Predicting and Understanding Initial Play[*]

Drew Fudenberg[†] Annie Liang[‡]

First version: November 14, 2017
This version: April 15, 2019

Abstract

We use machine learning to uncover regularities in the initial play of matrix games. We first train a prediction algorithm on data from past experiments. Examining the games where our algorithm predicts correctly, but existing economic models don't, leads us to add a parameter to the best performing model that improves predictive accuracy. We then observe play in a collection of new "algorithmically-generated" games, and learn that we can obtain even better predictions with a hybrid model that uses a decision tree to decide game-by-game which of two economic models to use for prediction.

[*] We are grateful to Alberto Abadie, 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, equilibrium analysis is a poor predictor of the choices that participants make the first time they play a new game. Initial play does however have regularities, as for example shown by the fact that level-k models (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 algorithms to discover new regularities in initial play, and to improve upon existing models. We study the prediction of the action most likely to be played in a given game. Throughout, we evaluate *out-of-sample* performance, meaning we use different data for training the model and for testing it.[1] We report both the accuracy of the model and its *completeness*, which we take to be the percentage of the possible improvement over random guessing, as in Peysakhovich and Naecker (2017) and Kleinberg, Liang and Mullainathan (2017).[2] Our improvements on existing theories of initial play are of interest in their own right, but our methods for using machine learning to extend and inform modeling are more general. Their success here suggests that using machine learning techniques to inform modeling may be useful in other domains within economics as well.

Our investigation proceeds in the following steps, which we first briefly summarize, and then explain in more detail below.

(A) First, we train a decision tree to predict play in some past experiments. We study the games where machine learning models predict well and existing models do not, which leads us to formulate a one-parameter extension of level-1 play, level-1(α), that makes better predictions.

(B) Next we run experiments on games with randomly determined payoffs, and use that data to algorithmically generate new games that are designed to display behaviors that are not captured by level-1(α).

(C) We then elicit play on the algorithmically generated games and train decision trees on the new data. These decision trees suggest that, in the new games, whether an action is part of a Pareto-dominant Nash equilibrium (henceforth PDNE) is a good predictor of whether it will be played.

(D) Neither the level-1(α) model nor PDNE performs well when evaluated on

[1] Increasing the flexibility of model—e.g. by adding additional parameters—results in weakly better in-sample fit (where the training and testing data are the same). But increased flexibility need not result in higher out-of-sample fit, as more complex models are more likely to overfit to the training data.

[2] Camerer, Ho and Chong (2004)'s related "economic value" compares the expected payoff that results from best-responding to a theory's forecast to the payoff that subjects actually obtained; this measure cannot be computed without a prediction of the entire distribution of play.

the combined data set of all games (lab, randomly-generated, and algorithmically-generated), but we obtain substantially better predictions by training a hybrid model that decides when to make the level-1(α) prediction and when to make predictions based on PDNE.

We now go into more detail for each of the steps above.

(A) Where and why do our decision trees perform better than level-1?

The initial data set we consider consists of play in symmetric 3×3 matrix games from six experimental game theory papers. In 72% of these games, the modal action was the action that maximizes expected payoff against the uniform distribution, i.e. the *level-1* action.[3] Although the level-1 model performs quite well, our relatively crude machine learning techniques (decision trees built on a set of features that describe strategic properties of the available actions) lead to a substantial improvement.[4] To understand the regularities that allow this improvement, we then examine the 9 (out of 86) games where play is predicted correctly by our algorithm, but not by level-1. Each of these games has an action whose average payoffs closely approximate the level-1 action, but with 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, as if they are risk averse. This leads us to extend the level-1 model to level-1(α), which predicts the level-1 action when dollar payoffs u are transformed under $f(u) = u^\alpha$ (so that the usual level-1 model is level-1(1)).[5] The performance of this model shows how atheoretical prediction rules fit by machine learning algorithms can help researchers discover interpretable and portable extensions of existing models.

(B) Algorithmic Experimental Design

The strong performance of the level-1 prediction rule, and the even better performance of level-1(α), are interesting in their own right, but leave open the question of how widely these findings extend beyond our specific set of laboratory games. We would like to understand how generally the level-1 model is a good description of modal behavior, and also identify the games where it predicts poorly and what behaviors it misses. To do this, we need data on play in new games.

Our first step was to construct games with randomly generated payoffs. We found that the level-1(α) model was an even better predictor of play in these random

[3] The best-performing version of the Poisson Cognitive Hierarchy model, 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. See Section 3.2.

[4] See Table 2 in Section 3.2 for accuracy and completeness estimates.

[5] As we discuss in Section 3.2, allowing for risk aversion parameter to generate better predictions has many precedents in the experimental literature.

games than in the lab games, making the correct prediction 89% of the time.[6] In principle we could still identify new regularities by examining data on a sufficiently large set of randomly generated games, but it is more efficient to focus attention on games where behavior is less likely to conform to the predictions of level-1(α). To generate such games, we used an algorithmic approach: First, we trained a rule for predicting the frequency of level-1(α) play based on the game matrix. Then, we generated payoff matrices at random, filtered out all the games where the predicted frequency of level-1(α) play was over 50%, and repeated until we had a set of 200 games.

(C) Learning From the the New Data

We elicited play in these "algorithmically designed" games on Amazon Mechanical Turk (MTurk) with 40 subjects per game. The data from these games showed that the algorithmic game generation procedure was effective in producing games where level-1(α) performed poorly. Moreover, a decision tree trained on this data substantially outperforms level-1(α) on this data, suggesting that there are regularities in initial play that are not captured by level-1(α). Directly consulting this tree did not yield new insights, since the best decision tree was complex and hard to interpret. But a simple version of the decision tree (restricted to just two decision nodes) returns predictions consistent with Pareto Dominant Nash equilibrium (PDNE).

(D) Hybrid Models

Our findings from the new games demonstrate that level-1(α), while highly predictive of play in the lab games and randomly-generated games, is outperformed in other games by models such as PDNE that depend on both player's payoffs, and so are more suggestive of strategic behavior. This suggests that we could further improve both our predictions and our understanding of initial play by learning which games are well-predicted by level-1(α) and which games are better predicted by PDNE.

Thus, we combine the level-1(α) model and PDNE into a hybrid model that first chooses between the level-1(α) model and PDNE, and then makes the corresponding prediction. To do this, we train regression trees to forecast the accuracies of these two ways of making predictions, and then use the model with the higher predicted accuracy. Our combination of the easily-interpreted level-1(α) model and PDNE is a hybrid "meta-model" that uses an algorithmic structure to combine simple behavioral/economic models. This hybrid model outperforms either of its parts,

[6] As discussed in Section 4.1, this is partly because the games with randomly generated payoffs tended to be "strategically simpler": compared to the lab games, the games with random payoffs were more likely to be dominance solvable, more likely to include a strictly dominated action, and less likely to have three or more pure-strategy Nash equilibria.

which shows that there are useful methods that straddle the "behavioral versus algorithmic" dichotomy.

1.1 Background Information and Related Work

As the Crawford, Costa-Gomes and Iriberri (2013) survey shows, there is an extensive literature that models initial play in matrix games. Most of these papers use some variant of "cognitive hierarchies," whose starting point is the specification of a "level-0" or unsophisticated player who is assumed to assign equal probability to each action. The various models then use the level-0 type to build up a richer specification of play.[7]

The simplest model of initial play is "level-1," which assumes that the whole population plays a best response to level-0. As we will see, this model does a reasonably good job of predicting the most likely (i.e. modal) action in many games, but there is substantial room for improvement. Our goal is to identify alternative models that are not only better at predicting play, but also interpretable and portable. In this respect our work is analogous to the extensions of the Poisson Cognitive Hierarchy model proposed by Leyton-Brown and Wright (2014) and Chong, Ho and Camerer (2016), which modify the specification of level-0 play.[8] Our paper is 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 games, and Sgroi and Zizzo (2009) and Hartford, Wright and Leyton-Brown (2016), which develop deep learning techniques for predicting play. These papers differ from ours in that their emphasis is predictive accuracy, instead of 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

[7] Outside of the domain of matrix games, modelers sometimes specify other choices for level-0, for example Crawford and Iriberri (2007) study "truthful" level-0's in an incomplete-information auction.

[8] Leyton-Brown and Wright (2014) replaces the specification of level-0 from uniform play with 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."

settings.[9]

Our hybrid models are a form of "mixture of experts" (Masoudnia and Ebrahimpour, 2014). They are related to methods such as "model trees" (Quinlan, 1992), which are decision trees that select between various parameters of linear regression models, and to "logistic model trees" (Landwehr, Hall and Frank, 2005), which replace linear regression with logistic regression to adapt model trees to classification tasks.

2 Predictions and Their Performance

2.1 Prediction Task

Throughout the paper we consider only 3 × 3 matrix games. The set of games is $G = \mathbb{R}^{18}$, and we use g to denote a typical game.

The prediction task we study is a classification problem: given a game, we seek to predict the action most frequently chosen by the row player (i.e. the modal row-player action in the observed play). The classification rules for this task are easier to understand than those for predicting distributions, and thus allow for a clearer exposition of our methods.[10]

For this problem, a prediction rule is a mapping $f : G \to A_1$ from games to the set of row player actions.

2.2 Prediction Rules

We evaluate several rules for predicting the modal action in a game. We first consider 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. Predict at random from the set of row player actions that are part of a pure-strategy Nash equilibrium profile.

Level-1. Following Stahl and Wilson (1994, 1995), define a player to be "level-0" if he randomizes uniformly over his actions. The level-1 prediction rule assigns to each game the best response to a level-0 player—we will also refer to these best

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

[10] In an earlier version of this paper we considered the problem of predicting the distribution of play. Our results there suggested that hybrid models have potential to be useful for that problem as well, although the improvements were smaller than those we report here.

responses as *level-1 actions*. When the level-1 prediction is not unique, we randomize over the set of level-1 actions.

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.[11] The predicted distribution over actions is based on the assumption that the actual proportion of level-k players in the population is proportional to $\pi_\tau(k)$. We predict the mode of this aggregated distribution.

