PENN INSTITUTE for ECONOMIC RESEARCH
UNIVERSITY of PENNSYLVANIA

The Ronald O. Perelman Center for
Political Science and Economics (PCPSE)
133 South 36th Street
Philadelphia, PA 19104-6297

pier@econ.upenn.edu
http://economics.sas.upenn.edu/pier

PIER Working Paper
18-009

Predicting and Understanding Initial Play

DREW FUDENBERG
MIT

ANNIE LIANG
University of Pennsylvania
Department of Economics

April 30, 2018

https://ssrn.com/abstract=3076682

Predicting and Understanding Initial Play*

Drew Fudenberg† Annie Liang‡

First version: November 14, 2017
This version: April 30, 2018

Abstract

We take a machine learning approach to the problem of predicting initial play in strategic-form games, with the goal of uncovering new regularities in play and improving the predictions of existing theories. The analysis is implemented on data from previous laboratory experiments, and also a new data set of 200 games played on Mechanical Turk. We first use machine learning algorithms to train prediction rules based on a large set of game features. Examination of the games where our algorithm predicts play correctly, but the existing models do not, leads us to introduce a risk aversion parameter that we find significantly improves predictive accuracy. Second, we augment existing empirical models by using play in a set of training games to predict how the models' parameters vary across new games. This modified approach generates better out-of-sample predictions, and provides insight into how and why the parameters vary. These methodologies are not special to the problem of predicting play in games, and may be useful in other contexts.

*We are grateful to Colin Camerer, Vincent Crawford, Charles Sprenger, Emanuel Vespa, and Alistair Wilson for very helpful comments and suggestions, and to Microsoft Research and National Science Foundation grant 1643517 for financial support.

†MIT

‡University of Pennsylvania

1 Introduction

In most game theory experiments, the distribution of play the first time participants play a game is not well approximated by the predictions of equilibrium analysis. Initial play does however have regularities, as for example shown by the fact that level-k thinking (Stahl and Wilson, 1994), the Poisson Cognitive Hierarchy model (Camerer, Ho and Chong, 2004), and related models surveyed in Crawford, Costa-Gomes and Iriberri (2013) fit initial play reasonably well in many one-shot simultaneous-move games. We use machine learning to try to uncover new regularities, and to develop modifications that improve the predictions of existing models.

We take two approaches:

A. Look at games where machine learning models predict well but existing models don't, and try to see why. We construct a set of game features based on payoff matrices, and use machine learning algorithms to learn a prediction rule mapping features into play. We then examine the games where play is well predicted by our machine learning algorithm, but poorly predicted by existing models. The fact that the ML algorithm predicts well in these games suggests that there are potentially underlying regularities in play not captured by the existing models, so we study them in detail. From these games, we identify a single parameter extension to the existing models, which we find significantly improves predictive accuracy.

B. Use machine learning to guide the choice of model parameters, or the choice of the model itself. The Poisson Cognitive Hierarchy model (hereafter, the PCHM) has a single free parameter τ, which (loosely) describes the distribution of player sophistication. In the baseline model, the value of τ is assumed to be constant across games. We show that we can improve prediction by choosing τ based on structural features of the game (using a machine learning algorithm trained on auxiliary data). We then generalize the approach, using the machine learning model to choose which of two models (PCHM or a variation on level 1) is better suited for predicting play in different games.

These improvements on existing theories of initial play are of interest in their own right, but our methodologies for using machine learning to extend and inform modeling are more general. Their success here suggests that they may be useful in other problem domains within economics as well.

In this paper we consider two datasets of game play. The first is play in 86 lab games from six past lab experiments. Since the games in this data set were designed for certain experimental goals, their payoff matrices tend to possess "strategically interesting" features. In order to determine the robustness of our results to these features, we augment this set of games with a new dataset of play by Mechanical Turk subjects in 200 games with randomly drawn payoffs.[1] Compared to the lab games, the games with random payoffs turn out to be more likely to be dominance solvable, more likely to include a strictly dominated action, and more likely to contain an action profile that is clearly best for both players. From prior work, we expect these

[1] See Erev et al. (2007) and Ert, Erev and Roth (2011) for earlier work measuring the performance of models using a random sample of games.

differences to make initial play in the new games somewhat easier to predict, and indeed this is what we find in the subsequent analysis.

We study two sorts of predictions tasks: predicting the action played—where error is minimized by predicting the modal action in each game—and predicting the distribution of actions. In both cases, we evaluate the performance of various prediction rules by their *completeness*, which we take to be the percentage of the possible improvement over random guessing (Peysakhovich and Naecker, 2017; Kleinberg, Liang and Mullainathan, 2017).[2]

We begin with the task of predicting the realized action, which is a classification problem, and has the advantage that the associated algorithms are easier to interpret. The modal action turns out to be level 1 in 72% of the lab games; correspondingly, simply predicting the level 1 action performs well, achieving 80% of the attainable improvement over random guessing. We find that the best-performing version of PCHM, which extends the level k model by assuming that types best respond to a Poisson distribution over lower level types, is equivalent to the level 1 model when its free parameter τ is estimated from training data. Thus, the PCHM achieves the same performance as level 1 on this task. In our set of random games, the modal action is level 1 in 87% of the games, and again the level 1 model and PCHM make the same predictions; here, they achieve a completeness measure of 88%.

We then create a large set of game features, including indicators for whether each action satisfies certain strategic properties (level 1, level 2, part of a Nash equilibrium, etc.), and train decision trees to use these features to predict play. When predicting the lab data, we find that the best decision tree with two decision nodes reproduces the level 1 prediction, but the best out-of-sample predictions are made by a tree with three decision nodes. We then examine the 9 (out of 86) lab games where the modal action is not level 1, but is correctly predicted by the best decision tree. It turns out that in each of these games, there is an action whose average payoffs closely approximate the level 1 action, and which additionally yields lower variation in possible payoffs. Players are more likely to choose this "almost" level 1 action than the actual level 1 action.

One explanation for this behavior is that players maximize a concave function over game payoffs. Motivated by this finding, we modify the level 1 prediction by assuming that agents have utility function for money payoffs of $u(x) = x^\alpha$. We find that estimating the single parameter α substantially improves out-of-sample prediction error, and improves upon the predictive accuracy of our feature-based prediction rule. This suggests that atheoretical prediction rules fit by machine learning algorithms can be used to help craft interpretable parametric models that fit better than the current state of the art. Extending the level 1 model in this way also generates better predictions in the random games (as compared to the benchmark of the level 1 model

[2] Camerer, Ho and Chong (2004)'s related "economic value" compares the expected payoff that results from best-responding to a theory's forecast, versus the payoff that subjects actually obtained. This measure is in payoffs while the completeness measure that we use evaluates predictive accuracy only. We note that it is not clear how to compute the economic value in our first prediction task, as we would need to generate a best response from a prediction of the most likely action.

and PCHM), although here the decision tree model performs slightly worse.

We then turn to the (more frequently studied) problem of predicting the distribution of play. As a naive baseline, we consider prediction of uniform play, where each action is played 1/3 of the time.[3] The PCHM obtains a significant improvement over this baseline, achieving 50% of the possible improvement in predicting the lab data set, and 78% of the possible improvement in predicting the Mechanical Turk data set. Several proposed variations do better still: adding a risk aversion parameter improves the predictive accuracy of the PCHM, as does replacing the assumption of exact maximization with logit responses, as in Stahl and Wilson (1994, 1995) and Leyton-Brown and Wright (2014). The latter approach (which we refer to as LPCHM) is improved further if we assign probability zero to level 0 players in the population (while allowing them to exist in the perceptions of players of higher levels). Notice that although adding additional parameters always improves in-sample fit, it need not reduce out-of-sample predictive error.[4] These methods attain 61-77% of the achievable improvement over guessing at random, and 78-84% of the possible improvement in predicting the Mechanical Turk data set

We next explore a new way to use game features for prediction. Specifically, we note that the distributional predictions of the PCHM and its variants are sensitive to the choice of the parameter τ (the Poisson rate parameter). Moreover, as has been noted in prior work (Camerer, Ho and Chong, 2004), the best-fitting value of τ varies substantially from game to game. Variation in the best-fit value of τ across games suggests that better predictions can be made by allowing for heterogeneity in τ. To accommodate parameter heterogeneity without eliminating the PCHM's predictive content, we propose a way to estimate the appropriate value of τ for a given game from other data. Our approach here is to learn a predictive function from game features to best-fit values of τ, and use this function to predict heterogeneity in values of τ in out-of-sample games. This technique turns out to appreciably improve prediction for both datasets we consider. Moreover, examining a simple decision tree (with two decision nodes) used to predict τ helps us to understand which features correlate with variation of the best-fit values of τ. We find that τ tends to be high in games that have a relatively "obvious" action, meaning that either the level 1 action is a much better response to the uniform distribution than any other action, or that some action profile gives a particularly high payoff. We repeat this procedure of introducing parameter heterogeneity to the LPCHM and the LPCHM without level 0 players, and again obtain improvements in predictive accuracy.

In addition to helping us choose model parameter values, a generalization of this method can also help us choose between models. As an illustration, we consider two models, PCHM and a variation of level 1. We learn a classification rule for predicting which model fits better (based on structural features of the payoff matrix alone). For each out-of-sample game, we first use

[3] We also consider prediction of uniform play over the actions consistent with Nash equilibrium; this turns out not to improve upon the naive baseline.

[4] Throughout, when we say *in-sample*, we mean that the data used for training and testing are the same. Increasing the flexibility of a model always allows for higher in-sample fit. When we say *out-of-sample*, we mean that different data is used for training and for testing. Increasing the flexibility of a model need not result in higher out-of-sample fit; in particular, more complex models are prone to overfitting to the training data.

the classification rule to select Model A or Model B, and then use the corresponding estimated model to predict play. This approach improves predictive accuracy over both of Model A and Model B applied in isolation.

Finally we consider another way to search for predictable patterns that aren't captured by current models: We incentivize MTurk participants to predict play by other individuals. Specifically, subjects were shown a set of games and asked, for each game, to pick the action that they thought was most frequently played. We find that in most cases the "naive crowd prediction rule"—which predicts in the two tasks, respectively, the modal crowd prediction and the distribution of crowd predictions—does better than the PCHM. Notice that the payoff matrices are not an input into the crowd rule: all information about the game itself is derived from the perception of the crowd subjects. Moreover, we find that the distribution of these predictions is significantly different than the distribution of play, so the predictions are not simply "proxy plays." These results point to the potential for using human predictions to develop better models and further improve predictions.

1.1 Background Information and Related Work

As the Crawford, Costa-Gomes and Iriberri (2013) survey shows, there is an extensive literature on initial play in matrix games. Most if not all of these papers use some variant of "cognitive hierarchies" in that their starting point is the specification of a "level 0" or unsophisticated player, who most often is assumed to play a uniform distribution. The various cognitive hierarchy models then use the level 0 type to build up a richer specification of play. The simplest such model is "level 1," which assumes that the whole population plays a best response to level 0. This is too stark a model to be a good fit for the observed distribution of play in most games, but we we will see it does a good job of predicting the most likely (i.e. modal) action.

Work on initial play has had the twin goals of offering an alternative to Nash equilibrium as a way of predicting play and of providing a model of how people think in these settings. Our goal is to find simple and portable ways of making good predictions, which led us to focus on tractable models from the literature, in particular the PCHM. Our paper is closest to the improvements of the PCHM proposed by Leyton-Brown and Wright (2014) and Chong, Ho and Camerer (2016): Leyton-Brown and Wright (2014) replaces the specification of level 0 from uniform play to a weighted linear model based on five game features, and Chong, Ho and Camerer (2016) defines the level 0 player to randomize only over actions that are "never-worst." Our paper is also similar in spirit to Fragiadakis, Knoepfle and Niederle (2016), which tries to identify the subjects whose play has regularities that are not captured by cognitive hierarchies.

Our paper is also related to other papers that have focused on improving prediction of play in games, including Ert, Erev and Roth (2011), which compares the performance of various models of social preference (and their combinations) for predicting play in a class of extensive-form game experiments, and Sgroi and Zizzo (2009) and Hartford, Wright and Leyton-Brown (2016), which develop deep learning techniques for predicting play. These papers differ from the present

paper in that their objective is predictive accuracy alone, and not on deriving conceptual lessons or portable models.

There is also an extensive literature on the prediction of play in repeated interactions with feedback, where learning plays an important role; see e.g. Erev and Roth (1999), Crawford (1995), Cheung and Friedman (1997) and Camerer and Ho (1999). In this paper, we consider only initial play, leaving open the question of how machine learning methods can contribute to our understanding of play in repeated settings.[5]

Costa-Gomes and Weizsacker (2007) compare elicited beliefs over play with play itself, and find that players both approximately act like level 1 players and also believe others to act like level 1 players. This is related to our section 6.1 on crowd predictions, where we find that reported beliefs can be used as inputs into predicting play. Relatedly, DellaVigna and Pope (2017) show the ability of untrained human subjects to predict economic behaviors in a different context, that of forecasting the efficacy of different experimental incentives.

2 Data and Experiments

Throughout the paper we consider only 3x3 matrix games. The set of payoff matrices is identified with $G = \mathbb{R}^{18}$, and we use g to mean a typical payoff matrix. The set of row player actions is A_{row}, the set of column player actions is A_{col}, and the set of action profiles is $A = A_{\text{row}} \times A_{\text{col}}$. Finally, we use $u_{\text{row}} : A \to \mathbb{R}$ and $u_{\text{col}} : A \to \mathbb{R}$ to mean the row player's and the column player's payoffs respectively.

Below, we describe two data sets of game play. Our first data set, presented in Section 2.1, aggregates play across several past lab experiments. Since the games in this data set were designed for certain experimental goals, they are not a random sample but tend to possess strategically interesting features. In order to determine the robustness of our results to these designs, we augment the lab games with a novel data set, described in Section 2.2, that consists of play by Mechanical Turk subjects in games with randomly drawn payoffs.

2.1 Lab Data

Our data on play in laboratory experiments consists of all 3x3 matrix games in the data set collected (and used) by Kevin Leyton-Brown and James Wright (see e.g. Leyton-Brown and Wright (2014)). This data includes 40-147 observations of play in each of 86 symmetric three-by-three normal-games. Table 1 below lists the number of games, as well as the number of observations of play, from each paper.

[5] Camerer, Nave and Smith (2017) uses machine learning to predict play in a repeated bargaining game.

Paper	Games	Total # of Observations
Stahl and Wilson (1994)	10	400
Stahl and Wilson (1995)	12	576
Haruvy, Stahl and Wilson (2001)	15	869
Haruvy and Stahl (2007)	20	2940
Stahl and Haruvy (2008)	18	1288
Rogers, Palfrey and Camerer (2009)	17	1210
Total	**86**	**6887**

Table 1: Original sources for the lab play data.

