DICE

DISCUSSION PAPER

NO 437

Algorithmic Cooperation: A Comparison with Human Play in the Infinitely Repeated Prisoner's Dilemma

Bernhard Kasberger
Simon Martin
Hans-Theo Normann
Tobias Werner

June 2026

hhu
Heinrich Heine
Universität Düsseldorf

IMPRINT

DICE DISCUSSION PAPER

Published by:
Heinrich-Heine-University Düsseldorf,
Düsseldorf Institute for Competition Economics (DICE),
Universitätsstraße 1, 40225 Düsseldorf, Germany
www.dice.hhu.de

Editor:
Prof. Dr. Hans-Theo Normann
Düsseldorf Institute for Competition Economics (DICE)
Tel +49 (0) 211-81-15125, E-Mail normann@dice.hhu.de

All rights reserved. Düsseldorf, Germany 2026.

ISSN 2190-9938 (online) / ISBN 978-3-86304-436-7

The working papers published in the series constitute work in progress circulated to stimulate discussion and critical comments. Views expressed represent exclusively the authors' own opinions and do not necessarily reflect those of the editor.

Algorithmic Cooperation: A Comparison with Human Play in the Infinitely Repeated Prisoner's Dilemma*

Bernhard Kasberger† Simon Martin‡
Hans-Theo Normann§ Tobias Werner¶

June 2026

Abstract

Reinforcement learning algorithms play an increasingly important role in economic situations. These situations are often strategic, and the artificial intelligence may or may not be cooperative. We compare human and algorithmic cooperation rates in the infinitely repeated two-player prisoner's dilemma and study which strategies they choose to cooperate and punish deviations. Through a sequence of computational Q-learning and human-player experiments, we find that our Q-learning algorithms tend to cooperate less than humans, particularly when cooperation is risky or not incentive-compatible. Algorithms often use different strategies than humans, leading to distinct on- and off-path behavior.

Keywords: Artificial intelligence, cooperation, Q-learning, repeated prisoner's dilemma
JEL Codes: C72, C73, C92, D83

*We are grateful to our editor, Emmanuel Vespa, as well as three referees for very helpful comments. We further thank Maria Bigoni, Joe Harrington, Itzhak Rasooly, and Yaroslav Rosokha for valuable advice. Seminar participants and conference audiences at the following venues gave useful feedback on the paper: Aarhus University, Bielefeld University, University of Düsseldorf, University of Freiburg, LEE Workshop on Human-AI Interaction at Jaume I University, University of Linz, University of Maastricht, University of Münster, University of Potsdam, University of Salzburg, University of Vienna, Max Planck Institute for Research on Collective Goods, Purdue University, Annual Meeting of the German Economic Association (Regensburg), BECCLE (Bergen), BSE Summer Forum Workshop on Computational and Experimental Economics (Barcelona), ESA World Meeting (Lyon), Thurgau Experimental Economics Meeting on Technology and Human Behaviour (Kreuzlingen).

†bernhard.kasberger@jku.at. Department of Economics, Johannes Kepler University Linz and Austrian Institute of Economic Research

‡simon.martin@univie.ac.at. Department of Economics, University of Vienna; also affiliated with CEPR and CESifo

§normann@hhu.de. Düsseldorf Institute for Competition Economics (DICE), Heinrich-Heine-Universität Düsseldorf and Max Planck Institute for Research on Collective Goods, Bonn

¶tobias.felix.werner@soton.ac.uk. Department of Economics, University of Southampton

1 Introduction

Despite cooperation often increasing welfare, incentivizing agents to cooperate may be difficult. The prisoner's dilemma distills the essential incentives and rewards of such social dilemmas: The Pareto-efficient outcome is supported by dominated strategies, so each individual has a strong incentive to free-ride on the other player. Theoretically, it is well understood that the possibility of future interaction, or repetition, is essential for establishing cooperation among self-interested players: Future encounters can be used to incentivize compliance through the threat of punishment. However, as there are myriad equilibria for sufficiently high discount factors and uncooperative equilibria persist, it becomes an empirical exercise to study how the repeated prisoner's dilemma is played. The vast experimental literature (see our literature review below) has addressed the determinants, forms, and levels of cooperation for human players.

In this paper, we study whether Q-learning[1] algorithms or humans achieve higher cooperation rates in the repeated prisoner's dilemma and how they differ in implementing cooperation (strategy choice). Contrasting the levels of cooperation is useful because, without a human benchmark, any statements about the cooperative potential of algorithms are of limited value. Moreover, comparing strategies between algorithms and humans helps clarify when human and algorithmic behavior look alike and when they diverge. In our computational experiments, we place the algorithms into the same economic environments implemented in laboratory experiments and analyze their behavior with the tools used to study human behavior (see also Dal Bó and Fréchette, 2018). As with humans, we are interested in the determinants, forms, and levels of cooperation. In each of these dimensions, we draw on the experimental literature to understand the similarities and differences between algorithms and humans.

Our starting point is the finding by Calvano et al. (2020a) and Klein (2021) that reinforcement learning algorithms learn to collude in repeated

[1]Q-learning (Watkins and Dayan, 1992) is a form of reinforcement learning and widely studied in economics (Johnson et al., 2023, Asker et al., 2024). The principles of Q-learning are at the core of most of the more advanced (deep) reinforcement learning algorithms used in the field. For example, Deep Q-Networks (DQN) and Double Q-Learning extend its simple but effective value estimation process.

pricing games. Since collusion is a form of cooperation where punishment is used to deter deviations, and humans are also known to be capable of collusion, we investigate whether such algorithms cooperate more or less than human actors and whether the punishment strategies differ. To this end, we study Q-learning agents with a similar algorithm specification as Calvano et al. (2020a) and Klein (2021). With the prisoner's dilemma, we use a stage game with considerably fewer actions than a standard pricing game, allowing us to analyze the adopted punishment strategies. Our two main treatment variables are the reward for mutual cooperation (three different levels) and the discount factor (four different values). In addition, Q-learning has three hard-coded hyperparameters: memory, learning rate, and exploration rate.[2] We adopt the baseline hyperparameters from Calvano et al. (2020a). Although we vary them in robustness analyses, we do not treat them as primary treatment variables because they either lack a clear economic interpretation or are not pre-specified for human subjects. Following Calvano et al. (2020a), our main specification uses partner matching, where the same pair of algorithms interacts throughout training; in Section 7, we explore stranger matching, where algorithms are rematched across supergames within a group.

Besides the computational experiments, we run human-player laboratory experiments to collect data for previously unexplored parameter combinations. These results are of independent interest.

Our first finding is that humans tend to cooperate more on average than the Q-learning reinforcement algorithms. With the baseline hyperparameters, humans cooperate strictly more in all of our twelve treatments, and significantly so in eight of them. In particular, algorithms never cooperate for low discount factors and reward parameters, while humans achieve low but positive cooperation rates. Hence, humans cooperate significantly more in environments where cooperation is very risky or not incentive-compatible. However, the comparison becomes ambiguous for algorithms that are programmed to be more exploratory (in the sense of randomly try-

[2]Memory specifies how many previous rounds of play inform the algorithm's action choice. The learning rate governs how strongly recent payoff feedback updates the algorithm's value estimates. The exploration rate controls how frequently the algorithm deviates from its currently preferred action to try alternatives. All three are formally defined in Section 3.2.2.

ing more actions). Humans cooperate even more than algorithms for low and intermediate reward parameters and discount factors, but algorithms cooperate more when both the discount factor and the reward parameter are very high. It remains an open question whether these differences reflect an intrinsic property of tabular Q-learning or whether alternative design choices (e.g., different state representations or matching protocols) could close the gap.