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 available actions. For each action, we define an indicator variable for whether the action has each of the following properties: whether it is part of a pure-strategy Nash equilibrium, whether it is part of a pure-strategy Pareto-dominant Nash equilibrium (i.e. its payoffs Pareto-dominate the payoffs of all other Nash equilibria),[12] 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)) or maximizes the minimum row player payoff (*pessimistic* in Costa-Gomes, Crawford and Broseta (2001)). We also include a score feature for how many of the above properties each action satisfies as a richer expression of how appealing the action seems.

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 that 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

[11] Throughout, we take τ to be a free parameter and estimate it from the training data.

[12] Note that a unique Nash equilibrium is always Pareto-dominant.

on the first, and so forth. When we say “best decision tree, ” we mean the tree grown using this method that achieves the highest out-of-sample accuracy. Using other algorithms such as random forests and 2-layer neural nets does not yield improvements for this data set (see Appendix B), and the outputs of these alternative algorithms are harder to interpret, so we focus on decision trees in the main text.

2.3 Performance Measure

An observation is a pair (g, a) consisting of a game g and the action a most frequently chosen by subjects in the role of the row player in that game, i.e. the modal row-player action. Given a set $\{(g_i, a_i)\}_{i=1}^n$ of n games and their modal actions, we measure the accuracy of prediction rule f using

$$\frac{1}{n}\sum_{i=1}^{n} \mathbb{1}(a_i = f(g_i)).$$

This is the fraction of games g_i in which the predicted modal action $f(g_i)$ is indeed the observed modal action a_i in that game.[13]

We call the *ideal* prediction rule the rule that assigns to each game the observed modal action in that game, and so predicts perfectly. This benchmark is idealized because it uses knowledge of the test set, and also because the modal action in our data may not be the one we would have seen with more data. In Appendix D we report completeness measures relative to two alternative benchmarks that do not have these features.[14] We use the prediction rule that corresponds to guessing uniformly at random as a naive baseline; this yields an expected accuracy of 1/3.

Unless explicitly stated otherwise, we report tenfold cross-validated prediction accuracies. This means that we divide the games into ten folds, use the games in nine of the folds for training, and use the remaining games for testing. The reported accuracy is averaged across the different choices of test fold. The reported standard errors for the cross-validated prediction accuracies are the standard deviation of prediction accuracies across choices of test sets, divided by $\sqrt{10}$, because we use 10 folds (see Hastie, Tibshirani and Friedman (2009) for procedural details).[15] Some

[13] We consider a related accuracy measure in Appendix C, where accuracy is the number of instances of play that are predicted correctly. With this accuracy measure, it is more important to correctly predict the modal action in games where the modal action is played more frequently. The performance ranking of the models could in principle change, but we find that it stays the same.

[14] The associated completeness measures are higher for all models—and in some cases substantially higher—so the completeness measures that we report in the main text should be understood as conservative estimates.

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

of our prediction algorithms do not require estimation from a training set, and for these prediction algorithms we report bootstrapped standard errors.[16]

3 Laboratory Games

3.1 Laboratory Data

Our data on play in laboratory experiments consists of all 3×3 matrix games in a data set collected 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 3×3 normal-games.[17] Some of these observations were row players and some were column players, but since the games we consider are symmetric, we label all observed actions as row-player actions. Table 1 lists the number of games and the number of observations from each paper.

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.

3.2 Results

Table 2 reports the accuracies and completeness measures of our prediction rules on the lab data. When evaluating the PCHM, the best-performing τ (estimated from training data) returns the level-1 prediction rule, so we report the performance of

[16] We re-sampled our data 100 times and evaluated the model on each of these data sets. We report the standard deviation of the prediction accuracies.

[17] Our data set does not have individual-level subject identifiers.

these two models together.[18,19]

	Accuracy	Completeness
Guess at random	0.33	0%
Uniform Nash	0.42	13%
	(0.05)	
Level-1/PCHM	0.72	58%
Decision tree	0.77	66%
	(0.04)	
Ideal prediction	1	100%

Table 2: Predicting the modal action in lab data.

We find that the PDNE rule and the uniform Nash prediction rule are only slightly better than guessing at random. In contrast, the level-1 model achieves a substantial improvement, increasing completeness to 58%. The decision tree (reported in Appendix E.1) based on game features performs better still, achieving a completeness of 66%.

Out of the 86 lab games, modal play is level-1 in 62 of the games. Moreover, there are nine games in which the modal action is not level-1 but *is* correctly predicted by the decision tree. The performance of the decision tree on those nine games gives us reason to believe that there is a systematic pattern to play in these games, beyond what is already captured by the level-1 model. We thus examine these games, displayed in Figure 1, and search for additional regularities.

[18] 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.

[19] PCHM (and other variants we consider) better fit the *distribution* of actions, as we showed in an earlier version of the paper.

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

Examining these games reveals a common feature: In each game, some action that is not level-1 yields an expected payoff against uniform play that is nearly as high as the level-1 payoff, and moreover has lower variation in possible row payoffs. Consider the first game in Figure 1. 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 choosing action a_1 yields significantly lower variation in possible row player payoffs.[20] 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.

We can modify the level-1 model to account for this regularity. Specifically, because the departure from level-1 behavior is consistent with a risk averse utility function over payoffs, we consider an alternative model in which players maximize against a uniform distribution of opponents' play (as in level-1), but the dollar payoffs u are transformed under $f(u) = u^\alpha$. We call the resulting model level-1(α); the standard level-1 model is nested as $\alpha = 1$. Table 3 compares the prediction error of level-1(α) with the original model.[21] We find that introducing this risk aversion parameter reduces prediction error substantially, achieving the prediction error of the best decision tree (with an estimated value $\alpha^* = 0.625$).

By focusing our attention on the 9 games where the tree predicted correctly but level-1 did not, our machine learning model allowed us to detect a new empirical regularity. Thus, the success of level-1(α) demonstrates how atheoretical prediction rules can help us identify parametric extensions of existing models that generate better predictions.

	Accuracy	Completeness
Level-1	0.72	58%
Decision Tree	0.77	66%
	(0.04)	
Level-1(α)	0.79	69%
	(0.04)	

Table 3: Introducing risk aversion improves level-1.

Risk aversion strikes us as a natural interpretation of the α parameter, and there is substantial evidence that small stakes risk aversion is a better description of laboratory play choices than is risk neutrality. That said, risk aversion is only one interpretation, and risk aversion for such small stakes is hard to reconcile with

[20] Depending on which action the column player takes, the row player will receive one of $\{43, 91, 11\}$ if he (the row player) chooses a_3, compared to $\{47, 51, 28\}$ if he chooses a_1.

[21] Once again, the PCHM did not yield an improvement.

standard expected utility theory (see e.g. Rabin (2000)).[22]

4 Generating New Games

The strong performance of the level-1 prediction rule, and our subsequent extension to level-1(α), are interesting in their own right, but leaves open the question of whether this performance is special to our specific set of laboratory games. We would like to understand whether the level-1(α) model is *generally* a good description of modal behavior. If there are games in which it does not predict well, we would like to know what these are, and what behaviors the model misses. To answer these questions, we need a larger and more varied set of games.

In a first attempt to generate such games (Section 4.1) , we constructed 200 games with randomly generated payoff matrices. These games do not have the special structure of the experimentally designed games, so they test the robustness of our findings, and also give us an opportunity to discover new behaviors.

We find that the level-1(α) model is an even *better* predictor of modal play in these randomly-generated games than in the laboratory games. This finding is reassuring, since it tells us that the performance of level-1(α) in the laboratory games was not a quirk of the design of these games. But it also means that studying play in random games is an inefficient way to uncover new regularities. If we want games in which the level-1(α) action is not modal, we need a more sophisticated approach for game generation.

One option would have been to hand-craft games where we conjectured that play would depart from level-1(α). Instead, we tried to learn this structure from our data. To do this, we trained a machine learning algorithm to predict the frequency of play of the level-1(α) action, and then selected games that achieved low predicted frequencies according to this algorithm. This "algorithmic game generation" is described in detail in Section 4.2.

4.1 Random Games

Our first auxiliary set of games consists of 200 payoff matrices generated 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

[22] Rabin suggested loss aversion as an explanation for apparent risk aversion, but loss aversion is not applicable when all of the gambles are in the gains domain, as in Holt and Laury (2002) and our data. Fudenberg and Levine (2006, 2011) instead explain small stakes risk aversion as a combination of a self control problem and the "narrow bracketing" proposed by Shefrin and Thaler (1988). More recently, Khaw, Li and Woodford (2018) explains small stakes risk aversion as a result of "cognitive imprecision."

generated games are not symmetric. We presented each of 550 MTurk subjects with a random subset of fifteen games, and asked them to play as the row player.[23]

Subjects faced the following incentives: On top of a base payment of \$0.35, they 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).[24,25]

Relative to the random games, the games played in lab experiments have more 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.

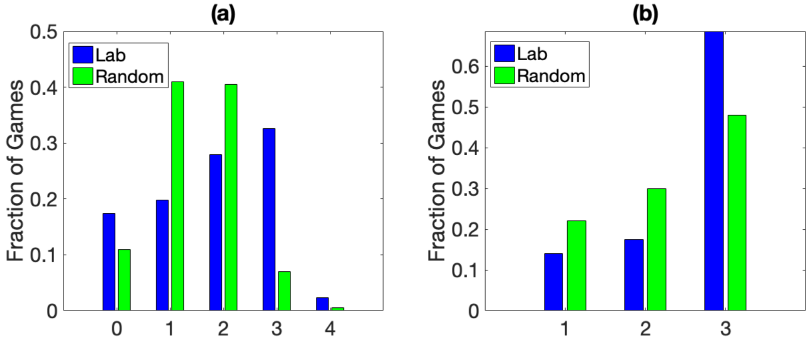


Figure 2: (a) Percentage of games with zero, one, two, three, or at least four pure strategy Nash equilibria; (b) Percentage of games with one, two, or three actions surviving iterated elimination of (pure-strategy) dominated actions.