The subject pool and payoff scheme differ across the six papers, but all of them use anonymous random matching without feedback: participants play each game only once, are not informed of their partner's play, and do not learn their own payoffs until the end of the session. Since our data set is limited to symmetric games, we label all observed actions (whether chosen by a column player or a row player) as row-player actions.

There is substantial variation in the distribution of play across games. For example, the fraction of subjects who chose the most frequently chosen action (the modal action) varies from 39.19% to 94.56%. Relatedly, there is large variation in how far the observed distribution differs from a uniform distribution over actions. The entropy of the observed distribution of play[6] varies from 0.5 (close to degenerate) to 1.09 (close to uniform). See Figure 1 for the distributions of both measures.

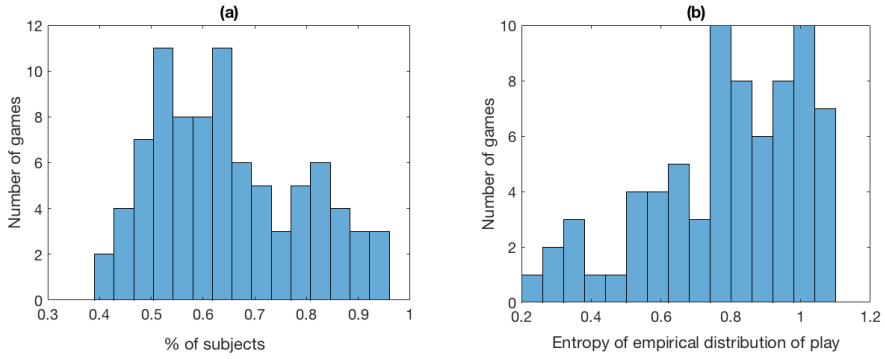


Figure 1: In our lab games: (a) % subjects who chose the modal action; (b) entropy of the distribution of play.

We show below the two games that achieved highest and lowest values according to these measures. The game with the lowest modal frequency (also the game whose distribution has the highest entropy) is:

[6]The entropy of frequency vector (p_1, p_2, p_3) is given by $\sum_{i=1}^{3} p_i \cdot \log(p_i)$.

	a_1	a_2	a_3	Frequency
a_1	21,21	93,13	45,29	32.43%
a_2	13,93	69,69	53,53	28.38%
a_3	29,45	53,53	61,61	39.19%

Although most subjects chose action a_3, play is close to uniform. In contrast, in the game with the highest modal frequency (also the game whose distribution has the lowest entropy), 95% of subjects chose the same action:

	a_1	a_2	a_3	Frequency
a_1	35,35	35,25	70,0	2.72%
a_2	25,35	55,55	100,0	94.56%
a_3	0,70	0,100	60,60	2.72%

It is important to note that all of these games were designed for specific experimental goals. For example, Stahl and Wilson (1994) write that: "Ten symmetric (3×3 games) were selected for a variety of characteristics: three were strict dominance solvable, two were weak dominance solvable, six had unique pure-strategy symmetric Nash equilibrium, while two had unique mixed-strategy Nash equilibria." To determine the robustness of our findings to design features such as these, we augment the laboratory data with a large data set of play in games with randomly generated payoffs, which we now describe below.

2.2 Random Games

We randomly generated 200 payoff matrices from a uniform distribution over $\{10, 20, \ldots, 90\}$[18]. This scale was chosen to match the lab experiments described above, although unlike in the previous section, the randomly generated games are not symmetric. We presented each of 550 Mechanical Turk subjects with a random subset of fifteen games, and asked them to play as the row player.[7]

Subjects were incentivized by the following payment scheme. On top of a base payment of \$0.35, subjects were told that one of the fifteen games would be chosen at random, and their action would be matched with another subject who had been asked to play as the column player. Their joint moves determined payoffs that were multiplied by \$0.01 to determine the subject's bonus winnings (ranging from \$0.10 to \$0.90). Subjects spent on average 7 minutes on the task, and the average payment was \$0.93, or \$8.14 an hour.[8] The minimum payment was \$0.45 and the maximum payment was \$1.25; the standard deviation of payments was \$0.23. The complete set of instructions can be found in Appendix B.3.[9]

[7] Each game was shown to 25-58 subjects, and the average number of responses per game was 41.25.

[8] This is a typical hourly wage for MTurk.

[9] In addition to eliciting play, we asked for subjects to volunteer a free-form description of how they made their decisions. Example answers can be found in Section B.4 of the Online Appendix.

To understand how the randomly generated games differ from the lab games, we compare various summary statistics of the two sets of payoff matrices. Relative to the random games, the games played in lab experiments are more likely to have a higher number of pure-strategy Nash equilibria and a higher number of rationalizable actions, as shown in Figure 2. These differences are large, suggesting that the set of lab games is indeed different from what we would expect in a random sample.

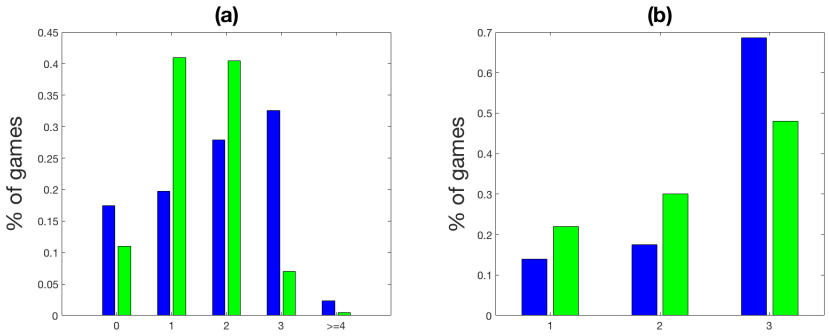


Figure 2: *Blue*—lab games; *green*—random games. (a) Percentage of games with zero, one, two, three, or at least four pure strategy Nash equilibria; (b) Percentage of games with one, two, or three actions surviving iterated elimination of (pure-strategy) dominated actions.

Appendix A.1 reports the distributions of several additional summary statistics, which further illustrate the differences between the sets of games: Lab games have payoffs with larger variation (higher variance, a larger maximum payoff, and a smaller minimum payoff), are less likely to be dominance solvable, less likely to include a strictly dominated action, and less likely to contain an action profile that is clearly best for both players (by various measures that we define). From prior work, we expect these differences to make initial play in the new games somewhat easier to predict, and indeed, this is what we find in the subsequent analysis. (Ideally we would use a sample of games that corresponds to the distribution of games that people face in the field, but we do not know what that distribution would be.)

As we saw with the lab data, there is substantial variation in how subjects played across the different (randomly generated) games. The percentage of subjects who chose the modal action varies from 36.84% to 100%, and the entropy of the observed distributions of play varies from 0.22 to 1.09. See Figure 3 for the distributions of both measures across games. As one might expect given the difference in the games, play in the MTurk data tends to be more concentrated on the modal action; for example, the modal action was chosen by over 75% of subjects in 38.50% of games in our MTurk data set, in contrast to only 25.58% of the lab games.

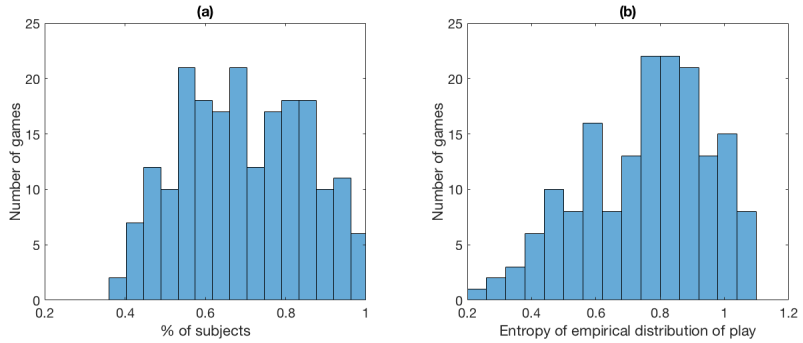


Figure 3: In our MTurk games: (a) % of subjects who chose the modal action; (b) entropy of the observed distribution of play.

3 Prediction Tasks and Measures of Performance

We use two prediction tasks to evaluate how well can we predict play. The first prediction task is a classification problem: given a specific instance of play of a fixed game, we seek to predict which action the row player chose. For this problem, a prediction rule is a mapping from games to row player actions

$$f : G \to A_{\text{row.}}$$

An observation is a pair (g_i, a_i) where g_i is the game played in observation i and a_i is the action that the row player chose.[10] Given a set of n observations $\{(g_i, a_i)\}_{i=1}^n$, we measure the error of prediction rule f using the *misclassification rate*

$$\frac{1}{n}\sum_{i=1}^{n} \mathbb{1}(\mathbf{a}_i \neq f(g_i)).$$

This is the fraction of observations in which the predicted action is not the action that was chosen in the given instance of play.

The *ideal* prediction rule for a given test set assigns to each game the (actual) modal action in that game. Note that this rule will be imperfect (i.e. have a strictly positive misclassification rate) unless the empirical distribution is a point mass in all games. The error achieved by this ideal rule is a lower bound on the best achievable error, but it need not be a tight bound. This is because the prediction rule uses knowledge of the data to be predicted, and so its error is not out-of-sample. (We discuss alternative benchmarks at the end of this section). The *naive* prediction rule guesses uniformly at random; this yields a misclassification rate of 2/3, which is also the best possible rate when the empirical distributions are uniform.[11]

[10] See Appendix B.1.1 for a related exercise where we take each observation to be a game and its modal action, instead of a given instance of play of a game.

[11] We can extend the definition of a prediction rule to any map $f : G \to \Delta(A_{\text{row}})$ and the misclassification rate

In the second prediction task, we seek to predict the distribution over (row player) actions in each game. A prediction rule for this problem is a mapping from payoff matrices to distributions over row player actions:

$$h : G \to \Delta(A_{\text{row}}).$$

An observation is a pair $(g_i, \mathbf{p}_i)$, where g_i is a (distinct) game and $\mathbf{p}_i$ is the distribution over (row-player) actions observed in that game. Error is assessed using *mean-squared error*

$$\frac{1}{n}\sum_{i=1}^{n}\frac{1}{3}\|\mathbf{p}_i - \hat{\mathbf{p}}_i\|^2,$$

where $\mathbf{p}_i$ is the frequency vector over actions observed for game i, $\hat{\mathbf{p}}_i$ is the predicted frequency vector, the norm is the Euclidean norm, and n is the number of games.

In this problem, the *ideal* prediction rule predicts the actual realized frequency vector for each game, and so has zero mean-squared error; the *naive* prediction of the uniform distribution has a mean-squared error that depends on the data. Again, the ideal rule is a lower bound that may not be attainable as it uses knowledge of the test data. In what follows, we use the gap between the errors of the naive and ideal prediction rules to calibrate the success of various prediction rules. Throughout, we separate the instances of play in lab games and random games, and assess error for each dataset separately.

Unless explicitly stated otherwise, we use tenfold cross-validation in computing prediction errors. For the first prediction task, this means that we divide the games into ten folds, use all observations of play associated with games in nine of the folds for training, and use the observations of play associated with games in the remaining fold for testing. The reported error is averaged across the different choices of test fold. For the second prediction task, we divide the games into ten folds, use all games in nine of the folds for training, and use all games in the remaining fold for testing; again, we report the average prediction error across choices of test set. The standard errors for the cross-validated prediction errors are estimated as the standard deviation of prediction errors across choices of test sets, divided by $\sqrt{10}$, because we use 10 folds. (see Hastie, Tibshirani and Friedman (2009) for a reference).[12] Some of our prediction algorithms are based on game matrices alone (and thus do not require estimation from a training set). For these prediction algorithms, we report bootstrapped standard errors.

This way of constructing the folds to do cross-validation is a more stringent test than a related cross-validation exercise, in which observations are pooled across games before being subdivided into folds (see e.g. Leyton-Brown and Wright (2014)). Under the alternative method, it is very likely that instances of play in every game appear in both the training and testing data. This substantially reduces the challenge of learning to predict play in the games *in* the data

to the average expected error $\frac{1}{n}\sum_{i=1}^{n}\mathbb{E}\left(\mathbb{1}(\mathbf{a}_i \neq f(g_i))\right)$.

[12] This is a standard approach for computing the standard error of a cross-validated prediction error, although it ignores correlation across the folds.

set, but prediction rules learned in this way may lead to worse prediction of games outside of it. In contrast, our cross-validation approach above forces the test data and training data to consist of observations of play in *different* games. We do report results for this alternative "observation-level" cross-validation in Section B.1.4 of the Online Appendix. As expected, all prediction errors are higher; we find additionally that the qualitative results that we report below are unchanged.

In Appendix A.3 we consider two alternative benchmarks for "ideal" prediction. In the first, for each game, we split observations of play into training and testing sets, and use the empirical distribution from the training set to predict play in the test set. That is, our prediction of the modal action in a given game is the modal action *in that game* in the training data, and our prediction of the action distribution in a given game is the empirical distribution in the training data. This "table lookup" approach provides a consistent estimate of the best possible out-of-sample error (in contrast to the in-sample error that we report in the main text), but performs poorly with small samples. In the second, we use the empirical distribution of play in the full data set to predict play in a re-sampled data set. Completeness measures using these alternative benchmarks are qualitatively similar to those we report in the main text.

4 Predicting the Action Played

4.1 Approaches

We evaluate and compare several approaches for the problem of predicting the realized action in a given instance of play. We first consider approaches based on Nash equilibrium, the level-k models of Stahl and Wilson (1995), and the Poisson Cognitive Hierarchy model of Camerer, Ho and Chong (2004).

Uniform Nash. We evaluate a prediction rule based on the hypothesis that play is a uniform distribution over the set of actions that are consistent with a pure-strategy Nash equilibrium.[13] Formally, define the set of actions $a_i \in A_{\text{row}}$ such that (a_i, a_j) is a Nash equilibrium for some $a_j \in A_{\text{col}}$, and predict at random from this set.

Level 1. Following Stahl and Wilson (1994, 1995), define a player to be "level 0" if he randomizes uniformly over his actions, so that his distribution of play is given by

$$P_0(a_i) = 1/3 \qquad \forall\, i \in 1, 2, 3$$

The level 1 player best responds to a level 0 player, and the level 1 prediction rule assigns to each game its level 1 action. When the level 1 prediction is not unique, we randomize over the set of level 1 actions.[14]

[13] This prediction rule is considered previously in Leyton-Brown and Wright (2014).

[14] The level 1 prediction is unique in all of the lab games, but not in all of the random games.