Our main finding is that humans and Q-learning algorithms adopt different strategies to sustain and punish cooperation (given parameter combinations for which both humans and algorithms frequently cooperate). Dal Bó and Fréchette (2018) show that the most frequent cooperative strategies are tit-for-tat and grim trigger in human-player experiments. Strategies from these families are also adopted by human players in our experiments. While we find that algorithms also play tit-for-tat, they hardly ever select the grim trigger strategy under partner matching. Instead, algorithms play win-stay-lose-shift (Nowak and Sigmund, 1993), a strategy only rarely played by humans, and a hitherto undocumented strategy that cooperates if and only if both players defected in the previous round. Both algorithms playing this strategy leads to cycles, so cooperation is observed in only half of the rounds. We call this strategy win-shift-lose-shift. However, these strategy findings are sensitive to the matching protocol: under stranger matching with larger groups, win-shift-lose-shift disappears and grim trigger becomes prevalent among algorithms (see Section 7).

Related literature. The first strand of the literature we relate to studies human behavior in the indefinitely repeated prisoner's dilemma. Early laboratory experiments with human participants were conducted by Roth and Murnighan (1978). Dal Bó (2005) first implemented several supergames in the lab, each indefinitely repeated. The meta-study of Dal Bó and Fréchette (2018) summarizes the subsequent literature. Throughout the paper, we draw upon the insights and methods of this literature, such as Fudenberg et al. (2012) and Romero and Rosokha (2018), and structure the analysis similar to Dal Bó and Fréchette (2018).

The second related literature is the one on reinforcement learning algorithms in economics, which has so far largely focused on cooperation in the

sense of (socially undesirable) anti-competitive collusion in oligopoly games. Following an early study by Waltman and Kaymak (2008), Calvano et al. (2020a) and Klein (2021) show in simulation studies that Q-learning algorithms often learn to play collusive prices on-path and that *average* prices drop after a deviation and gradually increase again. However, it is difficult to describe the algorithms' strategies due to the relatively complex stage games, let alone how the distribution of strategies depends on the game parameters. In contrast, we analyze the repeated prisoner's dilemma (which can be seen as a pricing game with two stage-game actions). This allows us to get a more complete understanding of on-path and off-path behavior of Q-learning agents, using the strategy frequency estimation method (SFEM) by Dal Bó and Fréchette (2011). Moreover, our setting allows us to draw upon a rich set of experimental studies to compare human with algorithmic behavior.[3] Another difference is that while Calvano et al. (2020a) study the impact of many variables on the algorithms' propensity to collude, they do so with one variable at a time. In contrast, we explore treatment variation both in terms of parameters of the environment, as well as the hyperparameters of the Q-learning algorithm.

Schaefer (2022) and Boczoń et al. (2024a) also examine the determinants of cooperation among Q-learning algorithms in the repeated prisoner's dilemma. Schaefer (2022) calibrates a heuristic measure called the "kinetic log ratio" to explain the cooperation propensity. Boczoń et al. (2024a) test equilibrium selection focusing on the size of the basin of attraction of always defect and the effect of strategic uncertainty, using Q-learning agents with two internal states. In contrast, we compare cooperation rates and strategies between humans and Q-learning agents and also explicitly vary the memory length of the algorithms. Barfuss and Meylahn (2022) investigate the relevance of noise in sustaining cooperative outcomes for reinforcement learning algorithms. We add to this literature not only the systematic analysis of the determinants of cooperation but also of the strategies played by algorithms.

There are several additional approaches to studying the potential for

[3]While this study focuses on the algorithms themselves, simpler forms of reinforcement learning have been used to describe human learning in games (see, for instance Roth and Erev, 1995, Romero and Rosokha, 2019). Unlike Q-learning, these models are often myopic with limited foresight.

cooperation and learning strategies that contribute to cooperation. Hanaki et al. (2005) endogenize initial attractions in an adaptive model of strategy learning. Ioannou and Romero (2014) develop a methodology of belief learning, which is, among others, applied to the prisoner's dilemma we study as well. Romero and Rosokha (2019) show how reinforcement learning algorithms can be extended to continuous action spaces. While these papers also find that learning algorithms may achieve cooperation, they do not consider Q-learning in the form commonly used in the computer science literature. Their motivation lies instead in modeling human behavior and learning processes.

The computer science literature that studies artificial intelligence in strategic situations often prioritizes algorithmic performance in various games (Crandall and Goodrich, 2011, Crandall et al., 2018). Unlike these studies, we analyze a fundamental reinforcement learning algorithm, and apply tools from game theory and experimental economics to describe its behavior.[4] This approach sets our work apart from studies like that of Sandholm and Crites (1996), which focused solely on the performance of Q-learning in the prisoner's dilemma without considering variations in environmental parameters or employing the economic tools we use to understand behavior.

Related to our paper are the experiments on the interaction of humans and algorithms (Crandall et al., 2018). While their focus is on designing algorithms that successfully cooperate with humans, we study how Q-learning algorithms behave in a repeated prisoner's dilemma and compare their behavior directly to that of human subjects. Normann and Sternberg (2023) analyze a prisoner's dilemma experiment with three players where one of the players may or may not be a pre-programmed algorithm. In a market environment, Werner (2022) conducts lab experiments in which humans either play with other humans or against self-learned pricing algorithms. He finds that algorithms can be more collusive than humans. In

[4]The interaction and training of algorithms are closely connected to the environments examined in evolutionary game theory. For instance, Zhang (2018) demonstrates that errors can foster cooperation in finite populations. In Zhang (2018), players choose strategies (finite automata) to play the indefinitely repeated prisoner's dilemmas, and he shows that the WSLS strategy remains stable only under low levels of noise. Strategies such as the grim trigger and tit-for-tat appear to play a less central role than they typically do in the experimental literature.

Table 1: Stage game payoffs

	C	D
C	R, R	12, 50
D	50, 12	25, 25

contrast, we compare human-human with machine-machine interactions.

2 Economic environment and hypotheses

This section introduces the game, the types of strategies that we analyze, and formulates the research questions.

2.1 Game, memory and strategies

We study the infinitely repeated two-player prisoner's dilemma with perfect monitoring. There are two players who repeatedly play the stage game prisoner's dilemma and discount future payoffs with the common discount factor δ. In the stage game, each player either cooperates (C) or defects (D). The payoffs are in Table 1. R denotes the reward for mutual cooperation. The prisoner's dilemma arises when D is strictly dominant, (D, D) the unique stage-game Nash equilibrium, and (C, C) Pareto-efficient. We consider reward parameters that satisfy $31 < R < 50$, which also imply that mutual cooperation is Pareto-efficient in the repeated game.

