Games and Economic Behavior 76 (2012) 44–73

Contents lists available at SciVerse ScienceDirect

Games and Economic Behavior

www.elsevier.com/locate/geb

Learning in experimental 2 × 2 games ☆

Thorsten Chmura [a], Sebastian J. Goerg [b,c,*], Reinhard Selten [d]

[a] Nottingham University Business School, Nottingham, NG8 1BB, UK
[b] Max Planck Institute for Research on Collective Goods, 53113 Bonn, Germany
[c] Department of Economics, Florida State University, Tallahassee, FL 32306, USA
[d] BonnEconLab, Department of Economics, University of Bonn, 53113 Bonn, Germany

ARTICLE INFO

Article history:
Received 31 March 2011
Available online 30 June 2012

JEL classification:
C72
C91
C92

Keywords:
2 × 2 games
Experimental data
Learning
Impulse-matching
Self-tuning EWA
Reinforcement
Action-sampling

ABSTRACT

In this paper, we introduce two new learning models: action-sampling learning and impulse-matching learning. These two models, together with the models of self-tuning EWA and reinforcement learning, are applied to 12 different 2 × 2 games and their results are compared with the results from experimental data. We test whether the models are capable of replicating the aggregate distribution of behavior, as well as correctly predicting individuals' round-by-round behavior. Our results are two-fold: while the simulations with impulse-matching and action-sampling learning successfully replicate the experimental data on the aggregate level, individual behavior is best described by self-tuning EWA. Nevertheless, impulse-matching learning has the second-highest score for the individual data. In addition, only self-tuning EWA and impulse-matching learning lead to better round-by-round predictions than the aggregate frequencies, which means they adjust their predictions correctly over time.

1. Introduction

It is well known that rational learning, in the sense of Bayesian updating, leads to the stationary points of the Nash equilibrium (e.g., Kalai and Lehrer, 1993). But it is also known that actual human behavior does not necessarily converge to Nash equilibrium. In fact, a vast body of literature indicates situations in which standard theory does not perform as a good predictor for subjects' behavior in experiments (e.g., Brown and Rosenthal, 1990; Erev and Roth, 1998).

A recent publication by Selten and Chmura (2008) documents the predominance of behavioral stationary concepts regarding descriptive power. In the paper, the concepts of impulse-balance equilibrium (Selten and Chmura, 2008), payoff-sampling equilibrium (Osborne and Rubinstein, 1998), action-sampling equilibrium (Selten and Chmura, 2008), and quantal response equilibrium (McKelvey and Palfrey, 1995) outperform Nash equilibrium in describing the decisions of a population in twelve completely mixed 2 × 2 games. Moreover, payoff-sampling equilibrium and action-sampling equilibrium perform better than quantal response equilibrium does. In addition, the parameter-free concept of impulse-balance equilibrium performs equally well as the parametric concept of quantal response equilibrium.[1] Furthermore, Goerg and Selten (2009) show that the

☆ Financial support by the Deutsche Forschungsgemeinschaft and the German–Israeli Foundation for Scientific Research and Development is gratefully acknowledged. We would like to thank Emanuel Castillo and Christian Stracke for excellent research assistance. Christoph Engel, Andreas Nicklisch, and two anonymous referees provided very helpful comments.

* Corresponding author at: Department of Economics, Florida State University, 113 Collegiate Loop, Tallahassee, FL 32306-2180, USA.
E-mail address: goerg@coll.mpg.de (S.J. Goerg).

[1] For further discussions, please refer to Brunner et al. (2011) and Selten et al. (2011).

0899-8256/$ – see front matter
http://dx.doi.org/10.1016/j.geb.2012.06.007

advantage of impulse-balance equilibrium over Nash equilibrium is not limited to 2×2 games, but also present in cyclic duopoly games.

Presumably stationary behavior is a result of a learning process converging to a stationary distribution of actions for both players, which is, as the above studies demonstrate, not necessarily the Nash equilibrium. Therefore, constructing and testing simple learning models with the predicted stationary states of the better-performing concepts suggests itself. For the two behavioral stationary concepts of action-sampling equilibrium and impulse-balance equilibrium this is quite easy: both yield precise expression for stationary behavior.

The main purpose of this article is to introduce two new learning models, action-sampling learning and impulse-matching learning, which are based on the behavioral reasoning of action-sampling equilibrium and impulse-balance equilibrium, and test them in the environment of twelve repeated 2×2 games with mixed equilibria. Here, the learning rules have to meet two challenges: first, do they reproduce in simulations the aggregate behavior of a human population, and second, can they adequately describe the observed round-by-round behavior of a single individual? The difference between these two approaches is that in the simulations the learning curves of a population of agents is simulated, while for the analyses of the individual round-by-round behavior the actual experimental data from previous rounds is used to predict the next period.

For comparison, we include the models of reinforcement learning (Erev and Roth, 1998) and self-tuning experience-weighted attraction learning (Ho et al., 2007). We decided to compare our results with reinforcement learning, as it was the first model with application of rote learning to economics and it is the most cited learning model in economics. Self-tuning EWA was selected to cover a broader set of different learning variants, as it can describe weighted fictitious play, averaging reinforcement learning, and models in between.[2]

For the replication of the populations' behavior, we conduct simulations with the four learning models and the twelve 2×2 games experimentally investigated in Selten and Chmura (2008). The simulations replicate the exact situation of the experiments. In each simulation run, eight agents, four deciding as row players and four deciding as column players, are randomly matched in each round over 200 rounds. In each simulation run, one game is played and one learning model is applied. To judge the predictive power of the simulations on the population level, we compare the distribution of choices in the simulation runs with the experimental data from Selten and Chmura (2008).

To investigate how well the learning models predict individuals' round-by-round behavior, we separately evaluate the explanatory power of the learning models for each participant of the 2×2 experiments. For each of the 864 subjects, we compare the actual decision in every round with the decision predicted by the learning model given the subject's history. To judge the power of the learning models, we introduce three benchmarks which all learning models should beat. The first benchmark is the inertia rule, which predicts for each round the same choice as executed in the round before. The second benchmark is a random play with equal probability for each of the two decisions. In addition, if the learning theories describe subjects' behavior correctly over time, their predictions should be more accurate than the observed aggregate frequencies. Thus, as a more demanding benchmark, we include the empirical frequencies as a criterion.

As Erev et al. (2010) state, there are three obstacles for the learning literature: (1) small data sets, (2) problems of overfitting (Salmon, 2001; Hopkins, 2002), and (3) relatively small sets of models. We try to address these issues by (1) using the large data set of Selten and Chmura (2008) with twelve 2×2 games played by 864 subjects; (2) using theories with at most one parameter, adjusting the parameters over all games and applying only non-parametric analysis; and (3) applying and testing four different learning models. In fact, the results presented in this paper are the condensed summary of the analyses with 7 learning models. In addition to the already mentioned models, we introduce and test the concepts of payoff-sampling learning and impulse-balance learning. Because both concepts perform worse than the inertia benchmark and the random play benchmark, we do not cover these two learning models in more detail. More information about these two models can be found in Appendix A. To shed some additional light on the performance of self-tuning EWA, we include a non-parametric version of self-tuning EWA into our analyzes. We will refer to these results in the discussion. Additional information about the omitted concepts as well as the comparison of all 7 learning models can be found in Appendices A, B (Figs. 16 and 17), C (Figs. 18–23) and D (Tables 3–6).

Our results are twofold: our newly introduced models are able to capture the distribution of decisions of the experimental population much better in simulations than self-tuning EWA and reinforcement do, while self-tuning EWA describes the individual data in a much more accurate way. In the simulations, the learning models of impulse-matching learning and action-sampling learning have the smallest distance to the experimental data, while the concepts of self-tuning EWA and reinforcement learning have relatively high distances to the data. On the individual level with the actual individuals' histories, self-tuning EWA and impulse-matching have the highest scores. In addition, these two learning concepts are the only concepts that perform significantly better at describing individual round-by-round behavior than the overall empirical frequencies.

[2] We decided not to include a pure version of fictitious play in our analyses since a population of fictitious players would converge in the 2×2 games to the Nash equilibrium (Miyasawa, 1961, and Metrick and Polak, 1994), which is clearly outperformed by the stationary concepts of impulse-balance equilibrium and action-sampling equilibrium (Selten and Chmura, 2008). However, with action-sampling learning and self-tuning EWA, variants of fictitious play are included into our analyses. Furthermore, in this paper we solely focus on learning rules with at most one parameter. Thus, more elaborate versions of fictitious play with additional parameters like the three parameter model by Cheung and Friedman (1997) and the six parameter model of Chen et al. (2011) are ignored.

2. The learning models

In the following, we will introduce impulse-matching learning and action-sampling learning, which are based on the behavioral stationary concepts discussed in Selten and Chmura (2008). In addition to the new learning models, the more established concepts of reinforcement learning (c.p. Erev and Roth, 1998) and self-tuning EWA (Ho et al., 2007) are briefly explained.

Two of the discussed models, namely action-sampling learning and self-tuning EWA, are parametric concepts. In case of action-sampling learning, the parameter is the sample size. Self-tuning EWA is based on the multi-parametric concept of experience-weighted attraction learning (Camerer and Ho, 1999). Self-tuning EWA replaces two of the parameters with numerical values and two with functions. The remaining parameter λ "*measures sensitivity of players to attractions*" (Camerer and Ho, 1999, p. 835). The version of reinforcement learning examined here does not have a parameter and the initial propensities are not estimated from the data. All investigated learning rules will start with randomization of 0.5 in the first round. Only after all necessary information has been gathered does the corresponding learning rule determine the following decisions.[3]

The parameters of the parametric concepts are estimated to lead to the best fit over all data and over all games. For more details about the parameter estimation on the aggregate level, refer to the results in Section 4.2; for details on the estimation of the parameters on the individual level, refer to results in Section 5.2.

2.1. Impulse-matching learning

Impulse-matching learning relates to the concepts of impulse-balance equilibrium (Selten et al., 2005 and Selten and Chmura, 2008) and learning direction theory (Selten and Stoecker, 1986 and Selten and Buchta, 1999). After a decision and after the realization of the payoffs, the behavior is adjusted to experience. Selten and Buchta explain the concept by the example of a marksman aiming at a trunk: "*If he misses the trunk to the right, he will shift the position of the bow to the left and if he misses the trunk to the left he will shift the position of the bow to the right. The marksman looks at his experience from the last trial and adjusts his behavior* [...]." (Selten and Buchta, 1999, p. 86). Impulse-balance equilibrium and impulse-matching learning overcome the limitation of learning direction theory defining a direction only for ordered strategies, e.g., increasing a bid in an auction (cf. Ho et al., 2007) by shifting the probabilities of single actions.

To understand how impulse-matching learning works, suppose that in a period the first of two actions has been chosen and that this action was not the best reply to the action played by the other player. Then the player receives an *impulse* towards the second action. Originally, an impulse was defined as the difference between the payoff the player could have received for his best reply minus the payoff actually received given the decision by the other player in this period. However, the theory of impulse-matching learning is based on another impulse concept. Here, a player always receives an impulse from the action with the lower payoff to the one with the higher payoff. The resulting learning model is similar to the regret-based learning models, which have already been successfully tested by Marchiori and Warglien (2008). The name impulse-matching is due to the fact that this kind of learning leads to probability matching by a player if the probabilities p_1 and $(1 - p_1)$ on the other side are fixed, and the payoffs for the player is one if both players play the strategy with the same number (one or two) and zero otherwise (cf. Estes, 1954).

To incorporate loss aversion, the impulses are not calculated with the original payoffs, but with transformed ones. In games with two pure strategies and a mixed Nash equilibrium, each pure strategy has a minimal payoff and the maximum of the two minimal payoffs is called the pure-strategy maximin. This pure-strategy maximin is the maximal payoff a player can obtain for sure in every round and it forms a natural aspiration level. Amounts below this aspiration level are perceived as losses and amounts above this aspiration level are perceived as gains. In line with prospect theory (Kahneman and Tversky, 1979), losses are counted double in comparison to gains. Thus, gains (the part above the aspiration level) are cut to half for the computation of impulses. Fig. 1 is taken from Selten and Chmura (2008) and illustrates the transformation of the payoffs by the example of game 3.

[3] This means, for example, that for impulse-matching learning, impulses into both directions must have been experienced by the subject, and for reinforcement learning, payoff sums for both actions must have been collected.

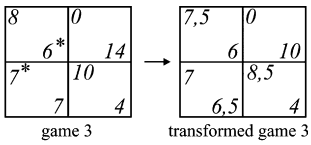


Fig. 1. Example of matrix transformation as given in Selten and Chmura (2008).

Impulse-matching learning can be described as a process in which a subject forms impulse sums. The impulse sum $R_i(t)$ is the sum of all impulses from j towards i experienced up to period $t-1$. The probabilities for playing action 1 and 2 in period t are proportional to the impulse sums $R_1(t)$ and $R_2(t)$:

$$p_i(t) = \frac{R_i(t)}{R_1(t) + R_2(t)}, \quad \text{for } i = 1, 2. \tag{1}$$

The impulses from action j towards action i in period t are as follows:

$$r_i(t) = \max[0, \pi_i - \pi_j], \tag{2}$$

for $i, j = 1, 2$ and $i \neq j$. Here, π_i is the transformed payoff for action i given the matched agent's decision and π_j the one for action j. Afterwards the impulse sums are updated with the new impulses:

$$R_i(t+1) = R_i(t) + r_i(t). \tag{3}$$

In the first round, all impulse sums are zero $R_1(1) = R_2(1) = 0$, and until both impulse sums are higher than zero, the probabilities are fixed to $p_1(t) = p_2(t) = 0.5$.

2.2. Action-sampling learning

Action-sampling learning relates to the idea of the action-sampling equilibrium of Selten and Chmura (2008). According to action-sampling equilibrium, a player takes in the stationary state a fixed-size sample of the pure strategies played by the other players in the past and optimizes against this sample. The process of action-sampling learning is a belief-based type of learning, which is very similar to fictitious play. In fact, the model by Chen et al. (2011) is a generalization of action-sampling learning, incorporating inertia, recency, and weighting of the grand mean. Action-sampling learning is much simpler and can be described as a version of fictitious play where only random periods are considered and not the whole data from the history.

In the process of action-sampling learning, the agent randomly takes a sample $A(t)$ with replacement of n earlier actions $a_1, \ldots, a_n$ of the other player. Let $\pi_i(a_j)$ be the payoff of action i if the opponent plays action a_j. For $i = 1, 2$ let $P_i(t) = \sum_{j=1}^{n} \pi_i(a_j)$ be the sum of all payoffs of the player for using her action i against the actions in this sample.

Therefore, in period t, the player chooses her action according to

$$p_i(t) = \begin{cases} 1 & \text{if } P_i(t) > P_j(t), \\ 0.5 & \text{if } P_i(t) = P_j(t), \\ 0 & \text{else}, \end{cases} \tag{4}$$

for $i, j = 1, 2$ and $i \neq j$.

At the beginning, the probabilities are set to $p_1 = p_2 = 0.5$, until both possible actions have been played by the opponent agents.

2.3. Reinforcement learning

The concept of reinforcement learning is one of the oldest and best established learning models in the literature; refer to Harley (1981) for an early application. For experimental economics, it was first formulated and introduced by I. Erev and A.E. Roth (Erev and Roth, 1998, and Roth and Erev, 1995). In the reinforcement model, a player builds up a payoff sum $B_i(t)$ for each action i according to the following formula:

$$B_i(t+1) = \begin{cases} B_i(t) + \pi(t) & \text{if action } i \text{ was chosen in } t, \\ B_i(t) & \text{else.} \end{cases} \tag{5}$$

Here $\pi(t)$ is the payoff obtained in period t. After an initial phase, in which both possible actions are used with equal probabilities, the probability of choosing action i in period t is given by:

$$p_i(t) = \frac{B_i(t)}{B_1(t) + B_2(t)}. \tag{6}$$

This model presupposes that all payoffs in a player's payoff matrix are non-negative and at least one payoff in each column and row is positive, a condition fulfilled by all twelve investigated games.[4] In the first round, the initial payoff sums $B_i(t)$ are zero and the player chooses both possible actions with equal probabilities $p_1 = p_2 = 0.5$. The initial phase ends as soon as both sums are positive, and only from then on, Eq. (6) is applied to determine the probabilities.

[4] For games with negative payoffs, this approach would not be adequate. To cope with negative payoffs, the model used by Erev and Roth (1998) replaces the payoff $\pi(t)$ in Eq. (1) by $\pi(t) - \pi_{\min}$, where $\pi_{\min}$ is the smallest possible payoff of the player.

Impulse matching learning and action-sampling learning are both based on behavioral stationary concepts. Reinforcement learning converges in 2×2 constant-sum games to Nash equilibrium, if they are pure or if they are mixed (Beggs, 2005). Therefore, we treat it in the following as the learning concept that corresponds to Nash equilibrium, although this holds only for one half of our games.

2.4. Self-tuning EWA

Self-tuning EWA was introduced by Ho et al. (2007). It is based on the experience-weighted attraction model (Camerer and Ho, 1999), but replaces all but one parameter of this model with functions or fixed values. Of all models discussed in this article, self-tuning EWA is the most sophisticated one because it can capture different types of learning. The decisions are made according to *attractions* $A_i(t)$ for each strategy i. Attractions are based on the payoffs $\pi(s_i, s^m(t))$ which a subject would have received for playing strategy s_i given the actual decision $s^m(t)$ by the matched player. The attraction updating function depends on an experience weight $N(t)$, a change-detector function $\phi(t)$, and the attention function $\delta(t)$:

$$A_i(t) = \frac{\phi(t)N(t-1)A_i(t-1) + [\delta(t) + (1-\delta(t))I(s_i, s(t))]\pi(s_i, s^m(t))}{N(t)}. \tag{7}$$

Here $I(s_i, s(t))$ is an indicator function equal to 1 for $s(t) = s_i$ and 0 otherwise. An experience weight is applied to each attraction and it is defined as

$$N(t) = N(t-1)\phi(t) + 1, \quad \text{with } N(0) = 1. \tag{8}$$

The change-detector function $\phi(t)$ weights lagged attractions and represents "*a player's perception of how quickly the learning environment is changing*" (Ho et al., 2007, p. 182). It is defined as

$$\phi(t) = 1 - \frac{1}{2}S(t) \tag{9}$$

with $S(t)$ being the so called *surprise index*, which measures the deviation of the matched players' recent decisions from all previous decisions.[5] $S(t)$ is the quadratic distance between the cumulative history vector $h_k^m(t)$ and the immediate history vector $r_k^m(t)$ for the k strategies of the matched player m. The cumulative history vector gives the relative frequency over all rounds and is defined as

$$h_k^m(t) = \frac{\sum_{\tau=1}^{t} I(s_k^m, s^m(\tau))}{t}. \tag{10}$$

The immediate history vector gives the relative frequency in the recent rounds. For 2×2 games with mixed equilibria, it is defined as

$$r_k^m(t) = \frac{\sum_{\tau=t-W+1}^{t} I(s_k^m, s^m(\tau))}{2}. \tag{11}$$

The surprise index is given as:

$$S(t) = \sum_{k=1}^{2} \left(h_k^m(t) - r_k^m(t)\right)^2. \tag{12}$$

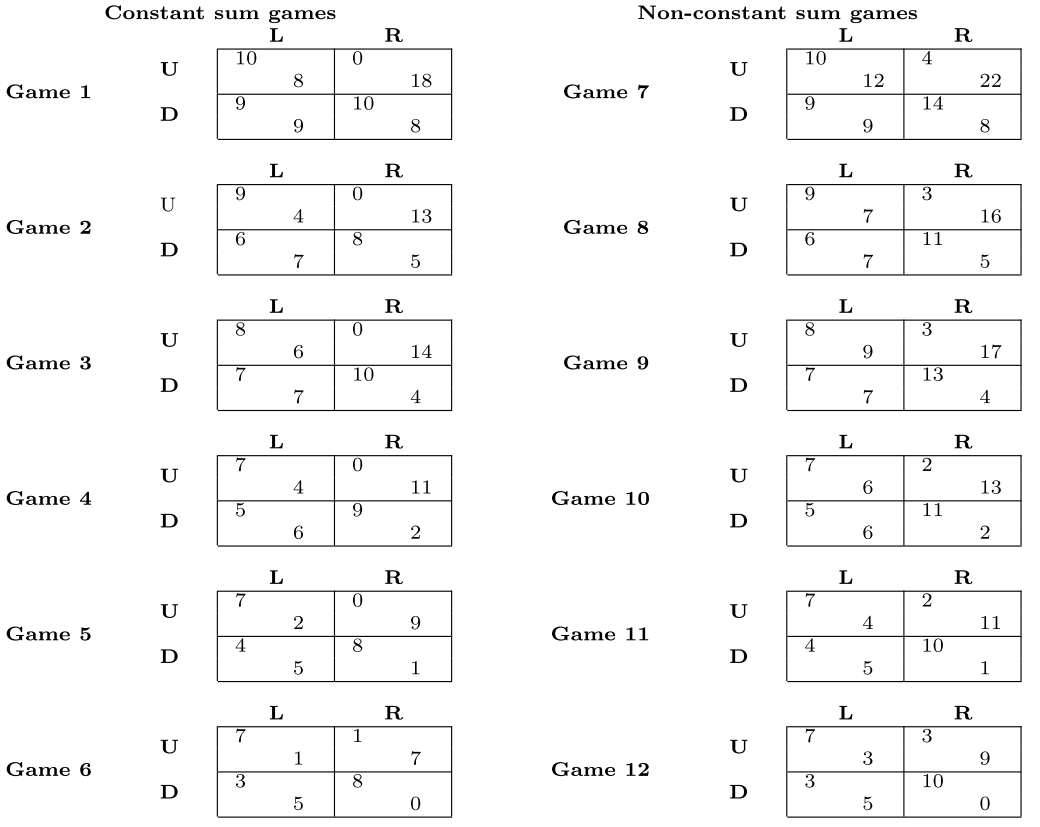
The attention function $\delta(t)$ generates a weight for foregone payoffs and turns the attention to strategies which would have yielded higher payoffs. In games with a unique mixed-strategy equilibrium, these payoffs are weighted with $1/W$, with W being the numbers of strategies played in equilibrium. Thus, in our 2×2 games with mixed equilibria, it is set to be $W = 2$.

$$\delta(t) = \begin{cases} \frac{1}{W} & \text{if } \pi(s_j, s_m(t)) \geqslant \pi(t), \\ 0 & \text{else.} \end{cases} \tag{13}$$

The attention function $\delta(t)$ of self-tuning EWA captures the idea of learning direction theory (Selten and Stoecker, 1986) that subjects have a tendency to move into the direction of the strategy which was ex-post the best response. This is done by shifting the attention and thus the probability towards the strategy with the highest payoff. This is similar to the process of impulse-matching learning. The resulting probability of playing action i in period t, depending on the attractions, is calculated as a logit response function:

$$p_i(t) = \frac{e^{\lambda A_i(t-1)}}{\sum_{j=1}^{2} e^{\lambda A_j(t-1)}}. \tag{14}$$

[5] The experiments were played with random matching and thus no identification of single players is possible. Therefore, we assume that all matched players are perceived as one average player.