Poisson Cognitive Hierarchy Model (PCHM). Following Camerer, Ho and Chong (2004), define level 0 and level 1 as above and define the play of level k players, $k \geq 2$, to be the best responses to a perceived distribution

$$g_k(h) = \frac{\pi_\tau(h)}{\sum_{l=0}^{k-1} \pi_\tau(l)} \qquad \forall\ h \in \mathbb{N},\ h < k, \tag{1}$$

over (lower) opponent levels, where π_τ is the Poisson distribution with rate parameter τ. The predicted distribution over actions is based on the assumption that the actual proportion of level k players in the population is $\pi_\tau(k)$. We predict the modal action according to this aggregated distribution. Throughout, we take τ to be a free parameter and estimate it from the training data, allowing different values of τ for each dataset.

Prediction rules based on game features. In addition to the methods described above, we introduce prediction rules based on features that describe strategic properties of the game matrix. Specifically, for each action, we define an indicator variable for whether the action has each of the following properties: whether it is part of a Nash equilibrium, whether it is part of an action profile that maximizes the sum of player payoffs (*altruistic* in Costa-Gomes, Crawford and Broseta (2001) and *efficiency* in Leyton-Brown and Wright (2014)), whether it is part of a Pareto dominant Nash equilibrium, whether it is level k (for each $k \in \{1, 2, \ldots, 7\}$), and whether it allows for the highest possible row player payoff (*optimistic* in Costa-Gomes, Crawford and Broseta (2001) and *max-max* in Leyton-Brown and Wright (2014)). We include additionally a score feature for how many of the above properties each action satisfies. The higher an action's score, the more compelling a choice it is.

We use a *decision tree algorithm* to learn predictive functions from these features to outcomes. Decision trees recursively partition the feature space and learn a (best) constant prediction for each partition element. We consider trees which use only a single feature to determine the split at each node, and use the standard approach of building up the decision tree one node at a time using a greedy algorithm. Thus, the first node is the best single split, the second node is the best second split conditional on the first, and so forth. Appendix B.1.3 reports predictions of the action played in lab data using a random forest algorithm and finds no substantial improvement. For a closely related prediction task,[15] we compare the decision tree model against lasso logistic regression and a 2-layer neural net, in addition to the random forest algorithm. Prediction errors using these alternative algorithms are comparable to those from the decision tree, and we expect that the same would be true on on our other data sets. Since the outputs of these alternative algorithms are harder to interpret, we focus on decision trees in the main text.

[15] For computational reasons, it is easier to consider the prediction problem in which each game is considered a single observation and modal actions are predicted (see Appendix B.1.1).

4.2 Results

Table 2 reports the misclassification rates and completeness measures for these prediction rules when predicting the distribution of play in the lab data, where the error attained by the naive prediction rule is set to 0 and the error attained by the ideal rule is set to 1.

When evaluating the PCHM, the best-performing τ (estimated from training data) turns out to correspond to predicting the level 1 action, so we report the performance of these two models together.[16,17]

	Error	Completeness
Naive benchmark (guess at random)	0.6667	0%
Best constant prediction (always guess a_1)	0.6013	18.96%
	(0.0797)	
Uniform Nash	0.5507	33.66%
	(0.0055)	
Level 1/PCHM	0.3889	80.55%
	(0.0079)	
Prediction based on game features	0.3652	87.42%
	(0.0057)	
Ideal prediction	0.3218	100%

Table 2: Predicting the realized action in lab data.

We find that the uniform Nash prediction rule improves slightly over guessing at random, and achieves a completeness measure of approximately 34%. The level 1 model achieves a substantially larger improvement, increasing completeness to 81%. Finally, the prediction rule based on game features does better still, achieving a completeness of 87%.[18]

We now ask what the prediction rule based on game features looks like, and why this decision tree outperforms the level 1 model. As a first pass, we examine the best decision tree under a severe parsimony constraint—that the decision tree use only two decision nodes. This "2-split" decision tree turns out to reproduce the level 1 model: the predicted action is the action that best responds to a uniform distribution over column player actions. In this sense, the level 1

[16] We find that prediction error is minimized at all values of τ in the interval $(0, 1.25]$. The values of τ in this range all yield prediction of the level 1 action for the games in our data sets.

[17] PCHM (and other variants we consider) better fit the *distribution* of actions, but we defer this discussion to Section 5.

[18] These qualitative results are robust to a modification of the prediction task that defines an observation to be a game and modal action pair, instead of a game and a given instance of play. There are 6887 observations in our main exercise and 86 observations under this alternative approach. The main reason to set an instance of play as the unit of observation is because it assigns greater importance to guessing the modal action in games in which the modal action is chosen more frequently. For example, if the modal action is chosen in 90% of observations for game 1, but only in 35% for game 2, then our main approach penalizes misprediction of the modal action in game 1 more than misprediction of the modal action in game 2. See Appendix B.1.1 for details.

model is the best "simple" prediction model.[19,20]

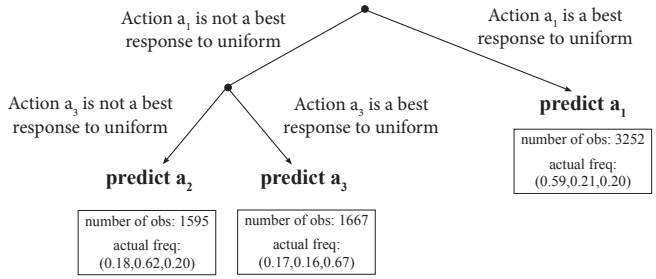


Figure 4: Best 2-split decision tree for predicting the realized action in lab data.

As we allow for additional complexity by increasing the number of decision nodes n, the best n-split decision tree builds on the level 1 model. Large values of n quickly result in overfitting. The decision tree with the best out-of-sample prediction (shown below) has $n = 3$, and appends a single additional criterion to the level 1 model: it agrees with the level 1 model except that even if action a_1 is level 1, it is not predicted if the number of reasons to choose a_2 is sufficiently large. In this case, action a_2 is predicted instead.

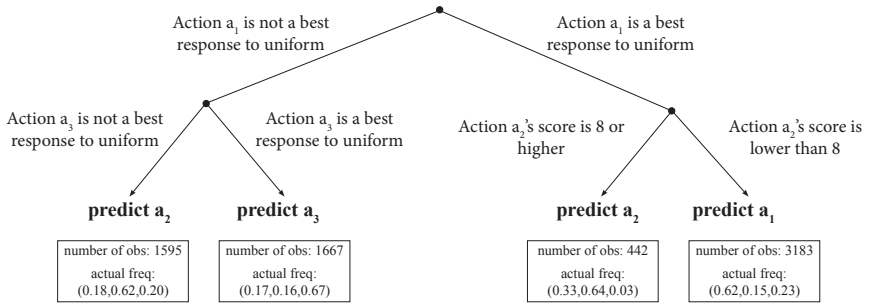


Figure 5: Best decision tree for predicting the realized action in lab data.

Out of 24 lab games in which the modal action is *not* level 1, there are nine games in which the modal action is correctly predicted by the decision tree above. This gives us reason to believe that the there is a systematic pattern to play in those nine games, beyond what is already captured by the level 1 model. We thus examine these games, displayed below in Figure 6, and search for additional regularities.

[19]This statement should be interpreted with respect to the features that we have defined. It may be that there is a new feature, outside of our set, that would allow for an even more predictive 2-split decision tree.

[20]The selection of level 1 features is robust to the choice of prediction task: in the problem of predicting the realized distribution of play in lab data, the best 2-split decision tree again picks out the level 1 features (see Section B.1.2 of the Online Appendix).

	a_1	a_2	a_3	Actual Freq:
a_1	*47,47*	*51,44*	*28,43*	51%
a_2	44,51	11,11	43,91	19%
a_3	**43,28**	**91,43**	**11,11**	30%

	a_1	a_2	a_3	Actual Freq:
a_1	*45,45*	*50,41*	*21,40*	81%
a_2	41,50	0,0	40,100	6%
a_3	**40,21**	**100,40**	**0,0**	13%

	a_1	a_2	a_3	Actual Freq:
a_1	**0,0**	**35,55**	**100,30**	34%
a_2	*55,35*	*40,40*	*20,0*	65%
a_3	30,100	0,20	0,0	0%

	a_1	a_2	a_3	Actual Freq:
a_1	15,15	0,0	0,100	0%
a_2	*0,41*	*90,90*	*10,0*	56%
a_3	**100,0**	**0,21**	**20,20**	44%

	a_1	a_2	a_3	Actual Freq:
a_1	**20,20**	**30,40**	**100,30**	35%
a_2	*40,30*	*40,40*	*60,0*	65%
a_3	30,100	0,60	40,40	0%

	a_1	a_2	a_3	Actual Freq:
a_1	1,1	0,10	0,100	0%
a_2	*10,0*	*90,90*	*10,5*	62%
a_3	**100,0**	**5,10**	**20,20**	38%

	a_1	a_2	a_3	Actual Freq:
a_1	**35,35**	**39,47**	**95,40**	11%
a_2	*47,15*	*51,51*	*67,15*	82%
a_3	40,100	15,67	47,47	7%

	a_1	a_2	a_3	Actual Freq:
a_1	10,10	10,15	10,100	2%
a_2	*15,10*	*80,80*	*15,0*	57%
a_3	**100,10**	**0,15**	**30,30**	41%

	a_1	a_2	a_3	Actual Freq:
a_1	**25,25**	**30,40**	**100,31**	44%
a_2	*40,30*	*45,45*	*65,0*	52%
a_3	31,100	0,65	40,40	4%

Figure 6: The most frequently played action (in *italics*) is predicted by the decision tree. The level 1 action is in **bold**.

Examining these games reveals a common feature: In each of the games, some action that is not level 1 yields an expected payoff against uniform play that is comparable to the level 1 payoff, and moreover has lower variation in possible row payoffs. Consider for example the first game in Figure 6. Action a_3 is the level 1 action in this game, but the expected payoff to action a_1 is not much smaller (42 vs. 48.33), and choice of action a_1 yields significantly lower variation in possible row player payoffs.[21] In our data, more subjects choose action a_1 than action a_3. This behavior appears in all of the nine games shown above: subjects preferred actions that were "almost level 1" when those actions yielded lower variation in payoffs.

With knowledge of this regularity, we can modify the level 1 model to incorporate it. Specifically, because the departure from level 1 behavior is consistent with a risk averse utility function over payoffs, we suppose that dollar payoffs u are transformed under $f(u) = u^\alpha$, which adds one parameter to PCHM. The standard assumption that players are expected value maximizers is nested as $\alpha = 1$. This revised model has two free parameters (τ, α), and as before, we can estimate these free parameters on training data and evaluate the estimated model out-of-sample.

[21] Depending on which action the column player takes, the row player will receive any of 43, 91, and 11 if he (the row player) chooses a_3, compared to 47, 51 and 28 if he (the row player) chooses a_1.

Table 3 compares the prediction error of this modified PCHM with the original model.[22] We find that introduction of risk aversion reduces prediction error substantially, achieving the prediction error of the best decision tree.

	Error	Completeness
Level 1/PCHM	0.3889	80.55%
	(0.0079)	
Prediction rule based on game features	0.3652	87.42%
	(0.0057)	
Level 1/PCHM with Risk Aversion	0.3642	87.71%
	(0.0093)	

Table 3: Introduction of risk aversion reduces prediction error.

Table 4 presents prediction errors for the MTurk data set. Relative to the lab data, absolute prediction errors are lower under each of the approaches considered, as anticipated in Section 2.2. However, most results are qualitatively similar: We again find that the PCHM and level 1 predictions coincide, and that the best 2-split decision tree generates the level 1 prediction. In contrast to the lab data, here the level 1 model outperforms the best decision tree (achieving a relatively high completeness measure of 88%).[23] Despite the strong performance of the level 1 model, adding a single parameter for risk aversion again yields an improvement in prediction: the level 1 model with risk aversion attains 91% of the achievable improvement over random guessing.

5 Predicting the Distribution of Play

Now we turn from the task of predicting the most likely action to the problem of predicting the empirical distribution over actions. Our baseline model here is the one-parameter PCHM described in Section 4.1. We consider variations on this model in Section 5.1 that add one or two parameters, as well as simpler models that assume only level 1 behavior. In addition to these variations on existing models, we introduce a new way of using parametric models to make predictions, which we describe in Section 5.2.

In this problem, predicting distributions using decision trees and lasso regression improves upon uniform Nash and the naive baseline, but does not improve upon PCHM. We report these errors in Appendix B.2.

[22] The estimated values of the free parameters are $\tau = 1$ and $\alpha = 0.6250$.

[23] Notice that although the level 1 model can always be reproduced by the decision tree algorithm given the set of features we have defined, the estimated tree varies depending on the training data. Table 4 thus says that it would be better to simply force the decision tree to be the level 1 model, instead of giving it the flexibility to learn alternative models from our feature set. Note also that there may well be other feature sets and other learning algorithms that would do better than the level 1 model here.

	Error	Completeness
Naive benchmark (guess at random)	0.6667	0%
Best constant prediction (always guess a_2)	0.6482	4.87%
	(0.0137)	
Uniform Nash	0.4722	51.21%
	(0.0075)	
Prediction based on game features	0.3430	85.23%
	(0.0050)	
Level 1/PCHM	0.3323	88.05%
	(0.0065)	
Level 1/PCHM with Risk Aversion	0.3211	91%
	(0.0046)	
Ideal prediction	0.2869	100%

Table 4: Predicting the realized action in MTurk data.

5.1 Existing Models and Some Variations

We consider several modifications of existing models of initial play:

PCHM with Risk Aversion. Motivated by the success of risk aversion in improving the prediction of level 1, we add risk aversion to the PCHM by transforming payoffs to $f(u) = u^\alpha$. The standard version of PCHM is returned for parameter choice $\alpha = 1$. This model has two free parameters τ and α, which we estimate on the training data and test out-of-sample.

