\rho \in [0, 1]$ discounts previous experiences.

Parametric memory-1 behavior In our main learning model, we assume that all players use the same semi-grim mixed strategy after the initial round in all supergames and all treatments. It is indeed possible both that actual play is not semi-grim, or that memory-1 behavior differs across treatments. In the first extension, which we call *Learning with memory-1* we let all memory-1 behaviors $(\sigma_{CC}, \sigma_{DC}, \sigma_{CD}, \sigma_{DD})$ be free parameters, thus increasing the total number of parameters to 7.

Flexible memory-1 behavior In the next step, we also allow these memory-1 behaviors depend on Δ^{RD}. This model we call *Initial round learning with flexible memory-1.* In this case we have

$$\sigma_h = \frac{1}{1 + \exp(-(\alpha^h + \beta^h \cdot \Delta^{RD}))}$$

In total this model has 11 parameters, but allows for the possibility that people, for example, cooperate more after a DC history if Δ^{RD} is high.

Learning at all memory-1 histories. So far, we have restricted learning to the initial round, and kept behavior at non-initial rounds constant, both across time and treatments. We can extend the learning dynamic we have for the initial round to all memory-1 histories. The "full learning" model tracks experience $e_i(h, t)$ at each memory-1 history h, where t now is a time variable running over all rounds and all supergames. Experience at h is only updated when h occurs, and when it does it depends on the individual's payoff for the rest of the supergame $V_i(t)$ according to

$$e_i(h,t+1) = \begin{cases} \lambda \cdot a_i(t) \cdot V_i(t-1) + \rho e_i(h,t) & \text{if } h(t) = h \\ e_i(h,t) & \text{if } h(t) \neq h \end{cases}$$

where $h(t)$ is the memory-1 history at time t.

Of course, at time t, the individual does not know what $V_i(t)$ will turn out to be. Instead, the probabilities of cooperation are only updated in the beginning of each supergame, and remain constant in its subsequent rounds. So the probability to cooperate at memory-1 history h is given by

$$p_i(h,t) = \begin{cases} \frac{1}{1+\exp(-(\alpha^h+\beta^h\cdot\Delta^{RD}+e_i(h,t))} & \text{if } r(t) = 1 \\ p_i(h,t-1) \quad \text{if } r(t) > 1 \end{cases}$$

where $r(t)$ denotes the round at time t. This learning model increases the number of parameters from 6 to 11. A last variation of this model allows for two different learning rates: Learning in the initial round happens with $\lambda_{initial}$, and learning for the memory-1 histories is reinforced with $\lambda_{memory-1}$.

Table 5 shows a comparison with our main learning model and all the different variations considered in this subsection.

Heterogeneous agents. It is commonly found that there is a lot of heterogeneity in individual behavior. To allow for this, we now consider a finite mixture extension of our learning model. We assume that there are two different types, with different parameters and one variable deciding the share of the two types in the population. In sample, this of course improves predictions a little bit. Out of sample, however, there is no clear improvement.

When we consider the individual one-step ahead predictions, we will see that introducing heterogeneity does slightly improve predictions. One reason that we find so little evidence of type heterogeneity may be that the learning model with a single type has endogenous heterogeneity that can account some of the observed heterogeneity: If an individual by chance defects in the initial round a few periods, they are likely to get a positive payoff in those supergames, thus reinforcing defection.

[13] In contrast, adding a constant share of AllD players, if anything slightly decreased the accuracy of out of sample predictions.[14]

Pure strategy belief learning. The pure-strategy belief learning model in Dal Bó and Fréchette (2011) assumes that all participants follow either TFT or AllD. Each participant has beliefs about how common TFT and AllD are in the population, which they update based (only) on opponents' moves in the initial rounds, and uses them to calculate the expected values from playing TFT or AllD. Given these values, the participant's choice of whether to play TFT or AllD in the following supergame is given by a logistic best reply function. We extend this model to allow for across treatments prediction, increasing the original 6 parameters to 8. A more complete description of the model can be found in the online appendix.

Pure strategy reinforcement learning In the pure strategy reinforcement learning model, we instead consider reinforcement learning over the pure strategies AllD, Grim, and TFT. Each of the pure strategies σ start with an initial attraction $A_k(1)$. At the begging of each supergame s, the individual samples the pure strategy to use according to

$$p_k(s) = \frac{\exp\left(\lambda A_k(s)\right)}{\sum\limits_{l \in \{TFT, AllD, Grim\}} \exp\left(\lambda A_l(s)\right)}$$

where $p_k(s)$ is the probability of using pure strategy k in supergame s, and λ denotes the sensitivity. Let $k(s)$ denote the pure strategy used in supergame s, then after supergame s, the attraction of the pure strategy used is updated according to

$$A_k(s+1) = \begin{cases} A_k(s) + V(s) & \text{if } k(s) = k \\ A_k(s) & \text{otherwise.} \end{cases}$$

To allow for adjustment to Δ^{RD}, the initial attractions are give by linear functions

[13] Our data does not include individual characteristics such as gender, major, or cognitive ability. Proto, Rustichini and Sofianos (2019) find that more intelligent subjects are quicker to adjust their play to feedback.

[14] This may be surprising in light of past findings that this form of heterogeneity is useful in predicting the next action played. See our discussion of this in Section 8.

of Δ^{RD}, i.e.,

$$A_k(1) = \alpha_k + \beta_k \Delta^{RD}.$$

Lastly, we extend the model to allow for trembling hand errors ε when implementing the pure strategy used. In other words, if an individual is following TFT and the previous history is DC, the probability of cooperating is $1 - \varepsilon$, instead of 1, and similarly for other pure strategies and histories. In total this model has 7 parameters without trembling hand errors, and 8 with.

Results In table 5 we see a comparison with the different alternatives considered in this subsection. The results for the time-path prediction problem show a similar relationship between the models, and can be found in appendix xx.

Model	Average Cooperation
Constant	0.0484 (0.0014)
OLS on (δ, g, l)	0.0192 (0.0008)
OLS on Δ^{RD}	0.0183 (0.0006)
GBT	0.0157 (0.0005)
Lasso	0.0154 (0.0006)
Pure strategy belief learning w/o trembles	0.0196 (0.0007)
Pure strategy belief learning w/ trembles	0.0196 (0.0007)
Pure strategy reinforcement learning w/ trembles	0.0180 (0.0007)
Learning with semi-grim and recency	0.160 (0.0006)
Learning 2 types	0.0158 (0.0006)
Learning at all h and two learning rates	0.0148 (0.0005)
Learning at all h	0.0148 (0.0005)
Learning with semi-grim	0.0147 (0.0005)
Learning with flexible memory-1	0.0146 (0.0005)
Learning with memory-1	0.0146 (0.0005)

Table 5: Out of sample prediction loss (MSE) of average cooperation

The best performing learning model based on pure strategies is reinforcement learning and includes trembles. It is however just barely better than a linear function of Δ^{RD}, and according to a pairwise test not significantly so.