Table 4 reports prediction accuracies for this new data set. We find that level-1(α) again improves upon the level-1 model.[26] Moreover, both models perform very well—in fact, achieving *higher* predictive accuracies than they did on the lab data. The level-1(α) model predicts the modal action correctly in 92% of new instances, and achieves 88% of the achievable improvement over random guessing. (Note that

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

[24] We restricted the subject pool to MTurk participants in the United States who had an approval rate of 75% or higher. Subjects spent an average of seven minutes on the task, and the average payment was \$0.93, or \$8.14 an hour. (This is a typical hourly wage for MTurk.) 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 the Online Appendix.

[25] In addition to eliciting play, we asked subjects to volunteer a free-form description of how they made their decisions. A selection of answers can be found in the Online Appendix.

[26] The value of α estimated on this data set is $\alpha = 0.41$.

in contrast to the lab data, the level-1 variants are not outperformed by the best decision tree.[27])

	Accuracy	Completeness
Guess at random	0.33 (0.02)	0%
Uniform Nash	0.57 (0.03)	36%
Decision Tree	0.86 (0.02)	79%
Level-1	0.87 (0.01)	81%
Level-1(α)	0.92 (0.02)	88%
Ideal prediction	1	100%

Table 4: Predicting the modal action in the random games.

The improved performance of level-1 here may be due to differences between the games that were crafted by experimenters and those with randomly generated payoffs, as discussed above. A second possibility is that the improvement is driven by differences between the laboratory subjects and the MTurk subjects. Indeed, we might expect that MTurk subjects are less sophisticated about the strategic aspects of the game, and hence are more likely to choose the level-1 action. To separate this *subject-based* explanation from the previous *game-based* explanation, we ran another experiment in which we asked MTurk subjects to play the lab games. In this new data, the level-1 model achieved a prediction accuracy of 0.68, which is much closer to the prediction accuracy of 0.72 we found for the lab games (Table 2) than the accuracy in the random games of 0.87 (Table 4). This suggests that the improved performance of level-1 on the new data set of randomly generated games is driven at least in part by the difference in the strategic structures of the games—our subsequent results will reinforce this view.[28]

[27] 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 use 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.

[28] Many authors have considered how much behavior in laboratory experiments resembles behavior on MTurk. While there are some differences, the consensus seems to be that the two types of data are similar. See e.g. Paolacci, Chandler and Ipeirotis (2010) "experimenters should consider Mechanical Turk as a viable alternative for data collection"; Rand (2012) "...evidence that data collected (on MTurk) is valid, as well as pointing out limitations" ; Mullinix et al. (2015) "The results reveal considerable similarity between many treatment effects"; Thomas and Clifford

Collectively, these results reveal that the structure of the laboratory games made level-1 play *less* prevalent, which suggests that subjects are most likely to depart from level-1 play exactly in games that are "strategically interesting." Thus, to identify regularities in play beyond level-1(α), we need more games that will induce such behaviors. One approach would be to hand-craft games along the lines of the original lab games, or to select games with specific features expected to lead to interesting findings, as in Stahl (2000). Instead, as described in subsection 4.2, we automated the game generation procedure by conjecturing many different strategic features that could be relevant, and then using machine learning to select which games were more likely to induce departures from level-1(α) play.

4.2 Algorithmic Experimental Design

We first trained an algorithm on the data in both the lab games and the randomly-generated games to predict the frequency which which the level-1(α^*) action was played. Throughout, we fix $\alpha^* = 0.625$ (our estimate of α from Section 3.2).[29]

For training, we used *bootstrap-aggregated* decision trees (also known as *bagged* decision trees).[30] These trees were built on a feature set describing various strategic properties of the game (see Appendix A.2 for the complete feature set), chosen based on our conjectures of what might determine the attractiveness of the level-1(α^*) action.

For example, one feature we thought might matter is whether the level-1 action is part of a pure-strategy Nash equilibrium. Another feature is the difference between the sum of possible row player payoffs given play of the level-1 action and the next

(2017) "...insufficient attention is no more a problem among MTurk samples than among other commonly used convenience or high-quality commercial samples, and...that employing rigorous exclusion methods consistently boosts statistical power without introducing problematic side effects." Finally, Snowberg and Yariv (2018) find that behavior in their MTurk data is closer to that in their nationally-representative survey data than is the behavior in their student data.

[29] We needed to fix the value of α since we could not anticipate the best-fit value of α for play on the yet-to-be designed games.

[30] This algorithm generates bootstrap samples of the training data and uses each sample to train a tree. The predictions of the different trees are averaged for out-of-sample prediction. Bagged trees are generally considered more predictive but less interpretable than the single decision tree (Breiman, 1994). To generate games where level-1(α^*) is not common, we care only about successful predictions, so we use tree ensembles, while elsewhere we rely on single decision trees to make it easier for us to interpret our results.

highest row sum. In the game below, this "row sum gap" takes a value of 20:

	a_1	a_2	a_3	Row Sum
a_1	$40, 40$	$20, 30$	$0, 20$	60
a_2	$30, 20$	$20, 20$	$100, 10$	150
a_3	$20, 0$	$10, 100$	$100, 100$	130

Yet another feature is whether the game contains a Nash equilibrium that yields "high payoffs" (specifically, at least 75% of the largest payoff sum[31]) and is not level-1—for example, the action profile (a_3, a_3) above.

After training a tree ensemble to predict the frequency of play of the level-1(α^*) action, we used it to generate a new data set of symmetric games. We started by randomly generating a set of 200 games whose row player payoffs were selected from the empirical payoff distribution from the lab data set, with the column player payoffs chosen symmetrically. Then, we applied our algorithm to predict the frequency of play of the level-1(α^*) action in those games. We eliminated all games in which the predicted frequency was larger than 1/2, and randomly generated new games to replace them, repeating this procedure until all games were predicted to have less than 1/2 frequency of play of the level-1(α^*) action.[32,33]

A typical game generated by the algorithm is the following:

	a_1	a_2	a_3
a_1	$90, 90$	$30, 80$	$45, 30$
a_2	$80, 30$	$55, 55$	$37, 5$
a_3	$30, 45$	$5, 37$	$70, 70$

Note that this game has three pure-strategy Nash equilibria: (a_1, a_1), (a_2, a_2), and (a_3, a_3). The level-1(α^*) action is a_2, but the expected payoff of a_1 against uniform play is close to the payoff from a_2, and a_1 is also part of a Pareto-dominant Nash equilibrium.

In general, while the randomly-generated games were strategically simple, the algorithmically designed games exaggerate strategic complexity. For example, Figure

[31] We chose the cutoff 75% somewhat arbitrarily, although in the subsequent Section 6 we introduce variations on this feature that use different cutoffs.

[32] The threshold 1/2 was chosen somewhat arbitrarily. Our tree ensemble very rarely predicted frequencies lower than 0.4, so our choice of 1/2 was guided by our desire to both have a low threshold and also have sufficiently many instances where the frequency of level-1(α^*) is predicted to be below the threshold.

[33] Our approach is related in spirit to adversarial machine learning Huang et al. (2011) and generative adversarial networks Goodfellow et al. (2014) in that we are generating instances to trick the level-1(α^*) model. Here, though, our goal is to design new instances for data collection.

3 replicates Figure 2 with the new games added in, and shows that the distribution of the number of pure-strategy Nash equilibria in the new games (as well as the number of rationalizable actions) first-order stochastically dominates the corresponding distribution in the lab games.

We elicit play in these new games on MTurk (using an identical experiment to the previous section), collecting 40 observations per game.

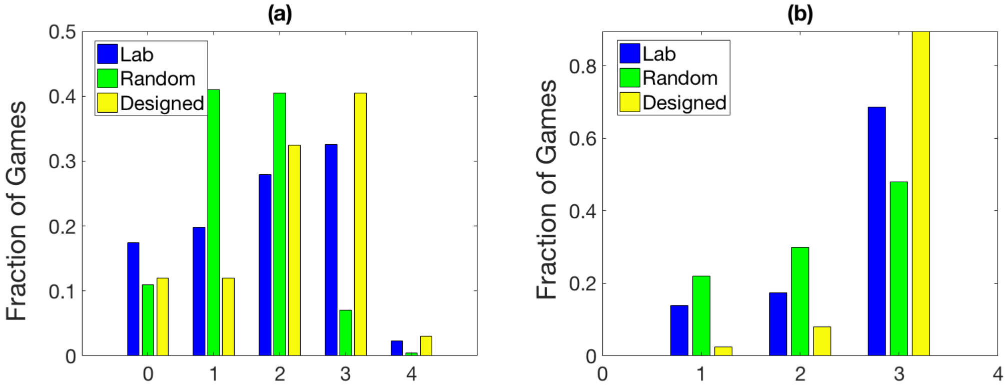


Figure 3: (a) Percentage of games with zero, one, two, three, or four pure strategy Nash equilibria (no games had more than four Nash equilibria); (b) Percentage of games with one, two, or three actions surviving iterated elimination of (pure-strategy) dominated actions.

5 Preliminary Lessons from the New Data

Table 5 reports the prediction accuracies of our best decision tree and of the models used above. We evaluate these approaches first on the new set of algorithmically designed games, and then separately on the full data set of games (consisting of the lab games, the randomly-generated games, and the algorithmically designed games).

	Algo Games Only		All Games	
	Accuracy	Completeness	Accuracy	Completeness
Guess at random	0.33	0%	0.33	0%
Uniform Nash	0.43	15%	0.49	24%
	(0.03)		(0.02)	
Level-1	0.36	5%	0.64	46%
Level-1(α)	0.38	7%	0.68	52%
	(0.02)		(0.02)	
Decision Tree	0.67	51%	0.70	55%
	(0.03)		(0.03)	
Ideal prediction	1	100%	1	100%

Table 5: Predicting the modal action

The algorithmically designed games were selected to be poor matches for the level-1 models, and we find that they succeed in this goal: the level-1(α) model correctly predicts the modal action in only 38% of games, achieving a completeness of 7%. (Recall that level-1(α) achieved an completeness of 58% for the lab games and 84% for the randomly-generated games.) In the aggregated data, the accuracy of level-1(α) is 0.66 and its completeness is 34%.[34]

The best decision tree in the algorithmically designed games is complex and hard to interpret, so we present the best 2-split decision tree instead, which achieves an accuracy of 0.62[35]; this is still substantially better than either uniform Nash or level-1(α) but below the 0.67 of the best tree. This 2-split tree, shown in Figure 4, is very simple: if there is a Pareto-dominant Nash equilibrium, the tree predicts it; otherwise the tree defaults to action a_3.[36]

[34] The best-performing value of α for the algorithmically designed games is 0.97, but given that play in these games is poorly predicted by the level-1(α) model, it is not clear that this parameter estimate has a meaningful economic interpretation.