The payoffs for the column players are shown in the lower right corner,
the payoffs for the row players are shown in the upper left corner.
Abbreviations used: L Left, R Right, U Up, D Down

Fig. 2. The twelve 2×2 games taken from Selten and Chmura (2008).

Here, λ is the response sensitivity and this parameter must be specified to fit the empirical data. To be consistent with the other models, we have chosen not to estimate any additional values and the simulations start with pure randomization with $p_1 = p_2 = 0.5$.

3. Games and experiments

Our comparison of the investigated learning rules is based on the data of Selten and Chmura (2008). In their study, twelve 2×2 games with pure equilibria in mixed strategies were experimentally investigated. To cover a broad set of games, six constant and six non-constant sum games were played. Fig. 2 shows the twelve games used in the experiment. The constant sum games are shown on the left side of the figure, and the non-constant sum games on the right side.

Note that the first six games have the same best-response structure as the second six games and that the concepts of action-sampling equilibrium and Nash equilibrium only depend on this best response structure. Thus, the predictions of Nash equilibrium are the same for the first and the second six games and the same holds true for action-sampling equilibrium. The predictions of Nash equilibrium, action-sampling equilibrium and impulse-balance equilibrium are given in Table 1.

All experiments were run at the BonnEconLab with students mainly majoring in economics or law. The experiment was programmed with RatImage developed by Abbink and Sadrieh (1995). The data was collected in 54 sessions with 16 subjects each. In every session, only one game was played and this game was known by all subjects. The games were played for 200 periods with matching groups consisting out of eight subjects. For each constant sum game, twelve independent matching groups were gathered; for each non-constant sum game, six independent matching groups were gathered. Overall, 864 subjects participated.

The role of the subjects was fixed for the whole experiment, thus four subjects in each matching group decided as column players and the other four as row players throughout the whole experiment. At the beginning of each round, row and column players were randomly matched. After every round, subjects received feedback about the other player's decision, their own payoff, the period number and their own cumulative payoff. Each participant received €5 for showing-up. In addition, the payoffs in the 200 periods were accumulated and transferred into Euro. The exchange rate was €0.016 Cent per

payoff point. An experimental session lasted between 1.5 and 2 hours and the average earning per subject was roughly €24, including show-up fee.

4. Simulating the behavior of populations

In this section, we investigate whether the learning algorithms can replicate in simulations the aggregate distribution of actions generated by human subjects. In the following, we will first introduce our measurement of success for the simulations. Thereafter, we discuss the success of the different learning models in predicting/reproducing the aggregate distribution of behavior.

4.1. Measure of predictive success for the simulations

For this analysis, we conduct simulations keeping everything the same as in the experiment, except that instead of real participants now computer agents interact. Each agent interacts according to her history and to the same learning model over 200 rounds. In each round, eight agents with fixed roles, four deciding as row players and four as column players, are randomly matched and all agents act in accordance to the same learning rule.

After each round, they receive feedback about the matched agent's decision and their payoff. Since none of the learning models makes use of the round number and since the calculation of the cumulated payoff can be done by the agents themselves, this information is not provided to the agents. It is crucial that the agents do not receive more information than the subjects in the experiment did.

All learning models include stochastic elements. To avoid the influence of statistical outliers, 500 simulation runs per game are conducted. In each simulation run, all agents act in accordance with the same learning model. To measure the predictive success on the population level, we will compare the mean frequencies of U and L in the simulations with the mean frequencies obtained in the experiments by means of the quadratic distance. The mean quadratic distance Q is the average quadratic distance over all 12 games and over all 500 simulations. It is defined as

$$Q = \frac{1}{12}\sum_{i=1}^{12}\left(\frac{1}{500}\sum_{n=1}^{500}\left(s_{in}^{L} - f_i^{L}\right)^2 + \left(s_{in}^{U} - f_i^{U}\right)^2\right),$$

with s_{in}^{L} and s_{in}^{U} being the frequencies for L and U in game i and simulation run n. Respectively, f_i^{L} and f_i^{U} are the mean frequencies for L and U observed in the experiments with game number i. The frequencies of R and D need not be considered in view of $(s_{in}^{L} - f_i^{L})^2 = (s_{in}^{R} - f_i^{R})^2$ and $(s_{in}^{U} - f_i^{U})^2 = (s_{in}^{D} - f_i^{D})^2$. The predictive success of a learning model increases with a decreasing mean quadratic distance, i.e., the smaller the mean quadratic distance is, the better the learning theory fits the experimental data on the aggregate level.

4.2. Parameter estimates

The concepts of action-sampling equilibrium and self-tuning EWA have a parameter which needs to be adjusted to the experimental data. We decided to estimate for each learning model one parameter that minimizes the quadratic distance over all games.[6] To estimate the optimal parameter, we ran, for each parameter, 500 simulations per game and calculated the mean quadratic distance. Thereafter, the simulations were conducted with the parameter that yielded the smallest quadratic distance.[7]

The parameter of action-sampling learning is the size of the drawn samples. Fig. 3 gives the mean quadratic distances of action-sampling learning for $1 \geqslant n \geqslant 15$. A sample size of $n = 12$ leads to the smallest quadratic distance, which is the same sample size that also leads to the smallest distance for the stationary concept (cf. Brunner et al., 2011).

Fig. 4 gives the mean quadratic distances of self-tuning EWA for different lambdas. The left part gives the mean quadratic distances for all tested lambdas between 0 and 10, and the right part gives the quadratic distance for $0.2 < \lambda < 0.3$.

Each point in the graph represents the mean quadratic distance over all twelve games with 500 simulation runs per game with one specific lambda value. The value leading to the smallest quadratic distance is $\lambda = 0.2775$.

[6] One could fit the parameter of the parametric concepts for each game separately. We believe that this gives an unfair advantage to one-parameter theories over parameter-free ones. This especially holds for the case of 2×2 games, where only two relative frequencies are predicted. Adjusting a parameter separately for each game, so to speak, does half the job. One might use methods to adjust the fit of a theory to the number of parameters used, but this only makes sense if the non-adjusted performance of a model increased in case of parameters being estimated for each game separately. For our simulations, only the quadratic distance of action-sampling learning would benefit from such a procedure. The quadratic distance of self-tuning EWA (0.0805 vs. 0.0786) would change only slightly and this adjustment would not influence the relative ranking of quadratic distances. Therefore, we decided to estimate only one parameter for each theory.

[7] To speed up this procedure, round-by-round data were only saved for the simulations with the final parameter.

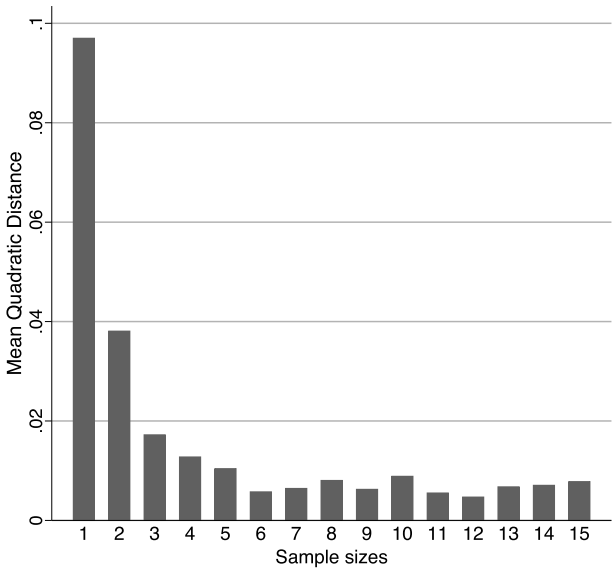


Fig. 3. Quadratic distances of action-sampling learning.

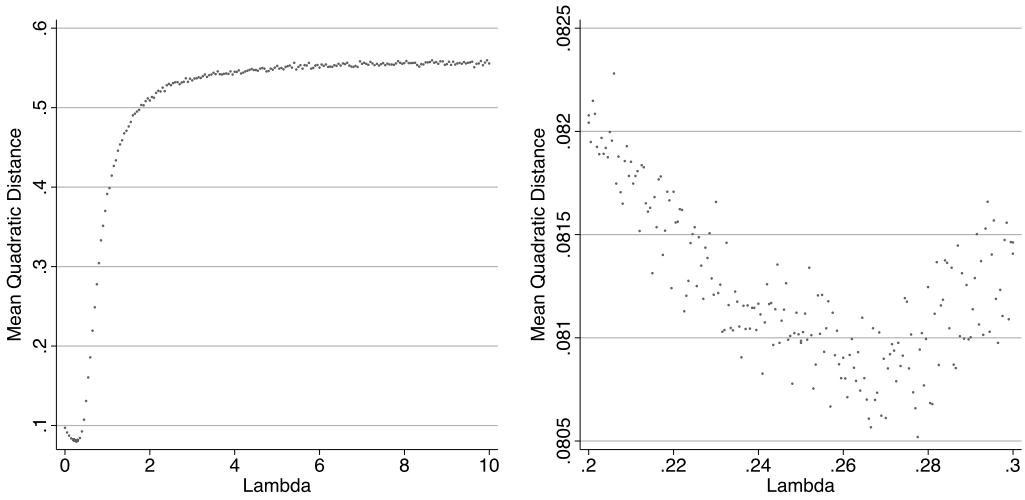


Fig. 4. Quadratic distances of self-tuning EWA for different lambdas. Each point represents the mean quadratic distance over 500 simulations per game. Left figure for $0 \leqslant \lambda \leqslant 10$ and right figure for $0.2 \leqslant \lambda \leqslant 0.3$.

4.3. Relative frequencies

Table 1 gives the observed mean frequencies for each learning type, mean frequencies predicted by the stationary concepts and the observed frequencies in the experiments. For the experimental games 1 to 6, the mean frequencies observed in a game are based on the observed frequencies in twelve independent matching groups; for games 7 to 12, they are based on the observed frequencies in six independent matching groups. Each matching group consists of eight subjects. For each learning type and game, the mean is based on 500 simulation runs, which produced 500 independent matching groups per game. Each matching group consists of eight agents. Fig. 5 gives the typical development of probabilities over time in game 7 for each of the learning types. This figure in combination with Table 1 already reveals some differences between the learning rules.[8]

It is surprising that self-tuning EWA yields relative frequencies very near to 0.5 for each of the twelve games. This is probably connected to the fact that, in our simulations, the whole population is of the same type and agents try to adjust to an inaccurate history, and by this process generate a new inaccurate history for themselves and the matched agents. Estimating the free parameter of this model jointly for all games is not a reason for this behavior. If we estimate the

[8] The course of probabilities for all games is given in Appendix C.1.

Table 1
Relative frequencies for playing left and up in the simulations, predicted by the stationary concepts and observed in the experiments for up and left.

		Impulse-matching learning	Action-sampling learning	Reinforcement learning	Self-tuning EWA learning	Impulse-balance equilibrium	Action-sampling equilibrium	Nash equilibrium	Selten and Chmura data
Game 1	L	0.574	0.710	0.345	0.499	0.580	0.705	0.909	0.690
	U	0.063	0.095	0.121	0.499	0.068	0.090	0.091	0.079
Game 2	L	0.495	0.571	0.333	0.477	0.491	0.584	0.727	0.527
	U	0.169	0.193	0.161	0.502	0.172	0.193	0.182	0.217
Game 3	L	0.770	0.763	0.503	0.541	0.765	0.774	0.909	0.793
	U	0.157	0.211	0.128	0.492	0.161	0.208	0.273	0.163
Game 4	L	0.714	0.711	0.587	0.548	0.710	0.719	0.818	0.736
	U	0.259	0.295	0.190	0.494	0.259	0.302	0.364	0.286
Game 5	L	0.632	0.639	0.566	0.524	0.628	0.643	0.727	0.664
	U	0.296	0.323	0.241	0.495	0.297	0.329	0.364	0.327
Game 6	L	0.602	0.596	0.666	0.527	0.600	0.596	0.636	0.596
	U	0.400	0.422	0.265	0.497	0.400	0.426	0.455	0.445
Game 7	L	0.637	0.709	0.380	0.564	0.634	0.705	0.909	0.564
	U	0.098	0.094	0.170	0.485	0.104	0.090	0.091	0.141
Game 8	L	0.563	0.572	0.396	0.540	0.561	0.584	0.727	0.586
	U	0.258	0.193	0.217	0.494	0.258	0.193	0.182	0.250
Game 9	L	0.767	0.762	0.525	0.600	0.764	0.774	0.909	0.827
	U	0.185	0.212	0.165	0.489	0.188	0.208	0.273	0.254
Game 10	L	0.726	0.711	0.640	0.587	0.724	0.719	0.818	0.699
	U	0.303	0.295	0.219	0.487	0.304	0.302	0.364	0.366
Game 11	L	0.648	0.640	0.609	0.572	0.646	0.643	0.727	0.652
	U	0.354	0.324	0.289	0.492	0.354	0.329	0.364	0.331
Game 12	L	0.605	0.596	0.560	0.578	0.604	0.596	0.636	0.604
	U	0.466	0.422	0.342	0.494	0.466	0.426	0.455	0.439

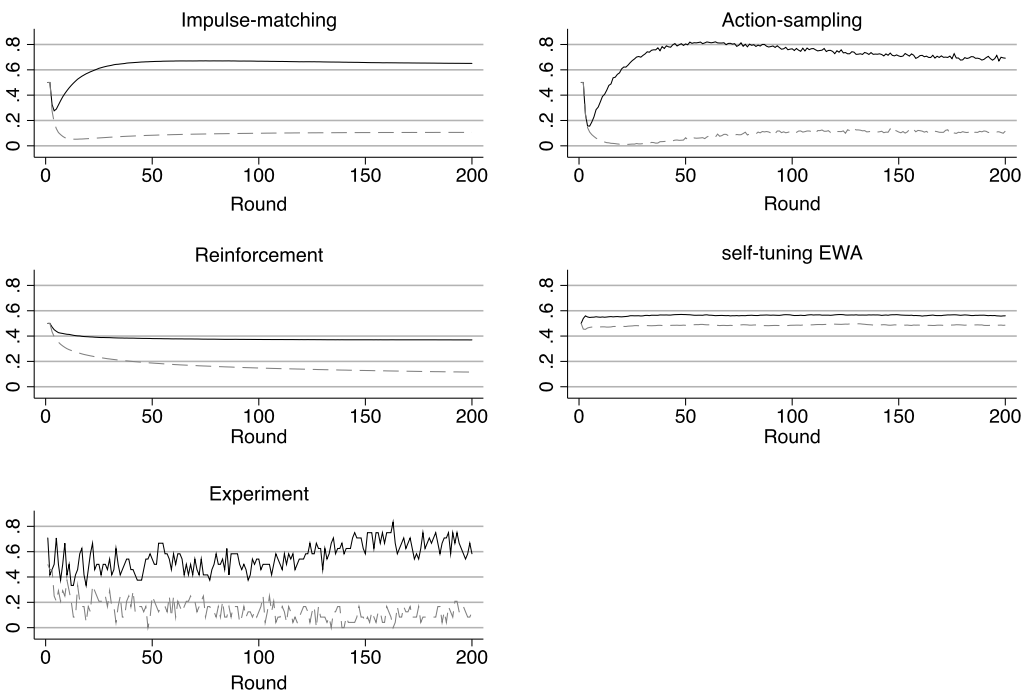


Fig. 5. Mean probabilities for left and right in the simulations runs and the experiment for game 7. The mean probability for left is given in black and the mean probability for up is given in gray.

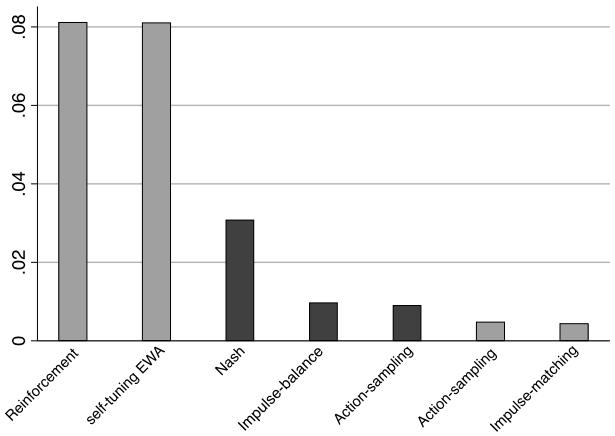


Fig. 6. Mean quadratic distances of the stationary concepts and the learning models to the observed behavior (stationary concepts in dark bars and learning models in light bars).

optimal λ for each game separately, the resulting mean quadratic distance to the data improves only marginally (0.081 vs. 0.079) and observed relative frequencies do not change much.

Impulse-matching learning and action-sampling learning are quite close to their stationary counterparts after 200 periods. The quadratic distances between impulse-matching learning and impulse-balance equilibrium, as well as the one between action-sampling learning and action-sampling equilibrium, are smaller than 0.001. If we treat reinforcement learning as the learning counterpart to Nash equilibrium, the difference in quadratic distances is 0.158. Self-tuning EWA has the highest distances towards all stationary concepts. This closeness results in high correlations between the frequencies of the simulations and corresponding stationary concepts. For impulse-matching learning and action-sampling learning, this is true for both players (pairwise correlation with $r > 0.9$ and $p < 0.01$ for row and column players), and for reinforcement only for row players (pairwise correlation with $r > 0.9$ and $p < 0.01$). Correlations between observed frequencies from the experiments and the simulations with action-sampling learning and impulse-matching learning are high (pairwise correlation with $r > 0.8$ and $p < 0.01$ for both players), but lower than the ones with the stationary concepts. The frequencies of reinforcement learning are correlated with the ones of the row player in the experiment (pairwise correlation with $r > 0.8$ and $p < 0.01$), but not for the ones of the column players. The frequencies of self-tuning EWA are not significantly correlated with the empirical data for either players.

4.4. Overall performance

Fig. 6 gives the mean of the quadratic distance between the experiment and simulations over all games and rounds for self-tuning EWA learning, reinforcement learning, action-sample learning, and impulse-matching learning. In addition, the figure gives the mean quadratic distances between of the stationary counterparts (if existing) and the data in black.[9]

We first turn our attention to the comparison of the simulations. The figure reveals a clear order of explanatory power. The order from worst to best (highest quadratic distance to lowest quadratic distance) is as follows: reinforcement learning, self-tuning EWA learning, action-sampling learning, and impulse-matching learning. Because of the high number of observations (6000 per learning type), the order given by Fig. 6 is statistically robust (for all $p < 0.01$ Fisher–Pitman permutation test for paired replicates). The difference between self-tuning EWA and reinforcement is very small and irrelevant. However, the similarity between the two quadratic distances does not mean that both theories make similar predictions. This can be seen, for example, in Table 1 and in Fig. 5. The figure demonstrates that the concepts of self-tuning EWA and reinforcement fail to describe the aggregate behavior in the 2×2 experiments, in contrast to the other concepts. The quadratic distance of self-tuning EWA is 18 times higher than the one of impulse-matching learning.

The quadratic distances of reinforcement learning and self-tuning EWA learning are significantly bigger not only over all games, but also for the subsets of constant sum games and non-constant sum games. However, reinforcement performs better in constant sum games than self-tuning EWA does, while self-tuning EWA performs better in non-constant sum games (both $p < 0.01$ Fisher–Pitman permutation test for paired replicates). While the quadratic distance of impulse-matching is stable in constant and non-constant sum games, the one of action-sampling learning is smaller in constant sum games. Thus, action-sampling learning performs significantly better in constant sum games, and impulse-matching learning performs significantly better in non-constant sum games (both $p < 0.01$ Fisher–Pitman permutation test for paired replicates).

Comparing the stationary concepts with the learning models reveals that self-tuning EWA and reinforcement learning are not only outperformed by impulse-matching learning and action-sampling learning, but by all stationary concepts.

[9] The mean quadratic distances of the stationary concepts are either taken from Selten and Chmura (2008) or from Brunner et al. (2011). There were some flaws in the paper by Selten and Chmura (2008). For a detailed discussion, refer to Brunner et al. (2011) and Selten et al. (2011).

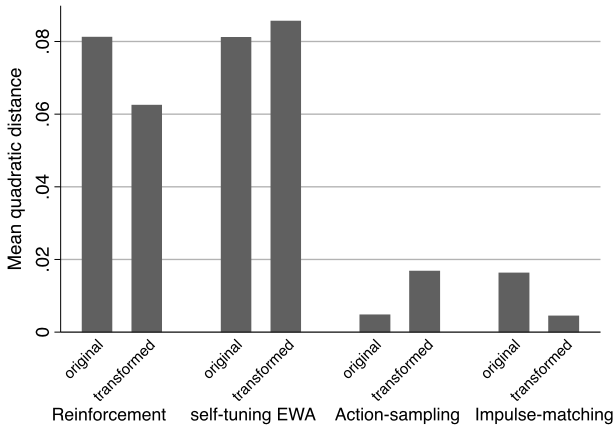


Fig. 7. Mean quadratic distance in original and transformed games.