PCHM with Logit Best Response (LPCHM). When fitting the standard PCHM to data, level 0 players not only determine the play of the level 1 players but also capture random play due to payoff shocks or errors, as well as any play that is not level k for some $k > 0$. (e.g. from errors or from payoff shocks). We can separate these roles by allowing for imperfect maximization by players. Specifically, following Stahl and Wilson (1994, 1995) and Leyton-Brown and Wright (2014), we replace the assumption of exact maximization with a logit best response. As before, define

$$P_0^r(a_i) = 1/3 \quad \forall\, a_i \in A_{\text{row}} \qquad\qquad P_0^c(a_j) = 1/3 \quad \forall\, a_j \in A_{\text{col}}$$

to be the distribution of play by a level 0 row player and a level 0 column player (respectively). Then for each level $k \geq 1$, recursively define

$$q_k^r(a_j) = \frac{1}{\sum_{h=0}^{k-1} \pi_\tau(h)} \sum_{h=0}^{k-1} \pi_\tau(h) P_h^c(a_j) \quad \forall\, a_j \in A_{\text{col}} \tag{2}$$

$$U_k^r(a_i) = \sum_{j=1}^{3} q_k^r(a_j) u_{\text{row}}(a_i, a_j) \quad \forall\, a_j \in A_{\text{row}} \tag{3}$$

$$P_k^r(a_i) = \frac{e^{\lambda U_k^r(a_i)}}{\sum_{a_i' \in A_{\text{row}}} e^{\lambda U_k^r(a_i')}} \quad \forall\, a_j \in A_{\text{row}} \tag{4}$$

and symmetrically define q_k^c, U_k^c, and P_k^c. The object q_k^r in (2) is the perceived distribution over opponent actions by a row player of level k; the expression for $U_k^r(a_i)$ in (3) is the expected row payoff to action a_i when the column player is level k; and the object P_k^r in (4) is the distribution over actions chosen by a row player of level k.

The key difference from PCHM is that players do not choose the payoff-maximizing action with probability 1; instead, they put decreasing (but positive) weight on actions that yield successively lower expected payoffs. The parameter λ controls the concentration of play on the best response: as $\lambda \to \infty$, play converges to probability 1 on the best response (returning the PCHM), and as $\lambda \to 0$, play converges to a uniform distribution over actions. This model, like the PCHM, assume that players correctly forecast the play of all lower types. We suppose that the logit parameter λ is constant across all players.

As before, the predicted distribution over actions is found by aggregating play across a population of players, where the proportion of level k players is $\pi_\tau(k)$. This approach has two free parameters τ and λ, which we estimate on the training data and test out-of-sample.

LPCHM, No Level-0 Players. Below we consider a further modification on LPCHM. Motivated by the suggestion that there are no true level 0 players in the population (see e.g. Crawford, Costa-Gomes and Iriberri (2013)), we consider a variation in which we remove level 0 players from the population, while allowing for them to exist in the perceptions of the other players.

Specifically, we construct the behavior of level-k players as in LPCHM, but fix the proportion of level 0 players in the population to be 0, reweighting the proportion of level $k \geq 1$ players to $\frac{\pi_\tau(k)}{\sum_{h \geq 1} \pi_\tau(h)}$. This approach has two free parameters τ and λ, which we estimate on the training data and test out-of-sample. Note that the "random play" that might be attributed to level 0 players in the baseline PCHM will here be attributed to the error caused by the logit response.[24]

Next, to combine random play with the frequency of level 1 behavior we reported in the previous section, we build on the level 1 model by adding logit best replies. This produces a non-degenerate prediction over actions that can be directly fit to empirical distributions.

[24] Note also that if we modified the baseline PCHM (with exact best responses) in the same way, by supposing that there are no actual level 0 players, then the model would assign probability 1 to the level 1 action in any game in which the level 1 action is unique and is played in a Nash equilibrium. This is a stark prediction that we do not expect to predict well, so we did not try to fit that variant to the data.

Logit Level 1. Let u_i be the expected row payoff to playing action a_i against a uniform distribution over column player actions. Predict action a_i with probability $e^{\lambda u_i} / \sum_{j=1}^{3} e^{\lambda u_j}$, where again the parameter λ controls the degeneracy of the best response. This approach has a single free parameter λ, which we estimate on the training data and test out-of-sample.

5.2 New Approach: Use ML to Choose Free Parameters

Recall that the PCHM has a single free parameter τ. In the standard application of this model, a single value of τ is learned for predicting play in all games. This value turns out to be $\tau = 0.81$ for the lab data and $\tau = 0.33$ for the MTurk data. If we allow for the value of τ to differ across games, however, the values of τ that best fit the observed distributions of play turn out to vary significantly across the games in our data sets (as they did also across the games studied in Camerer, Ho and Chong (2004)). See below for the distribution of best-fit values of τ across the 86 lab games and the 200 random games (where we impose an artificial constraint that $\tau \leq 2$). Additional details on these parameter estimates can be found in Appendix B.2.4.[25]

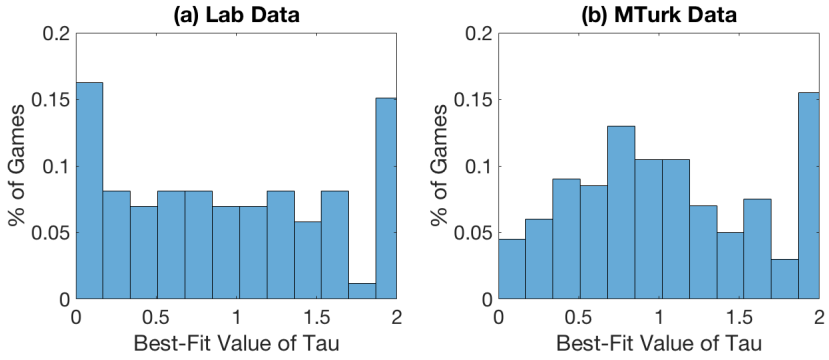


Figure 7: The best-fit value of τ varies substantially across games.

The variation seen in Figure 7 suggests that there are potential gains to prediction by allowing for game-specific values of τ. We need, however, to anticipate these values from properties of the game matrix alone, and in particular cannot base these parameter choices directly on actual data of play.

Our approach for prediction of τ is to posit a set of game features that describe strategic properties of the game matrix, and then use machine learning algorithms to train predictive functions from the set of game features to values of τ. The features that we use, reported in Appendix A.2, describe various properties of the game, such as the number of pure strategy

[25] The higher density of $\tau \approx 0$ estimates in the lab data may reflect the differences between the lab games and the random ones we used on MTurk. In particular, the games in the lab experiments tend to be more strategically complex, which might lead more participants to play in a way that does do not fit the PCHM. When forced to match this kind of game play with a value of τ, the best fit can be the uniform distribution (τ=0).

NE and the number of strictly dominated actions, as well as a count function that tracks the number of actions that satisfy a list of properties. This approach is illustrated in Figure 8.

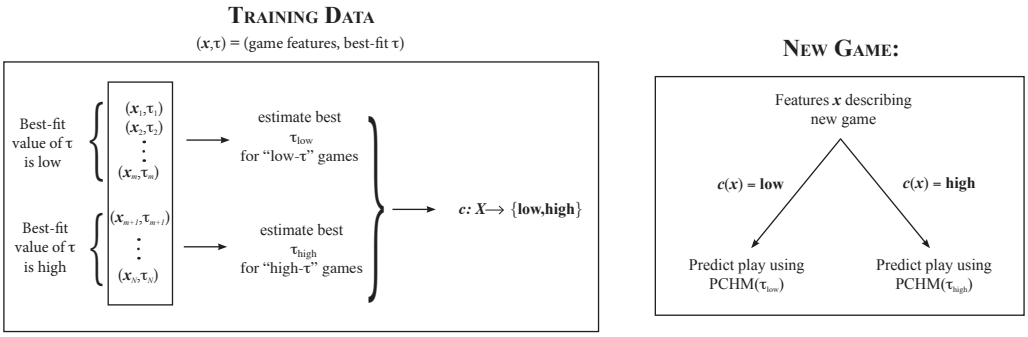


Figure 8: Illustration of our approach for prediction of parameter heterogeneity in PCHM.

For each game in our training data, we first find the value of τ that best fits the observed distribution of play. We then learn a function that takes as input game features, and outputs a classification for the game as either low or high τ. (Somewhat arbitrarily, we define low τ as $\tau < 1$, and high τ as $\tau \geq 1$. In the lab data, 54% of games have best-fit values of τ that are less than 1, and in the set of random games, 46% of games have best-fit values of τ that are less than 1. See Appendix B.2.3 for robustness checks to other split points.)

We pool all training games with a low value of τ, and learn the best single value τ_{low} for predicting play in these "low-τ" games. We similarly pool all training games with a high value of τ and estimate τ_{high} for predicting play in these high-τ games. We then train a decision tree to classify games into low- or high-τ based on their payoff matrices. When presented with a new game, our approach is to first predict whether τ is low or high (using the decision tree), and then to predict play using PCHM with the corresponding estimate of τ (τ_{low} or τ_{high}).

Note that this approach generates predictions for τ using only game features and not observations of play. Since all reported errors are cross-validated, the additional flexibility that comes from permitting heterogeneity does not guarantee an improvement in prediction.

This approach can be extended to incorporate parameter heterogeneity into other models in a similar way. For example, choosing LPCHM as the baseline model, we can estimate a best-fit pair of parameters $(\tau_{low}, \lambda_{low})$ for the set of low τ games and $(\tau_{high}, \lambda_{high})$ for the set of high τ games. Out of sample, we first predict whether τ is low or high, and then predict using the corresponding (τ, λ) pair.

5.3 Results

Below we evaluate the proposed models on our two data sets. All prediction errors are tenfold cross-validated, and we additionally report a completeness measure as before. See Appendix

A.4.2 for the parameter estimates.

We begin by considering the data set of lab play. Table 5 reports the prediction errors of the approaches described in Section 5.1, ordered from least to most predictive.

Method	Prediction Error	Completeness
Naive benchmark	0.0687	0%
Uniform Nash	0.0828	<0%
PCHM	0.0333	51.53%
	(0.0042)	
Logit Level 1	0.0265	61.43%
	(0.0040)	
PCHM with Risk Aversion	0.0259	62.30%
	(0.0028)	
LPCHM	0.0175	74.53%
	(0.0014)	
LPCHM, No Level-0 Players	0.0161	76.56%
	(0.0034)	
Ideal prediction	0	100%

Table 5: Predicting the distribution of play in lab data.

Predicting uniformly at random from Nash actions does not improve upon the naive baseline.[26] In contrast, the PCHM produces a significant improvement, achieving a completeness measure of 50%. Our proposed variations on PCHM do better still, attaining 61-77% of the achievable improvement over guessing at random. The best performance is achieved by LPCHM under the assumption that there are no level 0 players in the population; this approach achieves more than a 20% improvement in our completeness measure over the classic PCHM.

We show next that we can continue to improve upon these performances by predicting heterogeneity in parameter values (see Table 6). We demonstrate these improvements specifically for three models: the PCHM, the LPCHM, and the adaptation of the LPCHM that drops level 0 players. In particular, we find that the Heterogeneous-PCHM improves the PCHM prediction error from 0.0333 to 0.0262 (resulting in an increase of completeness from 52% to 62%). The estimated class-specific values of τ for this model are $\tau_{low} = 0.44$ and $\tau_{high} = 1.44$.

[26] We consider variations on the Nash prediction rule based on selecting the risk-dominant or Pareto-dominant equilibrium in Appendix B.2.1. These more sophisticated methods do not improve on the Uniform Nash prediction.

Method	Prediction Error	Completeness
PCHM	0.0333	51.53%
	(0.0042)	
Heterogeneous-PCHM	0.0262	61.86%
	(0.0019)	
LPCHM	0.0175	74.53%
	(0.0014)	
Heterogeneous-LPCHM	0.0165	75.98%
	(0.0030)	
LPCHM, No Level-0 Players	0.0161	76.56%
	(0.0034)	
Heterogeneous-LPCHM, No Level-0	0.0157	77.15%
	(0.0024)	

Table 6: Predicting heterogeneity in τ improves accuracy in predicting play in lab data.

Our results above extend to the prediction of play in the MTurk games: see Tables 7 and 8. We again find that predicting uniformly at random from Nash actions does worse than guessing at random, while the other models we consider attain significant improvements. We also again find that introduction of heterogeneity in τ yields improvements in prediction, although the sizes of these improvements are smaller than for the lab data.

Method	Prediction Error	Completeness
Naive	0.0838	0%
Uniform Nash	0.1283	<0%
PCHM	0.0186	77.80%
	(0.0038)	
PCHM with Risk Aversion	0.0173	79.36%
	(0.0014)	
LPCHM	0.0153	81.74%
	(0.0018)	
Logit Level 1	0.0134	84.01%
	(0.0008)	
LPCHM, No Level-0 Players	0.0133	84.13%
	(0.0009)	
Ideal prediction	0	100%

Table 7: Predicting the distribution of play in MTurk data.

There are, however, a few minor differences in the outcomes of predicting play in the two data sets. First, all absolute prediction errors are lower for the MTurk data. Additionally,

we see in Table 7 a marked improvement in the performance of the level 1 model with logit best replies.[27] We see two potential reasons for these differences. First, the Mechanical Turk subjects may be less sophisticated than lab subjects, and thus more likely to play the level 1 action. If so, this would improve the predictive accuracy of the level 1 models and result in lower absolute prediction errors (since play is more concentrated). Second, as noted in Section 2.2, the randomly generated games are strategically simpler than the lab games. If the level 1 action is a more compelling choice in these games, this would again improve the predictive accuracy of level 1 models and lower absolute prediction errors (independently of the sophistication of subjects).

In Appendix A.4.1, we try to distinguish by these subject-based and game-based explanations by supplementing our two main data sets with a third, in which we elicit play from MTurk subjects on our set of lab games.[28] We find that prediction errors for this new data set more closely resemble our results for the MTurk data than our results for the lab data; this suggests that the subject-based explanation plays a bigger role in determining the differences observed above between our primary data sets. However, the subject-based explanation is not the complete story, as for example it is still easier to predict the play of MTurk subjects on the MTurk games than the play of MTurk subjects on the lab games.

Method	Prediction Error	Completeness
PCHM	0.0186	77.80%
	(0.0038)	
Heterogeneous-PCHM	0.0159	81.03%
	(0.0006)	
LPCHM	0.0153	81.74%
	(0.0018)	
Heterogeneous-LPCHM	0.0150	82.10%
	(0.0016)	
LPCHM, No Level-0 Players	0.0133	84.13%
	(0.0009)	
Heterogeneous-LPCHM, No Level-0 Players	0.0131	84.37%
	(0.0010)	

Table 8: Predicting heterogeneity in τ improves accuracy in predicting play in MTurk data.

[27] This is consistent with our earlier observation that the modal action is more often level 1 in the random games data (88% vs. 72%).

[28] We told these subjects that their chosen action would be paired with that of a randomly chosen subject from the experiment to determine their payoff.

5.4 Understanding the Classification of Games by τ

Our results above show that the predictions of the PCHM and related models can be improved by using machine learning to guide choice of free parameters, but the decision tree models we used for this goal are complicated and hard to interpret. To better understand the classification of τ, we now study the decision tree model under a two decision node constraint. We focus here on the application of the Heterogeneous-PCHM and the Heterogeneous-LPCHM to the prediction of the lab data.