[35] The standard error of the accuracy is 0.02.

[36] When we report trees such as this one, we report the tree estimated on the full data set, since the trained tree potentially fluctuates across choices of training data. This 2-split tree was produced on seven of the ten training sets.

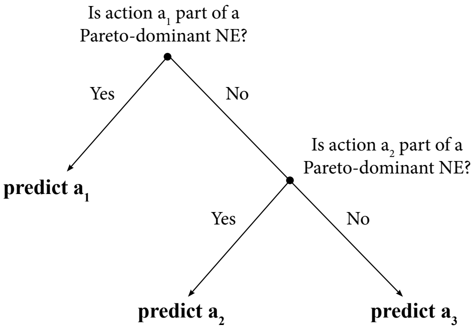

Is action a$_1$ part of a Pareto-dominant NE?
Yes
No
predict a$_1$
Is action a$_2$ part of a Pareto-dominant NE?
Yes
No
predict a$_2$
predict a$_3$

Figure 4: Best 2-split decision tree for the algorithmically designed games.

Motivated by this tree, we introduce the following rule:

Pareto-Dominant Nash Equilibrium (PDNE). We predict at random from the set of row player actions a_i such that (a_i, a_j) is a pure-strategy Nash equilibrium whose payoffs Pareto-dominate the payoffs in every other pure-strategy Nash equilibrium. If this set is empty, we predict an action uniformly at random.

This PDNE rule substantially outperforms level-1(α) on the algorithmically generated games, achieving an acccuracy of 0.65 and completeness of 48% (compare to 0.38 and 7%). It does not outperform level-1(α) on the set of all games, where it achieves an accuracy of 0.56 and completeness of 34% (compare to 0.68 and 52%).[37]

The differences in play and model fit across data sets highlights the importance of the experimental-design process for the resulting findings. It also raises the question of which distributions over games are the most economically relevant. We find this question difficult to answer, in part because 3×3 games are themselves a simplified representation of real-world interactions. In what follows we will report results on the combined set of all games.

Note also that while PDNE and level-1(α) respectively achieve accuracies of 0.56 and 0.68 on our full data set, the best decision tree achieves an accuracy of 0.70. This increased accuracy suggests that there is additional structure to discover. One possibility is that there are regularities beyond PDNE and level-1(α), but another possibility is that PDNE and level-1(α) are good predictors of play in different

[37] Note that the differences in the performance of PDNE across these data sets is not simply because there are more Pareto-Dominant NE in the algorithmically-generated games. In fact, the fraction of Pareto-Dominant NE is largest in the set of random games (70%), and comparable in the laboratory games (52%) and the algorithmically designed games (59%).

games, so that neither model on its own performs well on our aggregate data set. Table 6 provides evidence supporting the second hypothesis:

Level-1(α) PDNE	Right	Wrong
Right	155	115
Wrong	175	41

Table 6: There are many games where level-1(α) predicts correctly while PDNE does not, and vice versa.

This suggests that, if we can predict when PDNE is a good model of play and when level-1(α) is better, we can improve upon both component models. We explore this idea in the next section.

6 Hybrid Models

There are many possible ways combine level-1(α) and PDNE to make predictions. Perhaps the simplest is to use a "lexicographic rule" that predicts the PDNE when a PDNE exists and otherwise uses level-1(α). This rule improves on both of its components in the overall data set, due to its superior performance on the algorithmically generated games, but does worse than level-1(α) both on the lab games (which may have been designed to elicit non-Nash play) and also on the random games.[38]

We would like to find a better way to combine these two prediction rules, and moreover do so in a way that can be extended to combine arbitrary classification rules. To this end, we take the following approach: First, we estimate each model on the training data (if it has free parameters—note that PDNE does not). We then use the estimated model to predict the modal action in each game in the training data. Thus for each model we have a binary vector of accuracy outcomes ("correctly predicted" versus "incorrectly predicted") across the games in the training data. We then fit a regression tree[39] to predict a probability with which the model chooses the the correct action, based on the feature set described above in Section 4.2 (and reported in Section A.2). This returns, for each model, an algorithm that maps game features into a probability that the model's prediction is correct.

On out-of-sample games, we use the "accuracy prediction algorithms" to predict the probability of an accurate prediction under either model. We then select the

[38] The lexicographic rule has accuracy 0.72 on the combined data, 0.52 on the lab games, and 0.71 on the random games.

[39] *Regression trees* are decision trees where the predicted outcome is a real number.

model with the larger (predicted) accuracy, and use that model to predict the modal action. This procedure is depicted in Figure 6:

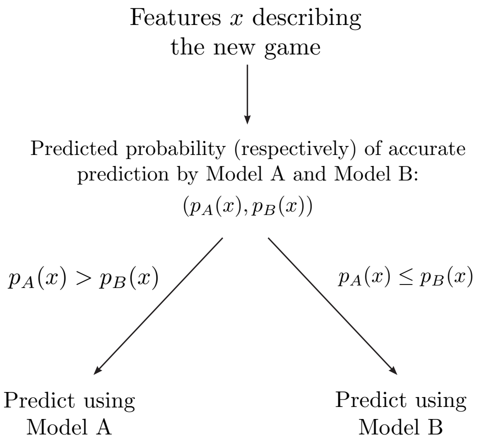

Features x describing the new game
Predicted probability (respectively) of accurate prediction by Model A and Model B:
$(p_A(x), p_B(x))$
$p_A(x) > p_B(x)$
$p_A(x) \leq p_B(x)$
Predict using Model A
Predict using Model B

Figure 5: Hybrid models

This model selection procedure is a form of “mixtures of experts” (Masoudnia and Ebrahimpour, 2014). There are many possible ways to use game features, and we do not claim that ours is optimal. We chose it because it is relatively simple to implement and interpret. Even with this simple formulation, we were able to achieve notable improvements in performance, but more sophisticated methods might do better still.

Hybrid models are closely related to *model trees* (Quinlan, 1992; Landwehr, Hall and Frank, 2005), which are decision trees whose branches lead to linear (or logistic) regression models. The hybrid models we use similarly embed models at the nodes of a decision tree, but our component models are simple economic/behavioral models. Our procedure is also related to the literature on *forecast combinations* (e.g. Timmermann (2006)), where different structural models are averaged using weights determined according to past performance.[40],[41]

In general, the regression trees used to predict the accuracies of the two component models can vary across folds of cross-validation. But for our hybrid model combining level-1(α) with PDNE, the best-cross validated prediction trees (reported

[40] For example, the weights might correspond to posterior probabilities as in Bayesian model averaging.

[41] For example, Negro, Hasegawa and Schorfheide (2016) combines different dynamic stochastic general equilibrium (DSGE) models for improvements in forecasting real GDP growth. Our work differs in that we assign a single model to each game, using properties of the game itself to determine this assignment, rather than assigning the same average to all of the games

in Appendix E.2.1) have only two splits each, and are the same on 9 of the 10 folds. The resulting rule for model assignment is depicted in Figure 6 below:

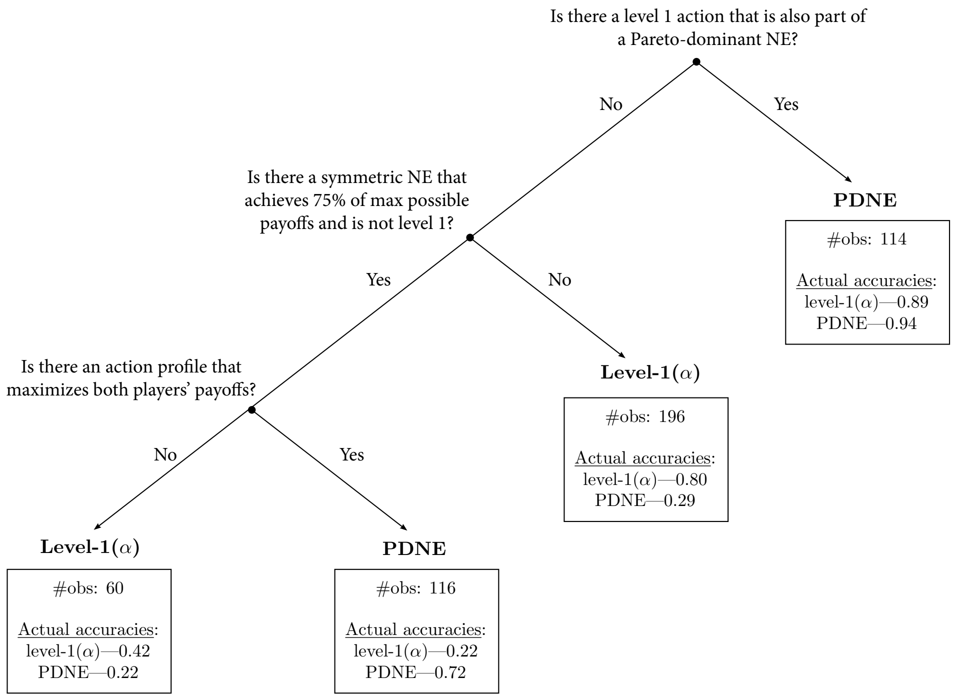


Figure 6: Assignment of games to models

This tree partitions the space of games into four classes. In two of these classes, the tree predicts a PDNE.[42] In the other two classes, the tree uses level-1(α). Of the games assigned to level-1(α), 74 games have a PDNE, so the the tree does not always pick the PDNE model even when a Pareto-dominant Nash equilibrium exists.[43]

The specific feature of whether the symmetric NE achieves 75% of the max possible sum of player payoffs was chosen somewhat arbitrarily, but the prediction accuracy of the hybrid model is essentially unchanged when we replace 75% with 70%

[42] Note that when there is a profile that maximizes both player's payoffs, it is guaranteed to be a PDNE, so the tree only uses PDNE to make its prediction when there is a PDNE to predict. Note also that a unique Nash equilibrium is by definition a PDNE.