In contrast, the learning models of action-sampling and impulse-matching perform very well. Both learning models have higher predictive success than the other learning models and additionally a higher predictive success than all stationary concepts.

4.5. Original vs. transformed games

The concept of impulse-matching learning is applied to the transformed game rather than the original one. This transformation is an essential part of impulse-matching learning and impulse-balance equilibrium (Selten and Chmura, 2008 and Goerg and Selten, 2009), because both concepts involve a fixed loss-aversion. Losses with respect to the pure-strategy maximin are counted double. While double counting of losses with respect to the pure-strategy maximin is an essential part of impulse-matching learning, it is ignored by the other concepts (reinforcement learning, action-sampling learning, self-tuning EWA). This raises the question whether the good performance of impulse-matching learning is an artifact of the incorporation of loss aversion. To investigate this point, we apply all learning models to the transformed and to the original matrices.

Fig. 7 shows the overall mean quadratic distances for self-tuning EWA learning, reinforcement learning, payoff-sampling learning, impulse-balance learning, action-sampling learning, and impulse-matching learning applied to the original games and to the transformed games, which are again based on 500 simulation runs per game and learning model.

It can be seen that impulse-matching learning and reinforcement learning perform better when applied to the transformed games, whereas self-tuning EWA learning and action-sampling learning do less well. While the improvement of impulse-matching learning in transformed games is expected, the benefit of applying reinforcement learning to transformed games is unexpected. This improvement is substantial, in the original game the quadratic distance is nearly 1.3 times higher than in the transformed ones.

The theory of Roth and Erev (1995) applies a transformation of the original game by replacing the payoff of a player by its difference to the minimal value in her matrix. The transformation used here is different since it involves double weights for losses with respect to the pure-strategy maximin. However, in Selten and Chmura (2008), no improvement of the predictive power of the Nash equilibrium was observed when applied to the transformed game rather than to the original one. It is interesting that the picture looks different for the simulations over 200 rounds with reinforcement learning, although it corresponds very much with Nash equilibrium (Beggs, 2005).

Although reinforcement learning improves when applied to the transformed matrix, it still performs significantly worse than impulse-matching learning. Therefore, and because of self-tuning EWA and action-sampling learning performing worse in the transformed matrices, we can conclude that the good performance of impulse-matching learning is not driven by the transition to the transformed matrices alone.

4.6. Changes over time

Learning processes are always dependent on time and history, and therefore it is of interest to check whether our above results remain stable over time. To check stability of the order of explanatory power over time, we compare the first hundred periods with the second hundred periods. Fig. 8 gives the mean quadratic distances for periods 1–100 (left) and 101–200 (right) for the six learning models. The basis of the comparison is always the observed mean frequencies for the corresponding rounds (either round 1–100 or 101–200) in the experiments.

It is easy to recognize that in the second half of the simulation runs the explanatory power of self-tuning EWA and reinforcement learning decreases significantly, while the one of impulse-matching learning improves significantly (all Fisher–Pitman permutation test for paired replicates $p < 0.01$). The concept of action-sampling learning is rather stable over time

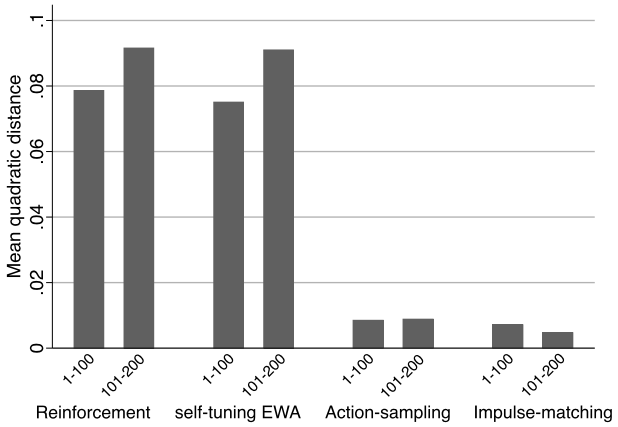


Fig. 8. Mean quadratic distance over time.

and the statistically significant disimprovement of action-sampling learning ($p < 0.01$) is economically negligible, with an increase of the quadratic distances of only 0.0004.

The ranking of concepts by mean quadratic distances is stable over time, the overall ranking for round 1–200 is the same in rounds 1–100 and 101–200.

5. Individual round-by-round predictions

In this part, we investigate how well the learning rules describe the individual behavior of the subjects in the 2×2 experiments. To judge the performance on the individual level, we compare the individual decisions in every round of the experiment with the predicted decisions or predicted probability by the learning rule, given the history of the subject.

5.1. Measure of predictive success for the individual round-by-round behavior

To measure the predictive success of the learning theories describing the behavior of a single individual, we apply the quadratic scoring rule on each of the 864 subjects for each learning rule. The quadratic scoring rule was first introduced by Brier (1950) in the context of weather forecasting. The rationale behind the quadratic scoring rule is that for each round a score is determined, which evaluates the nearness of the predicted probability distribution to the observed outcome.

In Selten (1998), the quadratic scoring rule is axiomatically characterized. The characterizing properties of the quadratic scoring rule, as described in Selten (1998), are: symmetry, elongational invariance, incentive compatibility, and neutrality. Symmetry means that the score of a theory must not depend on the numbering of the decision alternatives. Elongational invariance assures that the score of a theory is not influenced by adding or leaving an alternative which is predicted with a probability of zero. Incentive compatibility requires that predicting the actual probabilities yields the highest score. Finally, neutrality means that in the comparison of two theories, of which one is right in the sense that it predicts the actual probabilities, and the other is wrong, the score for the right theory does not depend on which of the two theories is the right one. This means that the score does not prejudge one of the theories depending on the location of the theory in the space of probability distributions.

We apply the quadratic scoring rule to measure the predictive success of a theory for every period and subject and then calculate the mean over subjects, rounds, and games. Accordingly, a score depending on the predicted probabilities and the actually observed action is computed. In order to compute the score, the observation is interpreted as a frequency distribution, where for the chosen action the relative frequency is one, and for the action not chosen it is zero.

The quadratic score $q(t)$ of a learning theory for subject choosing action i in period t is given as[10]:

$$q(t) = 2p_i(t) - p_i^2(t) - \big(1 - p_i(t)\big)^2.$$

Here $p_i(t)$ is the predicted probability of the learning theory. The predicted probability of the learning theory is calculated by applying the theory's learning algorithm to the whole playing history of this player. If no history yielding a positive number smaller than 1 for $p_i(t)$ is available, the player randomizes with $p_i(t) = 0.5$. This rule provides an initial phase. As soon as both probabilities are positive, they will remain positive forever.

If a player decides completely in line with the prediction of the theory, he receives a score of 1; if he decides in complete contrast to the prediction of the theory, he receives a score of -1. The mean score $\bar{q}$ is given as the mean of $q(t)$ over all

[10] If the decision maker has n choices it is defined as: $q(t) = 2p_i(t) - \sum_{j=1}^{n} p_j^2(t)$. The formula in the text holds for the special case of $n = 2$.

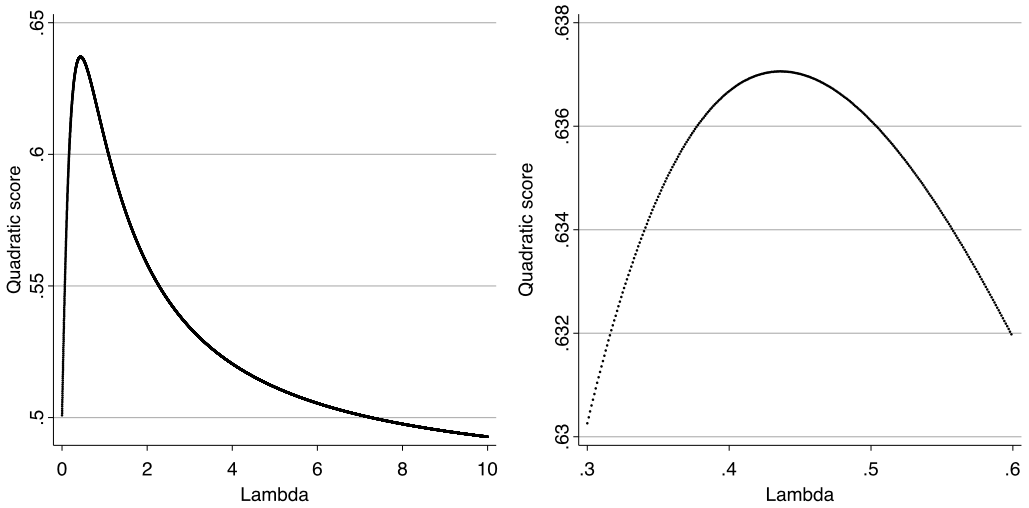


Fig. 9. Quadratic scores of self-tuning EWA for different lambdas, each point represents the mean quadratic score for all 864 subjects in left figure for $0 \leqslant \lambda \leqslant 10$ and right figure for $0.3 \leqslant \lambda \leqslant 0.6$.

200 rounds and all 864 subjects. Of course, $\bar{q}$ must be in the closed interval between -1 and $+1$. Thus, in contrast to our measurement for the success on the aggregate level, the success of a theory on the individual level increases with the score.

The concept of action-sampling learning always yields a probability of 1, 0 or 0.5 for one of the possible actions. Which action is chosen depends on the randomly drawn sample. Therefore we calculate the probability of drawing a sample that commands playing action 1 or action 2 as the predictions of this concept.

In addition to the investigated learning, rules we introduce three benchmarks. The first one is a heuristic which we call the inertia rule. This rule commands to "*do exactly the same as in the preceding round*". This does not apply to the first period in which both possible actions are chosen with equal probabilities. The player is required to repeat the decision of the preceding period even if he deviated from this rule in the past. Obviously, the inertia rule is not a serious decision rule, but it serves as a benchmark that every learning rule should beat. The second benchmark is the score an agent would receive if he decided randomly between the two actions with $p = 0.5$. In this case, the score would be 0.5 and again every learning rule should beat this benchmark. The third benchmark is the aggregated observed frequencies taken from the experiments. If a learning theory adequately describes the adjustments over time, it should yield higher scores than these stationary probabilities, which do not depend on this information.

5.2. Parameter estimates

On the individual level, we calculated for each parametric learning theory one parameter, which leads to the highest mean quadratic score over all 864 subjects. Fig. 9 gives the mean quadratic scores for the parameter lambda of self-tuning EWA between 0 and 10 (left side) and for lambda between 0.3 and 0.6 (right) side. The highest mean quadratic score is reached for $\lambda = 0.436$.

Fig. 10 gives the mean quadratic scores of action-sampling for sample sizes between 1 and 15. The optimal sample size for action-sampling is $n = 6$. Note that the optimal sample size of action-sampling learning for the performance on the aggregate level ($n = 12$) also performs very well on the individual level; it leads to the second-highest mean quadratic score.

5.2.1. Overall mean quadratic scores

Fig. 11 gives the mean quadratic scores in the 108 independent observations for each learning model. The randomization benchmark is included as a horizontal line. The figure reveals a clear order of predictive success, from best to worst: self-tuning EWA, impulse-matching learning, empirical frequencies, reinforcement learning, action-sampling learning, randomization benchmark, and inertia benchmark.

Applying a two-sided permutation test for the pairwise comparison of the mean scores over all independent observations reveals that the order given by the graph is statistically robust. All pairwise comparisons between two learning models over all games are at least significant on the 1% level. Table 2 gives all test results over all games (top), over the constant-sum games (middle), and over the non-constant sum games (bottom). This ranking is robust over all games, as well as for the subsets of constant sum games and non-constant sum games.

All reported learning models perform significantly better than the inertia and randomization benchmarks. However, only impulse-matching learning and self-tuning EWA perform significantly better than the aggregated empirical frequencies in describing the individual round-by-round behavior. No significant difference between reinforcement learning and the empirical frequencies are observed, and action-sampling learning performs even worse than the empirical frequencies.

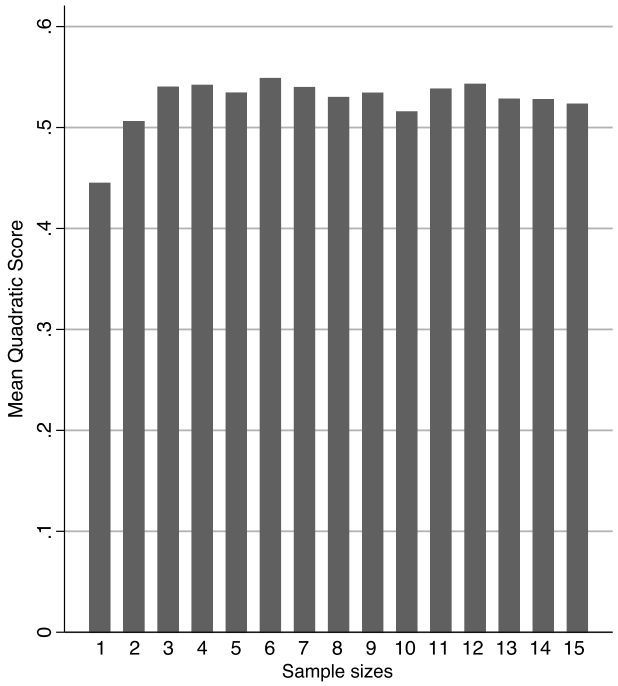


Fig. 10. Quadratic scores of action-sampling learning for different sample sizes.

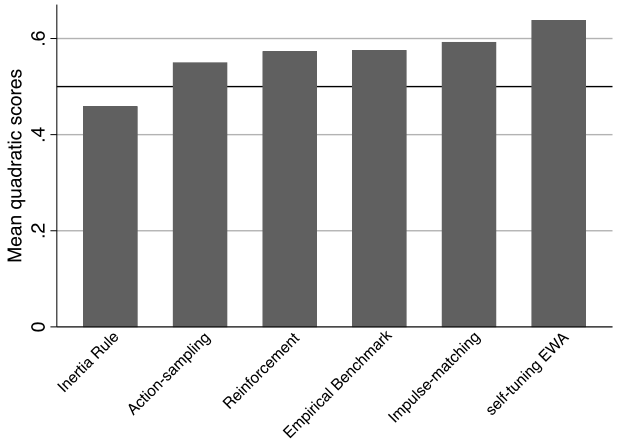


Fig. 11. Mean quadratic scores, over all 108 independent observations. The solid line gives the random rule benchmark.

Although the inertia benchmark performs rather badly in describing subjects' behavior, it does not imply that low inertia rates are observed. On the contrary, in line with Erev and Haruvy (2005) and Erev et al. (2010), very high inertia rates are observed. In 74% of all cases, subjects stick to their previous decision. On the other side, this means that, in 26% of all cases, inertia predicts exactly the opposite of observed behavior and receives the lowest score of all models (-1). Nevertheless, 13% of the subjects are best described by the inertia rule. Overall, roughly 8% of the subjects are best described by reinforcement and action-sampling learning, 23% by impulse-matching learning, and the majority of 47% is best described by self-tuning EWA.[11]

5.3. Mean quadratic scores over time

To conclude our analysis, we now take a look at the quadratic scores over time. Fig. 12 gives the mean quadratic scores for rounds 1–100 and rounds 101–200. In the first and the second half of the experiments, the same order of

[11] Please note that this calculation of proportion is problematic: it depends on the number and the performance of included learning concepts. We prefer the mean quadratic score as it does not depend on the competing learning concepts, but report these fractions for completeness.

Table 2
Two-sided Significances in Favor of Row Concepts, Monte-Carlo approximation of the two-sided Fisher–Pitman permutation test for paired replicates.

	Impulse matching	Empirical frequencies	Reinforcement	Action sampling	Random benchmark	Inertia benchmark
Self-tuning	**1%**	**1%**	**1%**	**1%**	**1%**	**1%**
EWA	*1%*	*1%*	*1%*	*1%*	*1%*	*1%*
learning	1%	1%	1%	1%	1%	1%
Impulse		**1%**	**1%**	**1%**	**1%**	**1%**
matching		*1%*	*1%*	*1%*	*1%*	*1%*
learning		5%	10%	1%	1%	1%
Empirical			**n.s.**	**1%**	**1%**	**1%**
frequencies			*n.s.*	*1%*	*1%*	*1%*
			n.s.	1%	1%	1%
Reinforcement				**1%**	**1%**	**1%**
learning				*1%*	*1%*	*1%*
				10%	5%	1%
Action					**1%**	**1%**
sampling					*1%*	*1%*
learning					5%	1%
Randomization						**1%**
benchmark						*1%*
						1%

Notes: Above: all 108 experiments; Middle: 72 constant-sum game experiments; Below: 36 non-constant sum game experiments.

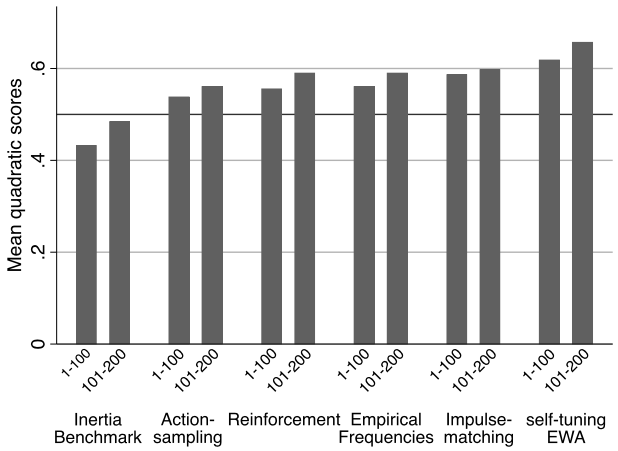


Fig. 12. Mean quadratic scores in the first and second half of the experiments.

success is present as over all rounds. For rounds 1–100, all differences between the scores, except for the one of reinforcement learning and the empirical benchmark, are highly significant (all Fisher–Pitman permutation test for paired replicates $p < 0.01$). As for the overall comparison, no significant difference between the mean quadratic scores of reinforcement learning, and the empirical benchmark is observed. Although the order remains the same over time, in the second half of the experiments the differences between impulse-matching learning, reinforcement learning and the empirical benchmark decrease. The difference between reinforcement learning and impulse-matching learning is no longer significant, while impulse-matching learning still performs significantly better than the empirical benchmark.

Fig. 13 gives the development of the mean quadratic scores per round over time. Impulse-matching learning has the fastest increase of scores in the very early rounds. In the first 10 rounds, impulse-matching learning performs better than self-tuning EWA. The performance of self-tuning EWA increases continuously over time, leading to higher scores per round after round 10, and after 25 rounds, the overall quadratic score of self-tuning EWA is above the one of impulse-matching. The score of reinforcement learning also increases continuously over time, but at a slower rate, approaching the performance of impulse-matching learning only in the last 50 rounds. In addition, the graph reveals the increase of inertia over the rounds. Over time, the score of the inertia benchmark approaches the score of the pure randomization benchmark (0.5). In the last periods, inertia has a higher predictive power than randomization, but still performs worse than the learning concepts.

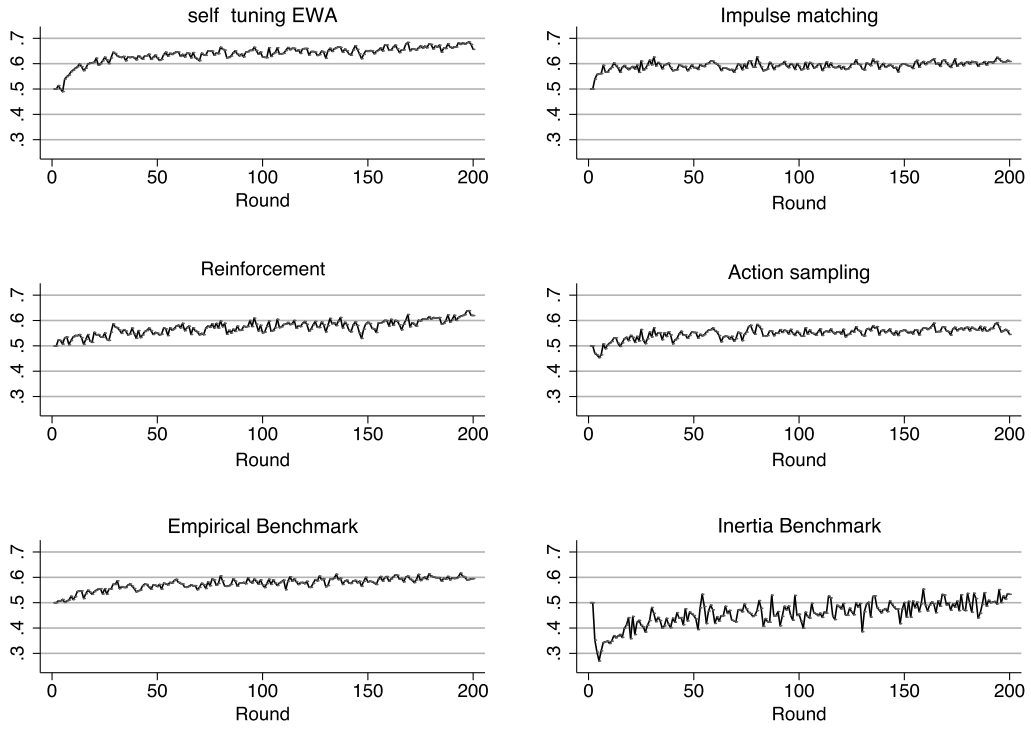


Fig. 13. Mean quadratic scores per round over time for the learning models and benchmarks.

6. Summary and discussion

In this article, the models of impulse-matching learning and action-sampling learning have been introduced. Together with reinforcement learning and self-tuning EWA, they were applied and tested in the environment of 12 repeated 2×2 games.