We restrict attention to finite-memory strategies of the infinitely repeated prisoner's dilemma. This stands in contrast to the theoretical textbook treatment of repeated games with perfect monitoring, where players can condition their actions on the entire history of past play. However, as arbitrarily long histories require unbounded memory, such strategies cannot be implemented by finite algorithms in general and Q-learning in particular. Thus, we consider Markov strategies where the states are the action profiles of the past k rounds; $k \in \mathbb{N}$ is each player's memory. For example, a memory-one strategy specifies behavior for the four states CC, CD, DC, and DD. Throughout, we use the first letter to indicate player 1's action in a specific state, e.g., player 1 played C and player 2 played D in the previous round in state CD. For the analysis of laboratory experiments, Fudenberg et al. (2012) suggest 20 plausible strategies, which are at most

memory-three. The results in Dal Bó and Fréchette (2019) imply that participants only use a few strategies, and these are up to memory two. Thus, the restriction to low levels of memory does not seem overly restrictive in comparison to human actors.

The following low-memory strategies are particularly relevant in our context. The first is 'always defect' (AllD), which prescribes playing D, the strictly dominant stage-game action in any round. (AllD, AllD) always forms a subgame perfect Nash equilibrium (SPNE). 'Always cooperate' (AllC) prescribes playing C regardless of the previous round's outcome and is never an SPNE. These strategies do not show any reward and punishment behavior and can be implemented with zero memory. In contrast to AllC and AllD, 'Tit-For-Tat' (TFT) is reciprocal and cooperative: TFT begins with C in period one and mimics the rival's action subsequently. TFT has a minimal memory of one and (TFT, TFT) is generically not an SPNE. A strategy with punishment that potentially forms an SPNE is 'grim trigger' (GT): A player starts by cooperating but defects whenever either player played D in the previous round. This version of GT requires a minimal memory of one. More forgiving than GT is the strategy 'win-stay, lose-shift' (WSLS, sometimes also referred to as 'perfect TFT;' Nowak and Sigmund, 1993). A player following WSLS starts with cooperation and then cooperates if and only if both players chose the same action in the previous round, which makes it a memory-one strategy. (WSLS, WSLS) can be an SPNE, and, unlike TFT and GT, can correct erroneous defections.

We also consider strategies that condition on the action profiles of the past $k > 1$ rounds. For example, a trigger strategy that punishes for two rounds regardless of the actions during the punishment rounds (T2) is a memory-2 strategy. There are also memory-2 versions of TFT and WSLS, such as TF2T (cooperate unless the opponent defected in both of the last two rounds), and WSLS with two rounds of punishment. Similar extensions are possible for memory three, such as T3 and TF3T.

2.2 Research questions and hypotheses

Our hypothesis and exploratory questions are organized in two sets: first, the determinants of algorithmic cooperation, and then the comparison with

humans. Two key economic variables in our repeated prisoner's dilemma are the discount factor δ and the payoff of mutual cooperation R. The experimental literature has shown that cooperation among humans can be expected to increase in both parameters (Dal Bó and Fréchette, 2018). A higher discount factor δ increases the probability of future interactions and makes cooperation more attractive compared to short-run gains from defection. A larger reward payoff generally makes cooperation more attractive. We expect the same comparative statics to hold for algorithms.[5]

Hypothesis 1. *The cooperation rate among Q-learning algorithms increases in R and δ.*

Calvano et al. (2020a) find a non-monotone relationship between δ and collusion (cooperation) in their repeated differentiated Bertrand game. The relationship is positive for relatively high δ, however. Due to the different games, they cannot study variation in R and, therefore, not the interaction between R and δ. Online appendix S.3 contains additional hypotheses related to the determinants of cooperation that are inspired by the experimental literature.

Our first exploratory question relates to the levels of cooperation of algorithms and humans. Humans are able to sustain cooperation in lab experiments (Dal Bó and Fréchette, 2018) and reinforcement learning algorithms learn to cooperate (collude) in pricing games (Calvano et al., 2020a). We are interested in comparing the respective levels of cooperation.

Exploratory Question 1. *When are algorithms more or less cooperative than humans, and how does the comparison depend on the game parameters δ and R and on the hyperparameters of the algorithm?*

Our second exploratory question relates to the forms of cooperation. In laboratory experiments, humans mostly play the strategies always defect, tit-for-tat and grim trigger (Dal Bó and Fréchette, 2011, Fudenberg et al., 2012, Bigoni et al., 2015). How do algorithms cooperate on path and how do they punish deviations off path?

[5]The discount factor δ of an *infinitely* repeated game is typically represented through a random termination probability $1-\delta$ in an *indefinitely* repeated game with human subjects in the laboratory, which are strategically equivalent under risk neutrality. The parameter is identically represented in Q-learning through the formulation of the Bellman equation, and hence the way it is used to update the Q-matrix in each iteration.

Exploratory Question 2. *Which strategies do algorithms learn, and how do the strategies depend on the game parameters δ and R, and on the hyperparameters of the algorithm?*

3 Experimental Design

Our experiment compares prisoner's dilemmas in which both players are either Q-learning algorithms or humans. Before we describe the computational and human-player designs, we discuss the treatments of the game parameters δ and R.

3.1 Treatments

Our choice of game parameters is guided by two considerations: covering theoretically meaningful variation in the scope for cooperation, and enabling direct comparison with existing human experimental data.

We study a 3×4 design. We consider $R \in \{32, 40, 48\}$ and $\delta \in \{0.50, 0.75, 0.90, 0.95\}$, motivated by configurations also used in Dal Bó and Fréchette (2011), Ghidoni and Suetens (2022), and Romero and Rosokha (2018). The variants with $\delta = 0.95$ are particularly relevant to compare with the parameterization used in Calvano et al. (2020a), Klein (2021), and other studies using algorithmic simulations. In human experiments, a discount factor of $\delta = 0.95$ (and indeed $\delta = 0.9$) has only been studied for $R = 32$; see Table S.2 in the online appendix. By adding the variants $R = 40$ and $R = 48$ with the discount factors $\delta = 0.9$ and $\delta = 0.95$, our study adds to the literature on human cooperation independently of the algorithmic simulations.

From a theoretical perspective, collusion can be credibly sustained (as an SPNE) unless $\delta = 0.5$ and $R = 32$. Hence, "rational" players cooperate only if δ or R take higher values.[6] Increasing δ or R decreases the size of the basin of attraction of always defect ("sizeBAD"), which has been used as a measure of how robust cooperation is to strategic uncertainty (Dal Bó and

[6]Q-learning can lead to cooperative outcomes even in the absence of memory, in which case DD is the unique equilibrium profile (Asker et al., 2024, Banchio and Mantegazza, 2022, Dolgopolov, 2024). Hence, Q-learning is known to not always lead to "rational" outcomes.

Fréchette, 2011, Boczoń et al., 2024b).[7] SizeBAD has been used to explain why cooperation might increase in δ and R. Dal Bó and Fréchette (2018) find that human cooperation rates decrease in the size of the basin of AllD across experiments. A robust finding of the experimental literature is that cooperation is more likely when it can be supported as a (risk-dominant) equilibrium (Blonski et al., 2011).

3.2 The algorithmic Q-learning experiments

We introduce the Q-learning algorithms, for ease of exposition, only for memory one; the general case is in the online appendix. We then describe our computational implementation in detail.