The best performing model is the model with learning only in the initial round, and thereafter the same semi-grim behavior in all treatments and all supergames. According to pairwise tests, it is significantly better than all atheoretical prediction algorithms, and most learning models. The fact that it outperforms more complicated learning models, of which it is a special case, can be explained by overfitting. Similarly, with a large enough data set we would have expected the machine learning algorithms to at least match the performance of our learning model.

Taken together this suggests that our suggested learning model captures most of the predictable regularity in cooperation rates. The relevant learning and direct adjustment to Δ^{RD} happens only with respect to the initial round of each supergame. Introducing heterogeneity does not improve predictions, and neither does learning or flexibility at non-initial rounds. If fact, extending our model often seems to lead to slightly worse out of sample performance, most likely due to overfitting. We also see that assuming some kind of learning over pure strategies does not lead to good predictions.

6 Extrapolating to Longer Experiments

Due to practical constraints, experiments on the PD are of limited duration, but as researchers we are also interested in what would happen over a longer run. Our learning model lets us make predictions of what would happen in experiments with a a longer time horizon than those in our data set.

6.1 Extrapolating within observed sessions

Before we turn to the implications of the learning model for long run play, we want to test how well it can extrapolate to longer sessions than it is trained on. To do this, we use the same cross-validation folds as earlier, so that data from a given session is either in a training fold or a test fold but not both. We then use the first halves of the training sessions to estimate the parameters, and use the estimated model to predict the second half of the sessions in each test set. This a way of approximating how accurate our predictions would be for experiments that are twice as long as the ones in the sample.

Since the parameters estimated on the time path performed better in predicting overall average cooperation, we estimate the parameters of the different models on the time paths in the first half of the session and use them to predict the average cooperation in the second half.

In table 6, we see the cross-validated MSE where we predict the later half of experimental sessions with parameters estimated on different models.

Model	Second half Session avg	Second half time path
Constant Prediction	0.066 (0.002)	0.081 (0.002)
Lasso	0.028 (0.001)	0.044 (0.001)
GBT:time-path	0.028 (0.001)	0.040 (0.001)
Learning with semi-grim	0.024 (0.001)	0.037 (0.001)

Table 6: Prediction loss (MSE) from estimating on first half and evaluating on second half of the experimental sessions.

The table shows that the learning model is better at extrapolating to longer supergames than our atheoretical black-box algorithms. In principle this might be due to our particular ML implementations, but it is also true that atheoretical prediction algorithms can have trouble extrapolating to a slightly different settings. A more structured model that encodes some intuition or knowledge about the problem domain can sometimes better extrapolate to related prediction problems, and we suspect that this is the case here/

6.2 Extrapolating to hypothetical session lengths

We generate predictions for the treatments in Dal Bó and Fréchette (2011), since these capture a nice range of behavior. For each of the treatments, 1000 populations with 14 participants were simulated for 10 000 supergames, with randomly drawn supergame lengths. We then simulated the learning model with the average (across folds) parameters estimated on the time path in table 4 Using these simulations we can compute the median level of average cooperation, and its 90% confidence interval.

Δ^{RD}	δ	Q05	Mean	Q95
-0.32	0.50	0.00	0.01	0.05
-0.11	0.50	0.00	0.02	0.07
0.11	0.50	0.00	0.37	0.76
-0.07	0.75	0.00	0.07	0.42
0.14	0.75	0.18	0.49	0.79
0.36	0.75	0.50	0.76	0.99

Table 7: Simulated cooperation after 10,000 supergames, 14 participants per session.

Δ^{RD}	δ	Q05	Mean	Q95
-0.32	0.50	0.00	0.01	0.03
-0.11	0.50	0.00	0.01	0.03
0.11	0.50	0.01	0.39	0.60
-0.07	0.75	0.00	0.06	0.29
0.14	0.75	0.33	0.50	0.65
0.36	0.75	0.65	0.76	0.85

Table 8: Simulated cooperation after 10 000 supergames, 100 participants per session.

We see quite wide 90% intervals for intermediate values of Δ^{RD} in figure 8 due to the randomness of behavior and small population size. In the treatment $\Delta^{RD} = 0.11$, even after 10 000 supergames the 90% interval goes from 0% to 75%, and the average is just 37%. (With populations of 100 participants, the 90% interval is smaller but still substantial; it goes from 0.01% to 60%.). This randomness comes in part from random initial play in a finite population, and also from the randomness in the realized supergame lengths. Even if we increase the population size to 1000, the 90% interval is still from 24% to 56%. However, if we also let all the simulated supergames have the expected number of rounds, the 90% interval is only 41% to 48%.

The intervals are smaller in treatments where Δ^{RD} is farther away from .11 in either direction. For $\Delta^{RD} < 0$, we predict less than 50 % cooperation, and for $\Delta^{RD} = -0.32$ cooperation is almost certain to decrease. For $\Delta^{RD} = 0.14$ we see a slow increase in initial round cooperation to .49, and for $\Delta^{RD} = 0.36$ we predict relatively fast and certain convergence to a high cooperation rate.

Dal Bó and Fréchette (2011) estimate the learning model in subsection 5.6 on the individual level, and use those individual estimates to simulate behavior. They produce plots similar to the one below, but restricted to the initial round. Visual inspection suggests that our single model fits the data about as well, even though it is simpler and is designed to do prediction rather than in-sample replication.