The best 2-split tree for classification of τ for the PCHM turns out to use two features, both of which track different measures of the "obviousness" of the best action. Recall that in Section 4 we introduced the binary features of whether an action was level 1 or max-max. The most predictive features quantify the "degree" or "robustness" of these properties:

Level-1 payoff gap: For each action a_i, define v_i to be the row player's expected payoff when the column player randomizes uniformly over his actions. The difference between the expected payoff under the level 1 action and the expected payoff under the next best action is then:

$$max_{i\in\{1,2,3\}} v_i - max\left\{ v_j : j \neq \underset{i\in\{1,2,3\}}{\operatorname{argmax}}\, v_i \right\}$$

Max-max payoff gap: For each action $a_i \in A_1$, let $m_i = \max_{a_2 \in A_{\text{col}}} u_{\text{row}}(a_i, a_2)$ be the highest payoff that the row player could receive if he chose action a_i. Define

$$\max_{i\in\{1,2,3\}} m_i - \max\left\{ m_j : j \neq \underset{i\in\{1,2,3\}}{\operatorname{argmax}}\, m_i \right\}$$

to be the difference between the highest payoff that the row player could receive from the "max-max" action, versus from choosing the action that allows for the next highest possible payoff.

These features are combined for classification of τ in the following way:

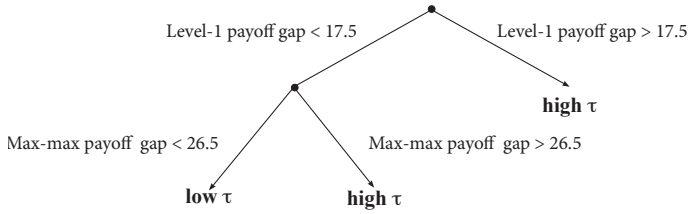


Figure 9: Best 2-split decision tree for classifying τ in PCHM.

This decision tree says that if either the level-1 payoff gap is sufficiently large (at least 17.5), or if the level-1 payoff gap is low but the max-max gap is large (at least 26.5), then we should predict that the game has a high value of τ. Otherwise—that is, if both the level-1 payoff gap

and the max-max payoff gap are low—we should predict that the game has a low value of τ. Intuitively, a large level-1 payoff gap and a large max-max payoff gap both provide evidence of an "obviously best" action. In contrast, if all actions have similar expected payoffs against uniform play, and all allow for similar "best possible" payoffs, then many actions are reasonable and play is less concentrated.

For example, consider the following game:

	a_1	a_2	a_3	Frequency
a_1	25,25	30,60	100,95	22.92%
a_2	60,30	31,31	51,30	41.67%
a_3	95,100	30,51	0,0	35.42%

Table 9: The distribution of play in this game is best fit by $\tau = 0$.

Here action a_1 is both level 1 and max-max, but its expected payoff against uniform play is only slightly larger than the expected payoff of action a_2, and its best payoff is only slightly larger than the best payoff to action a_3. In this way, action a_1 is "barely" level-1 and max-max. Correspondingly, the game has both a small level-1 payoff gap and also a small max-max payoff gap, and the algorithm classifies it (correctly) as low-τ.

We can repeat a similar exercise for classification of τ in LPCHM. Figure 10 shows the best decision tree for classifying τ in this model, under the constraint that only two decision nodes can be used. In addition to the max-max payoff gap feature already described above, this tree uses a feature for the number of actions that are level k for some value of k.

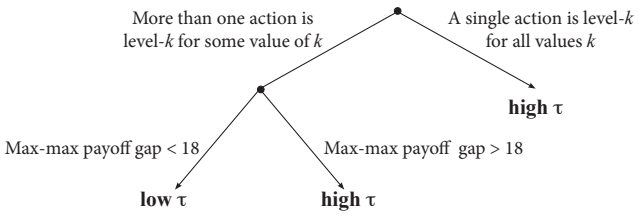


Figure 10: Best 2-split decision tree for classifying τ in LPCHM.

The tree is different from the previous one because τ plays a different role in the PCHM and the LPCHM, and so some games have a low (best-fit) value of τ for one model and a high value for the other. Nevertheless, the classification trees have some similarities. For example, notice that the presence of a single action that is level k for all orders k is again an indication that there is an obviously best action. The classification tree above predicts that τ is high in games with a single action that is level k for all k, and additionally, in games with multiple level k actions and a large max-max payoff gap. In contrast, games with multiple level k actions and a small max-max payoff gap are predicted to have a low value of τ; as before, these are games in

which multiple actions may seem equally compelling.

Because of the restriction to two splits, the decision trees shown above do not perform as well as the classification models used in our main analysis. We can assess the quality of this approximation by directly using the simpler trees above to choose parameters. For each game in our training data, we split games into low- or high-τ categories based on the decision tree above, and learn best-fit values τ_{low} and τ_{high} for games in either class. Out of sample, we first predict whether τ is low or high based on the tree, and then predict the distribution of play using PCHM with the corresponding value of τ. Below, we show that the 2-split models described above achieve approximately half of the improvement of the unconstrained version of Heterogeneous-PCHM and Heterogeneous-LPCHM (over PCHM and LPCHM). We leave open the question of whether there are other simple models for classification of τ that can perform better yet.

Method	Prediction Error	Completeness
PCHM	0.0333	51.53%
	(0.0042)	
Heterogeneous-PCHM, 2-split	0.0303	55.90%
	(0.0031)	
Heterogeneous-PCHM, unconstrained	0.0262	61.86%
	(0.0019)	
LPCHM	0.0175	74.53%
	(0.0014)	
Heterogeneous-LPCHM, 2-split	0.0165	75.98%
	(0.0025)	
Heterogeneous-LPCHM, unconstrained	0.0161	76.56%
	(0.0030)	

Table 10: Simple classification rules for τ improve beyond the baseline model, and achieve a substantial fraction of the improvement of the best classification rules.

5.5 Extension: "Meta-Models"

The approach used above is not special to predicting τ for the PCHM and LPCHM. In addition to helping us choose model parameter values, a generalization of the proposed approach can help us choose between models. This idea is depicted in Figure 11.

Suppose we have two models of play, Model A and Model B. We can first determine which model predicts better for each game in a training set (directly using the observed distribution of play), and cluster games into those better predicted by model A, and those better predicted by model B. We train a classification rule for predicting these groups, and also estimate the

two models on their respective sets of games. For each out-of-sample game, we first use the classification rule to select Model A or Model B, and then use the corresponding estimated model to predict play.

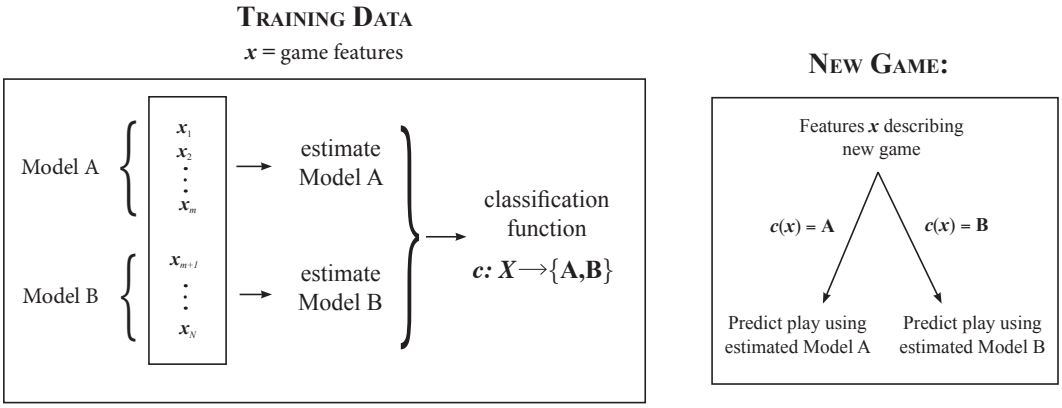


Figure 11: Use machine learning to choose between different models of play

Section 5.2 follows a special case of this in which Models A and B are the same model, estimated separately on two sets of games. Focusing on differences in the parameters of a given model is natural when there is significant variation in the most predictive value of this parameter across different games, as we saw with τ in the PCHM. But we can also choose Model A and Model B to be different models, using the machine learning algorithm to predict (for each game) which model predicts better. As an illustration, we set Model A to be PCHM and Model B to be Logit Level 1, and predict the lab data

Meta-Model: PCHM and Logit Level 1. We determine for each game in the training data the (in-sample) prediction error using PCHM and using Logit Level 1. Since Logit Level 1 has a better base performance, we bias the classification rule towards Logit Level 1. Specifically, we cluster games based on whether the PCHM error is less than half of the Logit Level 1 error. We estimate PCHM on the games for which this is the case, and we estimate Logit Level 1 on the remaining games. We also train a classification algorithm that predicts which of the two groups a new game would fall in. For out of sample games, we first select PCHM or Logit Level 1, and then we predict play using the corresponding (estimated) model.

This approach, which combines PCHM and Logit Level 1, turns out to outperform both models.[29]

[29]We note that the level 1 model with logit best replies does not take into account the column player's payoffs. The difference between the performance of this rule, and the performance of our best model (Heterogeneous LPCHM without level 0 players), provides some insight into the gains from modeling strategic thinking. The

Method	Prediction Error	Completeness
Naive benchmark	0.0687	0%
Uniform Nash	0.0828	<0%
PCHM	0.0333	51.53%
	(0.0042)	
Logit Level 1	0.0265	61.43%
	(0.0040)	
Meta-Model: PCHM and Logit Level 1	0.0231	66.38%
	(0.0056)	
Ideal prediction	0	100%

Table 11: The "meta-model" combining PCHM and Logit Level 1 outperforms both models.

Comparing these results with our previous methods, we see that this meta-model improves on PCHM and Heterogeneous-PCHM, but not on the variations of LPCHM. Since PCHM and Logit Level 1 were not our most predictive models, this is not surprising; we used them here because of their simplicity. We view the performance of our meta-model as a proof of concept, and leave the exploration of other combinations of models for future work.

6 Identifying New Feature Sets: Crowd Predictions

Our results so far have illustrated the potential for feature-based prediction rules to modify existing theories to make better predictions. Ultimately, these feature-based approaches are only as powerful as the features that we use. An interesting question for subsequent work is then what additional features might give useful insights into predicting play.

We conclude with an extended discussion of one very different approach for feature construction that does not explicitly use the payoff matrix. These features are instead based on human inputs, specifically the predictions of play by untrained human subjects.

6.1 Crowd Prediction Data

We asked human subjects on Mechanical Turk to predict play in 15 games from the 286 games described in Section 2. To the best of our knowledge, these subjects are untrained: the initial part of our experiment consisted of an introduction to matrix games, and we allowed subjects to proceed to the main experiment only after correctly answering a set of comprehension questions.[30]

difference in prediction errors is not large: 0.0265 for level 1 logit (61% completeness) and 0.0157 for Heterogeneous LPCHM without level 0 players (77% completeness). This suggests that even for the task of predicting distributions, a substantial component of the regularities in play can be described using level 1 thinking alone.

[30]The comprehension questions consisted of reporting the payoffs for a fixed action pair in two example games (see Appendix B.3). All subjects eventually answered both comprehension questions correctly.

In the main part of our experiment, each subject was shown a random subset of fifteen of the lab games (Section 2.1) or fifteen of the random games (Section 2.2). We informed subjects that these games had been played by real people, and asked them to predict the action that was *most likely to be chosen* by the row player. (We chose not to ask subjects to report probability distributions because we were skeptical about how accurate those predictions would be, and unlike in e.g. Nyarko and Schotter (2002) and Costa-Gomes and Weizsacker (2007), our focus was not on the extent to which play is a best response to stated beliefs. Instead we will study the usefulness of the aggregate distribution of a number of simpler reports.) To incentivize effort, we told subjects that on top of their base payment of $0.25, they would receive an additional $0.10 for every question they answered correctly. Figure 12 shows a typical question prompt presented to subjects, and the complete set of instructions can be found in Appendix B.3.

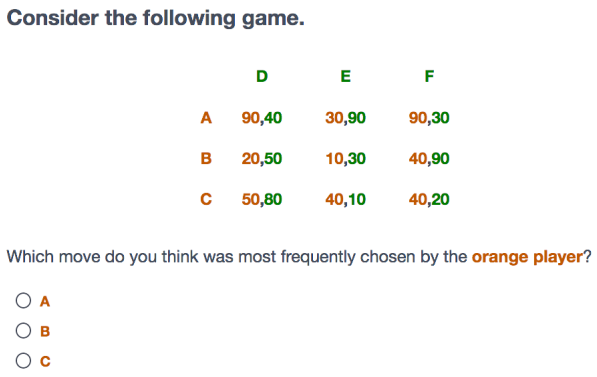


Figure 12: A typical question prompt presented to Mechanical Turk subjects in the single action treatment. The "orange player" is the row player.

A total of 250 subjects participated in the lab game prediction experiment, and 540 subjects participated in the random game prediction experiment. On average, approximately 40 crowd predictions were observed for each game.

6.2 Predictions and Results

We consider a simple and direct use of these crowd predictions: For every game g_i, let x_k^i be the fraction of subjects who predicted action a_k in game g_i. In the first prediction task, we predict the modal crowd prediction action $\text{argmax}_{k\in\{1,2,3\}} x_k^i$, and in the second, we predict the distribution (x_1^i, x_2^i, x_3^i). These naive crowd rules use only the perception of payoffs by (untrained) participants. Nevertheless, we find that they perform extremely well in both prediction problems and for both data sets; see Tables 12 and 13 below.[31] Specifically, the naive crowd

[31] Below, standard errors for the crowd prediction rule are bootstrapped standard errors with 100 resamples.

rule improves upon the PCHM in three of the four prediction problems,[32] and outperforms our most predictive model (LPCHM without level 0 players) in the MTurk data. The naive crowd rule does not, however, improve upon our best model-based approaches for predicting the lab data.[33]

	Lab Data		MTurk Data	
	Error	Completeness	Error	Completeness
Naive benchmark	0.6667	0%	0.6667	0%
Uniform Nash	0.5507	33.66%	0.4722	51.21%
PCHM	0.3838	82.02%	0.3159	92.36%
	(0.0197)		(0.0217)	
Crowd	0.3965	78.34%	0.3091	94.15%
	(0.0056)		(0.0067)	
Ideal prediction	0.3218	100%	0.2869	100%

Table 12: Crowd prediction of the action played

	Lab Data		MTurk Data	
	Error	Completeness	Error	Completeness
Naive benchmark	0.0687	0%	0.0838	0%
Uniform Nash	0.0828	<0%	0.1283	<0%
PCHM	0.0333	51.53%	0.0186	77.80%
	(0.0042)		(0.0038)	
LPCHM, No Level-0 Players	0.0161	76.56%	0.0133	84.13%
	(0.0034)		(0.0009)	
Crowd	0.0285	58.52%	0.0091	89.14%
	(0.0033)		(0.0008)	
Ideal prediction	0	100%	0	100%

Table 13: Crowd prediction of the distribution of play

[32] The naive crowd rule performs slightly worse than the PCHM in the problem of predicting the realized action in lab data, but its completeness measure is comparable.