3.2.1 The reinforcement learning algorithm

The decision-making process of a tabular Q-learning player is represented by a Q-matrix. The dimension of this Q-matrix depends on the player's memory, that is, how many past periods the player considers for the decision in the given period, and the number of possible actions. With memory one, the Q-matrix has four rows (one row for each state) and two columns (one for each action). The entries $Q(s, a)$ of the Q-matrix are the current approximations of the expected discounted utilities when choosing action a in state s. The players use their Q-matrices to choose actions and update their approximations of the long-run payoffs. For a given Q-matrix, the optimal strategy assigns to each state the action with the highest value.

Q-learning starts with some initial Q-matrix. At time t in state s, player i chooses the optimal ("greedy") action with probability $1 - \varepsilon_t$; the player exploits their knowledge as encoded in the Q-matrix. With complementary probability, the player explores other, possibly suboptimal, actions and chooses an action uniformly at random. This form of random exploration aims at balancing a trade-off for the algorithm. On the one hand, the player wants to exploit the knowledge it already has in the form of the Q-matrix.

[7] To define the basin of attraction of always defect, consider a hypothetical coordination game in which the players choose between the repeated-game strategies GT and AllD. In this game, the players believe that the opponent plays GT with probability p and AllD with probability $1 - p$. The basin of attraction of AllD is then defined as the maximum p that makes it still optimal for a player to play AllD.

On the other hand, the player has to explore the state space to improve the approximation of the profitability of other state-action combinations.

Irrespective of whether the action a was chosen through exploitation or exploration, the player obtains feedback through the stage-game payoff $\pi(s,a) \in \{12, 25, R, 50\}$ that is observed after both players' actions are realized. The player uses the payoff feedback in round t to update the estimate of the long-run payoff of choosing action a in state s according to

$$Q_{t+1}(s,a) = (1-\alpha) \underbrace{Q_t(s,a)}_{\text{old value}} + \alpha \Big(\underbrace{\pi(s,a)}_{\substack{\text{current} \\ \text{payoff}}} + \underbrace{\delta \max_{a' \in \{C,D\}} Q_t(s',a')}_{\substack{\text{estimated} \\ \text{long-run payoff}}} \Big).$$

The new value is a convex combination of the old value, and the current stage-game payoff π plus the best possible estimated long-run payoff in the next state s'. The weight put on the latter payoff is denoted by α and referred to as the *learning rate*. The next state s' is given by the actions from all players over the past k periods, which are commonly observable, including period t. Only after all players have chosen their actions, we calculate the payoffs from the current round, construct the continuation state, and update the Q-matrices. Importantly, each player updates only a single cell in each period.[8]

Besides the learning rate α, a key parameter is the exploration probability ε_t. Following common practice in the literature (e.g., Calvano et al., 2020a), we choose ε to decay over time; specifically, $\varepsilon_t = e^{-\beta t}$, where $\beta > 0$. The updating procedure in Q-learning also crucially depends on the discount factor δ, which we vary across treatments. While δ is given by

[8]The literature distinguishes between *synchronous* and *asynchronous* algorithms (Asker et al., 2024). Following Calvano et al. (2020a), we study asynchronous Q-learning where a single Q-value is updated in each period. Synchronous learning, on the other hand, would also update the Q-values associated with non-chosen actions. Asker et al. (2024) find that prices quickly converge to the static Nash equilibrium in a homogeneous pricing game with synchronous learning, whereas supracompetitive prices are observed for asynchronous algorithms. The analysis of Banchio and Mantegazza (2022) reveals the forces behind some of these results. They apply continuous-time methods and find that spontaneous coupling between the estimates of the algorithms leads to periodic coordination on cooperative actions. They derive a sufficient condition for such coupling not to occur. Similarly, Boczoń et al. (2024a) find that cooperation rates are higher for asynchronous algorithms in their repeated prisoner's dilemma. Therefore, our results primarily apply to asynchronous learning algorithms.

the environment that the algorithm is acting in, α and β are hyperparameters. They are not learned by the algorithm and not optimized over, but exogenously given by the researcher.[9] Another important parameter is ν, which is implied by β, and memory k. It denotes the expected number of times a cell in the Q-matrix is explored purely by randomness, disregarding optimality (Calvano et al., 2020a). The interest in this parameter stems from the fact that for a fixed β, the probability that a cell is visited by chance through exploration is smaller in larger state spaces (and hence for higher memory k). In our experiment, we keep ν constant across k to at least partially control for this interaction. Online appendix S.5 contains the formula of ν (Equation 5).

3.2.2 Details on the computation experiments

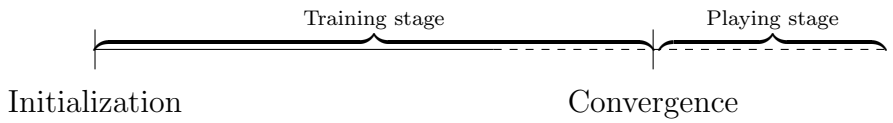
In our AI-based experiment, we distinguish for each run (parameterization) the training stage and the playing stage, as illustrated in Figure 1. In the training stage, two Q-learning algorithms repeatedly play the stage game in Table 1 and adjust their strategies according to the common discount factor δ, the learning rate α, and the exploration parameter β. The algorithms explore non-greedy actions with exogenous probability, where the probability decreases exponentially in time and according to the parameter β. The training ends when neither algorithm changes the policy in any state for 10^8 rounds, or after a total number of 10^9 rounds.[10] In the subsequent playing stage, lasting 1,000 periods, the two agents stay the same as before and the algorithms' initial actions are the optimal actions in the round of convergence, implying that the playing stage is a seamless continuation from training. After that, they play according to the learned strategies and no longer adjust the Q-values. Throughout the paper, we focus on the strategies learned upon convergence and deliberately abstract from the path toward convergence, as well as the strategies and payoffs

[9] This specification enables the most direct comparison with human subjects. Harrington (2022) and Abada et al. (2022) study unilateral incentives to use or improve algorithms.

[10] The necessity for such a tight convergence criterion arises in the context of $k = 3$ and $\nu = 1,000$, which features a large state space and substantial initial exploration that slows convergence times. In order to allow comparability across parameterizations, we use the same convergence criterion throughout.

along that path. As such, our results describe the behavior of fully trained algorithms and do not speak to the transient dynamics during learning.

Figure 1: Simulation outline.



Note: For each simulation, two new agents are initialized randomly. During the training stage, agents learn strategies as specified by the Q-learning algorithm. We evaluate cooperation rates and strategies upon convergence, i.e., in the playing stage. Agent pairs remain the same during the training and playing stage, and a new set of agents is initialized for each simulation. The entire simulation procedure is repeated 1,000 times for each parameterization.

We vary the hyperparameters in three-dimensional space with cardinality $3 \times 5 \times 5$. In the first dimension, memory k is either one, two, or three.[11] To account for the fact that a smaller β is needed to explore the state space sufficiently often in larger state spaces, we choose β as a function of memory. In particular, we choose $\beta(k)$ to keep ν constant for all k. We have $\nu \in \{4, 20, 100, 450, 1000\}$ in the second treatment dimension and $\alpha \in \{0.05, 0.10, 0.15, 0.20, 0.25\}$ in the third.[12] For each parameterization, we repeat 1,000 runs with a different random seed. In our main specification, we initialize the Q-matrix by adding a uniform random draw from the unit interval to $6/(1-\delta)$. We show that our main results are robust to different initializations of the Q-matrix in Section S.4 in the online appendix.