The newly-introduced learning models are based on the behavioral reasoning of action-sampling equilibrium and impulse-balance equilibrium, which had been successfully tested in experimental 2×2 games by Selten and Chmura (2008). Therefore the experimental dataset obtained by Selten and Chmura (2008) was used as a testbed for the learning models. The experimental data comprises aggregate and individual behavior in 12 completely mixed 2×2 games, 6 constant sum games with 12 independent subject groups each, and 6 non-constant sum games with 6 independent subject groups each. Each subject group consists of eight participants being randomly matched over 200 periods.

The learning models had to prove whether they could replicate the aggregate behavior of the experimental population and whether they could explain the individual round-by-round behavior of single subjects. For the comparison with the aggregate behavior of the population, 500 simulation runs per game and learning model were conducted. As in the experiment, 200 rounds with random matching and four agents deciding as row players and four agents as column players were simulated. Our measure of predictive power for the aggregate population is the quadratic distance between observed relative frequencies in simulation runs and the mean frequencies observed in the experiments. For the comparison with the individuals' behavior, the models were applied to the history of each participant. Then the actual decisions of every round were compared with the predictions of the learning models given the actual subject's history. For each subject and round, a quadratic score, a measurement for the accuracy of a prediction, was calculated and averaged over rounds, subjects, and games.

For our comparisons with the aggregate distribution of the experimental population and the individual round-by-round behavior, we can conclude two main results:

Main Result 1. *The models of impulse-matching learning and action-sampling learning are able to replicate with simulations the aggregate behavior of the experimental population.*

The comparison of the four models yields the following order of predictive success from best to worst: impulse-matching learning, action-sampling learning, reinforcement learning, self-tuning EWA learning. Due to the high number of simulation runs, this order is statistically robust, all pairwise comparisons are at least significant on the 1% level.

The predominance of the new models, impulse-matching learning and action-sampling learning, over the established models of reinforcement learning and self-tuning EWA is stable over time and across the different game types (constant

sum and non-constant sum games). A further interesting result is that for reinforcement learning the quadratic distance to the data is about 22% lower if applied to the transformed matrixes instead to the original ones.

Main Result 2. *On the individual level, self-tuning EWA outperforms all other learning concepts in predicting individuals' round-by-round behavior.*

Overall, the models of action-sampling learning, reinforcement learning, impulse-matching learning, and self-tuning EWA perform better than simple randomization with 0.5 and the inertia benchmark does. But only the models of self-tuning EWA and impulse-matching learning perform significantly better in describing round-by-round behavior than the aggregated frequencies from the experiment. The mean quadratic score of self-tuning EWA is significantly above the ones of all other investigated learning concepts, and impulse-matching learning has the second-highest score.

The good performance of self-tuning EWA on the individual level is remarkable, and not the results of the free parameter. A non-parametric version of self-tuning EWA[12] yields an only marginal lower quadratic score (0.624 vs. 0.637) and still performs significantly better than impulse-matching learning. However, on the aggregate level, the distance of the simulated populations to the experimental data increases even further (0.079 vs. 0.111), and therefore impulse-matching learning performs significantly better in this domain (both Fisher–Pitman permutation test for paired replicates $p < 0.01$). In addition, our data shows that over time inertia increases, and therefore adding an inertia component to the learning models might improve their predictive power. For example, the model by Chen et al. (2011) provides a multi-parameter generalization of action-sampling learning that takes inertia and recency into account.

We conclude that impulse-matching learning produces good results across fields of applications, while self-tuning EWA organizes existing individual data exceptionally well. Our results suggest that if one is interested in the aggregate behavior of a population in a certain 2×2 game without any prior information upfront (i.e., no information for parameter estimates), simulations with impulse-matching learning result in an extremely good approximation. In addition, impulse-matching learning provides good predictions for round-by-round behavior, but in this domain it is clearly outperformed by self-tuning EWA. Obviously, self-tuning EWA interprets actual history very well, while it fails to generate accurate behavior in the simulations. Interestingly, only the two concepts of impulse-matching learning and self-tuning EWA have components of regret-based learning and only these two concepts outperform the empirical frequency benchmark.

Appendix A. Additional learning models

A.1. Impulse-balance learning

The algorithm for impulse-balance learning is very similar to the one of impulse-matching learning. Only the calculation of impulses differs, therefore Eq. (A.1) replaces Eq. (2) of impulse-matching learning. All other equations ((1) and (3)) remain the same. In contrast to impulse-matching, a player receives only actual impulses. This means the player does not receive an impulse if his action was a best reply against the other player's decision. Thus, for impulse-balance learning, the impulses from action j towards action i in period t are as follows:

$$r_i(t) = \begin{cases} \max[0, \pi_i - \pi_j] & \text{if the chosen action is } j, \\ 0 & \text{else,} \end{cases} \tag{A.1}$$

for $i, j = 1, 2$ and $i \neq j$. Again, π_i is the transformed payoff for action i given the matched agent's decision and π_j the one for action j. Afterwards, the impulse sums are updated with the new impulses. In the first round, all impulse sums are zero $R_1(1) = R_2(1) = 0$ and until both impulse sums are higher than zero the probabilities are fixed to $p_1(t) = p_2(t) = 0.5$.

In fact, both learning rules are so similar that they lead to the same stationary points in 2×2 games. To illustrate this, we take a look at the structure of the investigated experimental 2×2 games, as introduced by Selten and Chmura (2008).

	L	**R**
U	$a_L + c_L$ b_U	a_R $b_U + d_U$
D	a_L $d_D + d_D$	$a_R + c_R$ b_D

Fig. 14. The structure of the experimental 2×2 games.

Fig. 14 shows the transformed payoffs, the payoffs for the column players are shown in the lower right corner, and the payoff for the row players are shown in the upper left corner. The following equations must be fulfilled: $a_L, a_R, b_U, b_D \geqslant 0$ and $c_L, c_R, d_U, d_D > 0$. In the following, p_U and p_D are the probabilities of the row player for U and D and q_L and q_R are

[12] A non-parametric self-tuning EWA can be obtained by replacing the equation for calculating the probabilities with $p_i(t) = \frac{A_i(t-1)}{\sum_{j=1}^{2} A_j(t-1)}$. We thank an anonymous referee for this suggestion.

the probabilities for L and R by the column player. In the following, we will only look at the row player; the behavior in equilibrium of the column player is calculated analogously.

In case of impulse-balance equilibrium, the expected impulses for each of both strategies must be the same. Here, the row player receives only an impulse towards U for the proportion of plays in which he would choose down (given by p_D) and the other player at the same time would have chosen L (given by q_L). Therefore the expected impulse for U is given by $p_D q_L c_L$. Applying the same reasoning leads to $p_U q_R c_R$ as the expected impulse for D of the row player. Thus the impulse-balance equation, which must be fulfilled in equilibrium, is given as:

$$p_D q_L c_L = p_U q_R c_R.$$

In case of impulse-matching equilibrium, the row player always receives an impulse of c_L towards U if the column player plays L. The column player does so with a probability of q_L. In addition the row player always receives an impulse of c_R towards D if the column player chooses R. The column player plays R with a probability of q_R. Impulse-matching equilibrium is reached if the ratio of the two probabilities of U and D is the same as the ratio of expected impulses for U and D.

$$\frac{p_U}{p_D} = \frac{q_L c_L}{q_R c_R}.$$

By transforming we obtain the impulse-balance equation of impulse-balance equilibrium:

$$p_D q_L c_L = p_U q_R c_R.$$

Therefore, impulse-matching equilibrium and impulse-balance equilibrium have the same mixed stationary points in the case of the described 2×2 games. However, for other types of games, both concepts do not necessarily lead to the same stationary points.

A.2. Payoff-sampling learning

Payoff-sampling learning relates to the stationary concept of Osborne and Rubinstein (1998), which was first applied to experimental data in Selten and Chmura (2008). The behavioral explanation of the stationary concept is that a player chooses her action after sampling each alternative an equal number of times, picking the action that yields the highest payoff.

To implement this behavior, payoff-sampling learning is based on samples from earlier periods. The samples are randomly drawn with replacement and fixed sample sizes. The agent draws two samples $(s_1(t), s_2(t))$ of earlier payoffs, one sample with payoffs from rounds in which she chose action 1, and one with payoffs from rounds in which she chose action 2. In the following, $S_1(t)$ and $S_2(t)$ denote the payoff sums in $s_1(t)$ and $s_2(t)$, respectively.

After the drawing of the samples, the cumulated payoffs $S_1(t)$ and $S_2(t)$ are calculated and the action with the higher cumulated payoff is played, if there is one. If the samples of both possible actions have the same cumulated payoff, the agent randomizes with $p_1 = p_2 = 0.5$.

$$p_i(t) = \begin{cases} 1 & \text{if } S_i(t) > S_j(t), \\ 0.5 & \text{if } S_i(t) = S_j(t), \\ 0 & \text{else}, \end{cases} \tag{A.2}$$

for $i, j = 1, 2$ and $i \neq j$.

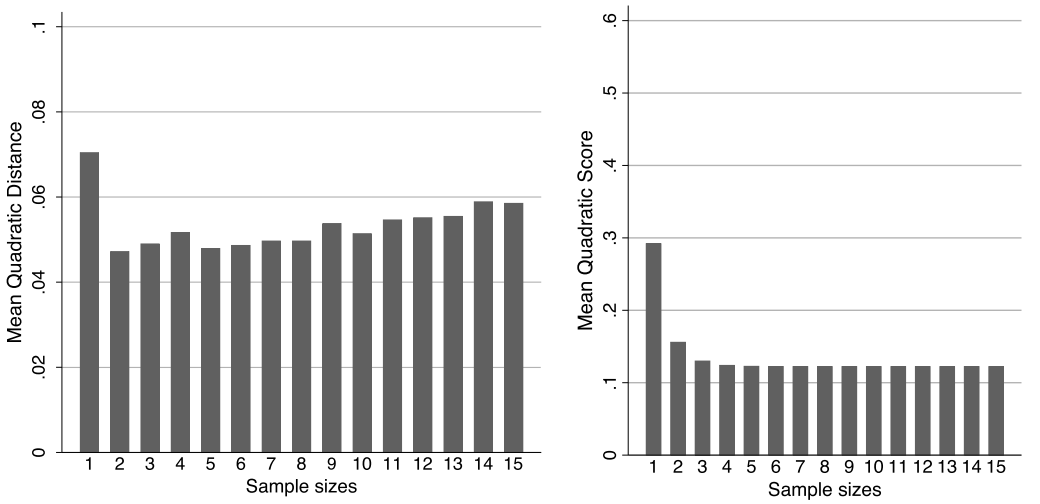


Fig. 15. Figure gives the mean quadratic distance (left) and the mean quadratic score (right) of payoff-sampling for different sample sizes.

As before, $p_i(t)$ is the probability of playing action i in period t. At the beginning and until positive payoffs for each action have been obtained at least once, the agent chooses both actions with equal probabilities, i.e., $p_1 = p_2 = 0.5$.

The optimal sample sizes are calculated analogously to the ones of action-sampling learning. Fig. 15 gives the mean quadratic distances and the mean quadratic scores for different sample sizes. For the quadratic distance, the optimal sample size is $n = 2$, and for the quadratic scores, it is $n = 1$.

A.3. Parameter-free self-tuning EWA

One of the referees suggested to test additionally a parameter-free version of self-tuning EWA and we are very grateful for this idea. This allows a better comparison between the similar action choice mechanisms of impulse-matching learning (based on impulses) and self-tuning EWA (based on attractions) without the bias of an estimated parameter. Therefore, the probability of playing action i in period t depending on the attractions is not calculated as a logit response function with the parameter λ. Instead, the probabilities are calculated with the relative attractions. Thus, Eq. (14) is replaced by Eq. (A.3):

$$p_i(t) = \frac{A_i(t-1)}{\sum_{j=1}^{2} A_j(t-1)}. \tag{A.3}$$

Everything else stays unchanged.

Appendix B. Comparison of all learning models

B.1. Simulations

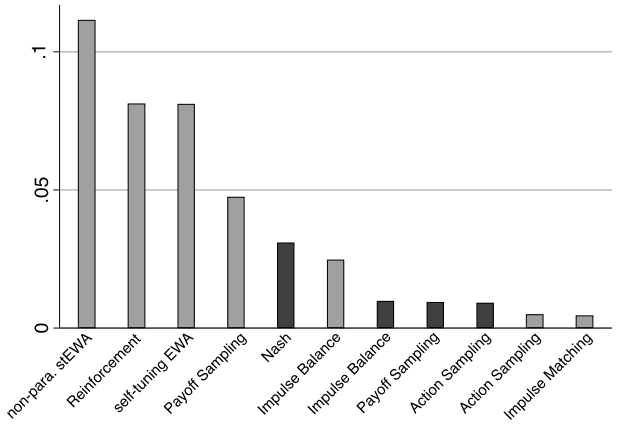


Fig. 16. Mean quadratic distance for all learning models and stationary concepts (dark grey).

B.2. Individual round-by-round behavior

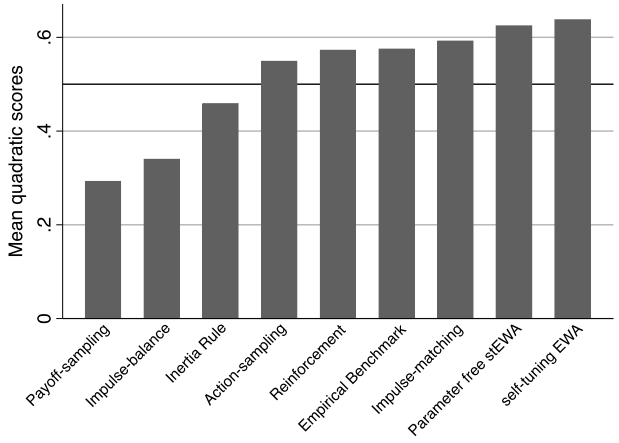


Fig. 17. Mean quadratic scores for all learning types.

Appendix C. Mean probabilities of simulations over time

C.1. Main learning models

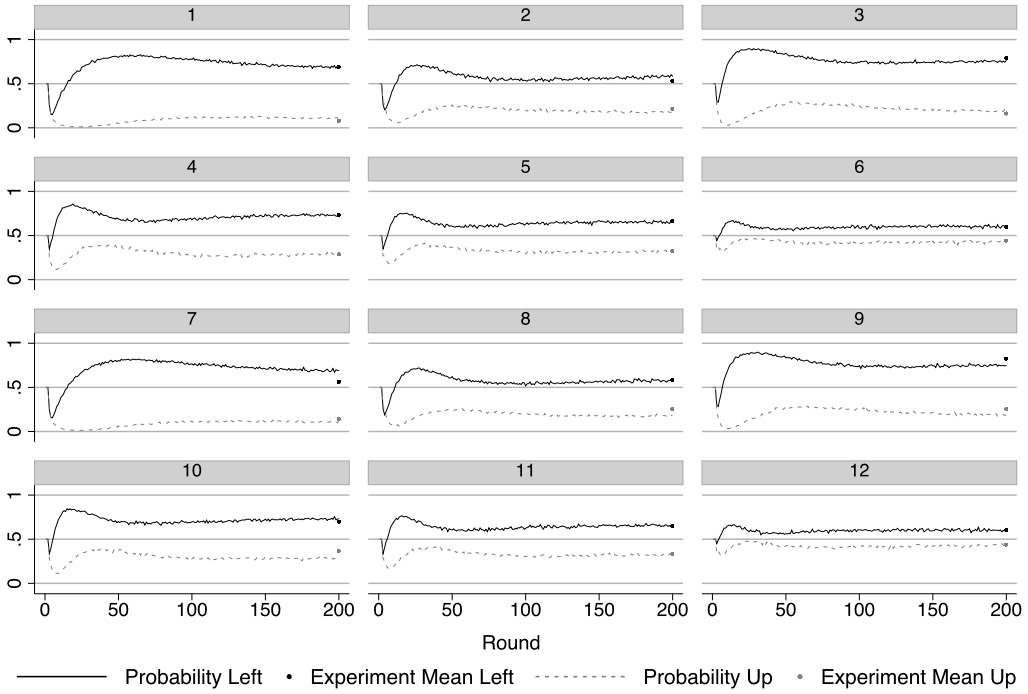


Fig. 18. Mean probabilities of *action-sampling learning* for left and up over 500 simulations runs per game, and frequencies in the experiments over 200 rounds.

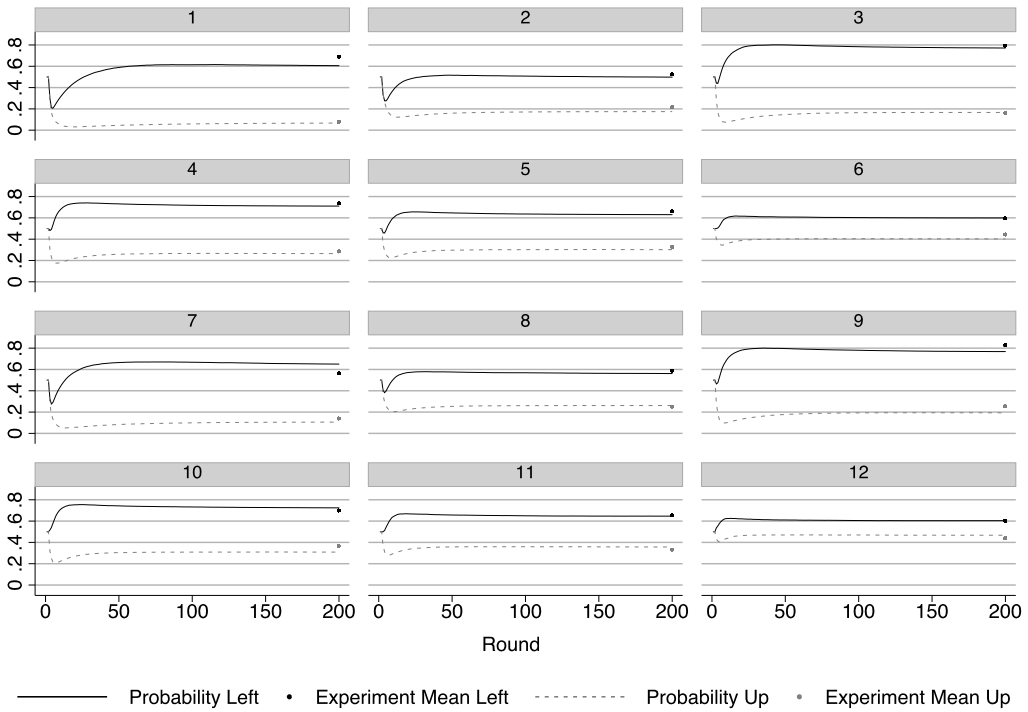


Fig. 19. Mean probabilities of *impulse-matching learning* for left and up over 500 simulations runs per game, as well as mean frequencies in the experiments over 200 rounds.

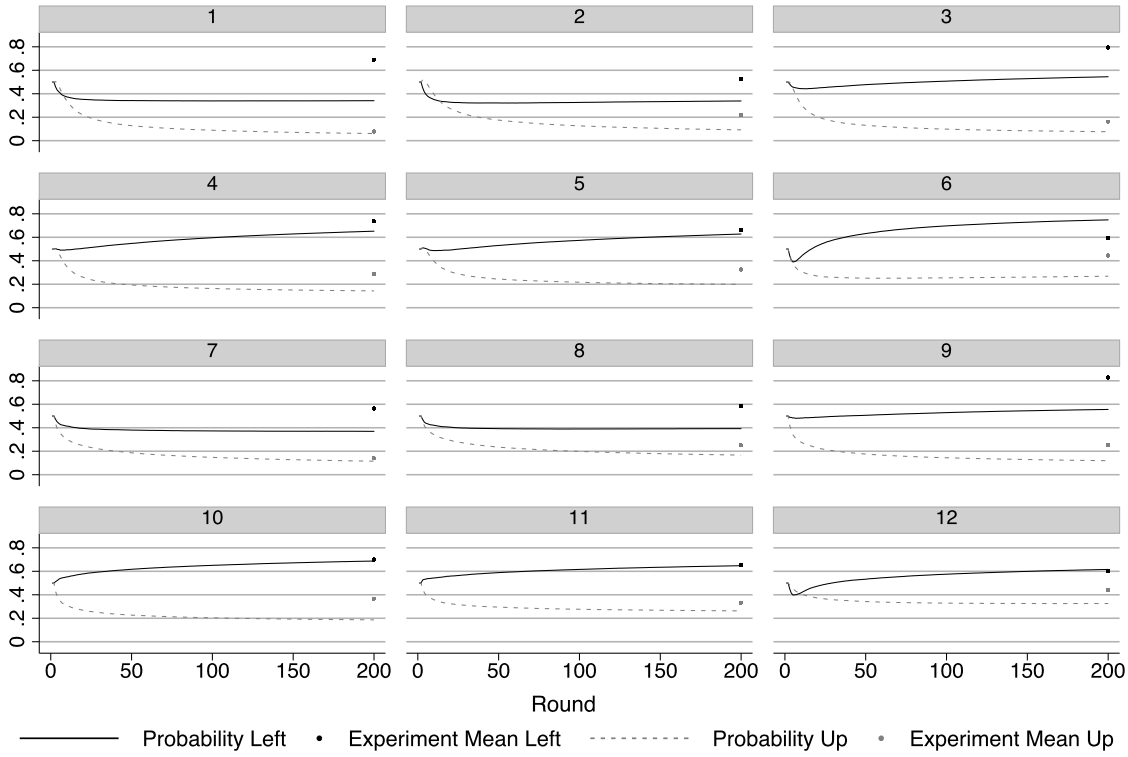


Fig. 20. Mean probabilities of *reinforcement learning* for left and up over 500 simulations runs per game, as well as mean frequencies in the experiments over 200 rounds.