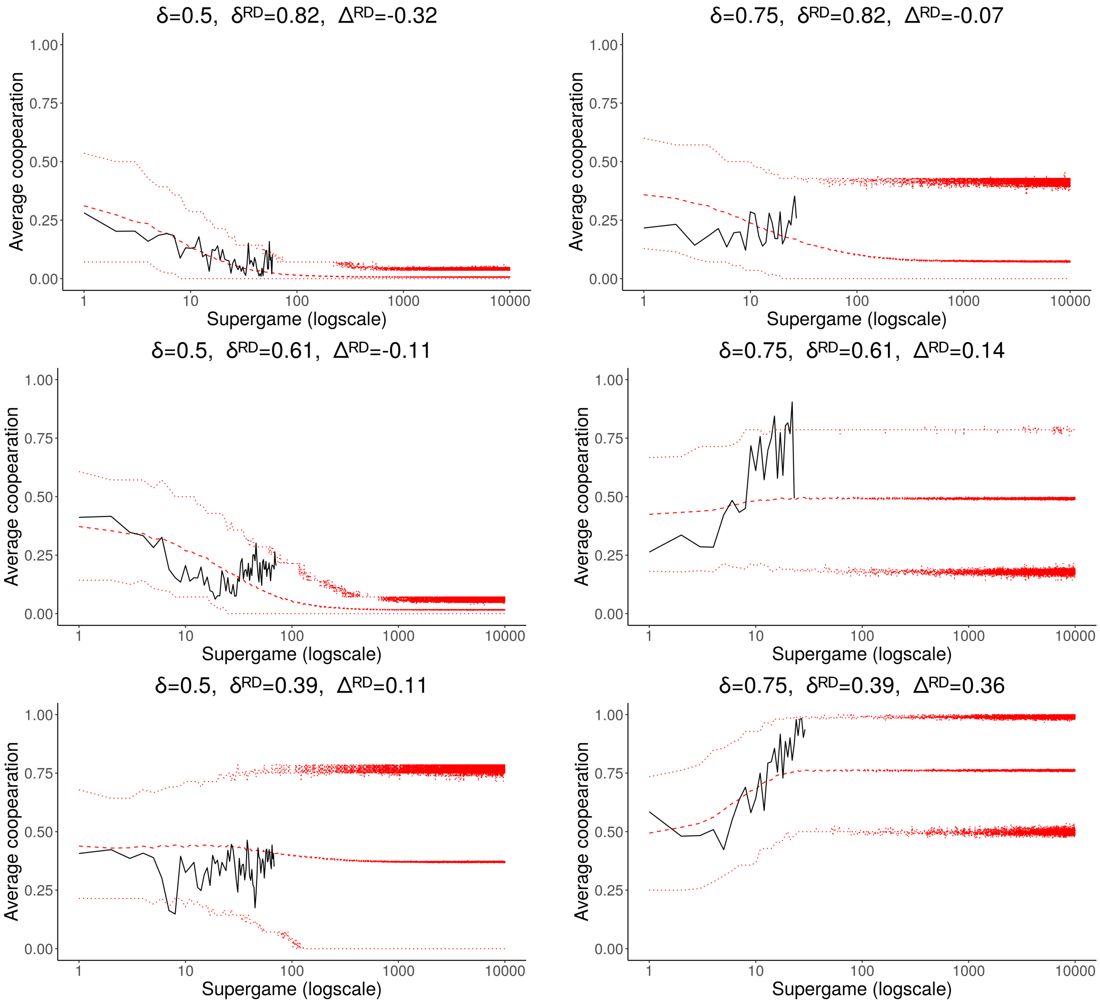

Figure 8: Predictions and actual behavior for six different treatments. The solid black line corresponds to the data, the red lines depict average cooperation and the middle 90% interval in 1000 simulated populations.

We get a broader picture of the long run predictions by replicating this exercise for all the 28 treatments in the data. In figure 9 we see the average cooperation after 10 000 supergames, predicted by simulating 1000 populations of size 16 for each treatment. We see that for Δ^{RD} between 0 and 0.3, even after 10 000 supergames, the learning model does not predict either very high or very low rates of cooperation.

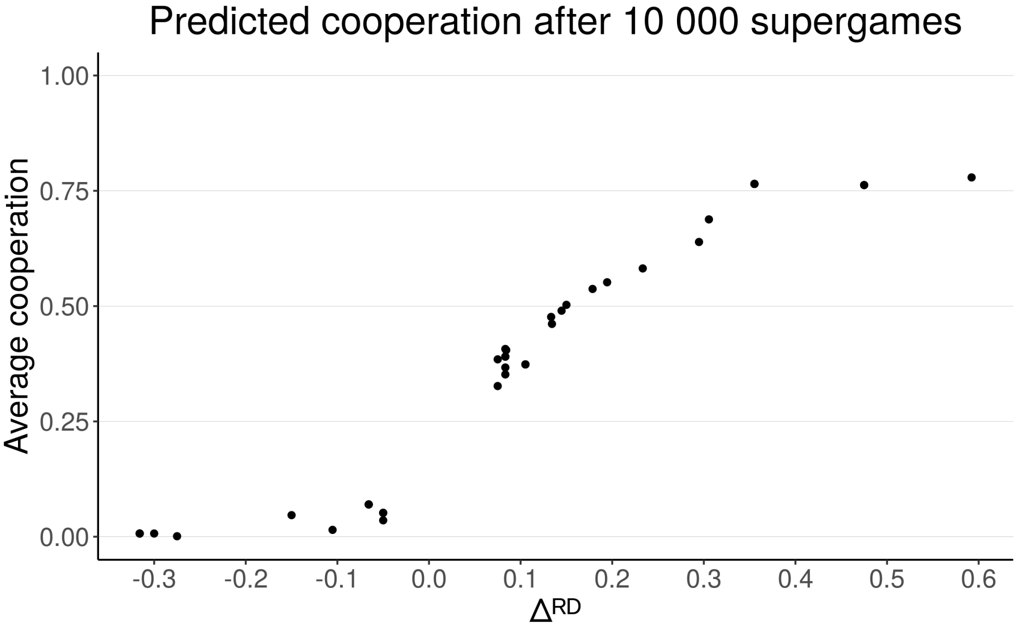

Figure 9: Predicted average cooperation after 10 000 supergames

7 Predicting the Next Action Played

Much of the existing literature has focused on determining the strategies used by the participants. Typically this is done with a finite mixture model, as in the SFEM of Dal Bó and Fréchette (2011), which assume that a finite number of strategies are used in the population. These models are often estimated using maximum likelihood, and their in-sample performance compared with information criteria. But because we want to compare the out-of-sample performance of our models with that of machine learning algorithms, we instead consider how well the models predict the next action taken by a participant.

We consider different numbers of types in the finite mixtures, and we also considered variants with a share of agents who always defect.[15] We compare the models to a naive benchmark that always predicts that $a_i(t+1) = a_i(t)$ and to a GBT.[16] As we show in Appendix F the simple six-parameter learning model predicts behavior about as well as a pure strategy model with 11 different pure strategies estimated separately for each treatment.

The naive benchmark does very well: In 84.3% percent of the cases, the participants simply repeat the action they took in the previous round. If we assume the GBT captures most of the predictable regularity in the data, there is roughly an additional 6.5% of the observations that we can hope to capture with a better model. In contrast, the pure strategy model performs poorly, both in terms of accuracy and prediction loss.

The best-performing model has three types, with learning in the initial round. and flexible memory-1 behavior. However, learning on its own captures most of the heterogeneity in behavior. Some of the heterogeneity not captured by a single learning type can be captured by introducing a share of AllD players, especially in terms of prediction loss. The 8 parameter model, learning with semi-grim and AllD, has an accuracy of 87.4%, which is similar to that of the 32-parameter three type flexible mixed strategy model, and of the pure strategy model that is estimated separately on each treatment. It is only 0.7% than the best performing learning model. Neglecting the influence of learning might lead researchers overemphasize the heterogeneity in "types" when the diversity of play is largely driven by differences in experience.

8 Conclusion

This paper studies how to predict cooperation rates in the experimental play of the prisoner's dilemma as a function of the game parameters and the number of

[15] Previous studies have consistently found that a substantial share of participants behaves roughly this way, perhaps because they fail to understand the existence of repetitive equilibria.

[16] We remove the first round of the first supergame when calculating the predictive performance of the naive predictions.

supergames played. We found that the key to predicting cooperation in a given match is the prediction of play in the initial round, and that this depends both on the game parameters and on the individual's experience in previous matches.

Our preferred learning model is very simple, as it holds play fixed except in the initial round of each supergame. This model only has 6 parameters, and one type of agent.

While the 6-parameter model is too stark to model the richness of actual behavior, it predicts average cooperation at least as well as the more complex machine learning algorithms or more complicated learning models. This may be due in part be due to the lack of enough data, but also comes from the way our learning model uses data on the realized lengths of the supergames.