[11] A memory length of up to $k = 3$ improves upon the existing economics literature. We cannot accommodate even higher memory due to the exponentially growing state space. Hettich (2021) demonstrates that algorithms using function approximation techniques like neural networks to represent the Q-matrix produce comparable outcomes to Calvano et al. (2020a). Given the simplicity of the action and state space in our environment, employing a tabular Q-learning algorithm with expanded memory is likely to cover most algorithmic behaviors.

[12] For comparison, Calvano et al. (2020a) focus on memory one and consider several values for β such that the implied ν is in $[4, 450]$. However, most of their analysis focuses on the case where $\nu \approx 20$, which will also be our main specification. The precise values we use in our baseline experiments and robustness checks are shown in Tables S.25 and S.26 in the online appendix.

3.3 The human lab experiments

The experiments involving human participants were run as standard lab experiments. The experimental design and instructions were identical to Dal Bó and Fréchette (2011), Romero and Rosokha (2018), Kartal and Müller (2021), and Ghidoni and Suetens (2022). In order to compare algorithmic cooperation with human cooperation for each treatment cell, we conduct the treatments $R = 40$ and $R = 48$ with the discount factors $\delta = 0.9$ and $\delta = 0.95$ as lab experiments. For the other treatments, ample lab data already exist (see Table S.2 in the online appendix for a complete list of experimental data from other studies that we use in this paper).

The experiments took place at the DICElab of the University of Düsseldorf and the PLEx at the University of Potsdam between December 2022 and May 2023. Subjects were recruited from the labs' subject pools using hroot (Bock et al., 2014). Upon arrival at the lab, participants randomly drew a token, assigning them a cubicle number. Printed instructions were distributed and summarized verbally. Participants were also given the opportunity to ask questions individually and privately. We ensured complete anonymity.

Subjects played several supergames. We aimed at a maximum of 15 supergames in each session unless the (pre-announced) time limit of two hours was exceeded. In that case, the supergame that was started before the two hours were up was the final supergame. The matching was fixed within a supergame, but random when a new supergame started. Sessions were conducted with twenty or thirty participants. The random matching across supergames was done within groups of ten subjects.

We pre-registered the human experiments and the hypotheses pertinent to human behavior at https://osf.io/zcv6x/, and we executed the experiments as registered. We had a total of 240 participants. Participants earned 22.54 euros on average.

4 Humans tend to cooperate more than algorithms

This section investigates whether humans or algorithms are more cooperative on average, and how the comparison depends on the game parameters

δ and R and on the algorithms' exploration. Throughout, the baseline specification involves $k = 1$, $\alpha = 0.15$, and $\nu = 20$, as in Calvano et al. (2020a). Section 6 reports cooperation rates beyond the baseline setup.

Figure 2 reports the average cooperation rates of algorithms and humans for each δ–R treatment. Figure 2a focuses on the algorithm's cooperation rate, which is the average over 1,000 periods after the algorithms' convergence, as explained in Section 3. Cooperation is far from dominant, let alone perfect: Even for high combinations of δ and R, cooperation rates do not exceed 60%. Algorithmic cooperation increases monotonically and substantially in both δ and R, with one exception: For $R = 48$, the shift from $\delta = 0.9$ to $\delta = 0.95$ leads to a *decrease* in cooperation (two-sided Mann-Whitney-U-test p-value < 0.001).[13] We will return to the non-monotonicity when we examine the learned strategies. Despite the non-monotonicity in δ for high values of R, we take the following result from Figure 2a, which is consistent with Hypothesis 1. (We investigate the dependence of the algorithms' cooperation rate on the game and hyperparameters more systematically in Section 6.)

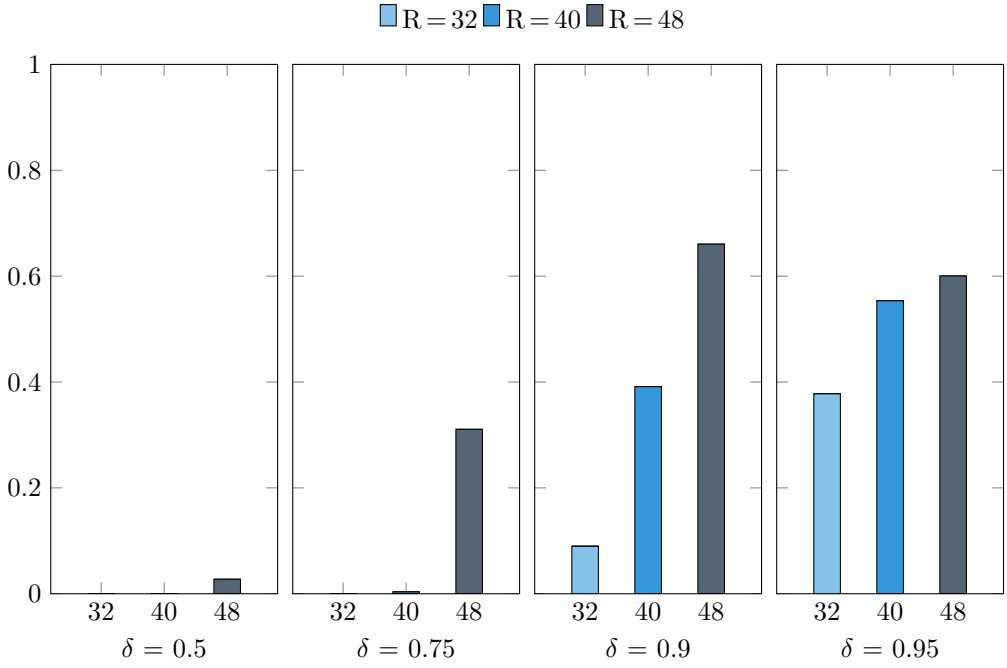
Result 1. *The average cooperation rate among Q-learning algorithms increases in R and δ.*

Figure 2b shows the human cooperation rates across all rounds and supergames from the laboratory experiments.[14] The figure includes previous experiments with all $\delta \leq 0.75$ or $R = 32$ treatments, and our experiments with $R \geq 40$ and $\delta \geq 0.90$; cooperation is relatively high for these δ–R treatments. In general, human cooperation is visibly higher than that of algorithms. There is non-zero cooperation in all treatments and the cooperation rates go up to 90%. While the cooperation rate increases in δ and R, humans interestingly exhibit the same non-monotonicity in the cooperation rate for $R = 48$ as δ increases from 0.9 to 0.95.

[13] We use a nonparametric rank sum test here and elsewhere because the normality assumption required to use the t-test is violated for the human data. Additionally, we conducted a skewness test for normality of algorithmic data (D'agostino et al., 1990); normality is rejected (p-value < 0.001) for 11 out of 12 parameterizations in the baseline setting.

[14] A detailed list of sources used for these calculations is provided in Table S.2 in the online appendix, and analysis across supergames is provided in Table S.28.