[43] We do not include the source of the game—lab-designed, algorithmically-designed, or randomly-generated—as a feature for the tree to use. Nevertheless it is possible that other features proxy for this, and the tree assigns games to models based on which data set the game belongs to. This turns out not to be the case: of the games assigned to PDNE, 13 come from the lab data set, 101 from the randomly-generated games, and 116 from the algorithmically-generated games.

or 80%. (The accuracy is the same up to two significant figures.) Our qualitative takeaway from this decision tree is that the important feature is whether there is a symmetric NE with "high" payoffs that does not include the level-1 action.

We report the accuracies of PDNE and level-1(α) on each of these four classes in Figure 6.[44] By inspecting the tree, we see that only a little accuracy is gained by using PDNE in the 114 games with a level-1 action that is part of a Pareto-dominant Nash equilibrium, as here both PDNE and level-1(α) predict quite well.[45] The gains from using PDNE are much greater in the other 116 games where it is used. In these games, PDNE is right 72% of the time while level-1(α) is worse than guessing at random. These games all contain a very good Nash equilibrium (Pareto-dominant, symmetric, yields maximal payoffs for both players) that does not correspond to the level-1 action. For example:

	a_1	a_2	a_3	Frequency of Play
a_1	$90, 90$	$30, 80$	$45, 30$	72%
a_2	$80, 30$	$55, 55$	$37, 5$	28%
a_3	$30, 45$	$5, 37$	$70, 70$	0%

In this game, action a_2 is level-1(α) but the action profile (a_1, a_1) is a Pareto-dominant Nash equilibrium and also maximizes both player's payoffs. We expect that PDNE will be a better prediction than level-1(α) in similar games beyond our data set.

Notice that the hybrid model is *not* guaranteed to improve upon the (out-of-sample) predictive performance of either base model, as it runs the risk of overfitting due to its greater complexity. Nevertheless, we find that "level-1(α) + PDNE" substantially improves upon the performance of both base models in the data set of all games. Moreover, for the lab data we used to begin our analysis, we find that the hybrid model weakly improves upon the level-1(α) model as well.[46]

[44] We set $\alpha = 0.41$, which is the estimate on the full data set. In practice the value of α fluctuates across the different choices of training data, so the prediction accuracies reported above are not exact.

[45] Note that there is a gap between the feature that describes whether the level-1 action is part of the Pareto-dominant Nash equilibrium and this hybrid model, because the latter predicts the level-1(α) action. Since the level-1(α) action and the level-1 action are not always the same, there are multiple instances in which the level-1(α) prediction is wrong even though the level-1 action is part of the unique Pareto-dominant Nash equilibrium.

[46] The hybrid model also outperforms both component models in the set of algorithmically generated games. The hybrid model does not improve on level-1(α) on the random games where level-1(α) already achieves a predictive accuracy of 91%.

	All Games		Lab Games	
	Accuracy	Completeness	Accuracy	Completeness
Guess at random	0.33	0%	0.33	0%
PDNE	0.56	34%	0.38	7%
Level-1(α)	0.68	52%	0.79	69%
	(0.02)		(0.02)	
Level-1(α) + PDNE	0.79	69%	0.82	73%
	(0.03)		(0.03)	
Ideal prediction	1	100%	1	100%

Table 7: The level-1(α) + PDNE hybrid model improves upon the performance of both component models.

Our analysis above demonstrates that we can improve predictions by combining two interpretable models. In principle, hybrid models can be built from a wide array of component models. For example, instead of combining two behavioral/economic models as we do here, we could combine a model such as level-1(α) with an algorithmic model, such as lasso or logistic regression. This kind of model would further blur the distinction between "behavioral" and "algorithmic" approaches. For more complex problem domains, such as predicting the distribution of play, we might consider hybrid models that combine two different structural models of play—for example, PCHM and a mixture-model of level-k types (as in Costa-Gomes, Crawford and Broseta (2001)). Yet another possibility is to combine a model based on the game matrix (as all of the approaches discussed so far are) with more "unconventional" models that use auxiliary data, such as crowd predictions. We leave pursuit of these other interesting hybrid models to future work.

7 Conclusion

This paper uses approaches from machine learning algorithms not only to improve predictions of initial play, but also to improve our understanding of it. We use these tools to develop simple and portable improvements on existing models.

One way we improve existing models is by studying games where machine learning algorithms predict well, but existing models do not. In Section 3, we showed how this exercise helped us realize that adding a risk aversion parameter to the level-1 model generates better out-of-sample predictions of the most likely action. We developed even better predictions by generating data on new games where level-1(α) performs poorly, identifying a simple alternative (PDNE) that does better on this new domain, and then using a hybrid model that learns which of the two sub-models should be applied to a given game.

Along with papers such as Leyton-Brown and Wright (2014), these results show how a combination of machine learning and behavioral models can improve the prediction and understanding of play in games. These methods are not special to the problem of predicting initial play in matrix games, so we expect that the proposed approaches can be used to improve prediction in other domains, both in game theory (e.g. the effect of learning and feedback on play in static games, or initial play in extensive-form games) and in other areas of economics such as decision theory, as well as in social science more generally.

We offer a few final comments on interpretations of our results as well as some potential future directions:

1) Although we studied a relatively large and diverse set of games compared to the literature, we restricted attention to the relatively simple setting of 3×3 matrix games. When the test set of games is small or less varied in structure, simple low-parameter models such as level-1(α) have an advantage over models with more parameters, which may overfit. In settings with more diverse behavior, richer models may perform better, just as the hybrid models improved on the level-1(α) model in predicting play in the algorithmically generated games.

2) Our finding that the performance ranking of our different models depends on which data set we examine raises an important caution about generalizing from experiments that were designed to highlight certain behaviors or to make specific points.

3) We did not use subject identifiers, so we could not predict or differentiate across the behavior of different subjects. Another interesting direction would be to use similar methods to categorize subjects (instead of games), assigning different groups of subjects different models of play as in Fragiadakis, Knoepfle and Niederle (2016).

4) We used hand-crafted features to train the rule for selecting between models. It is possible to simultaneously learn the prediction rule and the feature representation of the game, as in the deep learning methods of Hartford, Wright and Leyton-Brown (2016), but at present these techniques do not yield interpretable features.

5) 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 predictions of medium and long run behaviors. We leave this direction for subsequent work.

References

Breiman, Leo. 1994. "Bagging Predictors." University of California. 30

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. 9

Camerer, Colin, Teck-Hua Ho, and Juin-Kuan Chong. 2004. "A Cognitive Hierarchy Model of Games." The Quarterly Journal of Economics. 1, 2, 2.2, 2.2

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, 8

Costa-Gomes, M., V. Crawford, and B. Broseta. 2001. "Cognition and behavior in normal-form games: an experimental study." Econometrica. 2.2, 6

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

Crawford, Vincent P, and Nagore Iriberri. 2007. "Level-k auctions: can a nonequilibrium model of strategic thinking explain the winner's curse and overbidding in private-value auctions?" Econometrica, 75(6): 1721–1770. 7

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

Fragiadakis, Daniel E., Daniel T. Knoepfle, and Muriel Niederle. 2016. "Who is Strategic?" Working Paper. 1.1, 7

Fudenberg, Drew, and David K Levine. 2006. "A Dual-Self Model of Impulse Control." American Economic Review, 96(5): 1449–1476. 22

Fudenberg, Drew, and David K Levine. 2011. "Risk, delay, and convex self-control costs." American Economic Journal: Microeconomics, 3(3): 34–68. 22

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

Goodfellow, Ian, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. "Generative Adversarial Networks." NIPS. 33

Hartford, Jason, James Wright, and Kevin Leyton-Brown. 2016. "Deep Learning for Predicting Human Strategic Behavior." 1.1, 7

Haruvy, E., and D. Stahl. 2007. "Equilibrium selection and bounded rationality in symmetric normal-form games." Journal of Economic Behavior and Organization. 3.1

Haruvy, E., D. Stahl, and P. Wilson. 2001. "Modeling and testing for heterogeneity in observed strategic behavior." Review of Economic and Statistics. 3.1

Hastie, Trevor, Robert Tibshirani, and Jerome Friedman. 2009. The Elements of Statistical Learning. Springer. 2.3

Holt, Charles A, and Susan K Laury. 2002. "Risk aversion and incentive effects." American economic review, 92(5): 1644–1655. 22

Huang, Ling, Anthony D. Joseph, Blaine Nelson, Benjamin Rubinstein, and J.D. Tygar. 2011. "Adversarial Machine Learning." Proceedings of 4th ACM Workshop on Artificial Intelligence and Security. 33

Khaw, Mel Win, Ziang Li, and Michael Woodford. 2018. "Temporal discounting and search habits: evidence for a task-dependent relationship." Frontiers in Psychology, 9: 2102. 22

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, D.2

Landwehr, Niels, Mark Hall, and Eibe Frank. 2005. “Logistic Model Trees.” Journal of Machine Learning. 1.1, 6

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, 8, 2.2, 3.1, 7

Masoudnia, Saeed, and Reza Ebrahimpour. 2014. “Mixture of experts: a literature survey.” Artificial Intelligence Review, 42(2): 275–293. 1.1, 6

Mullinix, Kevin J, Thomas J Leeper, James N Druckman, and Jeremy Freese. 2015. “The generalizability of survey experiments.” Journal of Experimental Political Science, 2(2): 109–138. 28

Negro, Marco Del, Raiden B. Hasegawa, and Frank Schorfheide. 2016. “Dynamic prediction pools: An investigation of financial frictions and forecasting performance.” Journal of Econometrics. 41

Paolacci, Gabriele, Jesse Chandler, and Panagiotis G Ipeirotis. 2010. “Running experiments on Amazon Mechanical Turk.” Judgment and Decision Making, 5(5): 411–419. 28

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

Quinlan, J.R. 1992. “Learning with Continuous Classes.” Proceedings AI. 1.1, 6

Rabin, Matthew. 2000. “Risk Aversion and Expected-utility Theory: A Calibration Theorem.” Econometrica, 68(5): 1281–1292. 3.2

Rand, David G. 2012. “The promise of Mechanical Turk: How online labor markets can help theorists run behavioral experiments.” Journal of theoretical biology, 299: 172–179. 28