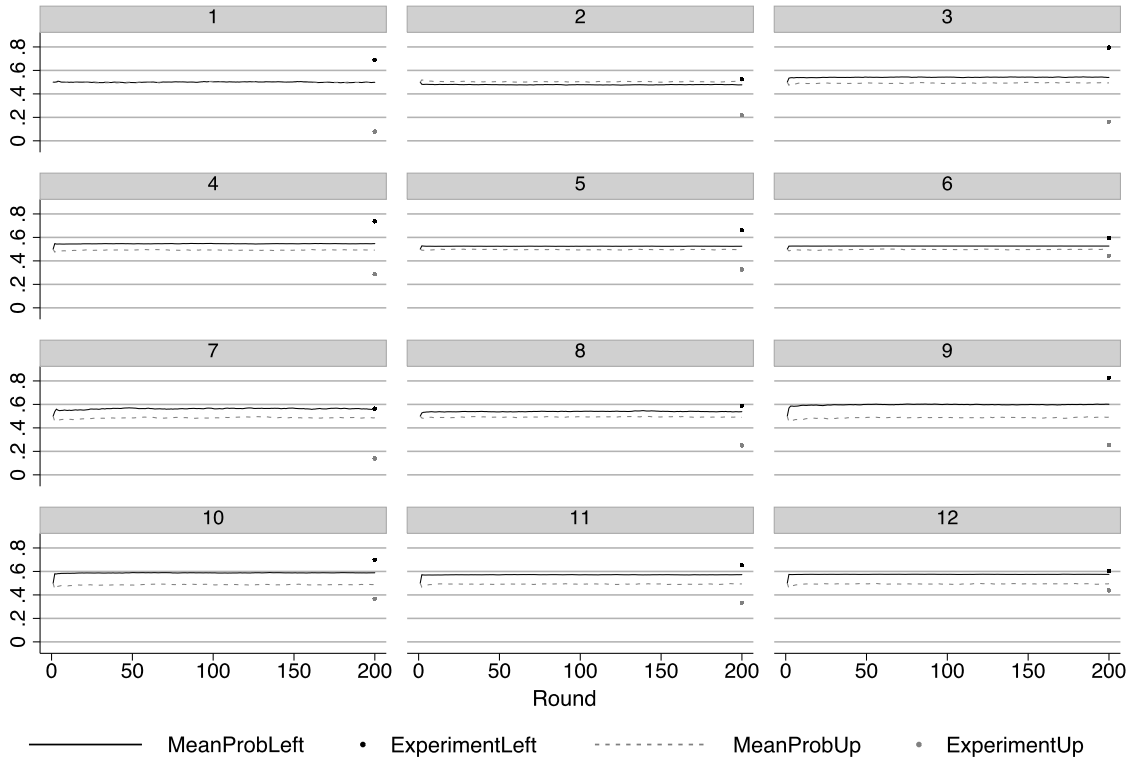


Fig. 21. Mean probabilities of *self-tuning EWA learning* for left and up over 500 simulations runs per game, as well as mean frequencies in the experiments over 200 rounds.

C.2. Excluded learning models

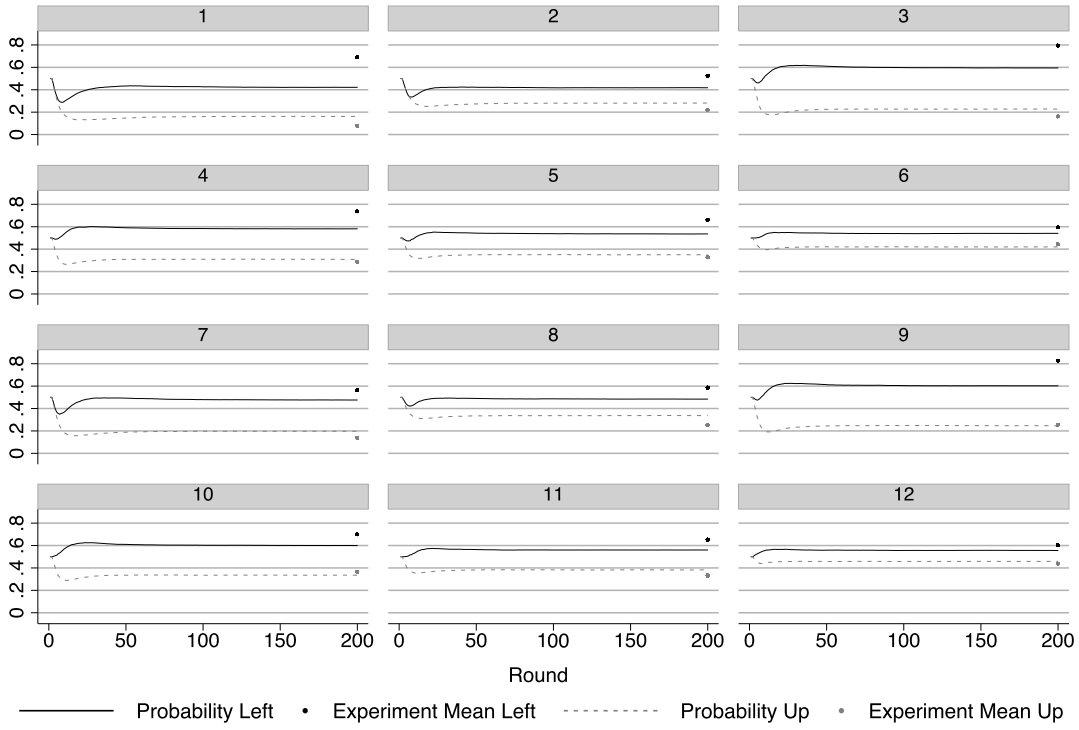


Fig. 22. Mean probabilities of *impulse-balance learning* for left and up over 500 simulations runs per game, as well as mean frequencies in the experiments over 200 rounds.

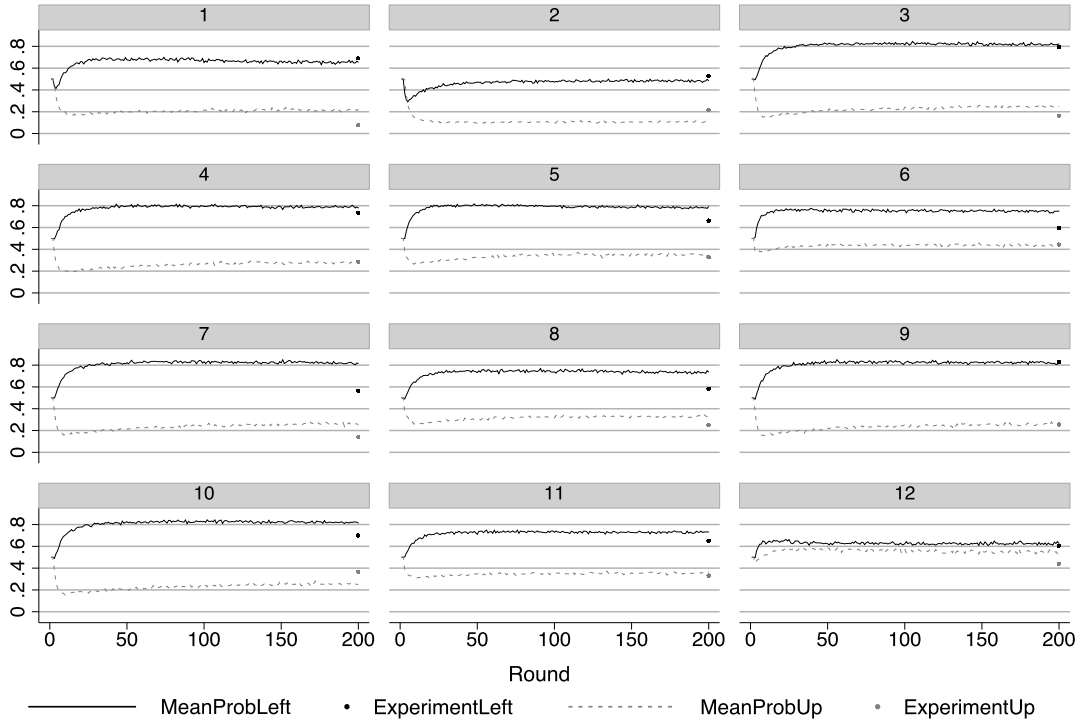


Fig. 23. Mean probabilities of *payoff-sampling learning* for left and up over 500 simulations runs per game, as well as mean frequencies in the experiments over 200 rounds.

Appendix D. Parameter estimates

Table 3
Quadratic distances of action-sampling learning for each game and each tested sample size. Line with the smallest mean is printed in bold.

n	Game number												Mean
	1	2	3	4	5	6	7	8	9	10	11	12	
1	0.2143	0.0805	0.2002	0.1023	0.0575	0.0127	0.1327	0.0708	0.1682	0.0578	0.0517	0.0153	0.0970
2	0.0660	0.0041	0.0839	0.0437	0.0214	0.0338	0.0205	0.0047	0.0903	0.0373	0.0185	0.0327	0.0381
3	0.0255	0.0035	0.0521	0.0069	0.0037	0.0129	0.0020	0.0059	0.0738	0.0007	0.0045	0.0150	0.0172
4	0.0103	0.0119	0.0181	0.0012	0.0198	0.0100	0.0021	0.0299	0.0131	0.0101	0.0161	0.0106	0.0128
5	0.0040	0.0088	0.0072	0.0061	0.0124	0.0032	0.0080	0.0252	0.0122	0.0249	0.0105	0.0024	0.0104
6	0.0018	0.0045	0.0030	0.0107	0.0010	0.0067	0.0152	0.0010	0.0141	0.0030	0.0008	0.0074	0.0058
7	0.0029	0.0087	0.0021	0.0050	0.0022	0.0008	0.0220	0.0047	0.0175	0.0079	0.0027	0.0007	0.0064
8	0.0053	0.0010	0.0026	0.0049	0.0098	0.0056	0.0279	0.0074	0.0040	0.0152	0.0084	0.0045	0.0080
9	0.0097	0.0036	0.0018	0.0006	0.0011	0.0009	0.0326	0.0085	0.0070	0.0075	0.0007	0.0013	0.0063
10	0.0144	0.0080	0.0024	0.0014	0.0021	0.0023	0.0348	0.0114	0.0113	0.0143	0.0023	0.0018	0.0089
11	0.0030	0.0041	0.0012	0.0009	0.0015	0.0017	0.0284	0.0070	0.0070	0.0090	0.0012	0.0013	0.0055
12	**0.0020**	**0.0032**	**0.0040**	**0.0012**	**0.0011**	**0.0012**	**0.0253**	**0.0039**	**0.0070**	**0.0057**	**0.0009**	**0.0009**	**0.0047**
13	0.0033	0.0062	0.0022	0.0011	0.0019	0.0046	0.0315	0.0062	0.0075	0.0108	0.0021	0.0037	0.0068
14	0.0044	0.0098	0.0012	0.0011	0.0013	0.0008	0.0370	0.0093	0.0090	0.0078	0.0021	0.0010	0.0071
15	0.0099	0.0050	0.0034	0.0012	0.0012	0.0027	0.0420	0.0098	0.0033	0.0123	0.0008	0.0021	0.0078

Table 4
Quadratic distances of payoff-sampling learning for each game and each tested sample size. Line with the smallest mean is printed in bold.

n	Game number												Mean
	1	2	3	4	5	6	7	8	9	10	11	12	
1	0.0577	0.0579	0.0775	0.0449	0.0581	0.0852	0.1550	0.0995	0.0558	0.0445	0.0614	0.0475	0.0704
2	**0.0306**	**0.0498**	**0.0298**	**0.0320**	**0.0375**	**0.0643**	**0.0885**	**0.0635**	**0.0344**	**0.0573**	**0.0388**	**0.0397**	**0.0472**
3	0.0177	0.0543	0.0237	0.0357	0.0464	0.0669	0.0844	0.0543	0.0340	0.0564	0.0595	0.0546	0.0490
4	0.0110	0.0567	0.0393	0.0381	0.0484	0.0710	0.0971	0.0633	0.0394	0.0528	0.0507	0.0520	0.0517
5	0.0096	0.0530	0.0396	0.0327	0.0470	0.0711	0.0845	0.0600	0.0259	0.0572	0.0390	0.0552	0.0479
6	0.0082	0.0603	0.0304	0.0393	0.0508	0.0672	0.0635	0.0684	0.0312	0.0506	0.0591	0.0544	0.0486
7	0.0079	0.0624	0.0415	0.0370	0.0474	0.0712	0.0804	0.0593	0.0301	0.0515	0.0534	0.0537	0.0496
8	0.0089	0.0597	0.0352	0.0367	0.0474	0.0699	0.0795	0.0495	0.0413	0.0552	0.0555	0.0569	0.0496
9	0.0083	0.0605	0.0338	0.0425	0.0490	0.0746	0.0917	0.0640	0.0426	0.0561	0.0605	0.0615	0.0538
10	0.0129	0.0639	0.0353	0.0453	0.0498	0.0695	0.0813	0.0537	0.0392	0.0535	0.0537	0.0585	0.0514
11	0.0217	0.0599	0.0428	0.0458	0.0570	0.0726	0.0845	0.0599	0.0406	0.0511	0.0638	0.0557	0.0546
12	0.0224	0.0644	0.0331	0.0386	0.0497	0.0740	0.0769	0.0681	0.0425	0.0609	0.0616	0.0692	0.0551
13	0.0218	0.0680	0.0477	0.0500	0.0510	0.0751	0.0758	0.0636	0.0382	0.0563	0.0590	0.0588	0.0554
14	0.0210	0.0655	0.0464	0.0551	0.0547	0.0793	0.0909	0.0558	0.0500	0.0635	0.0613	0.0628	0.0589
15	0.0197	0.0614	0.0462	0.0500	0.0546	0.0704	0.0992	0.0632	0.0475	0.0638	0.0639	0.0621	0.0585

Table 5
Quadratic distances of self-tuning EWA learning for each game, for $0 \geqslant \lambda \geqslant 10$ in steps of 0.05. Line with the smallest mean is printed in bold.

λ	Game number												Mean
	1	2	3	4	5	6	7	8	9	10	11	12	
0	0.2141	0.0817	0.1994	0.1029	0.0578	0.0132	0.1335	0.0705	0.1670	0.0582	0.0521	0.0153	0.0971
0.05	0.2133	0.0829	0.1914	0.0952	0.0552	0.0115	0.1248	0.0668	0.1479	0.0485	0.0462	0.0115	0.0913
0.1	0.2145	0.0853	0.1850	0.0904	0.0531	0.0107	0.1187	0.0643	0.1356	0.0416	0.0414	0.0088	0.0875
0.15	0.2141	0.0851	0.1798	0.0860	0.0508	0.0099	0.1167	0.0625	0.1236	0.0360	0.0371	0.0069	0.0841
0.2	**0.2141**	**0.0862**	**0.1766**	**0.0834**	**0.0503**	**0.0092**	**0.1146**	**0.0617**	**0.1159**	**0.0320**	**0.0349**	**0.0055**	**0.0820**
0.25	0.2146	0.0863	0.1736	0.0800	0.0492	0.0086	0.1191	0.0621	0.1104	0.0295	0.0336	0.0047	0.0810
0.3	0.2172	0.0859	0.1728	0.0783	0.0479	0.0082	0.1291	0.0638	0.1087	0.0282	0.0328	0.0046	0.0815
0.35	0.2195	0.0855	0.1738	0.0784	0.0484	0.0078	0.1491	0.0681	0.1101	0.0282	0.0351	0.0058	0.0842
0.4	0.2272	0.0831	0.1753	0.0782	0.0483	0.0077	0.2162	0.0792	0.1190	0.0321	0.0384	0.0078	0.0927
0.45	0.2527	0.0830	0.1828	0.0811	0.0493	0.0080	0.2952	0.0936	0.1452	0.0384	0.0454	0.0116	0.1072
0.5	0.3101	0.0789	0.1956	0.0845	0.0506	0.0083	0.3874	0.1324	0.1953	0.0531	0.0548	0.0178	0.1307
0.55	0.4119	0.0780	0.2217	0.0920	0.0515	0.0091	0.4488	0.1970	0.2426	0.0793	0.0710	0.0251	0.1607
0.6	0.4454	0.0804	0.2685	0.1043	0.0541	0.0096	0.4832	0.2491	0.2946	0.1125	0.0886	0.0369	0.1856
0.65	0.5173	0.0959	0.3170	0.1235	0.0586	0.0111	0.5315	0.3070	0.3423	0.1574	0.1192	0.0519	0.2194
0.7	0.5361	0.1229	0.3653	0.1500	0.0650	0.0128	0.5577	0.3684	0.3866	0.1972	0.1516	0.0713	0.2487
0.75	0.5766	0.1714	0.4162	0.1825	0.0701	0.0144	0.5666	0.3939	0.4186	0.2392	0.1899	0.0928	0.2777
0.8	0.5973	0.2228	0.4574	0.2225	0.0773	0.0182	0.5825	0.4201	0.4429	0.2678	0.2292	0.1156	0.3045
0.85	0.6203	0.2735	0.4948	0.2630	0.0881	0.0208	0.5991	0.4552	0.4623	0.3019	0.2725	0.1434	0.3329
0.9	0.6270	0.3004	0.5213	0.2972	0.1033	0.0246	0.5951	0.4666	0.4816	0.3192	0.3080	0.1710	0.3513
0.95	0.6355	0.3281	0.5354	0.3143	0.1205	0.0284	0.6129	0.4933	0.4966	0.3410	0.3353	0.1990	0.3700
1	0.6683	0.3541	0.5647	0.3507	0.1422	0.0328	0.6169	0.5103	0.5089	0.3577	0.3670	0.2217	0.3913

Table 5 (*continued*)