Our results lead to a clearer understanding of how and why the composite parameter Δ^{RD} influences cooperation rates: The parameter's main effect is on the probability of cooperation in the initial round of a match. Initial cooperation is positively reinforced when $\Delta^{RD} > .15$, so in these games the probability of cooperating in the initial round increases over the course of a session. Initial cooperation is negatively reinforced when $\Delta^{RD} < 0$, so here initial cooperation rates drift down. For intermediate values of Δ^{RD}, a participant's overall payoff is about the same regardless of how they play in the initial round, which is why in these games initial cooperation rates stay roughly constant throughout a session.

Our model lets us capture the effect of playing more supergames on average cooperation. One advantage of this is that we can predict what average cooperation rates would be with longer lab sessions (assuming the participants did not loose focus on the task).

In this paper we only consider the prisoner's dilemma with perfect monitoring. Many real-world settings have implementation errors or imperfect monitoring, and as shown by Fudenberg, Rand and Dreber (2012) in such cases people seem to use more complex strategies with longer memory. There are not yet enough experimental studies of these games to support the sort of analysis we do here, but once there are it would be useful to extend our analysis of average cooperation rates to this case.

We close with a novel comparative statics prediction inspired by our learning model. In the lab, there is typically a tradeoff between specifying high discount factors and having participants play many supergames. So consider varying δ and g holding Δ^{RD} fixed, and suppose that the number of supergames played is inversely proportional to their expected length, which $1/(1-\delta)$. Our model predicts that with more supergames and lower δ, there will be higher average cooperation if $\Delta^{RD} > .15$, and lower average cooperation if $\Delta^{RD} < 0$.

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: 1–43.

Athey, Susan, and Kyle Bagwell. 2001. "Optimal Collusion with Private Information." *The RAND Journal of Economics*, 32: 428–465.

Backhaus, T., and Y. Breitmoser. 2018. "God does not play dice, but do we? On the determinism of choice in long-run interactions."

Blonski, M., and G. Spagnolo. 2015. "Prisoners other Dilemma." *International Journal of Game Theory*, 44: 61–81.

Blonski, M., P. Ockenfels, and G. Spagnolo. 2011. "Equilibrium selection in the repeated Prisoner's Dilemma: Axiomatic approach and experimental evidence." *American Economic Journal: Microeconomics*, 3: 164–192.

Breitmoser, Y. 2015. "Cooperation, but No Reciprocity: Individual Strategies in the Repeated Prisoner's Dilemma." *American Economic Review*, 105: 2882–2910.

Camerer, C., and T. H. Ho. 1999. "Experience-weighted attraction learning in normal form games."

Cheung, Y., and D. Friedman. 1997. "Individual Learning in Normal Form Games :." *Games and Economic Behavior*, 19: 46–76.

Dal Bó, P. 2005. "Cooperation under the shadow of the future: Experimental evidence from infinitely repeated games." *American Economic Review*, 95: 1591–1604.

Dal Bó, P., and G. R. Fréchette. 2011. "The Evolution of Cooperation in Infinitely Repeated Games: Experimental Evidence." *American Economic Review*, 101: 411–429.

Dal Bó, P., and G. R. Fréchette. 2018. "On the Determinants of Cooperation in Infinitely Repeated Games: A Survey." *Journal of Economic Literature*, 56: 60–114.

Dal Bó, P., and G. R. Fréchette. 2019. "Strategy Choice in the Infinitely Repeated Prisoner's Dilemma." *American Economic Review*, 109: 3929–3952.

Engle-Warnick, J., and R. L. Slonim. 2006. "Learning to trust in indefinitely repeated games." *Games and Economic Behavior*, 54: 95–114.

Erev, I., and A. E. Roth. 1998. "Predicting how People Play Games."

Erev, I., and A. E. Roth. 2001. "Simple Reinforcement Learning Models and Reciprocation in the Prisoner's Dilemma Game." In *Bounded rationality: The adaptive toolbox.* , ed. Gerd Gigerenzer et al., Chapter 12. The MIT Press.

Fudenberg, D., and A. Liang. 2019. "Predicting and Understanding Initial Play." *American Economic Review*, 109: 4112–4141.

Fudenberg, D., D. G. Rand, and A. Dreber. 2012. "Slow to Anger and Fast to Forgive: Cooperation in an Uncertain World." *American Economic Review*, 102: 720–749.

Fudenberg, D., J. Kleinberg, A. Liang, and S. Mullainathan. 2020. "Measuring the Completeness of Theories."

Hanaki, N., R. Sethi, I. Erev, and A. Peterhansl. 2005. "Learning strategies." *Journal of Economic Behavior and Organization*, 56: 523–542.

Harrington, Joseph E. 2017. *The Theory of Collusion and Competition Policy.* The MIT Press.

Hastie, T., R. Tibshirani, and J. Friedman. 2009. *The Elements of Statistical Learning. Springer Series in Statistics*, New York, NY:Springer New York.

Honhon, Dorothée, and Kyle Hyndman. 2020. "Flexibility and reputation in repeated Prisoner's dilemma games." *Management Science*, 66: 4998–5014.

Ioannou, C. A., and J. Romero. 2014. "A generalized approach to belief learning in repeated games." *Games and Economic Behavior*, 87: 178–203.

Kruskal, W. 1987. "Relative Importance by Averaging Over Orderings." *The American Statistician*, 41: 6–10.

Lipovetsky, S. 2006. "Entropy criterion in logistic regression and Shapley value of predictors." *Journal of Modern Applied Statistical Methods*, 5: 94–105.

Lundberg, S. M., and S. Lee. 2017. "A Unified Approach to Interpreting Model Predictions." 4765—-4774.

Mishra, S. K. 2016. "Journal of Economics." *Journal of Economics Bibliography*, 3: 498–515.

Proto, E., A. Rustichini, and A. Sofianos. 2019. "Intelligence, personality, and gains from cooperation in repeated interactions." *Journal of Political Economy*, 127: 1351–1390.

Rand, D. G., and M. A. Nowak. 2013. "Human cooperation." *Trends in Cognitive Sciences*, 17: 413–425.

Romero, J., and Y. Rosokha. 2018*a*. "Constructing strategies in the indefinitely repeated prisoner's dilemma game." *European Economic Review*, 104: 185–219.

Romero, J., and Y. Rosokha. 2018*b*. "Mixed Strategies in the Indefinitely Repeated Prisoner's Dilemma." *SSRN Electronic Journal*.

Rotemberg, Julio, and Garth Saloner. 1986. "A Supergame-Theoretic Model of Price Wars during Booms." *American Economic Review*, 76: 390–407.

Wright, J. R., and K. Leyton-Brown. 2017. "Predicting human behavior in unrepeated, simultaneous-move games." *Games and Economic Behavior*, 106: 16–37.