(a) Algorithmic cooperation rates



(b) Human cooperation rates

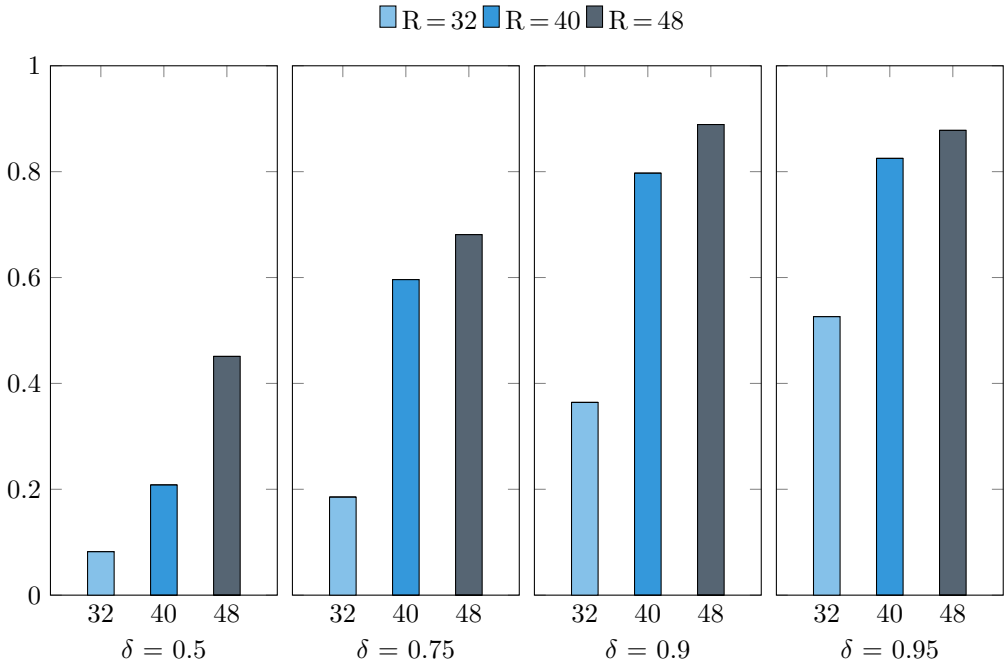


Figure 2: Cooperation rates by δ–R treatment.

Note: The figure reports the algorithmic cooperation rates for the baseline parameters $k = 1$, $\alpha = 0.15$, $\nu = 20$, computed over 1,000 periods after convergence. An observation is one simulation for algorithms (1,000 per parameterization), and a matching group for human experiments (between 6 and 11 per parameterization). The numerical values are in the online appendix in Table S.3 and Table S.4, respectively.

Figure 2 suggests that humans cooperate more on average than algorithms. For the baseline parameterization ($\alpha = 0.15$, $\nu = 20$, $k = 1$), Table 2 formalizes the relationship by testing the statistical significance of the different cooperation rates with a two-sided Mann-Whitney-U-test. We find that humans cooperate more in all treatments, although the difference is not always statistically significant. A second insight is that humans cooperate to some extent where the algorithm entirely fails to choose C, namely when $\delta = 0.5$ and in treatment ($\delta = 0.75$, $R = 32$). The difference is relatively minor in treatment ($\delta = 0.50$, $R = 32$) as humans also cooperate little in this treatment on average. On the other hand, humans cooperate with an average rate of about 60% significantly more in the ($\delta = 0.75$, $R = 40$) treatment, where algorithms cooperate at a mere rate of 0.35%. We see this as suggestive evidence that humans try to establish cooperation even in environments where it is hard to sustain. Third, the differences in the cooperation rates are also high for high δ–R treatments, although not all of the differences are statistically significant. Overall, humans cooperate more than algorithms with the baseline parameters.

Having said that, the conclusion that humans cooperate more than the algorithm is not generally tenable. For a second parameterization with significantly higher exploration ($\alpha = 0.15$, $\nu = 1,000$), Table 2 reports that algorithms cooperate more on average for the highest δ–R treatments.[15] When $\delta = 0.95$ and $R \geq 40$, the Q-learning algorithms cooperate significantly more than humans. However, perhaps surprisingly, average algorithmic cooperation declines in ν unless δ and R are very high ($R \geq 40$ and $\delta = 0.95$ or $R = 48$ and $\delta \geq 0.90$). It follows that humans still cooperate more unless $\delta = 0.95$ and $R \geq 40$. It appears that in environments in which it is relatively difficult to cooperate, humans establish more cooperation. On the other hand, in settings where collusion is relatively easy to sustain, algorithms that explore extensively cooperate more. We summarize our findings by answering Exploratory Question 1:

[15] In Section S.4.7 in the online appendix, we present heatmaps of average cooperation rates for all our α, ν and k combinations. Our main results on average cooperation rates are very robust in all of these dimensions.

Table 2: Difference human vs. algorithmic cooperation rates, $k = 1$, $\nu = 20$ and $\nu = 1000$

$\nu = 20$	δ	$R = 32$	$R = 40$	$R = 48$
	0.50	8.15^{***}	20.82^{***}	42.38^{***}
	0.75	18.54^{***}	59.28^{***}	37.08^{***}
	0.90	27.45^{***}	40.62	22.86
	0.95	13.85	27.16	27.80^{*}
$\nu = 1000$	δ	$R = 32$	$R = 40$	$R = 48$
	0.50	8.15^{***}	20.82^{***}	45.13^{***}
	0.75	18.54^{***}	59.63^{***}	41.15^{***}
	0.90	36.42^{***}	60.23^{***}	7.46
	0.95	44.02^{***}	-2.19^{***}	-3.51^{***}

Note: This table depicts the difference between cooperation rates of humans and algorithms, as well as the significance of a two-sided Mann-Whitney-U-test. For the Mann-Whitney-U-test, an observation is one simulation for algorithms (1,000 per each parameterization), and a matching group for human experiments (between 6 and 11 per parameterization).* $p < 0.05$, ** $p < 0.01$, *** $p < 0.001$

Result 2. *When cooperation is relatively hard to sustain, humans cooperate more than algorithms. The comparison is ambiguous in other cases. Even with high exploration, algorithms may cooperate significantly less than humans.*

Humans cooperate more than algorithms, even though the two are studied under different matching protocols. Humans are matched to another human for one supergame, and then matched to another participant for the next supergame. The Q-matrices remain matched throughout the learning and playing phases, which is a common approach in the recent literature on reinforcement learning algorithms in economics.[16] In Section 7, we investigate alternative matching protocols and find that the cooperation gap is robust, though the strategies that sustain algorithmic cooperation are sensitive to the matching environment.

[16]While the algorithm eventually stops exploring, there is evidence that humans continue randomization to learn or have a preference for randomization (Agranov and Ortoleva, 2022). Moreover, there is a substantial amount of randomization even in the late stages of human laboratory experiments (Romero and Rosokha, 2023).

Humans and algorithms learn to cooperate in very different ways. Humans do not have a parameter that determines the exploration of new strategies, and they can learn within and across supergames. Algorithmic learning, on the other hand, is strongly influenced by the exogenous exploration parameter. Additionally, for reinforcement learning algorithms, "learning to cooperate" and "cooperation" itself are separate issues due to the learning and playing phases. However, for humans, the data comprises both phases.