λ	Game number												Mean
	1	2	3	4	5	6	7	8	9	10	11	12	
1.05	0.6549	0.3660	0.5674	0.3721	0.1597	0.0389	0.6067	0.5028	0.5166	0.3629	0.3907	0.2482	0.3989
1.1	0.6669	0.3805	0.5811	0.3860	0.1899	0.0450	0.6210	0.5179	0.5293	0.3815	0.4077	0.2674	0.4145
1.15	0.6510	0.4054	0.5851	0.4044	0.2118	0.0514	0.6324	0.5332	0.5368	0.3921	0.4275	0.2883	0.4266
1.2	0.6692	0.3997	0.6050	0.4141	0.2309	0.0586	0.6089	0.5342	0.5410	0.3975	0.4381	0.3063	0.4336
1.25	0.6712	0.4257	0.6111	0.4312	0.2606	0.0665	0.6081	0.5517	0.5444	0.4076	0.4502	0.3199	0.4457
1.3	0.6713	0.4367	0.6168	0.4396	0.2871	0.0750	0.6154	0.5425	0.5548	0.4090	0.4619	0.3344	0.4537
1.35	0.6750	0.4333	0.6245	0.4518	0.3031	0.0832	0.6105	0.5439	0.5540	0.4172	0.4671	0.3455	0.4591
1.4	0.6737	0.4487	0.6285	0.4670	0.3234	0.0916	0.6131	0.5552	0.5558	0.4245	0.4767	0.3544	0.4677
1.45	0.6617	0.4614	0.6258	0.4639	0.3373	0.1036	0.6143	0.5418	0.5645	0.4259	0.4874	0.3636	0.4709
1.5	0.6741	0.4483	0.6321	0.4779	0.3529	0.1128	0.6160	0.5430	0.5640	0.4313	0.4898	0.3720	0.4762
1.55	0.6699	0.4659	0.6393	0.4822	0.3616	0.1218	0.6251	0.5523	0.5653	0.4360	0.4877	0.3786	0.4821
1.6	0.6784	0.4727	0.6459	0.4853	0.3772	0.1335	0.6532	0.5449	0.5694	0.4391	0.4994	0.3857	0.4904
1.65	0.6731	0.4546	0.6447	0.4876	0.3873	0.1426	0.6351	0.5743	0.5727	0.4429	0.5026	0.3922	0.4925
1.7	0.6843	0.4698	0.6498	0.4944	0.3942	0.1527	0.6169	0.5579	0.5783	0.4439	0.5008	0.3962	0.4949
1.75	0.6818	0.4678	0.6507	0.4966	0.4069	0.1629	0.6135	0.5568	0.5753	0.4470	0.5056	0.4016	0.4972
1.8	0.6762	0.4977	0.6602	0.5031	0.4051	0.1725	0.6224	0.5664	0.5790	0.4448	0.5077	0.4043	0.5033
1.85	0.6660	0.4854	0.6566	0.5014	0.4135	0.1810	0.6105	0.5634	0.5865	0.4516	0.5118	0.4088	0.5030
1.9	0.6932	0.4722	0.6612	0.5061	0.4162	0.1922	0.6418	0.5620	0.5822	0.4505	0.5108	0.4116	0.5083
1.95	0.6929	0.4902	0.6622	0.5109	0.4245	0.2013	0.6300	0.5684	0.5837	0.4497	0.5118	0.4141	0.5116
2	0.6593	0.4927	0.6659	0.5130	0.4245	0.2084	0.6183	0.5608	0.5806	0.4537	0.5139	0.4178	0.5091
2.05	0.6861	0.4835	0.6692	0.5066	0.4283	0.2167	0.6239	0.5756	0.5843	0.4551	0.5129	0.4195	0.5135
2.1	0.6762	0.4926	0.6656	0.5111	0.4321	0.2252	0.6114	0.5550	0.5893	0.4555	0.5131	0.4228	0.5125
2.15	0.6751	0.5101	0.6673	0.5180	0.4331	0.2355	0.6229	0.5695	0.5858	0.4577	0.5217	0.4256	0.5185
2.2	0.6876	0.5083	0.6679	0.5212	0.4411	0.2423	0.6338	0.5591	0.5883	0.4555	0.5202	0.4271	0.5210
2.25	0.6914	0.4847	0.6680	0.5182	0.4443	0.2486	0.6298	0.5633	0.5877	0.4605	0.5189	0.4288	0.5204
2.3	0.6885	0.5082	0.6654	0.5233	0.4555	0.2564	0.6356	0.5651	0.5895	0.4610	0.5193	0.4319	0.5250
2.35	0.6675	0.4973	0.6691	0.5254	0.4457	0.2629	0.6259	0.5521	0.5875	0.4599	0.5214	0.4333	0.5207
2.4	0.6807	0.5087	0.6727	0.5237	0.4471	0.2684	0.6605	0.5621	0.5899	0.4651	0.5240	0.4332	0.5280
2.45	0.6838	0.5169	0.6745	0.5262	0.4560	0.2742	0.6405	0.5679	0.5937	0.4627	0.5280	0.4357	0.5300
2.5	0.6846	0.4977	0.6697	0.5287	0.4612	0.2788	0.6361	0.5695	0.5863	0.4604	0.5293	0.4372	0.5283
2.55	0.6875	0.5153	0.6757	0.5291	0.4665	0.2849	0.6197	0.5625	0.5977	0.4654	0.5263	0.4389	0.5308
2.6	0.7064	0.5000	0.6671	0.5324	0.4612	0.2875	0.6395	0.5659	0.5908	0.4646	0.5275	0.4398	0.5319
2.65	0.6907	0.5125	0.6713	0.5327	0.4611	0.2930	0.6349	0.5653	0.5921	0.4636	0.5261	0.4417	0.5321
2.7	0.6761	0.5012	0.6789	0.5332	0.4653	0.2988	0.6047	0.5686	0.5895	0.4673	0.5283	0.4422	0.5295
2.75	0.6942	0.4888	0.6718	0.5346	0.4650	0.3036	0.6145	0.5800	0.5935	0.4646	0.5259	0.4436	0.5317
2.8	0.6661	0.4891	0.6754	0.5351	0.4780	0.3065	0.6332	0.5769	0.5934	0.4636	0.5314	0.4434	0.5327
2.85	0.6818	0.5109	0.6756	0.5374	0.4723	0.3106	0.6307	0.5854	0.5955	0.4681	0.5304	0.4445	0.5369
2.9	0.6743	0.4782	0.6792	0.5402	0.4720	0.3143	0.6237	0.5691	0.5935	0.4692	0.5314	0.4450	0.5325
2.95	0.6862	0.5027	0.6759	0.5346	0.4824	0.3182	0.6362	0.5625	0.5919	0.4717	0.5293	0.4466	0.5365
3	0.6675	0.4960	0.6785	0.5376	0.4738	0.3203	0.6317	0.5674	0.5966	0.4688	0.5282	0.4466	0.5344
3.05	0.6884	0.5088	0.6770	0.5424	0.4779	0.3246	0.6204	0.5663	0.5942	0.4672	0.5253	0.4475	0.5367
3.1	0.6885	0.4894	0.6759	0.5407	0.4852	0.3266	0.6258	0.5683	0.5901	0.4715	0.5323	0.4492	0.5370
3.15	0.7046	0.5049	0.6768	0.5397	0.4795	0.3319	0.6116	0.5694	0.5889	0.4691	0.5318	0.4490	0.5381
3.2	0.6819	0.5058	0.6753	0.5408	0.4800	0.3349	0.6123	0.5755	0.5923	0.4698	0.5304	0.4493	0.5373
3.25	0.6907	0.4957	0.6764	0.5433	0.4900	0.3375	0.6334	0.5664	0.5932	0.4679	0.5279	0.4505	0.5394
3.3	0.6817	0.5132	0.6784	0.5424	0.4882	0.3400	0.6297	0.5779	0.5971	0.4725	0.5289	0.4503	0.5417
3.35	0.6831	0.5034	0.6783	0.5434	0.4890	0.3413	0.6270	0.5537	0.5954	0.4700	0.5300	0.4512	0.5388
3.4	0.6687	0.5037	0.6779	0.5433	0.4910	0.3442	0.6463	0.5624	0.5963	0.4726	0.5327	0.4508	0.5408
3.45	0.6851	0.5271	0.6772	0.5458	0.4932	0.3473	0.6179	0.5626	0.5914	0.4725	0.5289	0.4519	0.5417
3.5	0.6983	0.5103	0.6739	0.5456	0.4963	0.3487	0.6387	0.5518	0.6011	0.4738	0.5375	0.4527	0.5440
3.55	0.6895	0.5127	0.6783	0.5439	0.4955	0.3518	0.6251	0.5601	0.5918	0.4737	0.5310	0.4522	0.5421
3.6	0.7086	0.5042	0.6789	0.5447	0.4948	0.3538	0.6405	0.5707	0.5962	0.4741	0.5306	0.4525	0.5458
3.65	0.6801	0.5223	0.6768	0.5465	0.4947	0.3566	0.6244	0.5540	0.5925	0.4752	0.5297	0.4534	0.5422
3.7	0.6863	0.4810	0.6782	0.5481	0.4981	0.3585	0.6208	0.5723	0.5962	0.4733	0.5326	0.4538	0.5416
3.75	0.6727	0.5206	0.6781	0.5480	0.4951	0.3600	0.6002	0.5663	0.5999	0.4793	0.5355	0.4536	0.5424
3.8	0.6850	0.5117	0.6796	0.5499	0.4978	0.3622	0.6145	0.5630	0.5920	0.4753	0.5313	0.4535	0.5430
3.85	0.6721	0.5122	0.6796	0.5475	0.4976	0.3643	0.6168	0.5691	0.5891	0.4746	0.5320	0.4538	0.5424
3.9	0.6963	0.5303	0.6771	0.5518	0.5003	0.3656	0.6017	0.5675	0.5935	0.4749	0.5328	0.4540	0.5455
3.95	0.6759	0.5056	0.6779	0.5514	0.5051	0.3672	0.5999	0.5615	0.5969	0.4736	0.5319	0.4551	0.5418
4	0.6775	0.5214	0.6768	0.5487	0.5047	0.3701	0.6191	0.5717	0.5891	0.4756	0.5342	0.4556	0.5454
4.05	0.6832	0.5152	0.6739	0.5488	0.5072	0.3720	0.6177	0.5596	0.5913	0.4808	0.5348	0.4549	0.5450
4.1	0.6937	0.5175	0.6769	0.5502	0.5050	0.3728	0.6269	0.5541	0.5983	0.4753	0.5367	0.4556	0.5469
4.15	0.6730	0.5118	0.6805	0.5505	0.5079	0.3750	0.5924	0.5613	0.5952	0.4772	0.5339	0.4560	0.5429
4.2	0.6694	0.5097	0.6802	0.5524	0.5052	0.3771	0.6071	0.5627	0.5977	0.4780	0.5334	0.4563	0.5441
4.25	0.6753	0.4976	0.6790	0.5529	0.5093	0.3786	0.6275	0.5653	0.5947	0.4776	0.5342	0.4560	0.5457
4.3	0.6792	0.5106	0.6759	0.5522	0.5054	0.3800	0.6101	0.5755	0.6010	0.4777	0.5327	0.4555	0.5463
4.35	0.6805	0.5217	0.6787	0.5518	0.5141	0.3831	0.6149	0.5632	0.5938	0.4787	0.5346	0.4561	0.5476
4.4	0.6972	0.5241	0.6812	0.5541	0.5057	0.3835	0.6116	0.5634	0.5956	0.4779	0.5331	0.4555	0.5486

(*continued on next page*)

Table 5 (*continued*)

λ	Game number												Mean
	1	2	3	4	5	6	7	8	9	10	11	12	
4.45	0.6888	0.5245	0.6784	0.5520	0.5062	0.3845	0.6105	0.5607	0.5969	0.4766	0.5344	0.4563	0.5475
4.5	0.6781	0.5327	0.6768	0.5540	0.5073	0.3864	0.6106	0.5642	0.5913	0.4789	0.5319	0.4563	0.5474
4.55	0.6837	0.5124	0.6779	0.5513	0.5084	0.3879	0.6020	0.5743	0.5935	0.4788	0.5334	0.4560	0.5466
4.6	0.6842	0.5034	0.6794	0.5538	0.5097	0.3900	0.6348	0.5654	0.6040	0.4765	0.5307	0.4564	0.5490
4.65	0.6767	0.5302	0.6752	0.5543	0.5130	0.3910	0.6307	0.5676	0.5963	0.4780	0.5315	0.4561	0.5501
4.7	0.6830	0.5243	0.6769	0.5536	0.5120	0.3926	0.6158	0.5647	0.5936	0.4795	0.5375	0.4566	0.5492
4.75	0.6793	0.5080	0.6766	0.5537	0.5123	0.3945	0.6037	0.5525	0.5989	0.4775	0.5358	0.4563	0.5458
4.8	0.6826	0.5066	0.6745	0.5562	0.5122	0.3953	0.6032	0.5594	0.5948	0.4757	0.5370	0.4574	0.5462
4.85	0.6923	0.5161	0.6779	0.5533	0.5100	0.3968	0.6127	0.5625	0.5957	0.4803	0.5343	0.4568	0.5491
4.9	0.6675	0.5221	0.6757	0.5553	0.5122	0.3982	0.6083	0.5682	0.5972	0.4794	0.5347	0.4569	0.5480
4.95	0.6955	0.5085	0.6789	0.5559	0.5176	0.3999	0.6225	0.5645	0.5937	0.4765	0.5360	0.4571	0.5505
5	0.6951	0.5255	0.6801	0.5541	0.5131	0.4007	0.6276	0.5637	0.6010	0.4808	0.5321	0.4567	0.5525
5.05	0.6909	0.5035	0.6772	0.5562	0.5167	0.4021	0.5991	0.5752	0.5963	0.4825	0.5337	0.4572	0.5492
5.1	0.6769	0.4975	0.6807	0.5548	0.5146	0.4030	0.6282	0.5703	0.5987	0.4807	0.5355	0.4573	0.5499
5.15	0.6893	0.5024	0.6798	0.5541	0.5216	0.4046	0.6010	0.5599	0.5957	0.4762	0.5362	0.4572	0.5482
5.2	0.6779	0.5290	0.6805	0.5531	0.5170	0.4051	0.6171	0.5710	0.5940	0.4795	0.5356	0.4573	0.5514
5.25	0.6991	0.5233	0.6796	0.5532	0.5191	0.4062	0.6179	0.5587	0.5946	0.4785	0.5374	0.4572	0.5521
5.3	0.6928	0.5229	0.6776	0.5546	0.5154	0.4078	0.6374	0.5604	0.5942	0.4817	0.5359	0.4575	0.5532
5.35	0.6930	0.4855	0.6739	0.5564	0.5160	0.4100	0.6369	0.5764	0.5885	0.4787	0.5333	0.4575	0.5505
5.4	0.7003	0.5377	0.6778	0.5579	0.5186	0.4104	0.6400	0.5643	0.5974	0.4793	0.5354	0.4581	0.5564
5.45	0.6911	0.5066	0.6767	0.5554	0.5185	0.4107	0.5957	0.5604	0.5900	0.4800	0.5333	0.4580	0.5480
5.5	0.6822	0.5400	0.6781	0.5557	0.5181	0.4124	0.6094	0.5532	0.5946	0.4800	0.5364	0.4564	0.5514
5.55	0.6979	0.5195	0.6744	0.5572	0.5213	0.4142	0.6186	0.5663	0.5971	0.4814	0.5368	0.4580	0.5536
5.6	0.6815	0.4897	0.6763	0.5568	0.5219	0.4143	0.6155	0.5615	0.5974	0.4816	0.5361	0.4576	0.5492
5.65	0.6850	0.5164	0.6769	0.5568	0.5189	0.4148	0.6374	0.5652	0.5975	0.4810	0.5358	0.4573	0.5536
5.7	0.6798	0.5389	0.6769	0.5559	0.5168	0.4160	0.6201	0.5589	0.5966	0.4818	0.5359	0.4574	0.5529
5.75	0.7158	0.5231	0.6769	0.5587	0.5205	0.4175	0.6193	0.5700	0.6011	0.4779	0.5371	0.4575	0.5563
5.8	0.6826	0.5057	0.6772	0.5569	0.5214	0.4184	0.5907	0.5590	0.5998	0.4825	0.5379	0.4576	0.5491
5.85	0.6727	0.5132	0.6784	0.5576	0.5218	0.4185	0.6153	0.5521	0.5959	0.4830	0.5352	0.4577	0.5501
5.9	0.6801	0.5171	0.6779	0.5586	0.5244	0.4200	0.6081	0.5702	0.6024	0.4827	0.5380	0.4572	0.5531
5.95	0.7059	0.5028	0.6799	0.5565	0.5229	0.4208	0.6361	0.5526	0.5958	0.4774	0.5354	0.4577	0.5536
6	0.6865	0.5041	0.6796	0.5584	0.5212	0.4214	0.6219	0.5516	0.5915	0.4773	0.5346	0.4582	0.5505
6.05	0.6817	0.5273	0.6767	0.5589	0.5230	0.4224	0.6363	0.5542	0.5986	0.4791	0.5354	0.4577	0.5543
6.1	0.6888	0.5239	0.6788	0.5584	0.5218	0.4227	0.6113	0.5503	0.5965	0.4810	0.5370	0.4582	0.5524
6.15	0.6819	0.5278	0.6768	0.5567	0.5236	0.4240	0.6198	0.5771	0.5906	0.4785	0.5384	0.4576	0.5544
6.2	0.6917	0.5123	0.6756	0.5592	0.5250	0.4244	0.6259	0.5372	0.5965	0.4772	0.5351	0.4576	0.5515
6.25	0.6838	0.5090	0.6771	0.5587	0.5262	0.4248	0.6047	0.5637	0.5948	0.4823	0.5361	0.4584	0.5516
6.3	0.6823	0.5209	0.6795	0.5574	0.5255	0.4262	0.5923	0.5602	0.5919	0.4826	0.5368	0.4576	0.5511
6.35	0.6883	0.5210	0.6763	0.5577	0.5235	0.4269	0.6164	0.5589	0.5994	0.4790	0.5359	0.4583	0.5535
6.4	0.6942	0.5115	0.6791	0.5566	0.5220	0.4266	0.6187	0.5538	0.5940	0.4797	0.5359	0.4583	0.5525
6.45	0.6781	0.5183	0.6782	0.5592	0.5236	0.4284	0.6294	0.5741	0.5942	0.4812	0.5386	0.4580	0.5551
6.5	0.6956	0.5192	0.6726	0.5559	0.5245	0.4293	0.6113	0.5542	0.5962	0.4800	0.5379	0.4576	0.5529
6.55	0.6923	0.5246	0.6770	0.5547	0.5265	0.4296	0.6283	0.5633	0.5931	0.4826	0.5373	0.4582	0.5556
6.6	0.6922	0.5215	0.6774	0.5571	0.5246	0.4311	0.6301	0.5568	0.5913	0.4808	0.5384	0.4579	0.5549
6.65	0.6861	0.5282	0.6783	0.5577	0.5258	0.4310	0.6266	0.5692	0.5970	0.4847	0.5358	0.4580	0.5565
6.7	0.6818	0.5183	0.6789	0.5571	0.5257	0.4318	0.6166	0.5597	0.5910	0.4788	0.5369	0.4578	0.5529
6.75	0.6801	0.5168	0.6738	0.5563	0.5278	0.4321	0.5995	0.5644	0.5957	0.4798	0.5377	0.4582	0.5518
6.8	0.6720	0.5192	0.6779	0.5546	0.5290	0.4327	0.6018	0.5558	0.5938	0.4803	0.5379	0.4584	0.5511
6.85	0.6974	0.5094	0.6775	0.5573	0.5266	0.4335	0.6105	0.5491	0.5945	0.4820	0.5370	0.4582	0.5528
6.9	0.6958	0.4949	0.6767	0.5559	0.5240	0.4336	0.6021	0.5597	0.5971	0.4804	0.5364	0.4580	0.5512
6.95	0.6875	0.5452	0.6816	0.5581	0.5283	0.4345	0.6320	0.5577	0.5967	0.4821	0.5364	0.4581	0.5582
7	0.7028	0.5257	0.6781	0.5561	0.5263	0.4352	0.6054	0.5599	0.5971	0.4774	0.5380	0.4582	0.5550
7.05	0.6831	0.5323	0.6786	0.5587	0.5252	0.4356	0.6324	0.5606	0.5994	0.4770	0.5389	0.4584	0.5567
7.1	0.6738	0.5319	0.6734	0.5585	0.5266	0.4358	0.6205	0.5700	0.5956	0.4784	0.5375	0.4581	0.5550
7.15	0.6982	0.5078	0.6757	0.5588	0.5249	0.4369	0.6151	0.5587	0.5935	0.4811	0.5381	0.4585	0.5539
7.2	0.6859	0.5398	0.6801	0.5591	0.5280	0.4375	0.6283	0.5574	0.5990	0.4800	0.5383	0.4576	0.5576
7.25	0.6875	0.5437	0.6754	0.5569	0.5274	0.4375	0.6121	0.5586	0.5949	0.4792	0.5368	0.4587	0.5557
7.3	0.6733	0.5103	0.6773	0.5600	0.5289	0.4377	0.6299	0.5557	0.5884	0.4819	0.5387	0.4580	0.5533
7.35	0.7007	0.5313	0.6752	0.5590	0.5262	0.4384	0.6102	0.5560	0.5908	0.4813	0.5374	0.4581	0.5554
7.4	0.7006	0.5217	0.6726	0.5570	0.5296	0.4387	0.5949	0.5512	0.5918	0.4794	0.5392	0.4580	0.5529
7.45	0.6720	0.5430	0.6771	0.5585	0.5260	0.4392	0.6135	0.5566	0.5966	0.4791	0.5384	0.4582	0.5549
7.5	0.7025	0.5322	0.6786	0.5591	0.5266	0.4398	0.5979	0.5596	0.5980	0.4796	0.5389	0.4581	0.5559
7.55	0.6829	0.5167	0.6760	0.5576	0.5281	0.4397	0.6122	0.5584	0.5952	0.4764	0.5390	0.4583	0.5534
7.6	0.7011	0.5094	0.6776	0.5575	0.5297	0.4412	0.6299	0.5467	0.5965	0.4791	0.5384	0.4580	0.5554
7.65	0.6889	0.5196	0.6762	0.5582	0.5306	0.4416	0.6211	0.5519	0.5882	0.4820	0.5373	0.4579	0.5545
7.7	0.6781	0.5286	0.6761	0.5584	0.5250	0.4417	0.6150	0.5605	0.5922	0.4814	0.5374	0.4589	0.5544
7.75	0.6772	0.5132	0.6764	0.5572	0.5280	0.4420	0.6243	0.5629	0.5896	0.4779	0.5376	0.4580	0.5537
7.8	0.6979	0.5191	0.6744	0.5586	0.5318	0.4422	0.6151	0.5613	0.5957	0.4783	0.5370	0.4582	0.5558

Table 5 (*continued*)