A Black box prediction results

A.1 Average cooperation

Table 9: Out of sample prediction loss (MSE) for per-session average cooperation

Model	Loss	SE	Relative improvment
Constant prediction	0.0484	(0.0014)	0.0%
OLS on (g, l, δ)	0.0192	(0.0008)	60.33%
OLS on Δ^{RD}	0.0183	(0.0006)	62.19%
GBT:average cooperation	0.0157	(0.0005)	67.56%
OLS Full	0.0154	(0.0006)	68.18%
Lasso	0.0154	(0.0006)	68.18%

A.2 Time path

Table 10: Out of sample prediction loss (MSE) for the Time Path of Cooperation

Model	Loss	SE	Relative improvement
Constant	0.0705	(0.0015)	0.0%
OLS on (g, l, δ)	0.0390	(0.0007)	44.68%
OLS on Δ^{RD}	0.0380	(0.0008)	46.1%
OLS Full	0.0335	(0.0007)	52.48%
Lasso	0.0334	(0.0007)	52.62%
GBT:time-path	0.0333	(0.0007)	52.77%

B Numerical Estimation of Learning Models

To simulate a decision, a number $r \sim Uniform(0, 1)$ is drawn, and if that number is lower than the probability of cooperation for the simulated individual, she cooperates, otherwise defects. Similarly, the type of each individual is decided by a random draw. By fixing the draws of these values r, we get a deterministic function.

The resulting function is however locally flat, which means that finding an optimum is difficult. To address this problem we first generate 30 candidate points using the following global differential evolution[17] optimization in parallel, using a 100 individuals with one common set of constant random numbers.

1. First a population is initialized: For each agent x, we pick 3 new agents a, b, c from the population of candidates and generate a new candidate x'. Each parameter x_i of x is updated with some probability CR (the cross-over probability), and if it is updated the new value is given by

   $$x'_i = a_i + F * (b_i - c_i).$$

   Once this is done, we compare the new value $f(x')$ with the old $f(x)$. If the this results in a lower loss, the new candidate replaces the old in the population, and otherwise it is thrown away.

2. After a fixed amount of time, the best candidate from this algorithm is used as a starting point for a Nelder-Mead algorithm that performs a local, gradient-free, optimization. This time using a different fixed realization of the random variables. The output of this local optimization is then returned as one candidate solution.

Once these 30 candidate points are found, they are each evaluated using a population size of 3,000, with a new fixed realization of the random variables for all 30 candidates. The best of these parameters are then returned as the solution.

[17] From the package BlackBoxOptim.jl

C The relative influence of game parameters and learning

The learning model assumes that both game parameters and experience influence initial round cooperation

$$p_i^{initial}(s) = \frac{1}{1 + \exp(-(\alpha + \beta \cdot \Delta^{RD} + e_i(s)))}.$$

We can thus interpret $\alpha + \beta \cdot \Delta^{RD}$ as the direct effect of the game parameters and $e_i(s)$ as the direct effect of learning. We here try to answer how much of the behavior is directly driven by learning and how much is driven by the game parameters, according to our learning model. Since these two values enter the expression in the same way, they are directly comparable.

We consider the last supergame of each experimental session. We consider the actual data and a simulated data set with 16 participants in each session. When we consider the actual data for an individual, we look at the initial round actions they took and their observed realized payoffs. From these values, we calculate the corresponding value for $e_i(s)$ in the last supergame. For the simulated data, we instead simulate the whole sequence of play, and use the simulated values to calculate $e_i(s)$.

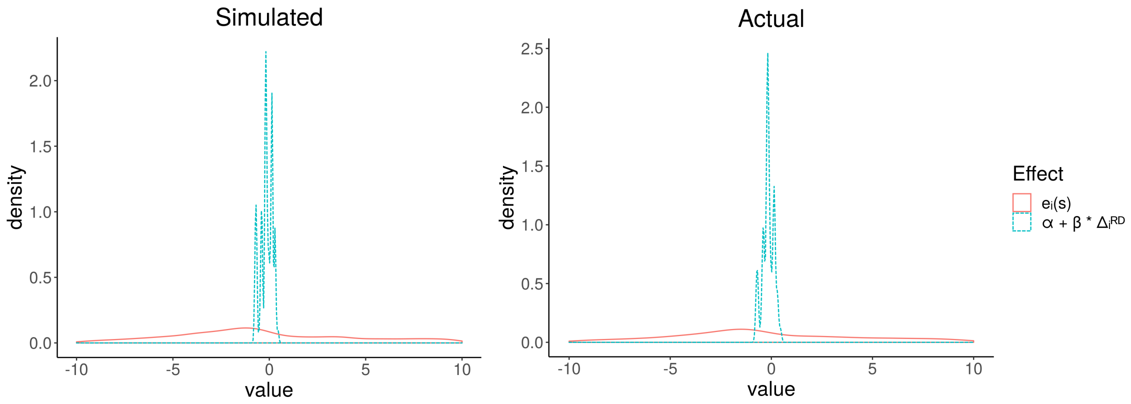

Figure 10: Across individuals variation in experience $e_i(s)$ and direct effect of the learning parameters $\alpha + \beta \cdot \Delta^{RD}$. Evaluated on the last supergame for each session. Simulated data to the left and actual data to the right.

The variation in the learning effect is much larger than the game parameter effect, and thus has a greater influence on the variation in behavior.

We can also consider the across treatment variation, averaging the $e_i(s)$ over the individuals in each treatment.

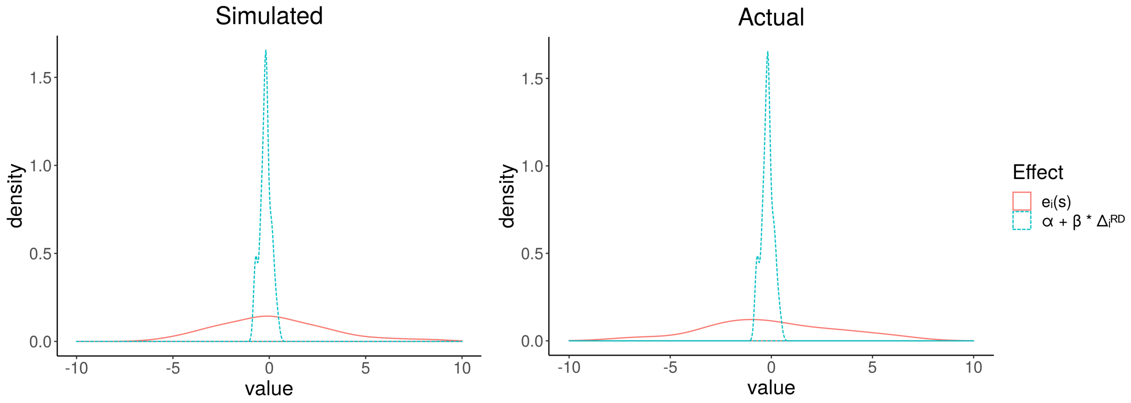

Figure 11: Across treatments variation in experience $e_i(s)$ and direct effect of the learning parameters $\alpha + \beta \cdot \Delta^{RD}$. Evaluated on the last supergame for each session. Simulated data to the left and actual data to the right.