[33] As another perspective on these comparisons, we report a measure based on *equivalent number of observations* (ENO) from Erev et al. (2007): we ask how many crowd predictions are needed to make as accurate a prediction as PCHM. Specifically, for each game, we sample n crowd predictions at random (without replacement), and construct a naive crowd prediction rule based on this re-sampled data. We find that for the lab data, PCHM's performance is equivalent to 23 crowd samples in the action prediction task, and 20 crowd samples in the distribution prediction task. For the MTurk data, PCHM's performance is equivalent to 16 crowd samples in the action prediction task, and 11 crowd samples for the distribution prediction task. This comparison again reveals the crowd prediction rule to be more effective relative to PCHM for our MTurk data set than for the lab data.

6.3 Do Subjects Predict Their Own Play?

A potential explanation for the performance of the naive crowd rule predictions is that subjects simply predict the actions that they themselves would choose. This hypothesis would imply that each prediction is equivalent to an observation of play, so that with sufficiently many predictions, the distribution of crowd predictions would approximate the distribution of play arbitrarily well.

To evaluate this hypothesis, we compare the distributions of play with the distributions of crowd predictions. Specifically, we test the null hypothesis that our samples of game play and samples of crowd predictions are drawn from the same distribution, using a Kolmogorov-Smirnov test to determine a p-value for each game. Under the hypothesis that crowd predictions and game play are indeed drawn from the same distribution, these p-values follow a uniform distribution. We find instead that for both the lab games and the random games, the observed distribution of p-values is statistically different from the uniform distribution (see Figure 13 below).[34]

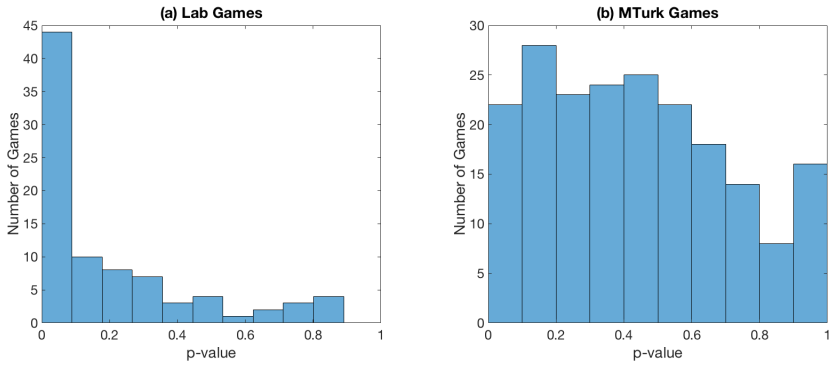


Figure 13: We reject the hypothesis that crowd predictions and game play are drawn from the same distributions.

Thus, the crowd systematically predicts some games better than others. The departure from uniform is especially large for the lab data, although the takeaway from this is not clear since the populations generating the predictions and game play are different: predictions are made by MTurk subjects, while game play is chosen by lab subjects. We therefore return to a supplementary data set mentioned previously in Section 5.3 (and described in more detail in Appendix A.4.1), in which Mechanical Turk subjects were asked to play the lab games. We repeat the analysis above, this time comparing the observed distribution of play by *MTurk subjects* in the lab games with the crowd predictions. The resulting distribution of p-values (shown below) is less different from uniform than the left panel of Figure 13, but still more so

[34] We reject that the distribution of p-values for the lab games is uniform with $p \approx 10^{-17}$ under a Kolmogorov-Smirnov test, and reject that the distribution of p-values for the random games is uniform with $p = 0.0027$. Our finding is similar in spirit to that of Costa-Gomes and Weizsacker (2007), who find (for a set of 14 lab games) that stated beliefs are closer to the uniform distribution than the actual distribution of play is.

than the right panel of Figure 13. Thus, rejection of our null hypothesis for the lab games was not entirely driven by differences in the subject pools.

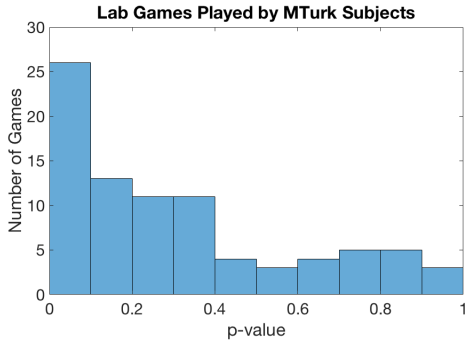


Figure 14: The distribution of p-values remains statistically different from uniform when we compare crowd predictions with MTurk play of lab games.

These results suggest that the use of human inputs can lead to improved predictions. We leave open the questions of what kind of games are most amenable to crowd prediction, and how human inputs might be more usefully leveraged than the naive aggregation rule considered here.

7 Conclusion

In studying initial play, as in other settings, there is a tradeoff between the predictive accuracy of a model and its interpretability. Our focus is not on predictive accuracy alone, and as we have shown in this paper, machine learning can be used not only to improve predictions of play, but also to improve our understanding of it, and to develop simple and portable improvements on existing models.

One way we do this is by studying games in which machine learning algorithms predict well, but existing models do not. This exercise helped us to realize that adding a risk aversion parameter to the level 1 model generates better out-of-sample predictions. As a second approach, we used machine learning methods to choose parameters for existing models. Specifically, we use game features to predict the best value of τ in (variations of) the Poisson Cognitive Hierarchy model. This approach emphasizes that the best model of play may differ systematically depending on the strategic structure of the game, and shows that machine learning methods may help us to identify the forces behind this heterogeneity. An extension of this approach allows us to classify games based on which of two models (PCHM and a variation of level 1) is better suited for predicting play. We plan to explore combinations of more diverse models, and the games to which they are most suited, in future work.

Along with papers such as Leyton-Brown and Wright (2014), these results suggest potential for interplay between machine learning methods and theory models to improve prediction and

understanding of play in games. Beyond our present setting of predicting play, we expect also that the proposed approaches can be used to improve the out-of-sample performance of other economic models.

Finally, we note that although many situations are intermediate between the "pure initial play" case we study here and the long-run outcomes studied in models of learning in games (Fudenberg and Levine, 1998), the distribution of initial play in a game can have a major role in determining the evolution of subsequent play. Thus, we expect that better modeling of initial play can improve prediction of medium and long run behaviors, leaving this direction for subsequent work.

A Appendix

A.1 Supplementary Material to Section 2.2

	Lab Games	Random Games
Dominance-solvable	0.1395	0.22
≥ 1 strictly dominated action	0.31	0.48
"Best for both" profile	0.1279	0.275
Variance of payoffs	901.7684	652.67
Max payoff	95.4070	88.3000
Min payoff	3.1977	11.8000
Max sum of payoffs	151.1512	153.1000
Min sum of payoffs	22.7209	46
Correlation between players' payoffs	0.074	0.029
Observations	**86**	**200**

Table 14: Comparison of summary statistics for the lab games and random games.

Table 14 compares the following summary statistics for the two sets of games:

– whether there exists an action profile a that is *best for both players*:

$$a \in \underset{a' \in A}{\operatorname{argmax}}\, u_{\text{row}}(a') \quad \text{and} \quad a \in \underset{a' \in A}{\operatorname{argmax}}\, u_{\text{col}}(a')$$

– the *number of games with an action that is (pure-strategy) strictly dominated*

– the *number of games that are (pure strategy) dominance-solvable*

– the *variance* of the payoffs $\frac{1}{18}\sum_{i=1}^{18}(g_i - \bar{g})^2$

– the *maximum individual payoff* $\max_i g_i$

– the *minimum individual payoff* $\min_i g_i$

– the *maximum total payoff* $\max_{a \in A} u_{\text{row}}(a) + u_{\text{col}}(a)$

– the *minimum total payoff* $\min_{a \in A} u_{\text{row}}(a) + u_{\text{col}}(a)$

– the *correlation between player payoffs.*

A.2 List of Features

Describing Specific Actions For each action a_i, we include an indicator variable for whether that action:

– is part of a *pure-strategy Nash equilibrium*

– is part of an action profile that *maximizes the sum of player payoffs*[35]

– is part of a *Pareto dominant Nash equilibrium*

– is *"max-max"*: $(a_i, a_2) \in \operatorname{argmax}_{a \in A} u_{\text{row}}(a)$ for some $a_2 \in A_{\text{col}}$.

[35] There exists $a_2 \in A_{\text{col}}$ such that $u_{\text{row}}(a_1, a_2) + u_{\text{col}}(a_1, a_2) = \max_{a \in A}(u_{\text{row}}(a) + u_{\text{col}}(a))$.

– is *level* k for each $k = 1, 2, \ldots, 7$

Additionally, we include a *score* feature for each action, which is the number of above properties that it satisfies.

Describing Properties of the Game Recall that $g \in \mathbb{R}^{18}$ describes the payoff matrix. We include the following properties of the payoff matrix:

– *number of pure strategy Nash equilibria*
– *number of actions that are strictly dominated by a pure strategy*
– *number of level 1 actions*
– *number of actions that are "max-max."*
– *number of actions that maximize total payoffs..*
– *number of actions that are both level 1 and "max-max."*
– *number of actions that are both level 1 and maximize total payoffs.*
– *number of actions that are both "max-max" and maximize total payoffs.*
– *number of actions that are level 1, "max-max," and maximize total payoffs.*

We additionally include various measures describing existence of an "obviously best" action. For each action a_i, let o_i be the number of properties it satisfies from the following list: level 1, best-for-both, and max-max. Additionally let $y_i = max_{a_2 \in A_{\text{col}}} u_{\text{row}}(a_i, a_2) + u_{\text{col}}(a_i, a_2)$ be the largest possible total payoff when the row player chooses a_i. We include as features:

– $max_k o_k - max\{o_l : l \neq argmax_k o_k\}$
– $max_{k \in \{1,2,3\}} o_k$
– $\max_{i \in \{1,2,3\}} y_i - \max\{y_j : j \neq argmax_{i \in \{1,2,3\}} y_i\}$
– the *level-1 payoff gap*, as defined in Section 5.4.
– the max-max payoff *gap*, as defined in Section 5.4.

A.3 Alternative Benchmarks

The idealized prediction errors that we use as a benchmark in the main text are an underestimate of the actual best possible prediction errors, since they do not involve out-of-sample testing. Below, we report completeness measures with respect to two alternative benchmarks. Our findings from the main text all extend to these measures; the main difference is that all of the completeness measures are higher.

A.3.1 Table Lookup Benchmark

We divide the full data set into five folds, using four folds for training and the fifth for testing. Note that unlike in our main cross-validation procedure (see Section 3), here the training and test sets contain observations of play in the same games. From the training set, we learn the modal action for each game, and use that as our prediction of play in the test set for the first prediction task. We use the empirical distribution of play in each game in the training data as our prediction of play for the second prediction task. This "table lookup" prediction

algorithm was used previously in Kleinberg, Liang and Mullainathan (2017) to establish an idealized benchmark against which to evaluate completeness.

The table lookup algorithm is a consistent estimator for the best possible out-of-sample error in both of our problems, and for this reason it is a natural benchmark. However, table lookup performs well only if we have a sufficient number of observations of play for each game, so we use this approach for our lab data set only. Table 15 reproduces Table 2 from the main text, replacing the idealized prediction error with the table lookup prediction error, and reporting new completeness measures.

	Error	Completeness
Naive benchmark (guess at random)	0.6667	0%
Best constant prediction (always guess a_1)	0.6013	19.22%
	(0.0797)	
Uniform Nash	0.5507	34.09%
	(0.0055)	
Level 1/PCHM	0.3889	81.63%
	(0.0079)	
Prediction based on game features	0.3652	88.60%
	(0.0057)	
Table Lookup	0.3264	100%
	(0.0040)	

Table 15: Predicting the realized action: completeness relative to a "table lookup" benchmark

Table 16 reproduces Table 5 from the main text, replacing the idealized prediction error with the table lookup prediction error, and reporting new completeness measures.

Method	Prediction Error	Completeness
Naive benchmark	0.0687	0%
Uniform Nash	0.0828	<0%
PCHM	0.0333	66.54%
	(0.0042)	
Logit Level 1	0.0265	79.32%
	(0.0040)	
PCHM with Risk Aversion	0.0259	80.45%
	(0.0028)	
LPCHM	0.0175	96.24%
	(0.0014)	
LPCHM, No Level-0 Players	0.0161	98.87%
	(0.0034)	
Table Lookup	0.0155	100%
	(0.0008)	

Table 16: Predicting the distribution of play: completeness relative to a "table lookup" benchmark

In both tables we see increases in all completeness measures; this change is especially pronounced in Table 16, where we now find our model-based prediction algorithms to achieve 99% of the reduction attained by our benchmark. This measure of completeness is likely an overestimate, just as our measure of completeness in the main text is an underestimate.

A.3.2 Bootstrap Benchmark

To construct our "bootstrap" benchmark, we learn for each game its empirical frequency in our (full) data set. We then re-sample 1000 data sets of equal size from the original data, and predicting play in these new data sets. We report the average prediction error across these data sets. Table 15 reproduces Table 2 and Table 4 from the main text (predicting the realized action); Table 18 reproduces Table 5 and 7 from the main text (predicting the distribution of play). Throughout, the idealized prediction error is replaced with this "resamples" or "bootstrap" benchmark, and new completeness measures are reported.

	Lab Data		MTurk Data	
	Error	Completeness	Error	Completeness
Naive benchmark	0.6667	0%	0.6667	0%
Best constant prediction	0.6013	18.79%	0.6482	5.04%
	(0.0797)		(0.0137)	
Uniform Nash	0.5507	33.32%	0.4722	53.01%
	(0.0055)		(0.0075)	
Level 1/PCHM	0.3889	79.80%	0.3323	91.14%
	(0.0079)		(0.0065)	
Prediction based on game features	0.3652	86.61%	0.3430	88.23%
	(0.0057)		(0.0050)	
Bootstrap Benchmark	0.3186	100%	0.3211	100%
	(0.0052)		(0.0046)	

Table 17: Predicting the realized action: completeness relative to a "bootstrap" benchmark

	Lab Data		MTurk Data	
Method	Error	Completeness	Error	Completeness
Naive benchmark	0.0687	0%	0.0838	0%
Uniform Nash	0.0828	<0%	0.1283	<0%
PCHM	0.0333	57.65%	0.0186	88.95%
	(0.0042)		(0.0038)	
Logit Level 1	0.0265	68.73%	0.0173	90.72%
	(0.0040)		(0.0014)	
PCHM with Risk Aversion	0.0259	69.71%	0.0153	93.45%
	(0.0028)		(0.0018)	
LPCHM	0.0175	83.39%	0.0134	96.04%
	(0.0014)		(0.0008)	
LPCHM, No Level-0 Players	0.0161	85.67%	0.0133	96.18%
	(0.0034)		(0.0009)	
Bootstrap Benchmark	0.0073	100%	0.0105	100%
	(0.0011)		(0.0010)	

Table 18: Predicting the distribution of play: completeness relative to a "bootstrap" benchmark

Note that the reason the ideal benchmarks are lower for the lab data set than the MTurk data set is that we have more observations per lab game. The lower benchmarks do not imply that the play in the lab games is easier to predict.