Where humans and Q-learning algorithms differ most strongly is in the learning phase. Humans need only a small number of rounds or a few supergames to cooperate; they belong to a generally cooperative species. In experimental games, humans are matched with many different opponents, so they are exposed to a much less stable environment than algorithms are. Also, they learn to cooperate throughout their entire lives, not only in the limited span of the experimental session. In contrast, reinforcement learning algorithms must start from scratch and require a large number of rounds to learn. Humans can interpret each other, play deliberately, and infer the intentions of their opponents. These differences between humans and algorithms appear to explain some of our results, such as why humans cooperate more when cooperation is relatively difficult to sustain. While the discount factor and reward parameter affect both human and algorithmic "learning to cooperate," the different natures of learning lead to different outcomes in supergames, such as the more forgiving nature of the strategies employed by the algorithm.

5 Algorithms cooperate differently than humans

We now analyze the strategies that algorithms learn to play and compare them to those used by humans. These strategies tell us how the players cooperate and how cooperation is sustained through punishment. We again focus on our baseline parameters but also study how the strategies depend on the game, learning, and exploration parameters. We start with memory one, in which case the strategies can be inferred from the Q-matrices. In the online appendix, we investigate algorithms with higher memory and estimate the frequencies with which certain strategies are being played.

5.1 Algorithmic strategies

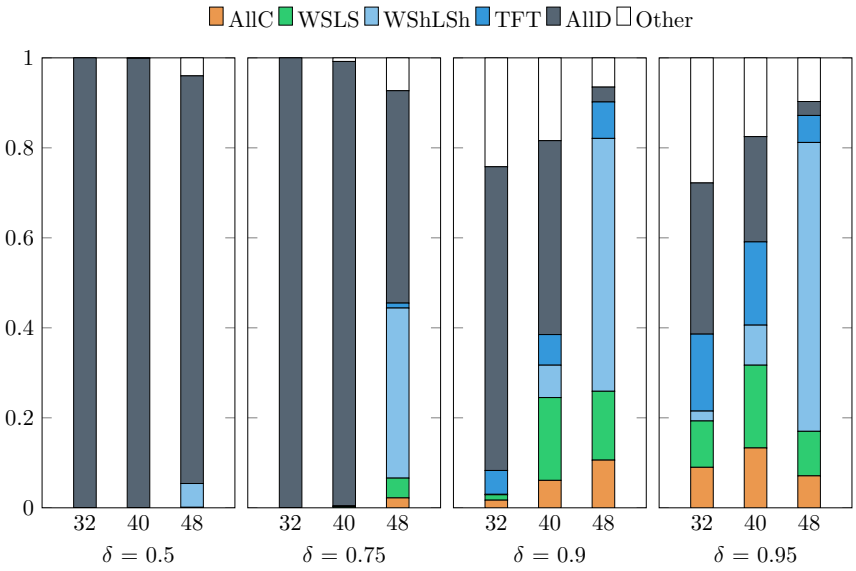
To analyze the algorithms' strategies, we classify them "manually," that is, we infer the strategies from the Q-matrix upon convergence as the state-wise maximizers. In our analysis, we allow the following 18 strategies, which include classic strategies such as tit-for-tat (TFT), grim trigger (GT), and win-stay-lose-shift (WSLS). First, we include the 16 memory-one strategies when ignoring the initial state (see Table S.1 in the online appendix).[17] We also include two strategies that take the initial state into account. The first is AllD', which plays grim trigger but starts in a state other than CC; because the initial state leads to immediate defection, AllD' is behaviorally equivalent to AllD and grouped with it. The second is AllC', which cooperates in every state except DD and starts in a state other than DD; on path against a cooperative opponent, DD is never reached, so AllC' is behaviorally equivalent to AllC and grouped with it. These two strategies are relatively frequent. For tractability, we work with these 18 strategies rather than the full set of 32 that arises when the initial action is also distinguished.

Figure 3a shows the distribution of the learned strategies when memory is hard-coded to one ($k = 1$). Consistent with low cooperation rates (Figure 2a), AllD dominates for low δ–R combinations. The share of AllD decreases in δ and R. For $R \geq 40$ and $\delta \geq 0.9$, cooperative strategies emerge more persistently: we mainly observe AllC, TFT, and WSLS. However, AllD is still the modal strategy for ($\delta = 0.90$, $R = 40$). Win-shift-lose-shift (WShLSh) is most common for $R = 48$, and it is even the modal strategy for $\delta \geq 0.9$ and $R = 48$ under partner matching. It prescribes defecting unless both players defected in the previous period. WShLSh is learned so often that the average cooperation rate actually decreases when δ is increased from 0.9 to 0.95 at $R = 48$. Note that WShLSh never forms a symmetric SPNE. Nevertheless, the algorithms learn the strategy when R is large. We discuss WShLSh in Section 5.4.

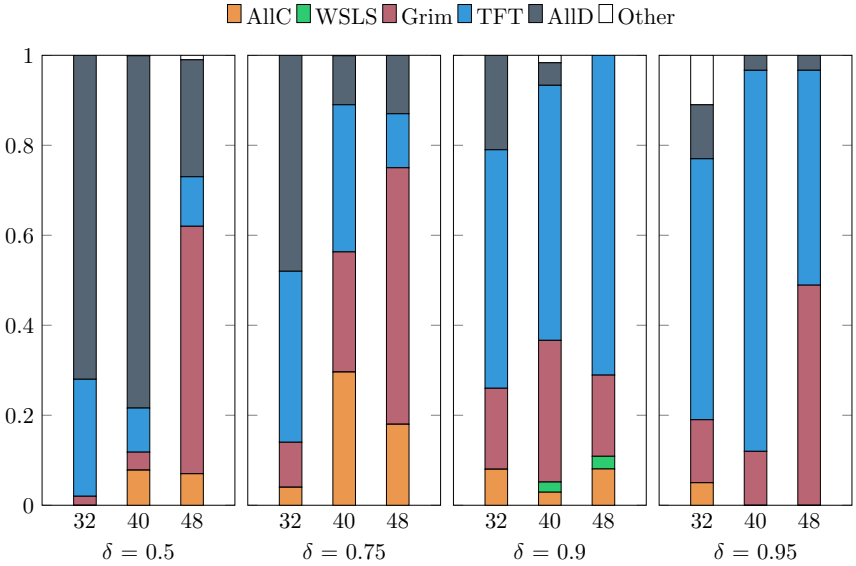
Result 3. *With memory one and partner matching, the most frequently*

[17] Since our classification conditions on the four memory-one states but not on the initial action, it pools strategies that differ only in round one. In particular, TFT and Suspicious TFT (STFT)—which both mimic the opponent's previous action but differ in whether they start with C or D—are grouped under the TFT label.

(a) Strategy choices of algorithms.



(b) Strategy choices of humans.



Note: The first figure reports the manual classification for $k = 1$ for the baseline parameters $\alpha = 0.15$, $\nu = 20$. The numerical values are available in Table S.6 in the online appendix. The second figure reports the distribution of strategies played by humans. We use the frequency estimates from Dal Bó and Fréchette (2019) for $\delta = \{0.5, 0.75\}$ with $R = \{32, 48\}$ and $\delta = 0.9$ with $R = 32$; from Dal Bó and Fréchette (2011) for $\delta = \{0.5, 0.75\}$ with $R = 40$; from Romero and Rosokha (2018) for $\delta = 0.95$ with $R = 32$; and our own data for $\delta = \{0.9, 0.95\}$ with $R = \{40, 48\}$. For the human-player data, GT and TFT include higher-memory variants of these strategies (see Table S.7 in the online appendix for details). Strategies with an estimated frequency below 5% are classified as "other."