Since there is considerable variation in $e_i(s)$ between the individuals in a treatment, but Δ_i^{RD} is the same for everyone in a treatment, the differences are smaller here than for the individual data. But we still see that the variation in the direct learning effect is larger than the variation in the direct game parameter effect.

To get a numerical estimate of the relative importance we can look at how much of the variation in predicted initial round cooperation is driven by the two effects. The total average variance in initial round cooperation is given by

$$Var(p|e,\Delta^{RD}) = \sum_{i\in I}\left(\frac{1}{1-\exp\left(-(\alpha+\beta\Delta^{RD}+e_i(s)\right)} - \overline{p}\right)^2 /|I|$$

where I is the set of all individuals, and $\overline{p}$ is the average predicted initial round cooperation. We can compare this to the variation in predicted cooperation from the direct learning effect and the direct game parameter effect respectively.

$$\begin{aligned} Var(p|\Delta^{RD}) &= \sum_{i\in I}\left(\frac{1}{1-\exp\left(-(\alpha+\beta\Delta^{RD})\right)} - \overline{p}(\Delta^{RD})\right)^2 /|I| \\ Var(p|e) &= \sum_{i\in I}\left(\frac{1}{1-\exp\left(-e_i(s)\right)} - \overline{p}(e)\right)^2 /|I|. \end{aligned}$$

To calculate the relative importance of Δ^{RD} we take the average of the variation introduced by Δ^{RD} alone, and the additional variation when it is added to the direct effect of $e_i(s)$ divided by the total variation.

$$\begin{aligned} \text{Relative Importance}(\Delta^{RD}) &= \frac{Var(p|\Delta^{RD}) + \Big(Var(p|e,\Delta^{RD}) - Var(p|e)\Big)}{2} / Var(p|e,\Delta^{RD}) \\ \text{Relative Importance}(e) &= \frac{Var(p|e) + \Big(Var(p|e,\Delta^{RD}) - Var(p|\Delta^{RD})\Big)}{2} / Var(p|e,\Delta^{RD}). \end{aligned}$$

This is the Shapley value of the two effects[18] This can be be done on either the individual level or the treatment level, where the probabilities $p_i(s)$ are first averaged for each session. In the table below we see the results.

[18] (Kruskal, 1987) and Mishra (2016) use the Shapley value to analyze regressions, (Lipovetsky, 2006) uses them for logistic regressions, and (Lundberg and Lee, 2017) for general machine learning algorithms.

Data	$Var(p\|e, \Delta^{RD})$	$Var(p\|e)$	$Var(p\|\Delta^{RD})$	Rel Imp $e_i(s)$	Rel Imp Δ^{RD}
Simulated individual	0.195	0.188	0.006	96.7%	3.3%
Actual individual	0.188	0.183	0.005	97.3%	2.7%
Simulated treatment	0.059	0.052	0.005	89.4%	10.6%
Actual treatment	0.072	0.064	0.005	90.9%	9.1%

Table 11: Relative importance measures.

We see that in both the simulated and actual data, $e_i(s)$ is responsible for roughly 96% of the variation in predicted individual behavior, and roughly 90% of the variation in predicted initial round cooperation between treatments, so in our model experience drives most of the variation initial round cooperation.

D Pairwise tests

Here we consider paired t-tests and paired signed Wilcox tests of whether the out of sample predictive MSE of the various models are significantly different. With the 10 different 10-fold cross-validation splits, we have 100 different test sets. Since we use the same splits for all predictive models, we can do paired tests. In the tables below paired tests between the initial round learning model and alternatives are shown for the average cooperation prediction task and for the time-path prediction task.

Model	Difference	T-test p-value	Sign test p-value
Pure strategy belief learning with trembles	-0.0049	p<0.001	p<0.001
OLS on Δ^{RD}	-0.0036	p<0.001	p<0.001
Pure strategy reinf. learning with trembles	-0.0033	p<0.001	p<0.001
Learning at all h	-0.0019	p<0.001	p<0.001
GBT	-0.001	p=0.006	p=0.015
Full OLS	-0.0007	p=0.036	p=0.041
Lasso	-0.0006	p=0.047	p=0.072
Learning at all h two rates	-0.0002	p=0.370	p=0.379
Learning with flexible memory-1	0.0001	p=0.698	p=0.648
Learning with flexible memory-1 and AllD	0.0001	p=0.856	p=0.573

Table 12: Differences and paired significance test with the main learning model (learning with semi-grim) for the average cooperation prediction task.

In the pairwise tests in table 12 we see that our main learning model, learning with semi-grim, is significantly better than almost all alternatives we consider with both tests at the 5% level, including most generalizations of the model. The differences with the generalizations of this model are small.

Model	Difference	T-test p-value	Sign test p-value
Pure strategy belief learning without trembles	-0.0103	p<0.001	p<0.001
Pure strategy reinf. learning with trembles	-0.0072	p<0.001	p<0.001
Pure strategy belief learning with trembles	-0.0062	p<0.001	p<0.001
OLS on Δ^{RD}	-0.006	p<0.001	p<0.001
Full OLS	-0.0016	p=0.001	p<0.001
Lasso	-0.0015	p=0.002	p=0.001
GBT	-0.0013	p=0.005	p=0.006
Learning with semi-grim and recency	-0.0012	p<0.001	p<0.001
Learning at all h	-0.0005	p=0.100	p=0.176
Learning at all h, two rates	-0.0005	p=0.055	p=0.139
Learning with memory-1	-0.0	p=0.991	p=0.728
Learning with flexible memory-1 and AllD	0.0004	p=0.267	p=0.128

Table 13: Differences and paired significance test with the main learning model (learning with semi-grim) for the time-path prediction task.

For predicting time paths, the picture is similar. Some of the generalizations appear slightly better, but not significantly so.

E Evaluation of the procedure on simulated data

To test our estimation approach, we simulate the data using three different models: learning with semi-grim, learning with semi-grim with individual parameters drawn from a normal distribution, and the pure strategy reinforcement learning model.

From each of these models we generate a simulated data set that mimics the data we have. Each session is simulated with an actual sequence of supergame lengths, and with 16 participants in each session. On each of these different data sets we perform the estimations from the main text, and report averages and standard deviations across the 10 different folds. The parameters for each model are taken as the average parameter estimates we got from on the actual data.

Model	Learning with SG	+ Noise	Pure strategy reinf learning
Learning with semi-grim	0.0080	0.0099	0.0081
Standard Deviation	0.0023	0.0054	0.0036
Pure strategy reinf. learning	0.0134	0.0136	0.0031
Standard Deviation	0.0023	0.0054	0.0036

Table 14: Averages and standard deviations of a 10-fold cross validation on different simulated data sets.

When we add noise to the learning model, we draw each individual's parameters from a normal distribution where $\alpha \sim N(-0.313, 0.5)$, $\beta \sim N(1.298, 1)$, $\lambda \sim N(0.196, 0.1)$, $p_{CC} \sim N(0.996, 0.1)$, $p_{CD/DC} \sim N(0.373, 0.1)$ and $p_{CC} \sim N(0.016, 0.1)$. The standard deviations were set ad-hoc to what we thought were reasonable sizes. The means are the estimated parameters from the main analysis. The sampled probabilities are then cut-off to be in the interval $(0, 1)$.