A.4 Supplementary Material to Section 5

A.4.1 Prediction of Mechanical Turk Play on Lab Games

In a new experiment, we asked 256 MTurk subjects to play as the row player in 15 (randomly selected) games from our lab game set. Subjects were told that their choices would be matched with those of "other subjects," but we were not explicit about who those subjects were. On top of a base payment, participants received a payoff bonus, depending on the actions they chose in the game.[36] There are in total 45 observations of play for each game.

We find that play in this new data set qualitatively resembles that of our original MTurk data set in the two ways described previously: level 1 models achieve (or improve upon) the performance of PCHM variations, and all absolute prediction errors are low (relative to prediction of the lab data set). This suggests that the strong performance of level 1 models, and also the lower absolute errors in our original MTurk data (relative to the lab data), are substantially driven by differences in subject populations. However, the absolute prediction errors are still lower for predicting play of random games by MTurk subjects than for predicting play of lab games by MTurk subjects (e.g. compare an error of 0.0134 achieved by Level 1 Logit for the MTurk data with an error of 0.0147 in Appendix A.4.1). This suggests that the subject-based explanation is not the entire story; part of the difference that we see is driven by variation in games.

Method	Prediction Error	Completeness
Naive	0.0382	0%
Uniform Nash	0.0642	<0%
PCHM	0.0242	36.65%
	(0.0023)	
PCHM with Risk Aversion	0.0202	47.12%
	(0.0017)	
PCHM with Logit BR	0.0153	59.95%
	(0.0048)	
PCHM with Logit BR, No Level 0 Players	0.0149	60.99%
	(0.0015)	
Logit Level 1	0.0147	61.52%
	(0.0015)	
Ideal prediction	0	100%

Table 19: Prediction of play of lab games by Mechanical Turk subjects

[36] We matched subjects' choices with the modal actions chosen by the lab subjects, and paid them according to the corresponding payoff profile.

A.4.2 Parameter Estimates (Pooling All Games)

The table below reports estimated parameters for each of the approaches considered in the main text. We report separately the parameter estimates for each of our data sets of play: lab games played by lab subjects (as introduced in Section 2.1), lab games played by Mechanical Turk subjects (as introduced above in Section A.4.1), and random games played by Mechanical Turk subjects (as introduced in Section 2.2). Parameter estimates are averaged across the multiple iterations of training.

	Lab Subjects	MTurk Subjects	
PCHM	$\tau = 0.81$	$\tau = 0.33$	$\tau = 0.94$
LPCHM	$\tau = 1.54$	$\tau = 1$	$\tau = 1.25$
	$\lambda = 0.17$	$\lambda = 0.11$	$\lambda = 0.17$
Risk-PCHM	$\tau = 0.75$	$\tau = 0.33$	$\tau = 0.90$
	$\alpha = 0.67$	$\alpha = 0.22$	$\alpha = 0.67$
LPCHM, No Level 0 Players	$\tau = 1.46$	$\tau = 0.35$	$\tau = 0.44$
	$\lambda = 0.14$	$\lambda = 0.05$	$\lambda = 0.09$
Logit Level 1	$\lambda = 0.02$	$\lambda = 0.02$	$\lambda = 0.03$
Logit-Risk Level 1	$\lambda = 0.08$	$\lambda = 0.09$	$\lambda = 0.08$
	$\alpha = 0.71$	$\alpha = 0.62$	$\alpha = 0.81$

Table 20: Parameter Estimates

When estimating Heterogeneous-PCHM on the lab data set, we find that $\tau_{low} = 0.44$ and $\tau_{high} = 1.44$. The same model, estimated using the MTurk dataset, yields $\tau_{low} = 0.67$ and $\tau_{high} = 1.55$. When estimating Heterogeneous-LPCHM on the lab data set, we find $(\tau_{low}, \lambda_{low}) = (0.83, 0.16)$ and $(\tau_{high}, \lambda_{high}) = (1.67, 0.21)$. The same model, estimated using the Mechanical Turk dataset, yields $(\tau_{low}, \lambda_{low}) = (0.67, 0.37)$ and $(\tau_{high}, \lambda_{high}) = (1.82, 0.11)$

References

Camerer, Colin, and Teck-Hua Ho. 1999. "Experienced-Weighted Attraction Learning in Normal Form Games." *Econometrica*. 1.1

Camerer, Colin, Gideon Nave, and Alec Smith. 2017. "Dynamic unstructured bargaining with private information: theory, experiment, and outcome prediction via machine learning." Working Paper. 5

Camerer, Colin, Teck-Hua Ho, and Juin-Kuan Chong. 2004. "A Cognitive Hierarchy Model of Games." *The Quarterly Journal of Economics*. 1, 2, 4.1, 4.1, 5.2, B.2.4, 28

Cheung, Yin-Wong, and Daniel Friedman. 1997. "Individual Learning in Normal Form Games: Some Laboratory Results." *Games and Economic Behavior*. 1.1

Chong, Juin-Kuan, Teck-Hua Ho, and Colin Camerer. 2016. "A Generalized Cognitive Hierarchy Model of Games." *Games and Economic Behavior*. 1.1

Costa-Gomes, Miguel, and Georg Weizsacker. 2007. "Stated Beliefs and Play in Normal-Form Games." *Review of Economic Studies*. 1.1, 6.1, 34

Costa-Gomes, M., V. Crawford, and B. Broseta. 2001. "Cognition and behavior in normal-form games: an experimental study." *Econometrica*. 4.1

Crawford, Vincent. 1995. "Adaptive Dynamics in Coordination Games." *Econometrica*. 1.1

Crawford, Vincent, Miguel Costa-Gomes, and Nagore Iriberri. 2013. "Structural Models of Nonequilibrium Strategic Thinking: Theory, Evidence, and Applications." *Journal of Economic Literature*. 1, 1.1, 5.1

DellaVigna, Stefano, and Devin Pope. 2017. "Predicting Experimental Results: Who Knows What?" *Journal of Political Economy*. 1.1

Erev, Ido, Alvin Roth, Robert Slonim, and Greg Barron. 2007. "Learning and equilibrium as useful approximations: Accuracy of prediction on randomly selected constant sum games." *Economic Theory*, 33(29-51). 1, 33

Erev, Ido, and Alvin Roth. 1999. "Predicting How People Play Games: Reinforcement Learning in Experimental Games with Unique, Mixed Strategy Equilibria." *American Economic Review*. 1.1

Ert, Eyal, Ido Erev, and Alvin Roth. 2011. "A Choice Prediction Competition for Social Preferences in Simple Extensive Form Games: An Introduction." *Games*. 1, 1.1

Fragiadakis, Daniel E., Daniel T. Knoepfle, and Muriel Niederle. 2016. "Who is Strategic?" Working Paper. 1.1

Fudenberg, Drew, and David Levine. 1998. *The Theory of Learning in Games.* MIT Press. 7

Hartford, Jason, James Wright, and Kevin Leyton-Brown. 2016. "Deep Learning for Predicting Human Strategic Behavior." 1.1

Haruvy, E., and D. Stahl. 2007. "Equilibrium selection and bounded rationality in symmetric normal-form games." *Journal of Economic Behavior and Organization*. 2.1

Haruvy, E., D. Stahl, and P. Wilson. 2001. "Modeling and testing for heterogeneity in observed strategic behavior." *Review of Economic and Statistics*. 2.1

Hastie, Trevor, Robert Tibshirani, and Jerome Friedman. 2009. *The Elements of Statistical Learning.* Springer. 3

Kleinberg, Jon, Annie Liang, and Sendhil Mullainathan. 2017. "The Theory is Predictive, but is it Complete? An Application to Human Perception of Randomness." Working Paper. 1, A.3.1

Leyton-Brown, Kevin, and James Wright. 2014. "Level-0 Meta-Models for Predicting Human Behavior in Games." *ACM Conference on Economics and Computation (ACM-EC).* 1, 1.1, 2.1, 3, 13, 4.1, 5.1, 7, B.1.4

Morris, Stephen, Rafael Rob, and Hyun Song Shin. 1995. "p-Dominance and Belief Potential." *Econometrica.* 37

Nyarko, Yaw, and Andrew Schotter. 2002. "An Experimental Study of Belief Learning Using Elicited Beliefs." *Econometrica.* 6.1

Peysakhovich, Alex, and Jeff Naecker. 2017. "Using Methods from Machine Learning to Evaluate Models of Human Choice Under Uncertainty." Forthcoming. 1

Rogers, B.W., T.R. Palfrey, and C.F. Camerer. 2009. "Heterogeneous quantal response equilibrium and cognitive hierarchies." *Journal of Economic Theory.* 2.1

Sgroi, Daniel, and Daniel John Zizzo. 2009. "Learning to play 3x3 games: Neural networks as bounded-rational players." *Journal of Economic Behavior and Organization.* 1.1

Stahl, Dale O., and Paul W. Wilson. 1995. "On players' models of other players: Theory and experimental evidence." *Games and Economic Behavior.* 1, 2.1, 4.1, 4.1, 5.1, B.2.4, 28

Stahl, D., and E. Haruvy. 2008. "Level-n bounded rationality and dominated strategies in normal-form games." *Journal of Economic Behavior and Organization.* 2.1

Stahl, D., and P. Wilson. 1994. "Experimental evidence on players' models of other players." *Journal of Economic Behavior and Organization.* 1, 2.1, 2.1, 4.1, 5.1

B Online Appendix

B.1 Supplementary Material to Section 4

B.1.1 Robustness Check

As a robustness check to Section 4, we consider a related exercise in which each observation is a pair (g_i, a_i), where g_i is one of the 86 games in the lab data set, and a_i is the modal action chosen in that game (so that there are 86 observations in total). Given a data set of n pairs $\{(g_i, a_i)\}_{i=1}^{n}$, we measure the error of a prediction rule $f : G \to A_{\text{row}}$ as before, using the misclasification rate.

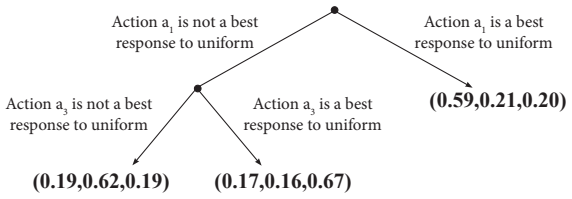
	Error	Completeness
Naive benchmark (guess at random)	0.6667	0%
Best constant prediction (always guess a_1)	0.5569	16.47%
	(0.0582)	
Uniform Nash	0.4302	35.47%
	(0.0596)	
Level 1	0.3256	51.16%
	(0.0471)	
Prediction based on game features	0.2326	65.11%
	(0.0449)	
Ideal prediction	0	100%

Table 21: Predicting the realized action in lab data

The best 2-split and 3-split decision trees are unchanged from Figures 4 and 5, and the 3-split decision tree again minimizes out-of-sample prediction error.

B.1.2 The Best 2-Split Decision Tree for Predicting Distribution of Play Also Uses Level 1 Features

Below, we show the best 2-split decision tree for the task of predicting the distribution of play. This tree closely resembles the one shown in Figure 5. In particular, the most predictive two features are again the level 1 features.



When action a_1 is level 1, the tree predicts a distribution that places majority weight on a_1, and similar for actions a_2 and a_3.

B.1.3 Other Prediction Algorithms

Here we show that the accuracy of predictions of the realized action in lab games is not significantly improved by using more sophisticated algorithms on the same feature set. We first consider a *random forest* algorithm, which grows decision trees using bootstrapped samples of the data, predicting based on a majority vote across the ensemble of trees—and find no improvement over the simpler decision tree model.

	Error	Completeness
Naive benchmark (guess at random)	0.6667	0%
Decision Tree	0.3652	87.42%
	(0.0057)	
Random Forest	0.3933	79.27%
	(0.0209)	
Ideal prediction	0.3218	100%

Table 22: Predicting the realized action in lab data

For the related prediction task described above in Appendix B.1.1, we also compare the decision tree model against a *2-layer neural net*—which feeds features (inputs) through a "layer" of nonlinear transformations, producing outputs that can be fed into the next layer—and *lasso logistic regression*—multinomial logisitic regression with a lasso regularization penalty on the coefficients. Again we find no substantial improvements in prediction accuracy.

	Error	Completeness
Naive benchmark (guess at random)	0.6667	0%
Decision Tree	0.2326	65.11%
	(0.0449)	
Random Forest	0.2321	65.19%
	(0.0255)	
2-Layer Neural Net	0.2564	61.54%
	(0.1043)	
Lasso Logistic Regression	0.2086	68.71%
	(0.0888)	
Ideal prediction	0	100%

Table 23: Predicting the realized action in lab data

B.1.4 Cross-Validation at the Observation Level

In the main text, we reported tenfold cross-validated errors, where the method of cross-validation was to divide the games into ten folds, use all observations of play associated with games in nine of the folds for training, and use the observations of play associated with games in the remaining fold for testing. We consider below an alternative approach to cross-validation, where we pool all of the observations of play and randomly split this pooled data into folds. This is the approach used in Leyton-Brown and Wright (2014).

Table 24 presents misclassification rates for both the lab data and the MTurk data.

	Lab Data		MTurk Data	
	Error	Completeness	Error	Completeness
Naive benchmark	0.6667	0%	0.6667	0%
Uniform Nash	0.5507	33.66%	0.4722	51.21%
	(0.0055)		(0.0075)	
Level 1/PCHM	0.3838	82.02%	0.3159	92.36%
	(0.0197)		(0.0217)	
Prediction rule based on game features	0.3360	95.88%	0.2984	96.97%
	(0.0056)		(0.0095)	
Ideal prediction	0.3218	100%	0.2869	100%

Table 24: Predicting the realized action using "observation-level" cross-validation

B.2 Supplementary Material to Section 5

B.2.1 Alternative Nash Predictions

In the main text, we consider uniform prediction over all actions that are part of a Nash equilibrium. This is not the only possible prediction model based on Nash equilibrium. For example, we can use stricter standards, such as predicting only actions that are part of a Pareto-dominant or a risk-dominant Nash equilibrium.[37] We consider below the following approaches:

1. **Predict actions that are part of a Pareto-dominant and risk-dominant Nash equilibrium:** here we predict an action with probability 1 if it is part of a Pareto-dominant and risk-dominant Nash equilibrium, and otherwise predict the uniform distribution.