λ	Game number												Mean
	1	2	3	4	5	6	7	8	9	10	11	12	
7.85	0.7037	0.5359	0.6781	0.5587	0.5299	0.4432	0.6161	0.5598	0.5983	0.4841	0.5356	0.4583	0.5585
7.9	0.6779	0.5386	0.6798	0.5561	0.5307	0.4432	0.6281	0.5581	0.5919	0.4780	0.5382	0.4583	0.5566
7.95	0.6732	0.5312	0.6739	0.5599	0.5307	0.4437	0.6226	0.5523	0.5933	0.4830	0.5396	0.4585	0.5552
8	0.6858	0.5290	0.6779	0.5593	0.5307	0.4446	0.6174	0.5507	0.5995	0.4807	0.5367	0.4586	0.5559
8.05	0.6911	0.5332	0.6788	0.5586	0.5305	0.4442	0.6391	0.5553	0.5957	0.4836	0.5374	0.4581	0.5588
8.1	0.7041	0.5111	0.6756	0.5591	0.5281	0.4444	0.6071	0.5648	0.6003	0.4823	0.5379	0.4589	0.5561
8.15	0.6852	0.5175	0.6791	0.5570	0.5320	0.4449	0.6284	0.5569	0.5902	0.4859	0.5370	0.4584	0.5560
8.2	0.6762	0.5182	0.6787	0.5612	0.5300	0.4446	0.6331	0.5569	0.5999	0.4802	0.5378	0.4585	0.5563
8.25	0.7006	0.5336	0.6767	0.5586	0.5302	0.4460	0.6092	0.5501	0.5934	0.4817	0.5378	0.4589	0.5564
8.3	0.6614	0.5273	0.6764	0.5612	0.5297	0.4465	0.6116	0.5586	0.5941	0.4801	0.5392	0.4582	0.5537
8.35	0.6792	0.5172	0.6756	0.5605	0.5312	0.4458	0.6008	0.5450	0.5898	0.4806	0.5368	0.4580	0.5517
8.4	0.6711	0.5323	0.6795	0.5571	0.5282	0.4466	0.6160	0.5517	0.5948	0.4817	0.5386	0.4587	0.5547
8.45	0.6760	0.5356	0.6766	0.5566	0.5286	0.4466	0.5927	0.5619	0.6019	0.4799	0.5365	0.4587	0.5543
8.5	0.6865	0.5206	0.6774	0.5611	0.5307	0.4464	0.6366	0.5496	0.5922	0.4835	0.5400	0.4586	0.5569
8.55	0.6896	0.5303	0.6763	0.5593	0.5323	0.4474	0.6140	0.5597	0.5972	0.4817	0.5387	0.4585	0.5571
8.6	0.6942	0.5305	0.6731	0.5614	0.5326	0.4486	0.6109	0.5624	0.6007	0.4796	0.5378	0.4578	0.5575
8.65	0.6874	0.5429	0.6767	0.5596	0.5306	0.4476	0.6267	0.5677	0.5995	0.4821	0.5377	0.4585	0.5598
8.7	0.6891	0.5160	0.6786	0.5594	0.5309	0.4481	0.6281	0.5523	0.5963	0.4825	0.5377	0.4583	0.5564
8.75	0.6949	0.5172	0.6780	0.5604	0.5318	0.4486	0.6361	0.5487	0.5917	0.4820	0.5391	0.4580	0.5572
8.8	0.6888	0.5257	0.6783	0.5585	0.5316	0.4487	0.6175	0.5438	0.5960	0.4788	0.5382	0.4586	0.5554
8.85	0.6811	0.5258	0.6776	0.5595	0.5305	0.4492	0.6469	0.5508	0.5954	0.4851	0.5386	0.4585	0.5582
8.9	0.7013	0.5373	0.6741	0.5602	0.5313	0.4493	0.6170	0.5569	0.5964	0.4807	0.5380	0.4586	0.5584
8.95	0.6704	0.5154	0.6786	0.5589	0.5317	0.4495	0.6108	0.5530	0.5978	0.4852	0.5371	0.4588	0.5539
9	0.6650	0.5338	0.6776	0.5586	0.5309	0.4495	0.6024	0.5638	0.5919	0.4860	0.5386	0.4584	0.5547
9.05	0.6886	0.5449	0.6733	0.5560	0.5316	0.4493	0.6175	0.5572	0.5938	0.4830	0.5388	0.4589	0.5577
9.1	0.6948	0.5063	0.6767	0.5565	0.5300	0.4504	0.6008	0.5583	0.5962	0.4785	0.5388	0.4580	0.5538
9.15	0.6838	0.5360	0.6763	0.5579	0.5321	0.4501	0.6264	0.5577	0.5922	0.4828	0.5390	0.4585	0.5578
9.2	0.6915	0.5231	0.6782	0.5583	0.5322	0.4502	0.5955	0.5425	0.6008	0.4817	0.5393	0.4582	0.5543
9.25	0.6813	0.5333	0.6778	0.5592	0.5329	0.4507	0.6274	0.5340	0.5963	0.4770	0.5381	0.4580	0.5555
9.3	0.6822	0.5234	0.6750	0.5589	0.5318	0.4507	0.6181	0.5560	0.5985	0.4823	0.5393	0.4591	0.5563
9.35	0.6909	0.5213	0.6769	0.5568	0.5336	0.4506	0.5928	0.5540	0.6041	0.4845	0.5380	0.4584	0.5551
9.4	0.6870	0.5217	0.6802	0.5584	0.5327	0.4508	0.6232	0.5608	0.5905	0.4842	0.5389	0.4581	0.5572
9.45	0.6790	0.5390	0.6768	0.5602	0.5340	0.4512	0.5985	0.5507	0.5980	0.4841	0.5393	0.4580	0.5557
9.5	0.6794	0.5281	0.6741	0.5565	0.5318	0.4513	0.6340	0.5480	0.5940	0.4837	0.5401	0.4585	0.5566
9.55	0.7019	0.5259	0.6803	0.5628	0.5319	0.4515	0.6117	0.5518	0.5895	0.4789	0.5396	0.4584	0.5570
9.6	0.6962	0.5154	0.6766	0.5588	0.5324	0.4518	0.6239	0.5568	0.5974	0.4842	0.5393	0.4588	0.5576
9.65	0.6671	0.5094	0.6796	0.5592	0.5319	0.4518	0.5956	0.5516	0.5841	0.4827	0.5388	0.4586	0.5509
9.7	0.6919	0.5154	0.6755	0.5575	0.5330	0.4525	0.5975	0.5617	0.5954	0.4833	0.5395	0.4586	0.5551
9.75	0.6954	0.5113	0.6788	0.5587	0.5284	0.4523	0.6125	0.5522	0.5970	0.4846	0.5400	0.4582	0.5558
9.8	0.6989	0.5426	0.6772	0.5589	0.5335	0.4525	0.6262	0.5412	0.5939	0.4772	0.5414	0.4587	0.5585
9.85	0.6812	0.5225	0.6774	0.5593	0.5332	0.4528	0.5889	0.5445	0.5991	0.4840	0.5392	0.4584	0.5534
9.9	0.6918	0.5301	0.6749	0.5603	0.5327	0.4523	0.6180	0.5458	0.6009	0.4772	0.5380	0.4586	0.5567
9.95	0.6818	0.5465	0.6795	0.5581	0.5306	0.4525	0.6338	0.5508	0.5973	0.4880	0.5403	0.4579	0.5598
10	0.6804	0.5202	0.6746	0.5605	0.5335	0.4529	0.6257	0.5462	0.5988	0.4806	0.5383	0.4583	0.5558

Table 6
Quadratic distances of self-tuning EWA learning for each game, for $0.2 \geqslant \lambda \geqslant 0.3$ in steps of 0.0005.

λ	Game number												Mean
	1	2	3	4	5	6	7	8	9	10	11	12	
0.2000	0.2147	0.0858	0.1753	0.0833	0.0499	0.0092	0.1172	0.0619	0.1155	0.0320	0.0346	0.0055	0.0821
0.2005	0.2138	0.0854	0.1762	0.0831	0.0494	0.0090	0.1161	0.0616	0.1165	0.0320	0.0347	0.0056	0.0819
0.2010	0.2143	0.0868	0.1761	0.0835	0.0501	0.0090	0.1159	0.0621	0.1158	0.0320	0.0346	0.0055	0.0821
0.2015	0.2151	0.0859	0.1761	0.0831	0.0496	0.0091	0.1167	0.0624	0.1152	0.0319	0.0345	0.0054	0.0821
0.2020	0.2145	0.0863	0.1756	0.0826	0.0502	0.0091	0.1163	0.0614	0.1152	0.0319	0.0346	0.0054	0.0819
0.2025	0.2143	0.0849	0.1755	0.0828	0.0500	0.0091	0.1151	0.0623	0.1160	0.0321	0.0349	0.0055	0.0819
0.2030	0.2156	0.0866	0.1755	0.0827	0.0501	0.0090	0.1147	0.0618	0.1156	0.0318	0.0349	0.0054	0.0820
0.2035	0.2138	0.0870	0.1755	0.0826	0.0499	0.0088	0.1157	0.0616	0.1152	0.0323	0.0346	0.0056	0.0819
0.2040	0.2140	0.0856	0.1760	0.0826	0.0502	0.0091	0.1152	0.0623	0.1163	0.0317	0.0345	0.0054	0.0819
0.2045	0.2139	0.0860	0.1754	0.0828	0.0503	0.0092	0.1172	0.0610	0.1154	0.0317	0.0342	0.0054	0.0819
0.2050	0.2148	0.0857	0.1751	0.0829	0.0503	0.0089	0.1166	0.0628	0.1154	0.0317	0.0344	0.0054	0.0820
0.2055	0.2140	0.0866	0.1757	0.0829	0.0499	0.0092	0.1166	0.0622	0.1150	0.0316	0.0343	0.0054	0.0820
0.2060	0.2147	0.0865	0.1761	0.0827	0.0500	0.0092	0.1187	0.0618	0.1152	0.0323	0.0349	0.0053	0.0823
0.2065	0.2146	0.0859	0.1756	0.0818	0.0502	0.0091	0.1163	0.0619	0.1143	0.0315	0.0344	0.0053	0.0817
0.2070	0.2142	0.0859	0.1758	0.0821	0.0500	0.0091	0.1172	0.0621	0.1151	0.0316	0.0340	0.0054	0.0819
0.2075	0.2131	0.0857	0.1756	0.0829	0.0501	0.0090	0.1166	0.0620	0.1149	0.0314	0.0341	0.0052	0.0817
0.2080	0.2140	0.0858	0.1748	0.0825	0.0497	0.0089	0.1157	0.0625	0.1148	0.0316	0.0343	0.0051	0.0817

(*continued on next page*)

Table 6 (*continued*)

λ	Game number												Mean
	1	2	3	4	5	6	7	8	9	10	11	12	
0.2085	0.2155	0.0864	0.1757	0.0825	0.0500	0.0089	0.1157	0.0614	0.1146	0.0314	0.0345	0.0054	0.0819
0.2090	0.2143	0.0862	0.1755	0.0823	0.0503	0.0092	0.1169	0.0622	0.1146	0.0316	0.0347	0.0053	0.0819
0.2095	0.2140	0.0874	0.1752	0.0825	0.0493	0.0089	0.1168	0.0615	0.1148	0.0312	0.0344	0.0052	0.0818
0.2100	0.2144	0.0862	0.1764	0.0824	0.0498	0.0089	0.1167	0.0616	0.1151	0.0315	0.0340	0.0052	0.0819
0.2105	0.2153	0.0856	0.1750	0.0832	0.0502	0.0090	0.1161	0.0607	0.1150	0.0313	0.0344	0.0052	0.0817
0.2110	0.2148	0.0857	0.1750	0.0820	0.0498	0.0089	0.1175	0.0626	0.1146	0.0314	0.0340	0.0051	0.0818
0.2115	0.2150	0.0874	0.1759	0.0817	0.0496	0.0090	0.1158	0.0618	0.1142	0.0314	0.0345	0.0053	0.0818
0.2120	0.2139	0.0865	0.1752	0.0822	0.0498	0.0093	0.1150	0.0620	0.1139	0.0311	0.0342	0.0051	0.0815
0.2125	0.2143	0.0860	0.1757	0.0825	0.0500	0.0090	0.1170	0.0623	0.1149	0.0310	0.0341	0.0052	0.0818
0.2130	0.2157	0.0855	0.1757	0.0823	0.0497	0.0092	0.1159	0.0619	0.1148	0.0315	0.0346	0.0052	0.0818
0.2135	0.2146	0.0863	0.1762	0.0821	0.0497	0.0092	0.1156	0.0617	0.1137	0.0313	0.0343	0.0051	0.0817
0.2140	0.2140	0.0856	0.1755	0.0824	0.0505	0.0090	0.1150	0.0627	0.1141	0.0311	0.0343	0.0052	0.0816
0.2145	0.2162	0.0862	0.1747	0.0823	0.0493	0.0091	0.1163	0.0614	0.1139	0.0309	0.0341	0.0052	0.0816
0.2150	0.2135	0.0852	0.1747	0.0822	0.0496	0.0089	0.1162	0.0616	0.1136	0.0312	0.0339	0.0051	0.0813
0.2155	0.2150	0.0868	0.1753	0.0820	0.0499	0.0090	0.1156	0.0619	0.1142	0.0311	0.0341	0.0051	0.0817
0.2160	0.2137	0.0870	0.1752	0.0816	0.0496	0.0091	0.1172	0.0620	0.1132	0.0306	0.0340	0.0051	0.0815
0.2165	0.2159	0.0860	0.1758	0.0816	0.0499	0.0090	0.1172	0.0625	0.1135	0.0311	0.0336	0.0052	0.0818
0.2170	0.2156	0.0864	0.1763	0.0818	0.0496	0.0090	0.1178	0.0618	0.1137	0.0308	0.0334	0.0052	0.0818
0.2175	0.2148	0.0870	0.1747	0.0812	0.0500	0.0090	0.1162	0.0611	0.1135	0.0305	0.0339	0.0051	0.0814
0.2180	0.2134	0.0865	0.1758	0.0823	0.0500	0.0089	0.1151	0.0626	0.1135	0.0309	0.0341	0.0052	0.0815
0.2185	0.2150	0.0865	0.1750	0.0821	0.0502	0.0089	0.1167	0.0628	0.1136	0.0306	0.0338	0.0052	0.0817
0.2190	0.2151	0.0861	0.1757	0.0824	0.0492	0.0088	0.1173	0.0618	0.1130	0.0311	0.0340	0.0052	0.0817
0.2195	0.2138	0.0855	0.1753	0.0819	0.0495	0.0089	0.1154	0.0618	0.1127	0.0307	0.0343	0.0051	0.0812
0.2200	0.2147	0.0875	0.1746	0.0817	0.0497	0.0091	0.1180	0.0614	0.1139	0.0307	0.0342	0.0051	0.0817
0.2205	0.2153	0.0862	0.1749	0.0825	0.0496	0.0090	0.1165	0.0618	0.1131	0.0307	0.0338	0.0052	0.0816
0.2210	0.2154	0.0862	0.1740	0.0826	0.0499	0.0091	0.1173	0.0616	0.1129	0.0306	0.0339	0.0052	0.0816
0.2215	0.2151	0.0867	0.1749	0.0825	0.0499	0.0090	0.1173	0.0612	0.1129	0.0306	0.0343	0.0051	0.0816
0.2220	0.2160	0.0864	0.1754	0.0818	0.0498	0.0089	0.1162	0.0623	0.1127	0.0308	0.0339	0.0051	0.0816
0.2225	0.2142	0.0860	0.1742	0.0812	0.0494	0.0090	0.1151	0.0621	0.1125	0.0309	0.0338	0.0051	0.0811
0.2230	0.2139	0.0848	0.1755	0.0816	0.0496	0.0089	0.1166	0.0616	0.1131	0.0304	0.0336	0.0050	0.0812
0.2235	0.2142	0.0848	0.1753	0.0817	0.0498	0.0088	0.1166	0.0622	0.1126	0.0305	0.0336	0.0052	0.0813
0.2240	0.2170	0.0858	0.1748	0.0818	0.0497	0.0087	0.1156	0.0620	0.1129	0.0304	0.0339	0.0051	0.0815
0.2245	0.2150	0.0857	0.1753	0.0815	0.0498	0.0088	0.1169	0.0622	0.1133	0.0305	0.0340	0.0051	0.0815
0.2250	0.2140	0.0875	0.1748	0.0811	0.0500	0.0088	0.1184	0.0623	0.1126	0.0303	0.0336	0.0051	0.0815
0.2255	0.2158	0.0858	0.1744	0.0814	0.0492	0.0088	0.1165	0.0614	0.1124	0.0308	0.0338	0.0049	0.0813
0.2260	0.2155	0.0870	0.1749	0.0811	0.0497	0.0090	0.1163	0.0614	0.1132	0.0304	0.0344	0.0050	0.0815
0.2265	0.2158	0.0869	0.1750	0.0816	0.0490	0.0087	0.1158	0.0620	0.1125	0.0301	0.0338	0.0050	0.0813
0.2270	0.2147	0.0859	0.1748	0.0811	0.0494	0.0086	0.1174	0.0617	0.1119	0.0302	0.0335	0.0050	0.0812
0.2275	0.2160	0.0857	0.1751	0.0812	0.0498	0.0089	0.1170	0.0616	0.1125	0.0304	0.0342	0.0049	0.0814
0.2280	0.2144	0.0859	0.1751	0.0820	0.0490	0.0088	0.1173	0.0624	0.1128	0.0304	0.0333	0.0051	0.0814
0.2285	0.2141	0.0860	0.1757	0.0814	0.0497	0.0087	0.1179	0.0626	0.1129	0.0303	0.0339	0.0048	0.0815
0.2290	0.2153	0.0861	0.1741	0.0811	0.0495	0.0088	0.1161	0.0622	0.1131	0.0302	0.0340	0.0050	0.0813
0.2295	0.2144	0.0862	0.1750	0.0815	0.0495	0.0089	0.1157	0.0621	0.1129	0.0300	0.0334	0.0049	0.0812
0.2300	0.2147	0.0867	0.1742	0.0817	0.0499	0.0088	0.1193	0.0628	0.1124	0.0304	0.0340	0.0050	0.0817
0.2305	0.2153	0.0866	0.1743	0.0816	0.0490	0.0089	0.1163	0.0621	0.1122	0.0300	0.0336	0.0048	0.0812
0.2310	0.2150	0.0861	0.1746	0.0813	0.0494	0.0088	0.1172	0.0615	0.1122	0.0303	0.0336	0.0051	0.0813
0.2315	0.2130	0.0861	0.1738	0.0811	0.0494	0.0087	0.1170	0.0624	0.1122	0.0298	0.0340	0.0049	0.0810
0.2320	0.2145	0.0863	0.1740	0.0813	0.0487	0.0089	0.1171	0.0614	0.1120	0.0300	0.0334	0.0050	0.0810
0.2325	0.2159	0.0853	0.1745	0.0819	0.0492	0.0089	0.1183	0.0627	0.1114	0.0303	0.0340	0.0049	0.0815
0.2330	0.2134	0.0870	0.1747	0.0812	0.0494	0.0089	0.1170	0.0622	0.1113	0.0303	0.0337	0.0047	0.0812
0.2335	0.2146	0.0862	0.1740	0.0810	0.0494	0.0086	0.1178	0.0619	0.1114	0.0299	0.0331	0.0047	0.0810
0.2340	0.2156	0.0861	0.1737	0.0813	0.0499	0.0087	0.1153	0.0616	0.1119	0.0303	0.0334	0.0048	0.0810
0.2345	0.2141	0.0856	0.1745	0.0813	0.0494	0.0089	0.1186	0.0617	0.1120	0.0299	0.0338	0.0048	0.0812
0.2350	0.2138	0.0860	0.1750	0.0808	0.0495	0.0086	0.1186	0.0620	0.1116	0.0298	0.0337	0.0047	0.0812
0.2355	0.2138	0.0853	0.1735	0.0809	0.0492	0.0089	0.1181	0.0624	0.1117	0.0301	0.0338	0.0048	0.0811
0.2360	0.2149	0.0854	0.1732	0.0810	0.0489	0.0087	0.1184	0.0615	0.1109	0.0296	0.0335	0.0049	0.0809
0.2365	0.2146	0.0860	0.1741	0.0812	0.0498	0.0087	0.1186	0.0615	0.1111	0.0298	0.0335	0.0049	0.0812
0.2370	0.2152	0.0856	0.1742	0.0810	0.0491	0.0086	0.1163	0.0617	0.1120	0.0301	0.0337	0.0049	0.0810
0.2375	0.2149	0.0859	0.1742	0.0810	0.0493	0.0086	0.1178	0.0625	0.1109	0.0298	0.0340	0.0049	0.0812
0.2380	0.2152	0.0867	0.1733	0.0815	0.0490	0.0086	0.1170	0.0618	0.1118	0.0294	0.0335	0.0048	0.0810
0.2385	0.2151	0.0865	0.1743	0.0799	0.0492	0.0087	0.1189	0.0618	0.1113	0.0296	0.0338	0.0048	0.0811
0.2390	0.2152	0.0860	0.1738	0.0816	0.0498	0.0089	0.1176	0.0618	0.1112	0.0294	0.0333	0.0050	0.0811
0.2395	0.2139	0.0861	0.1734	0.0811	0.0492	0.0087	0.1190	0.0619	0.1112	0.0299	0.0332	0.0049	0.0810
0.2400	0.2148	0.0869	0.1738	0.0804	0.0494	0.0087	0.1185	0.0621	0.1114	0.0298	0.0335	0.0047	0.0812
0.2405	0.2145	0.0856	0.1739	0.0813	0.0491	0.0086	0.1185	0.0620	0.1119	0.0297	0.0335	0.0049	0.0811
0.2410	0.2141	0.0862	0.1729	0.0812	0.0490	0.0087	0.1169	0.0619	0.1111	0.0298	0.0333	0.0049	0.0808
0.2415	0.2158	0.0863	0.1750	0.0804	0.0494	0.0085	0.1168	0.0620	0.1111	0.0295	0.0333	0.0049	0.0811
0.2420	0.2154	0.0870	0.1734	0.0809	0.0495	0.0087	0.1191	0.0617	0.1111	0.0299	0.0335	0.0047	0.0813

Table 6 (*continued*)