In table 14, we see that our estimation strategy can indeed distinguish between these different learning models.

F One Step Ahead Prediction and Maximum Likelihood Estimation

We here consider the question of how well we can predict the next action taken by a participant given their actions so far. If each participant uses a fixed strategy or learning rule, and the relative shares in the population are known, it should be possible to accurately predict the next action a given individual will take at a given history. Thus, following the literature, we assume that there are a finite number of different "strategic types" (i.e. strategies or learning rules) used in the population, and estimate the parameters and the shares of these strategies by maximum likelihood. We then see how well the different types match the individuals behavior up to that point, and then make the corresponding prediction.

We consider different possible models of the types, and compare those both to a naive benchmark that predicts the previous action taken by the individual, and to the predictions made by a gradient boosting tree. The different specifications we consider are pure strategies, learning with semi-grim, learning with memory-1, and learning at all h.

We focus on out of sample predictions. This allows us to compare models of different complexities, because overly complex models may be penalized by cross-validation.

Several interesting conclusions arise from this exercise. First, in contrast to the problem of predicting the populations behavior, explicitly modeling heterogeneity does improve predictions here. Moreover, as above, learning allows us to make better out of sample predictions. Finally, as in past work we see no evidence of the participants using strategies of memory greater than 1.

F.1 The General Prediction Problem

Consider the complete data set of observations

$$D = \{(h_i(t), a_i(t)) | i \in I, t \in T(i)\},$$

each pair consisting of the history and the action taken for individual $i \in I$, in time period t, where $T(i)$ denotes all the rounds played by individual i, and we track the game parameters Γ_i as part of the history. The action taken $a_i(t)$ is 1 for cooperation and -1 for defection.

A predictive model is a function $m : \mathcal{H} \to [0,1]$, where $\mathcal{H}$ is the space of all individual histories in an experimental session. This function predicts the probability that an individual with a given history cooperates. A model comes with a set of parameters θ and we write

$$m(h_i(t)|\theta) = \widehat{a}_i(t)$$

to the denote model m's predicted probability of cooperation given history $h_i(t)$.

Two different measures of predictive performance are used, prediction loss and accuracy. The prediction loss is based on the cross-entropy of the predicted probability of the taken action. For a data set $D' \subset D$, the average prediction loss is given by

$$\mathcal{L}(m|D',\theta) = \frac{-1}{|D'|} \sum_{(h_i(t),a_i(t))\in D'} \log(m(h_i(t)|\theta))\cdot\mathbb{1}\{a_i(t)=1\} + \log(1-m(h_i(t)|\theta))\cdot\mathbb{1}\{a_i(t)=-1\}.$$

or if we simplify the notation, by letting m and θ be implicit, with

$$\mathcal{L}(D') = \frac{-1}{|D'|} \sum_{(h_{isr},y_{isr})\in D'} \log(\widehat{y}_{isr})\cdot y_{isr} + \log(1-\widehat{y}_{isr})\cdot(1-y_{isr}).$$

The models are always optimized with respect to the prediction loss, however, it is also interesting to look at the accuracy of the predictions. The accuracy is the share of observations where the taken action was predicted to be the most likely, i.e.

$$\begin{aligned} Acc(m|D',\theta) = \frac{1}{|D'|} \sum_{(h_i(t),a_i(t))\in D'} \Big(& \mathbb{1}\{a_i(t)=1\}\cdot\mathbb{1}\{m(h_i(t)|\theta)\geq 0.5\} \\ & + \mathbb{1}\{a_i(t)=-1\}\cdot\mathbb{1}\{m(h_i(t)|\theta)<0.5\}\Big). \end{aligned}$$

F.2 Finite Mixture models

When estimating the models, we assume that the population can be divided into different types, where individuals of the same type behave in the same way. Depending on the model, these types are parameterized in different ways. The learning models presented are the same ones used in the main text, with the difference that experience at a given supergame is calculated using the actual observed data upto that supergame, and not from simulation.

Pure Strategy Model In this model we assume that each type σ^j follows a pure strategy with a fixed mistake probability ε_j. If we let $\omega^j : \mathcal{H} \to \{0,1\}$ denote a pure strategy, e.g., Tit for Tat or Grim, a type can be described by a tuple $(\omega^j, \varepsilon_j)$. We start with an exogenous list of 11 different pure strategies, taken from the pure strategies estimated to have positive share in Fudenberg, Rand and Dreber (2012)[19], and estimate the share ϕ_j and mistake probability ε_j and for each such pure strategy. [20] The mistake probabilities ε_j and the shares ϕ_j are explicitly estimated, while the 11 available pure strategies remain fixed. In the standard SFEM approach it is commonly assumed that a common error rate ε is used, we relax this assumption in order to give the pure strategy model a better chance of performing well.

Estimating any finite mixture model gives us a set of types and their relative shares. To make a prediction of $a_i(t)$ based on $h_i(t)$, we first calculate the probability of $h_i(t)$ under the different types. For simplicity, we represent the different types with σ^j for each type j.

$$\Pr(h_i(t)|\sigma^j) = \prod_{\tau<t} \sigma^j(h_i(t))^{\mathbb{1}\{a_i(t)=1\}} \cdot \Big(1 - \sigma^j(h_i(t))\Big)^{\mathbb{1}\{a_i(t)=-1\}}.$$

Given the estimated shares ϕ the conditional probability of individual i being of type j at time t is given by

[19] While Fudenberg, Rand and Dreber (2012) studies interactions with exogenous noise, these 11 strategies contain those strategies often found to be used in games without noise e.g. Dal Bó and Fréchette (2018)

[20] The online appendix provides a list of these strategies.

$$\Pr(\sigma^j | h_i(t)) = \frac{\phi^j \Pr(h_i(t)|\sigma^j)}{\sum_l \phi^l \Pr(h_i(t)|\sigma^l)}.$$

Given these estimated probabilities, the prediction of model m is given by

$$m(h_i(t)) = \sum_j \sigma^j(h_i(t)) \Pr(\sigma^j | h_i(t)).$$

F.3 Evaluating the Models

To evaluate out of sample performance we again use 10-fold cross-validation. Because we are now predicting individual and not aggregate play, here the partitions are at the level of individuals, so that each individual is in exactly one test set.

Furthermore, the splits are balanced over the treatments so that roughly 10% of the participants from each treatment are in each fold.