2. **Predict actions that are part of a Pareto-dominant Nash equilibrium:** here we predict an action with probability 1 if it is part of a Pareto-dominant Nash equilibrium, and otherwise predict the uniform distribution.

[37] For our setting of 3x3 games, we consider specifically (2/3)-dominance (Morris, Rob and Shin, 1995): (a_1, a_2) is a (2/3)-dominant Nash equilibrium if a_1 is a best response when the column player chooses a_2 with probability at least 2/3, and vice versa.

3. **Predict actions that are part of a risk-dominant Nash equilibrium:** here we predict an action with probability 1 if it is part of a risk-dominant Nash equilibrium, and otherwise predict the uniform distribution.

Method	Lab Data	MTurk Data
(1)	0.0906	0.0813
(2)	0.1641	0.1238
(3)	0.0906	0.0845

Table 25: Alternative prediction models based on Nash equilibrium

These errors are comparable to those of the uniform Nash prediction rule, and in particular are substantially worse than the PCHM.

B.2.2 Predicting the Distribution of Play Using Game-Based Prediction Rules

Below we consider predictive models based on the set of game-based features described in Appendix A.2, built using Lasso regression and a decision tree model. Both algorithms are predictive, but do not improve on the PCHM:

Method	Prediction Error	Completeness	Prediction Error	Completeness
Naive benchmark	0.0687	0%	0.0838	0%
Uniform Nash	0.0828	<0%	0.1283	<0%
Decision Tree	0.0426	37.99%	0.0232	72.32%
	(0.0205)		(0.0074)	
LASSO regression	0.0367	46.58%	0.0179	78.64%
	(0.0020)		(0.0049)	
PCHM	0.0333	51.53%	0.0173	79.36%
	(0.0042)		(0.0014)	
Ideal prediction	0	100%	0	100%

Table 26: Predictive rules built on game features are predictive, but do not outperform PCHM

B.2.3 Other Split Points

We consider different ways of classifying games: specifically, defining low-τ and high-τ with alternative split points, and allowing for three categories (low-, medium-, and high-τ). The two alternative split points we consider are the median value of the best-fit τ (in the full lab data set), and the mean value of the best-fit τ (again in the full lab data set). For determining three classes, we use the 33rd and 66th percentile as split points. The results shown below are similar to the main text.

Method	Prediction Error	Parameter Estimates
Heterogeneous-PCHM	0.0230	$\tau_{low} = 0.3333$
split at $\tau = 0.8889$ (median)	(0.0024)	$\tau_{high} = 1.3333$
Heterogeneous-PCHM	0.0248	$\tau_{low} = 0.4444$
split at $\tau = 0.9557$ (mean)	(0.0032)	$\tau_{high} = 1.3333$
Heterogeneous-PCHM	0.0258	$\tau_{low} = 0.2222$
3 categories	(0.0021)	$\tau_{med} = 0.8888$
(split at 33 and 66 percentile)		$\tau_{high} = 1.7778$

Table 27: Classify τ based on other split points

B.2.4 Best-Fit Values of Parameters for Individual Games

Below we report below the distribution of best-fit values of τ in each of our two data sets. Our parameter estimates for the games in Stahl and Wilson (1995) are similar to the estimates reported in Camerer, Ho and Chong (2004) (Table III):[38]

	Our estimate of τ	*Estimate of τ reported in Camerer, Ho and Chong (2004)*
Game 1	*3.23*	*2.93*
Game 2	*0*	*0*
Game 3	*1.21*	*1.40*
Game 4	*2.82*	*2.34*
Game 5	*2.02*	*2.01*
Game 6	*0*	*0*
Game 7	*8.08*	*5.37*
Game 8	*0*	*0*
Game 9	*1.21*	*1.35*
Game 10	*6.46*	*11.33*
Game 11	*8.48*	*6.48*
Game 12	*1.61*	*1.71*

Table 28: Comparison of our parameter estimates for lab games from Stahl and Wilson (1995) with the estimates reported in Camerer, Ho and Chong (2004).

In both data sets, the best-fit value of τ varies substantially across games. Panel (a) in Figure 15 below shows a histogram of best-fit values of τ across the games in our lab data set. In our main analysis in Section 5, we remove the right tail of estimates (as these increase the variance in the output of the prediction algorithm) by imposing a constraint that $\tau \leq 2$. Under this constraint, the median value of τ is 0.89 and the variance is 0.44 (For the unconstrained

[38] These parameter estimates should be interpreted with caution, since prediction error as a function of τ is a poorly behaved function with large discontinuities. Moreover, for some games, rather different values of τ turn out to yield prediction errors that closely approximate the global minimum. These curiosities explain some of the small differences in the parameter estimates below.

values, the median value of τ is 1.21, and the variance is 6.46.) The distribution of best-fit values of τ in this range is shown below in panel (b) of Figure 15.

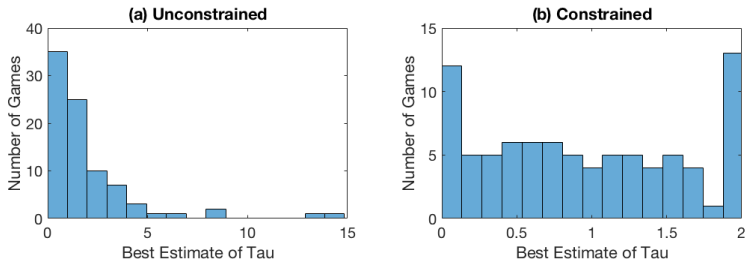


Figure 15: The best-fit τ varies substantially across lab games: in (a) we report the best-fit values of τ, in (b) we report the best-fit values of τ under the constraint that $\tau \leq 2$.

We additionally show the distributions of best-fit values of τ and λ across lab games for the LPCHM. The distribution of best-fit values of τ shifts right relative to Figure 15; this likely reflects that many of the games in which the best-fit value of τ was low had less concentrated distributions of play, and this can alternatively be captured using λ.

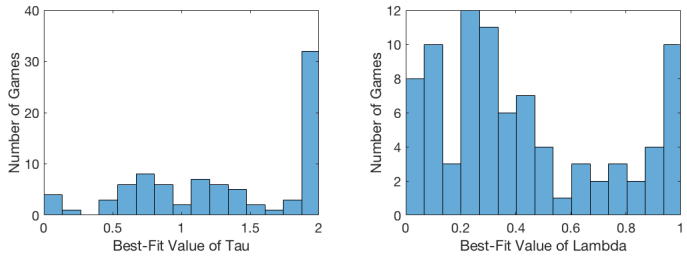


Figure 16: The best-fit values of τ and λ in the LPCHM vary across our set of lab games.

Finally, we show the distributions of best-fit values of τ and λ for the LPCHM with level 0 players removed. Note that unlike in the PCHM, here the parameter value $\tau = 0$ does not correspond to probability 1 of level 0 play.

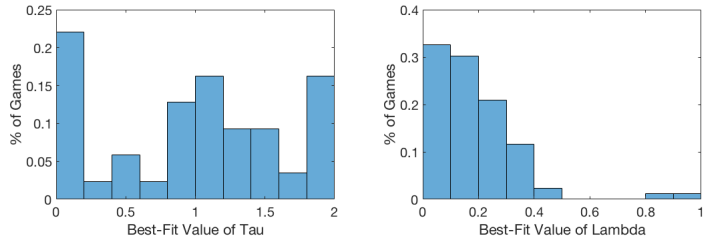


Figure 17: The best-fit values of τ and λ in the LPCHM with level 0 players removed vary across our set of lab games.

B.3 Experimental Instructions

The instructions provided to Mechanical Turk subjects in the experiments described in Sections 2.1 and 6.1 can be found below. With a few exceptions, instructions that were repeated across these experiments are only presented once.

B.3.1 Playing Random Games (Section 2.1): Initial Instructions

We are researchers interested in how people play a simple kind of game.

Rules of the game

There are two players. Each player is assigned to one of two roles: **orange** and **green**. Both players move only once, and they move at the same time. The orange player's move is to choose one of

A **B** **C**

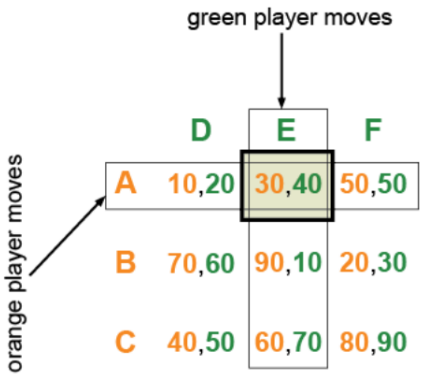
and the green player's move is to choose one of

D **E** **F**

Depending on which moves are chosen, each player wins a certain number of points. These points are displayed in a table like this one:

	D	E	F
A	10,20	30,40	50,50
B	70,60	90,10	20,30
C	40,50	60,70	80,90

To read this table, look at the row marked with the orange player's move, and the column marked with the green player's move. This determines a pair of numbers. For example, if the orange player moves **A** and the green player moves **E**, then you should look at **30**,**40**.



The **first number** is the number of points that the orange player wins, and the **second number** is the number of points that the green player wins.

Easy? Let us ask you a few questions to make sure you got it.

Comprehension Question 1/2

	D	E	F
A	50,40	90,30	20,70
B	30,10	40,90	20,60
C	60,10	50,80	80,40

You are the **orange player**. If you choose **A** and your partner chooses **F**, how many points will you win?

Comprehension Question 2/2

	D	E	F
A	90,90	40,30	70,30
B	70,60	30,30	40,70
C	50,40	80,10	90,30

You are the **green player**. If you choose **D** and your partner chooses **B**, how many points will you win in this game?

Great! You answered both questions correctly. Now let's move on to your main task.

Your task

We will show you fifteen games like the one described above. You will be asked to play the **orange player** in each of these games.

How you are paid

You will be paid a **base rate of $0.35** for completing the HIT. In addition, one of the fifteen games you play will be chosen at random. We will match you with another subject who has been asked to play as the orange player, and we will use your joint moves to determine the number of points you win. You will then receive **a bonus** of:

$0.01 x the number of points you won in that game

This bonus will range from $0.10-$0.90. Please allow up to a week to receive this.

We are almost ready to begin the exercise.

Please read through the following information and indicate your consent before continuing.

B.3.2 Playing Random Games (Section 2.1): Typical Question

Consider the following game.

	D	E	F
A	50,80	10,20	50,50
B	50,50	20,30	90,20
C	40,20	50,70	10,20

You are the **orange player**. What move do you choose?

- ○ A
- ○ B
- ○ C

B.3.3 Predicting the Most Likely Action (Section 6.1), Initial Instructions:

How well can you guess how people will play in games?

We are researchers interested in whether you can predict how people play in a simple kind of game. Real people were matched with a partner and asked to play the following two-player game:

Rules of the game

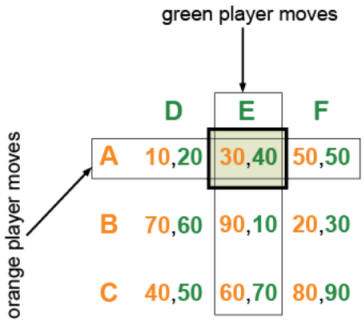
There are two players. Each player is assigned to one of two roles: **orange** and **green**. Both players move only once, and they move at the same time. The yellow player's move is to choose one of

A **B** **C**

and the green player's move is to choose one of

D **E** **F**

Depending on which moves are chosen, each player wins a certain number of points. These points are displayed in a table like this one:



The number of points the orange player wins is the **first number**, and the number of points the green player wins is the **second number**.

Easy? Let us ask you a few questions to make sure you got it.

Great! You answered both questions correctly. Now let's move on to your main task.

The challenge
Real people were asked to play games like the ones you just looked at. In each round of this HIT, we will show you the points table for one of these games, and ask you to guess which move was most frequently chosen by the **orange player**. There are fifteen total games.

How you are paid

You will receive **$0.25** no matter what for completing this HIT. But you will receive **$0.05** more for every round in which you correctly guess the move most frequently chosen. This means that you will win **a bonus of up to 0.75**. Please allow up to a week for the bonus to arrive.

You may only complete this HIT once. If you complete this HIT multiple times, you will be rejected.

We are almost ready to begin the exercise. Please read through the following information and indicate your consent before continuing.

B.3.4 Predicting the Most Likely Action (Section 6.1), Typical Question:

Consider the following game.

	D	E	F
A	45,45	50,41	21,40
B	41,50	0,0	40,100
C	40,21	100,40	0,0

Which move do you think was most frequently chosen by the orange player?

- ○ A
- ○ B
- ○ C

B.4 Explanation of Choices in Experiments

Subjects were asked to explain how they made their choices in a (free-form) text box. We show below selected answers from our experiments in which players were asked to choose an action:

- "I chose based on mutually beneficial numbers, followed by singular benificial [sic] numbers, and finished with whatever was left over."
- "Except the first question. I added the orange in each row(A,B,C) Then put it in order from highest to the least. I'm hoping I did this right :o)"
- "i count each value quickly. It is easy for me. Good game"
- "I assumed Green was aquisitive [sic] and non-sharing"
- "Without knowing what sort of patterns the partner displayed it's mostly guesswork. I assumed orange would avoid choosing rows where zero payoff was possible, and that green would similarly prefer not to bet on columns with a zero payoff. I assumed both would think the same way and be trying to achieve a good payoff, not just selecting the row or column with the highest possible payoff. Wheels within wheels."
- "i tried to figure out if there is obvious worst of all, then eliminate it"
- "I looked at what Green would probably pick and then based on that decided what Orange would pick when thinking about what the Green letter would likely be."

We show below selected answers from our experiments in which players were asked to predict the play of others:

- "I picked the lines that had the biggest looking numbers. People like big numbers."
- "I chose mostly the midrange digits for most and varied the low and high for mid and least."
- "I looked at the highest numbers and whether there were any zeroes in the line, because I figured that would be a huge deterrent for someone."
- "I chose the route of either placing the orange player in a strict profit maximizing role without taking into account the decisions of the green player, or I chose the best scenario for both the orange and green player."
- "I just picked what felt right at the particular game"
- "i was aware that the best way to choose orange move was relative to the best move for green but i don't think people that took this study was smart enough for considering that and they would choose first the move that had the biggest number."
- "i just tried to be logical"