Rogers, B.W., T.R. Palfrey, and C.F. Camerer. 2009. “Heterogeneous quantal response equilibrium and cognitive hierarchies.” Journal of Economic Theory. 3.1

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

Shefrin, Hersh M, and Richard H Thaler. 1988. “The behavioral life-cycle hypothesis.” Economic inquiry, 26(4): 609–643. 22

Snowberg, Erik, and Leeat Yariv. 2018. “Testing the waters: Behavior across participant pools.” National Bureau of Economic Research. 28

Stahl, Dale O. 2000. “Rule learning in symmetric normal-form games: theory and evidence.” Games and Economic Behavior, 32(1): 105–138. 4.1

Stahl, Dale O., and Paul W. Wilson. 1995. “On players’ models of other players: Theory and experimental evidence.” Games and Economic Behavior. 2.2, 2.2, 3.1

Stahl, D., and E. Haruvy. 2008. “Level-n bounded rationality and dominated strategies in normal-form games.” Journal of Economic Behavior and Organization. 3.1

Stahl, D., and P. Wilson. 1994. “Experimental evidence on players’ models of other players.” Journal of Economic Behavior and Organization. 1, 2.2, 3.1

Thomas, Kyle A, and Scott Clifford. 2017. “Validity and Mechanical Turk: An assessment of exclusion methods and interactive experiments.” Computers in Human Behavior, 77: 184–197. 28

Timmermann, Allan. 2006. “Forecast Combinations.” Handbook of Economic Forecasting. 6

Appendix

A Feature Sets

A.1 Features Describing Specific Actions

For each row player 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*.
- is part of a *Pareto-dominant pure-strategy Nash equilibrium* (its payoffs Pareto-dominate the payoffs in every other pure-strategy Nash equilibrium)
- is part of an action profile that is *Pareto-undominated*
- is *"max-max"*: a_i is played in the profile that maximizes the row player's payoff
- is *"max-min"*: a_i maximizes the minimum, over the column player's actions, of the row player's payoff
- is *level k* for each $k = 1, 2, 3$
- is part of a *"good" Nash equilibrium*, meaning that the sum of player payoffs in this Nash equilibrium is at least $3/4$ of the largest possible player payoff sum
- is part of a *symmetric good Nash equilibrium*

Additionally, we include a *score* feature for each action, which is the number of the following properties that it satisfies: part of a Nash equilibrium, level-1, level-2, level-3, level-4, level-5, level-6, level-7, part of a Pareto-dominant Nash equilibrium, part of an action profile that maximizes the sum of player payoffs.

A.2 Features Describing Properties of the Game

We define features for the following properties of the payoff matrix:

- number of pure strategy Nash equilibria
- number of actions that survive iterated elimination of strictly dominated pure strategies
- indicator for whether there is at least one action that is strictly dominated
- number of strictly dominated actions
- existence of an action that simultaneously maximizes both players' payoffs
- number of different actions that yield the maximal row player payoff (for some column player action)
- number of different actions that are part of an action profile that maximizes the sum of player payoffs
- number of different actions that are part of a Pareto-undominated Nash equilibrium
- number of different level-1 actions

- number of actions that are simultaneously level-1, achieve the highest possible row-player payoff (for some column player action), and achieve the highest possible sum of player payoffs (for some column player action)
- number of actions that are level-k for some $k \in \{1, 2, \ldots, 7\}$
- indicator for whether there is some row player payoff that is 100
- number of actions that yield a row player payoff of 100
- indicator for whether some level-1 action is also level 2
- indicator for whether some level-1 action also yields the largest possible row player payoff (*max-max*)
- indicator for whether some level-1 action maximizes the sum of player payoffs (*max-sum*)
- indicator for whether some level-1 action is *max-max* and also *max-sum*
- largest number n where some row player action satisfies n of the following properties: level-1, *max-max*, *max-sum*
- indicator for whether some level-1 action is part of a Pareto-dominant pure-strategy Nash equilibrium
- indicator for whether some level-1 action is also part of a pure-strategy Nash equilibrium
- indicator for whether there is a symmetric pure-strategy Nash equilibrium
- indicator for whether some Nash equilibrium achieves 75% of the largest possible sum of player payoffs[47]
- indicator for whether some Nash equilibrium achieves 75% of the largest possible sum of player payoffs, and includes the level-1 row player action
- indicator for whether some Nash equilibrium achieves 75% of the largest possible sum of player payoffs, and does not include the level-1 row player action
- indicator for whether some Nash equilibrium achieves 75% of the largest possible sum of player payoffs, and does not include any level-k row player action
- indicator for whether some symmetric Nash equilibrium achieves 75% of the largest possible sum of player payoffs
- indicator for whether some symmetric Nash equilibrium achieves 75% of the largest possible sum of player payoffs, and includes the level-1 row player action
- indicator for whether some symmetric Nash equilibrium achieves 75% of the largest possible sum of player payoffs, and does not include the level-1 row player action
- indicator for whether some symmetric Nash equilibrium achieves 75% of the largest possible sum of player payoffs, and does not include any level-k row player action
- indicator for whether the best sum of player payoffs in the matrix exceeds—by at least 20% of the max row player payoff in the matrix—the best payoff sum

[47] We note that in this feature and the others below using %'s, the % was chosen somewhat arbitrarily; future work may consider estimation of the optimal choice of what % to use.

when the row player chooses a level-k action.

- indicator for whether the best sum of player payoffs in the matrix exceeds—by at least 40% of the max row player payoff in the matrix—the best payoff sum when the row player chooses a level-k action.
- indicator for whether the best sum of player payoffs in the matrix exceeds—by at least 60% of the max row player payoff in the matrix—the best payoff sum when the row player chooses a level-k action.
- indicator for whether the *row sum gap*, defined as the difference between the sum of possible row player payoffs when the row player chooses his level-1 action (and the column player's action is allowed to vary), and the next highest row sum, is at least 30% of the max row player payoff in the matrix

B Other Prediction Algorithms

Here we consider more sophisticated algorithms for predicting the modal action in the set of all games. We 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 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.

	Lab Games Only		All Games	
	Accuracy	Completeness	Accuracy	Completeness
Guessing at Random	0.33	0%	0.33	0%
Decision Tree	0.77	66%	0.70	55%
	(0.04)		(0.03)	
Random Forest	0.74	61%	0.72	61%
	(0.03)		(0.02)	
2-layer Neural Net	0.76	64%	0.77	69%
	(0.02)		(0.01)	
Ideal prediction	1	100%	1	100%

The alternative algorithms underperform the single decision tree for predicting modal play in the lab games. They improve upon the single decision tree on our set of all games, but do not improve upon the performance of the hybrid model built on level-1(α) and PDNE. The outputs of these alternative algorithms are substantially less interpretable than the single decision tree, so we do not focus on them in this paper.

C Robustness Check: Predicting Each Instance of Play

As a robustness check, we repeat our main analysis on the full set of games for a related prediction task. Instead of predicting the modal action, we predict a given instance of play. For this problem, a prediction rule is still a map $f : G \to A_1$ from games to row player actions, but now each observation is a pair (g_i, a_i) where g_i is the game played in instance i and a_i is the action chosen in that instance of play. Thus we have many repetitions of each game corresponding to the different subjects we observe playing those games. Given a set of instances of play $\{(g_i, a_i)\}$, we again evaluate accuracy using the correct classification rate.

The naive rule is guessing at random, and again yields an expected accuracy of 1/3. The ideal prediction rule assigns the observed modal action to each game (as before), but now has an accuracy far from 1, since different subjects play different actions in the same game. Table 8 reports prediction accuracies and completeness measures on our set of all games. The ranking is qualitatively unchanged from the main text.

	Accuracy	Completeness
Guess at random	0.333	0%
Level-1	0.431 (0.01)	31%
Level-1(α)	0.449 (0.00)	37%
PDNE	0.552 (0.02)	39%
Decision Tree	0.563 (0.01)	70%
Ideal prediction	0.645 (< 0.01)	100%

Table 8: Hybrid models also improve predictive accuracy in predicting each instance of play.

D Alternative Ideal Benchmarks

In the main text we evaluated completeness relative to predicting the actual observed modal action in each game. This ideal benchmark is not attainable, and thus we under-estimate the completeness of the models we consider. Below we present completeness measures relative to two alternative ideal benchmarks. These completeness measures are not very different from the main text, but do suggest that some of the performances are closer to complete than the main text suggests. For example, the best completeness measure for predicting the modal action in the set of all games is 69% in the main text, but 78% and 92% relative to the two benchmarks we consider in this section.

D.1 Bootstrapped Benchmark

We construct a bootstrapped prediction benchmark as follows. First, we assign the observed modal action a_i to each game g_i . We test this prediction rule on bootstrap-resamples of our data. That is, for each game g_i, we sample n_i times with replacement from the empirical distribution of actions observed in that game, where n_i is the number of observations we have for that game. Our test data is then $\{(g_i, \hat{a}_i)\}$ where $\hat{a}_i$ is the modal resampled action in game g_i. We repeated this procedure 100 times and report the average prediction accuracy, along with the standard deviation of these prediction accuracies.

	Lab Games		Random Games		Algo Games		All Games	
	Acc	Complete	Acc	Complete	Acc	Complete	Acc	Complete
Guess at random	0.33	0%	0.33	0%	0.33	0%	0.33	0%
PDNE	0.38	8%	0.55	37%	0.65	58%	0.56	34%
Uniform Nash	0.42	15%	0.57	40%	0.43	18%	0.49	27%
	(0.03)		(0.03)		(0.03)		(0.02)	
Level-1	0.72	63%	0.87	79%	0.36	5%	0.64	53%
Level-1(α)	0.79	74%	0.91	97%	0.38	9%	0.68	59%
Decision Tree	0.77	71%	0.86	88%	0.67	62%	0.70	63%
	(0.04)		(0.02)		(0.03)		(0.03)	
Bootstrap	0.95	100%	0.93	100%	0.88	100%	0.92	100%
	(0.02)		(0.01)		(0.02)		(0.01)	

Table 9: Compare the lab game results to Table 2, the random game results to Table 4, and the final two columns to Table 5.