For each such partition k, we find the parameters θ_k^{train} with the smallest prediction loss on the training set,

$$\theta_k^{train} = \arg\min_{\theta \in \Theta} \mathcal{L}(m | D_k^{train}, \theta)$$

and calculate the prediction loss on the test set, $\mathcal{L}(m|D_k^{test}, \theta_k^{train})$. The prediction loss from the 10-fold cross-validation that will be reported, and used to compare the models, is given by averaging over all such splits,

$$\text{PredictionLoss}(m|D, K) = \frac{1}{K} \sum_{k=1}^{K} \mathcal{L}(m|D_k^{test}, \theta_k^{train})$$

and the accuracy is similarly given by

$$\text{Accuracy}(m|D, K) = \frac{1}{K} \sum_{k=1}^{K} Acc(m|D_k^{test}, \theta_k^{train}).$$

Depending on the model, we have to estimate it in slightly different ways. There is no canonical why to capture across treatment differences in the pure strategy model the way we do in the memory-1 mixed and learning model. Instead, we follow the literature and make a separate estimation for each treatment. For the memory-1 mixed and learning models however, we estimate one single finite mixture model for all treatments.

F.3.1 Results

In table 15 we see the prediction errors of the different models.

Model	N	AllD	Loss	Accuracy
Naive			0.435	84.3%
Pure Strategies			0.323	87.6%
Learning with semi-grim	1		0.328	87.2%
	1	Yes	0.308	87.4%
	3		0.294	87.9%
	3	Yes	0.288	87.9%
Learning with memory-1	1		0.328	87.2%
	1	Yes	0.307	87.4%
	3		0.289	88.1%
Learning with flexible memory-1	1		0.326	87.2%
	2		0.296	88.1%
	3		0.287	88.1%
Learning at all h	1		0.329	87.1%
	2		0.291	87.9%
	3		0.287	88.1%
GBT memory-1			0.225	90.88%
GBT memory-3			0.223	90.93%

Table 15: Out of sample prediction errors for predicting the next action taken by an individual.

As we see, a single type of the main learning model performs about as well as fitting 11 different pure strategy types on each treatment. Allowing for heterogeneity in the learning model, and dropping the semi-grim restriction improves it, but flexible memory-1 behavior that adjusts to Δ^{RD} does not, and neither does extending learning to all h.

F.4 Maximum Likelihoods

Our main analysis of OSAP focus on the prediction errors of the estimated models of the next action taken by individuals, since this allows for straightforward comparisons between models of different complexities. However, since it is more common in the literature to use to consider the likelihoods instead of predictive abilities, we report these likelihoods for completeness. For every history $h_i(t)$ the behavior of type j is captured by a function $\sigma^j : \mathcal{H} \to [0,1]$ that takes a history and assigns a probability to cooperate. Each model comes with set of parameters. We will go through the different models in the following subsections, but first present the general estimation procedure.

If we let $a_i(t) \in \{-1, 1\}$ denote the action taken by individual i at time t, the likelihood of the observed behavior for participant i if she was of type σ^j with parameters is given by

$$\mathrm{Pr}_i(\sigma^j|\theta_j) = \prod_{t\in T(i)} \sigma^j(h_i(t))^{\mathbb{1}\{a_i(t)=1\}}(1-\sigma^j(h_i(t)))^{\mathbb{1}\{a_i(t)=-1\}}.$$

Let $\theta = (\theta^j)_{j=1}^J$ denote the parameters of the different types, and let $\phi \in \Delta(J)$ denote their relative share. A is then a pair $m = (\theta, \phi)$, and its likelihood is

$$\mathcal{L}\left(m|\theta,\phi,I\right) = \sum_{i\in I} \log\left(\sum_{j=1}^{J} \phi^j \mathrm{Pr}_i(\sigma^j|\theta_j)\right).$$

The model is then estimated by maximum likelihood.

Our main learning model only has six parameters per type, and these six parameters are the same across treatments. In comparison, the pure strategy model incorporates 11 different pure strategies, each with a different mistake probability, and these are estimated separately for each of the 28 treatments. If we were to directly compare the pure strategy model's loglikelihoods with the initial round learning model's loglikelihoods, we would be comparing a model with 736 parameters and one with 6.

To make the comparison more meaningful, here we consider the models estimated separately on each treatment as well as on the overall data, and we include BIC values to compensate for model complexity.

We consider three versions of each model (except the 11-type pure strategy model): A single type, a single type plus an AllD type, and three types. Since the first period learning model does not include AllD as a subset, we also consider a version with three first period learning types and one AllD.

In table 16 we see the loglikelihoods, estimated using F.4, of the different models, estimated and evaluated on the full supergames, and in table 17 evaluated on the last third of the supergames in each session.

Model	N	AllD	Estimated on	Loglikelihood	BIC
Pure	11	No	Each Treat	-64950	136881
Learning with semi-grim	1	No	Each Treat	-65327	132649
	1	Yes	Each Treat	-61028	124715
	3	No	Each Treat	-60533	127714
	3	Yes	Each Treat	-59020	124689
	1	No	All Treat	-67457	134987
	1	Yes	All Treat	-63352	126803
	3	No	All Treat	-60677	121575
	3	Yes	All Treat	-58902	118048
Learning with memory-1	1	No	Each Treat	-64707	131740
	3	No	Each Treat	-56856	120694
	1	No	All Treat	-67077	134240
	3	No	All Treat	-59084	118424
Learning at all h	1	No	Each Treat	-64245	132148
	3	No	Each Treat	-56379	123730
	1	No	All Treat	-67532	135199
	3	No	All Treat	-58900	118204

Table 16: Maximum likelihood log-likelihoods evaluated on the complete set of supergames.

In the literature, it is common to focus on the latter part of the experiment, under the assumption that behavior then has become more stable.

Model	N	AllD	Estimated on	Loglikelihood	BIC
Pure	11	No	Each Treat	-15522	37002
Learning with semi-grim	1	No	Each Treat	-17634	36969
	1	Yes	Each Treat	-15560	34525
	3	No	Each Treat	-14961	33893
	3	Yes	Each Treat	-14642	36659
	1	No	All Treat	-17885	35837
	1	Yes	All Treat	-16626	33341
	3	No	All Treat	-15380	30960
	3	Yes	All Treat	-15099	30421
	1	No	Each Treat	-16662	35310
	3	No	Each Treat	-13927	34377
	1	No	All Treat	-17788	35620
	3	No	All Treat	-14943	30043
Learning at all h	1	No	Each Treat	-16429	34843
	3	No	Each Treat	-13874	34273
	1	No	All Treat	-18037	36119
	3	No	All Treat	-14498	29152

Table 17: Maximum likelihood log-likelihoods evaluated on the last third of the supergames.

As shown in the tables above, these maximum likelihood results on either all supergames or the last third are both consistent with the primary analysis. According to the BIC, the best model is the learning model that extends to all memory-1 histories, while the pure strategies model is one of the worst. However, we still achieve relatively good performance with the simpler learning model that keeps behavior after the initial round constant across treatments and individuals, especially if we include AllD.

We see the same ordering as for the predictions: mixed strategies are better than pure strategies, first period learning plus AllD is better than mixed strategies, fist period learning with flexible mixed strategies better still, and the best is the full learning model.

We also see that we accurately capture the between treatment variation within our models. The loglikelihood is often similar for the models estimated jointly for all treatment, with logistic functions of Δ^{RD} capturing the variation between treatments, and the ones estimated separately for each treatment. And the lowest BIC is given by such joint estimation.