λ	Game number												Mean
	1	2	3	4	5	6	7	8	9	10	11	12	
0.2425	0.2156	0.0863	0.1739	0.0808	0.0496	0.0086	0.1188	0.0616	0.1111	0.0294	0.0336	0.0048	0.0812
0.2430	0.2153	0.0863	0.1737	0.0809	0.0491	0.0088	0.1190	0.0625	0.1108	0.0295	0.0334	0.0048	0.0812
0.2435	0.2140	0.0863	0.1741	0.0807	0.0488	0.0087	0.1174	0.0630	0.1110	0.0295	0.0333	0.0047	0.0810
0.2440	0.2151	0.0858	0.1729	0.0808	0.0498	0.0087	0.1186	0.0629	0.1111	0.0297	0.0336	0.0048	0.0811
0.2445	0.2156	0.0872	0.1737	0.0809	0.0488	0.0086	0.1207	0.0627	0.1105	0.0294	0.0335	0.0047	0.0814
0.2450	0.2156	0.0860	0.1740	0.0810	0.0489	0.0089	0.1169	0.0620	0.1105	0.0296	0.0334	0.0048	0.0810
0.2455	0.2138	0.0864	0.1744	0.0803	0.0491	0.0087	0.1203	0.0620	0.1110	0.0293	0.0331	0.0048	0.0811
0.2460	0.2149	0.0868	0.1740	0.0804	0.0491	0.0086	0.1199	0.0617	0.1103	0.0293	0.0337	0.0047	0.0811
0.2465	0.2150	0.0870	0.1750	0.0804	0.0492	0.0087	0.1188	0.0623	0.1109	0.0295	0.0336	0.0048	0.0813
0.2470	0.2158	0.0868	0.1742	0.0808	0.0491	0.0087	0.1173	0.0616	0.1103	0.0296	0.0330	0.0048	0.0810
0.2475	0.2144	0.0865	0.1729	0.0812	0.0497	0.0089	0.1178	0.0630	0.1106	0.0290	0.0333	0.0048	0.0810
0.2480	0.2141	0.0856	0.1733	0.0803	0.0492	0.0082	0.1180	0.0616	0.1112	0.0292	0.0338	0.0049	0.0808
0.2485	0.2150	0.0857	0.1725	0.0804	0.0489	0.0086	0.1197	0.0631	0.1109	0.0293	0.0333	0.0047	0.0810
0.2490	0.2156	0.0874	0.1740	0.0805	0.0491	0.0085	0.1183	0.0621	0.1107	0.0295	0.0329	0.0048	0.0811
0.2495	0.2151	0.0866	0.1741	0.0809	0.0494	0.0086	0.1179	0.0621	0.1104	0.0292	0.0332	0.0048	0.0810
0.2500	0.2143	0.0856	0.1743	0.0809	0.0493	0.0086	0.1195	0.0621	0.1104	0.0290	0.0331	0.0047	0.0810
0.2505	0.2154	0.0864	0.1739	0.0808	0.0494	0.0085	0.1178	0.0626	0.1101	0.0292	0.0334	0.0048	0.0810
0.2510	0.2161	0.0858	0.1739	0.0807	0.0492	0.0085	0.1184	0.0636	0.1103	0.0294	0.0329	0.0047	0.0811
0.2515	0.2163	0.0846	0.1737	0.0798	0.0490	0.0088	0.1186	0.0635	0.1106	0.0292	0.0332	0.0048	0.0810
0.2520	0.2163	0.0863	0.1733	0.0809	0.0493	0.0083	0.1211	0.0623	0.1106	0.0294	0.0336	0.0047	0.0813
0.2525	0.2156	0.0864	0.1735	0.0803	0.0488	0.0086	0.1186	0.0623	0.1103	0.0295	0.0333	0.0048	0.0810
0.2530	0.2151	0.0858	0.1731	0.0797	0.0488	0.0083	0.1197	0.0621	0.1102	0.0287	0.0327	0.0047	0.0808
0.2535	0.2148	0.0862	0.1746	0.0804	0.0494	0.0085	0.1164	0.0621	0.1099	0.0295	0.0340	0.0048	0.0809
0.2540	0.2167	0.0858	0.1733	0.0805	0.0492	0.0084	0.1210	0.0624	0.1102	0.0290	0.0331	0.0048	0.0812
0.2545	0.2152	0.0864	0.1739	0.0806	0.0490	0.0086	0.1198	0.0612	0.1100	0.0290	0.0337	0.0047	0.0810
0.2550	0.2180	0.0861	0.1735	0.0800	0.0490	0.0084	0.1211	0.0620	0.1092	0.0290	0.0334	0.0047	0.0812
0.2555	0.2139	0.0861	0.1742	0.0798	0.0489	0.0085	0.1211	0.0626	0.1095	0.0288	0.0330	0.0048	0.0809
0.2560	0.2155	0.0866	0.1734	0.0803	0.0490	0.0084	0.1204	0.0623	0.1100	0.0287	0.0332	0.0047	0.0810
0.2565	0.2160	0.0866	0.1741	0.0808	0.0489	0.0085	0.1197	0.0627	0.1098	0.0289	0.0335	0.0046	0.0812
0.2570	0.2146	0.0860	0.1732	0.0799	0.0492	0.0085	0.1175	0.0627	0.1093	0.0289	0.0335	0.0047	0.0807
0.2575	0.2155	0.0851	0.1731	0.0807	0.0493	0.0086	0.1208	0.0631	0.1097	0.0290	0.0337	0.0048	0.0811
0.2580	0.2136	0.0863	0.1737	0.0796	0.0487	0.0081	0.1204	0.0630	0.1107	0.0289	0.0331	0.0050	0.0809
0.2585	0.2148	0.0862	0.1735	0.0797	0.0489	0.0085	0.1211	0.0635	0.1098	0.0287	0.0331	0.0047	0.0810
0.2590	0.2152	0.0849	0.1734	0.0795	0.0487	0.0082	0.1211	0.0627	0.1095	0.0291	0.0331	0.0048	0.0809
0.2595	0.2156	0.0854	0.1731	0.0802	0.0489	0.0084	0.1192	0.0623	0.1098	0.0290	0.0329	0.0047	0.0808
0.2600	0.2157	0.0869	0.1735	0.0798	0.0494	0.0086	0.1189	0.0622	0.1095	0.0288	0.0328	0.0048	0.0809
0.2605	0.2147	0.0864	0.1732	0.0801	0.0494	0.0084	0.1190	0.0623	0.1101	0.0285	0.0330	0.0044	0.0808
0.2610	0.2140	0.0858	0.1734	0.0803	0.0487	0.0083	0.1190	0.0629	0.1094	0.0287	0.0333	0.0047	0.0807
0.2615	0.2152	0.0859	0.1734	0.0801	0.0492	0.0084	0.1201	0.0625	0.1092	0.0286	0.0336	0.0048	0.0809
0.2620	0.2151	0.0863	0.1735	0.0802	0.0490	0.0085	0.1207	0.0625	0.1096	0.0287	0.0332	0.0046	0.0810
0.2625	0.2148	0.0862	0.1734	0.0800	0.0490	0.0085	0.1200	0.0623	0.1094	0.0287	0.0334	0.0046	0.0809
0.2630	0.2150	0.0866	0.1731	0.0801	0.0488	0.0083	0.1193	0.0625	0.1094	0.0284	0.0331	0.0048	0.0808
0.2635	0.2158	0.0867	0.1725	0.0802	0.0493	0.0085	0.1203	0.0624	0.1094	0.0287	0.0327	0.0046	0.0809
0.2640	0.2146	0.0862	0.1727	0.0796	0.0492	0.0086	0.1184	0.0640	0.1090	0.0287	0.0333	0.0046	0.0807
0.2645	0.2160	0.0867	0.1741	0.0801	0.0490	0.0085	0.1214	0.0624	0.1091	0.0286	0.0327	0.0046	0.0811
0.2650	0.2153	0.0861	0.1728	0.0803	0.0488	0.0084	0.1199	0.0624	0.1089	0.0287	0.0334	0.0046	0.0808
0.2655	0.2144	0.0850	0.1733	0.0797	0.0491	0.0085	0.1206	0.0626	0.1089	0.0288	0.0328	0.0047	0.0807
0.2660	0.2141	0.0855	0.1728	0.0799	0.0487	0.0084	0.1191	0.0627	0.1095	0.0286	0.0333	0.0047	0.0806
0.2665	0.2154	0.0857	0.1731	0.0796	0.0488	0.0083	0.1179	0.0625	0.1090	0.0288	0.0331	0.0046	0.0806
0.2670	0.2145	0.0863	0.1737	0.0797	0.0488	0.0083	0.1225	0.0629	0.1092	0.0287	0.0334	0.0047	0.0810
0.2675	0.2152	0.0854	0.1724	0.0801	0.0484	0.0085	0.1203	0.0628	0.1090	0.0286	0.0332	0.0046	0.0807
0.2680	0.2146	0.0844	0.1734	0.0799	0.0490	0.0085	0.1204	0.0635	0.1086	0.0286	0.0332	0.0047	0.0807
0.2685	0.2163	0.0867	0.1726	0.0791	0.0485	0.0085	0.1211	0.0639	0.1087	0.0287	0.0336	0.0045	0.0810
0.2690	0.2147	0.0863	0.1727	0.0798	0.0487	0.0085	0.1191	0.0619	0.1091	0.0283	0.0336	0.0048	0.0806
0.2695	0.2149	0.0870	0.1738	0.0800	0.0490	0.0085	0.1189	0.0635	0.1087	0.0283	0.0334	0.0045	0.0809
0.2700	0.2156	0.0861	0.1733	0.0790	0.0488	0.0086	0.1180	0.0625	0.1090	0.0283	0.0333	0.0047	0.0806
0.2705	0.2153	0.0863	0.1725	0.0788	0.0491	0.0084	0.1210	0.0643	0.1083	0.0285	0.0332	0.0046	0.0809
0.2710	0.2160	0.0857	0.1732	0.0796	0.0486	0.0081	0.1220	0.0626	0.1091	0.0283	0.0333	0.0046	0.0809
0.2715	0.2143	0.0864	0.1726	0.0800	0.0490	0.0084	0.1216	0.0638	0.1089	0.0284	0.0334	0.0046	0.0810
0.2720	0.2174	0.0858	0.1731	0.0796	0.0491	0.0084	0.1205	0.0629	0.1085	0.0281	0.0332	0.0047	0.0809
0.2725	0.2160	0.0858	0.1728	0.0797	0.0485	0.0081	0.1196	0.0638	0.1086	0.0283	0.0335	0.0046	0.0808
0.2730	0.2153	0.0871	0.1731	0.0797	0.0484	0.0085	0.1232	0.0617	0.1088	0.0283	0.0329	0.0047	0.0810
0.2735	0.2155	0.0862	0.1729	0.0800	0.0489	0.0086	0.1209	0.0628	0.1083	0.0281	0.0334	0.0047	0.0809
0.2740	0.2155	0.0868	0.1730	0.0798	0.0486	0.0084	0.1212	0.0626	0.1085	0.0285	0.0331	0.0048	0.0809
0.2745	0.2144	0.0858	0.1735	0.0800	0.0490	0.0084	0.1244	0.0635	0.1090	0.0286	0.0330	0.0046	0.0812
0.2750	0.2157	0.0860	0.1740	0.0793	0.0489	0.0083	0.1240	0.0633	0.1084	0.0284	0.0332	0.0046	0.0812
0.2755	0.2154	0.0871	0.1724	0.0798	0.0493	0.0085	0.1205	0.0626	0.1085	0.0282	0.0333	0.0046	0.0809
0.2760	0.2152	0.0852	0.1732	0.0790	0.0485	0.0085	0.1252	0.0629	0.1084	0.0283	0.0331	0.0047	0.0810

(*continued on next page*)

Table 6 (*continued*)

λ	Game number												Mean
	1	2	3	4	5	6	7	8	9	10	11	12	
0.2765	0.2139	0.0856	0.1726	0.0790	0.0486	0.0084	0.1221	0.0634	0.1087	0.0284	0.0333	0.0047	0.0807
0.2770	0.2162	0.0853	0.1718	0.0791	0.0489	0.0082	0.1214	0.0623	0.1087	0.0281	0.0330	0.0047	0.0807
0.2775	**0.2166**	**0.0853**	**0.1719**	**0.0798**	**0.0486**	**0.0084**	**0.1174**	**0.0637**	**0.1087**	**0.0281**	**0.0332**	**0.0046**	**0.0805**
0.2780	0.2167	0.0868	0.1720	0.0793	0.0487	0.0083	0.1223	0.0633	0.1080	0.0284	0.0330	0.0045	0.0809
0.2785	0.2156	0.0862	0.1730	0.0792	0.0486	0.0082	0.1231	0.0635	0.1090	0.0281	0.0329	0.0046	0.0810
0.2790	0.2156	0.0858	0.1728	0.0791	0.0484	0.0081	0.1224	0.0625	0.1085	0.0281	0.0333	0.0047	0.0808
0.2795	0.2158	0.0870	0.1730	0.0787	0.0483	0.0085	0.1224	0.0633	0.1084	0.0285	0.0335	0.0047	0.0810
0.2800	0.2166	0.0862	0.1741	0.0792	0.0486	0.0084	0.1243	0.0633	0.1085	0.0280	0.0332	0.0046	0.0812
0.2805	0.2143	0.0857	0.1729	0.0791	0.0485	0.0084	0.1223	0.0623	0.1086	0.0279	0.0335	0.0046	0.0807
0.2810	0.2150	0.0848	0.1730	0.0797	0.0491	0.0081	0.1221	0.0617	0.1087	0.0284	0.0328	0.0048	0.0807
0.2815	0.2161	0.0863	0.1729	0.0789	0.0488	0.0085	0.1233	0.0638	0.1083	0.0282	0.0336	0.0046	0.0811
0.2820	0.2174	0.0868	0.1724	0.0797	0.0491	0.0083	0.1252	0.0629	0.1084	0.0281	0.0334	0.0046	0.0814
0.2825	0.2159	0.0861	0.1730	0.0794	0.0484	0.0082	0.1208	0.0645	0.1090	0.0279	0.0327	0.0046	0.0809
0.2830	0.2157	0.0858	0.1724	0.0796	0.0488	0.0084	0.1259	0.0630	0.1080	0.0280	0.0335	0.0047	0.0812
0.2835	0.2163	0.0866	0.1724	0.0797	0.0486	0.0082	0.1242	0.0624	0.1090	0.0282	0.0337	0.0048	0.0812
0.2840	0.2160	0.0859	0.1722	0.0787	0.0487	0.0085	0.1279	0.0642	0.1086	0.0278	0.0334	0.0046	0.0814
0.2845	0.2159	0.0870	0.1724	0.0799	0.0484	0.0083	0.1257	0.0640	0.1085	0.0281	0.0336	0.0047	0.0814
0.2850	0.2160	0.0854	0.1725	0.0794	0.0490	0.0081	0.1243	0.0629	0.1083	0.0283	0.0336	0.0047	0.0810
0.2855	0.2183	0.0854	0.1717	0.0792	0.0485	0.0083	0.1267	0.0636	0.1081	0.0286	0.0329	0.0047	0.0813
0.2860	0.2162	0.0846	0.1726	0.0790	0.0488	0.0084	0.1247	0.0624	0.1079	0.0280	0.0331	0.0047	0.0809
0.2865	0.2157	0.0866	0.1723	0.0793	0.0490	0.0082	0.1209	0.0637	0.1077	0.0282	0.0337	0.0048	0.0809
0.2870	0.2171	0.0861	0.1730	0.0790	0.0488	0.0083	0.1261	0.0648	0.1081	0.0278	0.0335	0.0047	0.0814
0.2875	0.2175	0.0856	0.1728	0.0785	0.0486	0.0082	0.1233	0.0635	0.1083	0.0281	0.0332	0.0046	0.0810
0.2880	0.2151	0.0873	0.1718	0.0796	0.0486	0.0082	0.1266	0.0646	0.1079	0.0282	0.0332	0.0047	0.0813
0.2885	0.2152	0.0867	0.1724	0.0789	0.0482	0.0082	0.1256	0.0630	0.1085	0.0277	0.0327	0.0048	0.0810
0.2890	0.2154	0.0850	0.1722	0.0791	0.0490	0.0082	0.1280	0.0642	0.1077	0.0280	0.0334	0.0048	0.0813
0.2895	0.2148	0.0859	0.1729	0.0793	0.0484	0.0081	0.1252	0.0632	0.1077	0.0280	0.0335	0.0048	0.0810
0.2900	0.2164	0.0856	0.1720	0.0793	0.0486	0.0083	0.1245	0.0636	0.1077	0.0280	0.0333	0.0048	0.0810
0.2905	0.2152	0.0855	0.1727	0.0785	0.0488	0.0083	0.1273	0.0640	0.1075	0.0282	0.0329	0.0048	0.0811
0.2910	0.2158	0.0868	0.1721	0.0790	0.0492	0.0084	0.1273	0.0628	0.1080	0.0279	0.0334	0.0047	0.0813
0.2915	0.2172	0.0860	0.1730	0.0793	0.0482	0.0084	0.1274	0.0642	0.1082	0.0279	0.0336	0.0046	0.0815
0.2920	0.2154	0.0858	0.1722	0.0787	0.0487	0.0083	0.1264	0.0635	0.1078	0.0281	0.0332	0.0047	0.0811
0.2925	0.2177	0.0853	0.1726	0.0786	0.0488	0.0083	0.1276	0.0645	0.1075	0.0276	0.0335	0.0046	0.0814
0.2930	0.2146	0.0869	0.1726	0.0786	0.0485	0.0082	0.1253	0.0641	0.1075	0.0280	0.0331	0.0048	0.0810
0.2935	0.2157	0.0870	0.1729	0.0787	0.0486	0.0082	0.1297	0.0635	0.1080	0.0279	0.0334	0.0047	0.0815
0.2940	0.2172	0.0871	0.1730	0.0786	0.0486	0.0081	0.1290	0.0650	0.1073	0.0277	0.0336	0.0047	0.0817
0.2945	0.2147	0.0857	0.1728	0.0787	0.0487	0.0081	0.1259	0.0646	0.1075	0.0278	0.0331	0.0048	0.0810
0.2950	0.2165	0.0865	0.1731	0.0791	0.0489	0.0081	0.1261	0.0645	0.1078	0.0280	0.0334	0.0048	0.0814
0.2955	0.2178	0.0850	0.1729	0.0793	0.0484	0.0081	0.1281	0.0646	0.1087	0.0279	0.0333	0.0048	0.0816
0.2960	0.2145	0.0865	0.1723	0.0791	0.0483	0.0082	0.1268	0.0642	0.1079	0.0278	0.0339	0.0048	0.0812
0.2965	0.2153	0.0861	0.1726	0.0786	0.0486	0.0083	0.1241	0.0641	0.1078	0.0276	0.0337	0.0048	0.0810
0.2970	0.2158	0.0856	0.1724	0.0793	0.0490	0.0083	0.1262	0.0648	0.1075	0.0281	0.0331	0.0047	0.0812
0.2975	0.2147	0.0862	0.1725	0.0792	0.0482	0.0083	0.1260	0.0638	0.1085	0.0279	0.0331	0.0048	0.0811
0.2980	0.2188	0.0856	0.1719	0.0787	0.0485	0.0080	0.1279	0.0648	0.1076	0.0279	0.0332	0.0049	0.0815
0.2985	0.2178	0.0860	0.1717	0.0791	0.0486	0.0082	0.1288	0.0646	0.1082	0.0277	0.0334	0.0046	0.0816
0.2990	0.2153	0.0865	0.1731	0.0790	0.0488	0.0080	0.1242	0.0639	0.1079	0.0279	0.0335	0.0048	0.0811
0.2995	0.2160	0.0864	0.1726	0.0789	0.0486	0.0082	0.1287	0.0641	0.1077	0.0281	0.0335	0.0048	0.0815
0.3000	0.2164	0.0862	0.1722	0.0792	0.0483	0.0082	0.1285	0.0649	0.1070	0.0277	0.0334	0.0049	0.0814

References

Abbink, Klaus, Sadrieh, Abdolkarim, 1995. RatImage – research assistance toolbox for computer-aided human behavior experiments. SFB discussion paper B-325, University of Bonn.

Beggs, A.W., 2005. On the convergence of reinforcement learning. J. Econ. Theory 122, 1–36.

Brier, Glenn W., 1950. Verification of forecasts expressed in terms of probability. Mon. Weather Rev. 78 (1), 1–3.

Brown, James N., Rosenthal, Robert W., 1990. Testing the minimax hypothesis: A re-examination of O'Neill's game experiment. Econometrica 58, 1065–1081.

Brunner, Christoph, Camerer, Colin F., Goeree, Jacob K., 2011. Stationary concepts for experimental 2×2 games: Comment. Amer. Econ. Rev. 101 (2), 1029–1040.

Camerer, Colin F., Ho, Teck H., 1999. Experience-weighted attraction learning in normal form games. Econometrica 67, 827–874.

Chen, Wei, Liu, Shu-Yu, Chen, Chih-Han, Lee, Yi-Shan, 2011. Bounded memory, inertia, sampling and weighting model for market entry games. Games 2 (1), 187–199.

Cheung, Yin-Wong, Friedman, Daniel, 1997. Individual learning in normal form games: Some laboratory results. Games Econ. Behav. 19, 46–76.

Erev, Ido, Ert, Eyal, Roth, Alvin E., 2010. A choice prediction competition for market entry games: An introduction. Games 1 (2), 117–136.

Erev, Ido, Haruvy, Ernan, 2005. Generality, repetition, and the role of descriptive learning models. J. Math. Psychol. 49, 357–371.

Erev, Ido, Roth, Alvin E., 1998. Predicting how people play games: Reinforcement learning in experimental games with unique mixed strategy equilibria. Amer. Econ. Rev. 88 (4), 848–881.

Estes, William K., 1954. Individual behavior in uncertain situations: An interpretation in terms of statistical association theory. In: Thrall, Robert M., Coombs, C.H., Davis, R.L. (Eds.), Decision Processes. Wiley, New York, pp. 127–137.

Goerg, Sebastian J., Selten, Reinhard, 2009. Experimental investigation of stationary concepts in cyclic duopoly games. Exper. Econ. 12 (3), 253–271.

Harley, Calvin B., 1981. Learning the evolutionarily stable strategy. J. Theor. Biol. 89 (4), 611–633.
Ho, Teck H., Camerer, Colin, Chong, Juin-Kuan, 2007. Self-tuning experience-weighted attraction learning in games. J. Econ. Theory 133, 177–198.
Hopkins, E., 2002. Two competing models of how people learn in games. Econometrica 2002 (70), 2141–2166.
Kahneman, Daniel, Tversky, Amos, 1979. Prospect theory: An analysis of decision under risk. Econometrica 47 (2), 263–291.
Kalai, Ehud, Lehrer, Ehud, 1993. Rational learning leads to Nash equilibrium. Econometrica 61 (5), 1019–1045.
Marchiori, Davide, Warglien, Massimo, 2008. Predicting human interactive learning by regret-driven neural networks. Science 319, 1111–1113.
McKelvey, Richard D., Palfrey, Thomas R., 1995. Quantal response equilibria for normal form games. Games Econ. Behav. 10 (1), 6–38.
Metrick, Andrew I., Polak, Ben, 1994. Fictitious play in 2×2 games: A geometric proof of convergence. Econ. Theory 4, 923–933.
Miyasawa, K., 1961. On the Convergence of the Learning Process in a 2×2 Non-Zero-Sum Game. Economic Research Program. Princeton University, Research Memorandum, 33.
Osborne, Martin J., Rubinstein, Ariel, 1998. Games with procedurally rational players. Amer. Econ. Rev. 88 (4), 834–847.
Roth, Alvin E., Erev, Ido, 1995. Learning in extensive-form games: Experimental data and simple dynamic models in the intermediate term. Games Econ. Behav. 8, 164–212.
Salmon, Timothy C., 2001. An evaluation of econometric models of adaptive learning. Econometrica 69 (6), 1597–1628.
Selten, Reinhard, 1998. Axiomatic characterization of the quadratic scoring rule. Exper. Econ. 1, 43–62.
Selten, Reinhard, Abbink, Klaus, Cox, Ricarda, 2005. Learning direction theory and the winner's curse. Exper. Econ. 8, 5–20.
Selten, Reinhard, Buchta, Joachim, 1999. Experimental sealed bid first price auctions with directly observed bid functions. In: Budescu, David, Erev, Ido, Zwick, Rami (Eds.), Games and Human Behavior: Essays in the Honor of Amnon Rapoport. Lawrenz Associates, Mahwah, NJ, pp. 79–104.
Selten, Reinhard, Chmura, Thorsten, 2008. Stationary concepts for experimental 2×2-games. Amer. Econ. Rev. 98 (3), 938–966.
Selten, Reinhard, Chmura, Thorsten, Goerg, Sebastian J., 2011. Stationary concepts for experimental 2×2-games: Reply. Amer. Econ. Rev. 101 (2), 1041–1044.
Selten, Reinhard, Stoecker, Rolf, 1986. End behavior in sequences of finite prisoner's dilemma supergames. J. Econ. Behav. Organ. 7, 47–70.