	Accuracy	Completeness
Guess at random	0.33	0%
Level-1(α)	0.68	59%
	(0.02)	
PDNE	0.56	39%
Level-1(α) + PDNE	0.79	78%
	(0.03)	
Bootstrap	0.92	100%
	(0.01)	

Table 10: Compare to Table 7

D.2 Table Lookup Benchmark

Following Kleinberg, Liang and Mullainathan (2017) we consider a “table lookup” benchmark, defined as follows: We divide the observations of play for each game g_i into three folds and randomly select two of these folds for training. Based on this data, we learn the prediction rule that assigns the modal action to each game in the training data, and use this rule to predict the modal action in the remaining fold. We report the average prediction accuracy across the three choices of test fold in Table 11. Although this approach will converge to the idealized benchmark of 1 given enough data, since we use only a limited number of observations, it is in fact possible to beat the table lookup benchmark, and indeed our model beats the benchmark for the set of randomly-generated games.

	Lab Games		Random Games		Algo Games		All Games	
	Acc	Complete	Acc	Complete	Acc	Complete	Acc	Complete
Guess at random	0.33	0%	0.33	0%	0.33	0%	0.33	0%
PDNE	0.38	9%	0.55	42%	0.65	76%	0.56	46%
Uniform Nash	0.42	16%	0.57	46%	0.43	24%	0.49	32%
	(0.03)		(0.03)		(0.03)		(0.02)	
Level-1	0.72	68%	0.87	104%	0.36	7%	0.64	62%
Level-1(α)	0.79	81%	0.91	112%	0.38	9%	0.66	66%
Decision Tree	0.77	77%	0.86	102%	0.67	81%	0.65	64%
	(0.04)		(0.02)		(0.03)		(0.03)	
Table Lookup	0.90	100%	0.85	100%	0.75	100%	0.83	100%
	(0.01)		(0.02)		(0.03)		(0.03)	

Table 11: Compare the lab game results to Table 2, the random game results to Table 4, and the final two columns to Table 5.

	Accuracy	Completeness
Guess at random	0.33	0%
Level-1(α)	0.68	70%
	(0.01)	
PDNE	0.56	46%
Level-1(α) + PDNE	0.79	92%
	(0.03)	
Table Lookup	0.83	100%
	(0.01)	

Table 12: Compare to Table 7

E Decision Trees

E.1 Prediction of Modal Action in Lab Games

The decision tree with lowest out-of-sample error is shown below:[48],[49]

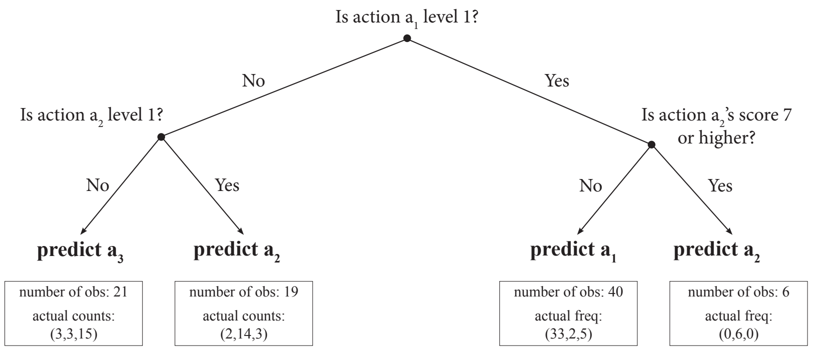


Figure 7: Best decision tree for predicting the realized action in lab data.

This tree appends a single additional criterion to the level 1 model: it agrees with the level-1 model unless there are sufficiently many other reasons to choose a_2. (The score variable, described in Section A.1 ranges from zero to 10, and a_2 is predicted if its score is at least seven.) In that case, even if action a_1 is level-1, action a_2 is predicted instead.

Note that features are indexed to labelled actions, so the tree does not need to treat them symmetrically. The favored treatment of a_1 here reflects the fact that this was the most common modal action in the lab data.

E.2 Used in Hybrid Models

E.2.1 Supplementary Material to Section 6

Below we report the trees used to predict accuracy of the level-1(α) prediction (Figure 8) and accuracy of the PDNE prediction (Figure 9) in the level-1(α) + PDNE hybrid model.

[48] 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.

[49] The tree in Figure 7 was produced for eight of the ten training sets.

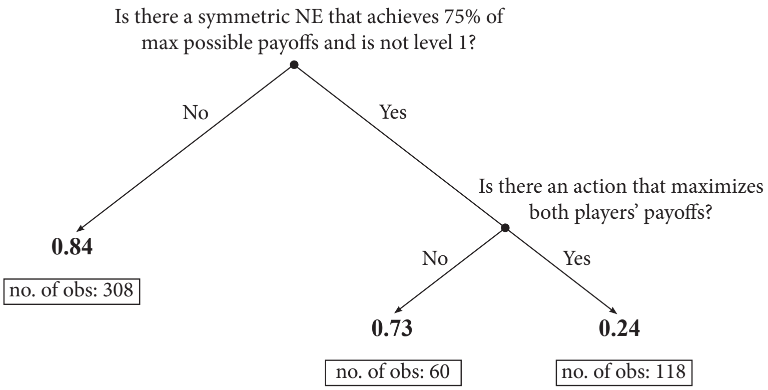

Is there a symmetric NE that achieves 75% of max possible payoffs and is not level 1?
No
Yes
0.84
no. of obs: 308
Is there an action that maximizes both players' payoffs?
No
Yes
0.73
no. of obs: 60
0.24
no. of obs: 118

Figure 8: Predicted probability that the level-1(α) prediction is correct in **bold**.

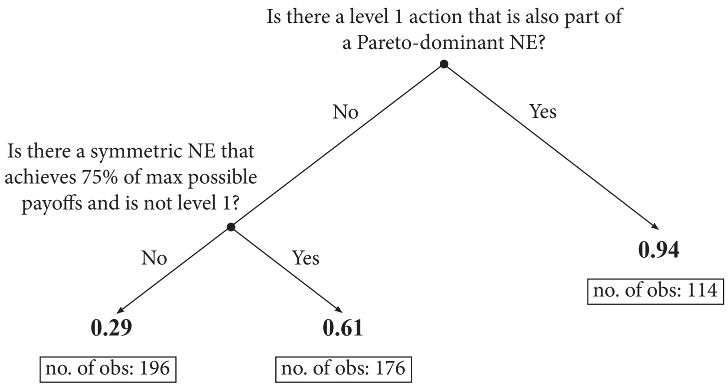

Is there a level 1 action that is also part of a Pareto-dominant NE?
No
Yes
Is there a symmetric NE that achieves 75% of max possible payoffs and is not level 1?
No
Yes
0.29
no. of obs: 196
0.61
no. of obs: 176
0.94
no. of obs: 114

Figure 9: Predicted probability that the PDNE prediction is correct in **bold**.

The first tree predicts the probability that whether the level-1(α) model will choose the modal action. For example, if the game does not have a symmetric NE with high payoffs (75% of max possible) that does not include the level-1 action, then the level-1(α) action is predicted to be modal 84% of the time.[50] The level-1(α) model is predicted to perform worst when there is a symmetric NE that maximizes both players' payoffs but does not contain the level-1 action: In this case, the level-1(α) action is predicted to be correct only 24% of the time.

The second tree predicts the probability that the PDNE prediction will be correct. The model is predicted to perform well when the Pareto-dominant NE includes

[50] Roughly this means that in 84% of games in the training sample with this property, the level-1(α) action was modal.

the level-1 action, and also when there is a symmetric NE that achieves high payoffs (this is almost always a Pareto-dominant NE in our data). We do not know whether this is true more generally or whether it is a special feature of our set of games.

E.3 Lab Games Only

We report below the analogue of Figure 6—which chooses between the level-1(α) model and PDNE—for the data set consisting only of the lab games.

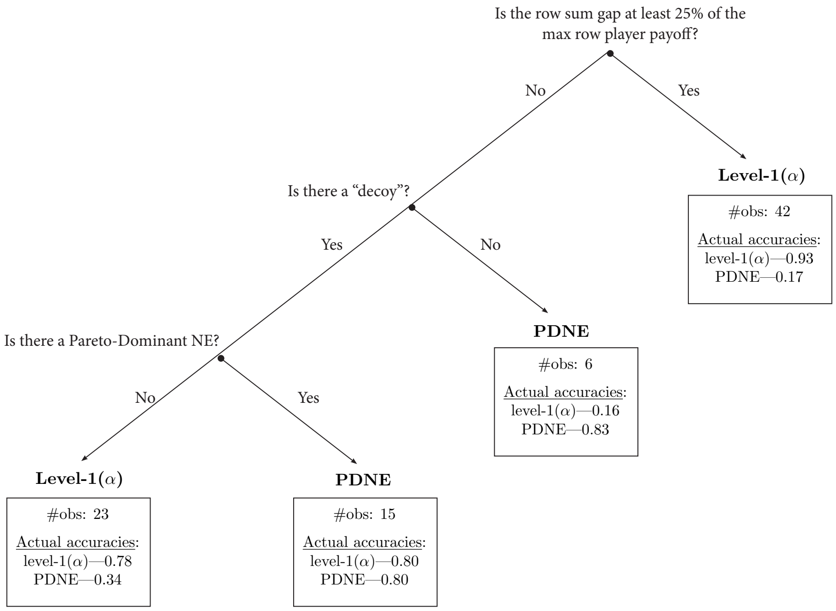


Figure 10: Assignment of games to level-1(α) or PDNE (lab games only)—compare to Figure 6.

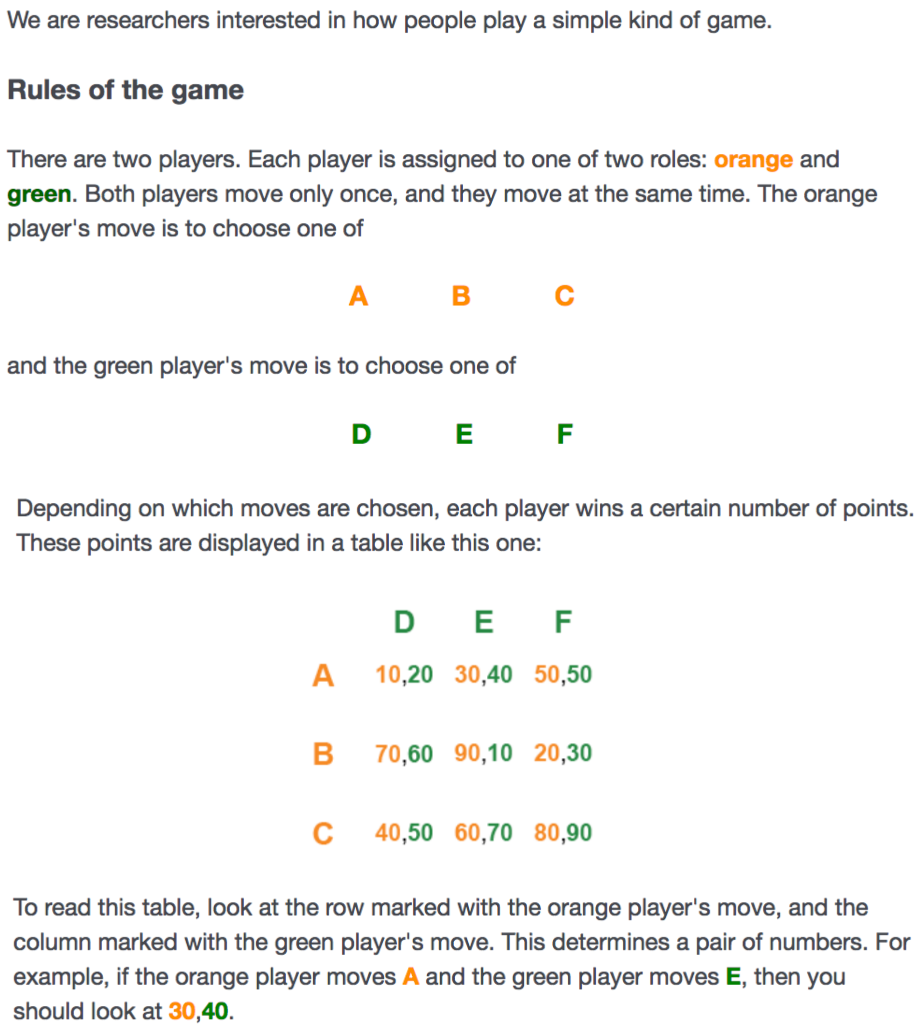
Above, the feature "is there a decoy" refers to the indicator for whether the best sum of player payoffs in the matrix exceeds—by at least 60% of the max row player payoff in the matrix—the best payoff sum when the row player chooses a level-k action.

For Online Publication

F Experimental Instructions

The instructions provided to Mechanical Turk subjects in the experiment described in Section 4 are reproduced below. With a few exceptions, instructions that were repeated across these experiments are only presented once.

F.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**

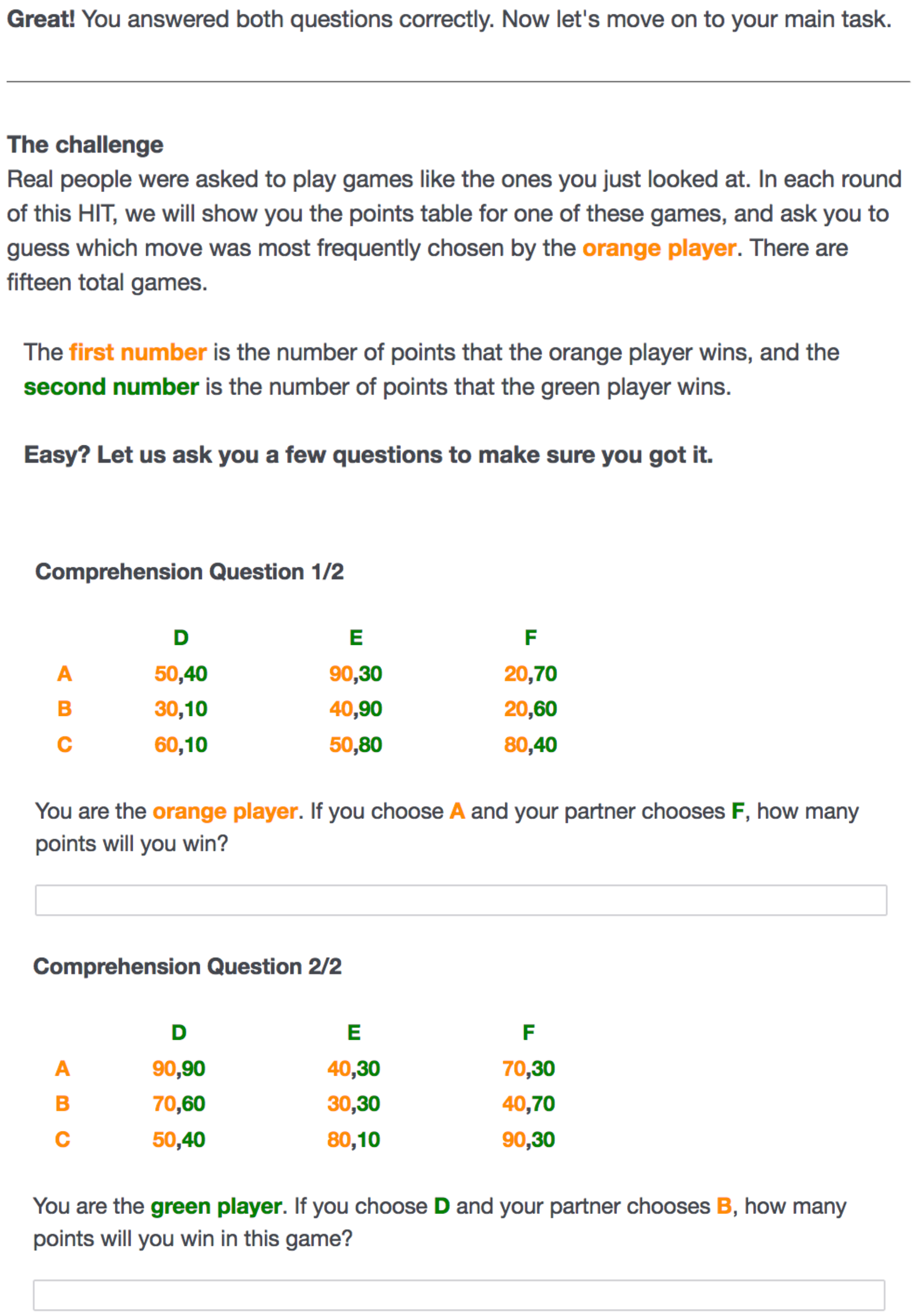
and the green player's move is to choose one of

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**.



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.

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

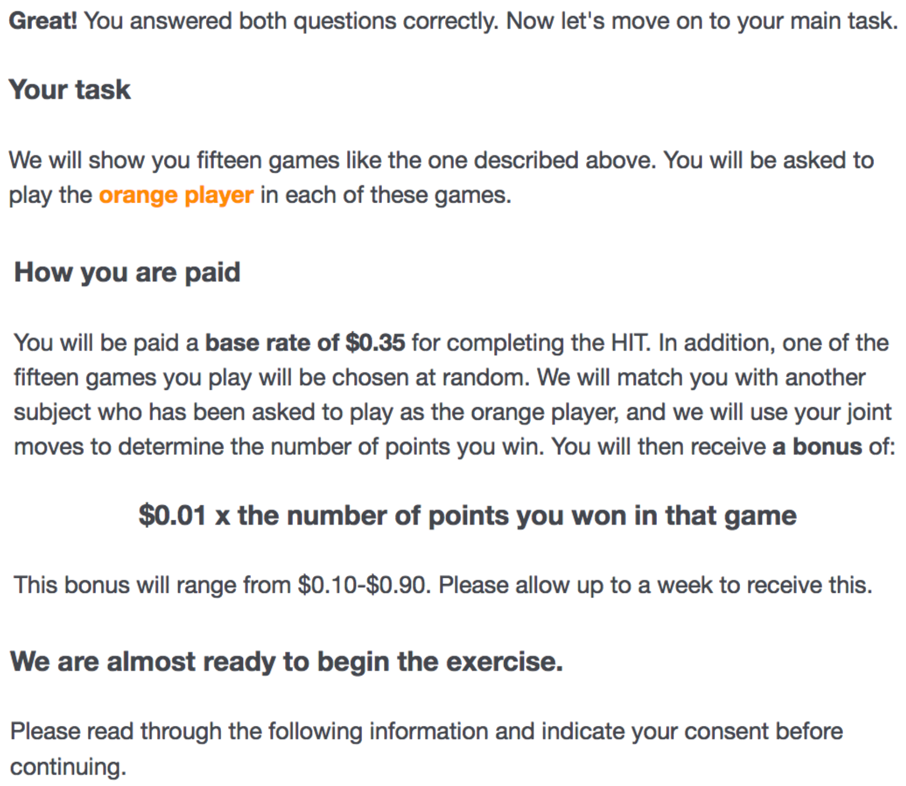
	D	E	F
A	50,40	90,30	20,70
B	30,10	40,90	20,60
C	60,10	50,80	80,40

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:

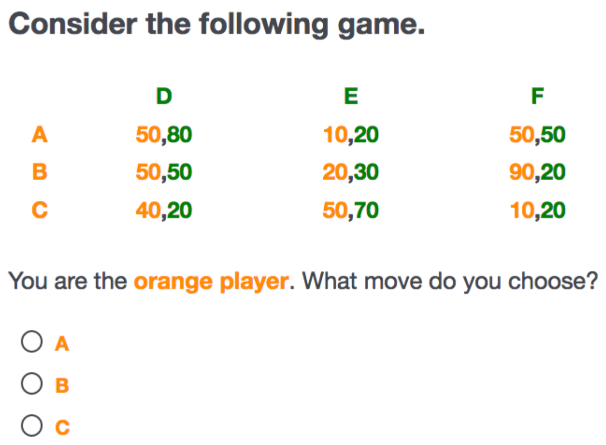
$0.01 x the number of points you won in that game

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.

F.2 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

G 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."