Figure 3: Strategy choices by δ–R treatment.

learned strategies by the algorithms are AllD, WSLS, TFT, and WShLSh.

These strategy findings depend on the matching protocol. Under stranger matching with larger matching groups, WShLSh disappears and grim trigger becomes the dominant cooperative strategy (see Section 7).

5.2 Comparison to humans

How do the strategies implemented by humans differ from those chosen by the algorithms? We employ the Strategy Frequency Estimation Method (SFEM) of Dal Bó and Fréchette (2011) to estimate human strategies; see the online appendix for details. The results are shown in Figure 3b. The most popular strategies among human players are AllD, TFT, and GT across treatments and experiments.[18] Consistent with the respective cooperation rates, some humans play cooperative strategies even for low δ–R combinations, while algorithms basically always defect for these. Some humans even always cooperate but most cooperate only if the other player also cooperates. For higher δ–R treatments, humans mainly choose GT and TFT. Together, they account for 83% to 95% of the strategy estimates (when $\delta \geq 0.9$ and $R \geq 40$). This exceeds the share of TFT and GT in the most cooperative games in Dal Bó and Fréchette (2018). It appears that human subjects learn not to cooperate unconditionally, despite the high cooperation rates. For these treatments, algorithms mainly learn WSLS, TFT and WShLSh. However, the frequency of TFT is higher among humans, and algorithms play WSLS and WShLSh, two strategies that humans do not play at all. AllC is played by both player types but less often overall than these other strategies.

Result 4. *Humans mainly cooperate by playing grim trigger and tit-for-tat. Under partner matching, algorithms mainly cooperate by playing win-shift-lose-shift and win-stay-lose-shift.*

Both humans and algorithms frequently cooperate through strategies that do not form subgame-perfect Nash equilibria. For humans this is

[18] In their meta study, Dal Bó and Fréchette (2018) find that, across experiments, humans tend to adopt AllD, TFT, and GT. Romero and Rosokha (2023) found that experienced humans tend to use the pure strategies AllD, TFT, and GT also when they have the option to play mixed strategies.

mainly TFT and for algorithms this is WShLSh (and to a lesser extent TFT). However, it has been argued that TFT is empirically not a bad strategy (Axelrod, 1984). In contrast, WShLSh leads to cycles between CC and DD and can be easily exploited. Hence, humans choose strategies that implement higher cooperation levels and that are more robust to exploitation. On the other hand, algorithms learn WSLS in certain treatments, which is subgame-perfect and potentially restarts mutual cooperation (in contrast to GT and mutual TFT). In the online appendix, Section S.2.3, we assess whether the differences between strategies used by algorithms and by humans are statistically significant. Log-likelihood ratio tests reject the hypothesis that algorithmic strategies are fully captured by the set of human strategies. This indicates that restricting attention to human strategies entails a loss of information: the strategies employed by the algorithms are not, in general, well approximated by the set of human strategies for these parameterizations. This conclusion is consistent with the observation that algorithms frequently employ WSLS and WShLSh, strategies that humans never use.

5.3 Hyperparameters and algorithms' strategies

To further contrast humans' and algorithms' strategies, we analyze the dependence of the algorithmic strategies on the hyperparameters ν and α. Recall that a higher ν implies more exploration, and that a higher α leads to stronger updates of the Q-values based on current payoff feedback. Figure 4 illustrates the relationship between these hyperparameters and the most frequently observed strategies. The figure reports pooled means for $\delta \in \{0.90, 0.95\}$ and $R \in \{40, 48\}$.

Figure 4a depicts the impact of ν on the strategies. A first observation is that the prevalence of AllD drops initially in ν but reaches a steady level of around 22%. Second, WSLS increases monotonically in ν and becomes the modal strategy for $\nu \geq 100$. Third, WShLSh is always among the top three strategies in terms of frequency but falls in ν for ν sufficiently high. Lastly, TFT consistently accounts for between 5 and 15% of the data.

Even with high exploration parameters, algorithms cooperate mainly through WSLS and WShLSh. However, the modal strategy becomes WSLS,

which is subgame-perfect in contrast to TFT. One might conclude that algorithms that explore extensively play the repeated prisoner's dilemma in some "better" way than humans. However, WSLS is learned only when the economic incentives of cooperation are high. For example, when $R = 32$, $\delta = 0.95$, and $\nu = 1,000$, 92% of the algorithms play AllD, despite WSLS being an SPNE. Therefore, the algorithm's learned strategies qualitatively depend on the *economic* environments and not only the algorithms' hyperparameters. The strategy choices of humans appear more consistent across treatments.

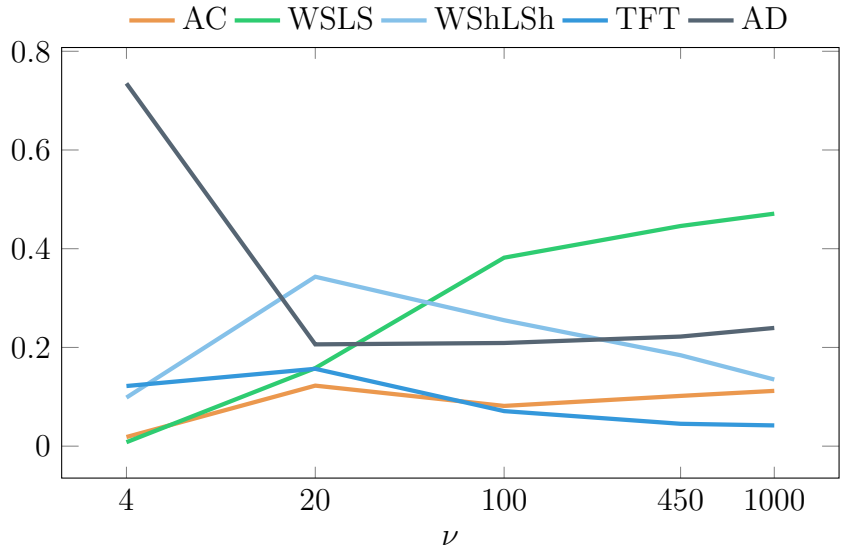
Figure 4b illustrates the influence of α on the algorithms' strategies. A first insight is that for low levels of α, AllD dominates with a frequency of around 70%. However, as soon as α exceeds the lowest level, this frequency drops to around 20%. A second observation is that WSLS largely increases while TFT decreases in α. Altogether, once α is sufficiently high, increasing it further has only a moderate effect on strategies.

In the online appendix, we estimate a multinomial logit regression to systematically investigate the effect of the parameters on the strategies for $k = 1$. The baseline category combines AllD with all other unlisted strategies, and we use the full dataset rather than restricting it to the data depicted in Figure 4. Relative to the baseline, δ and R are positively associated with the cooperative strategies AllC, WSLS, WShLSh, and TFT. The learning parameter α has no statistically significant effect on the strategies. The effect of ν on TFT, WShLSh, and AllC is negative significant. The effect on WSLS is positive but insignificant, due to non-linear interaction with δ and R. We conclude as follows.

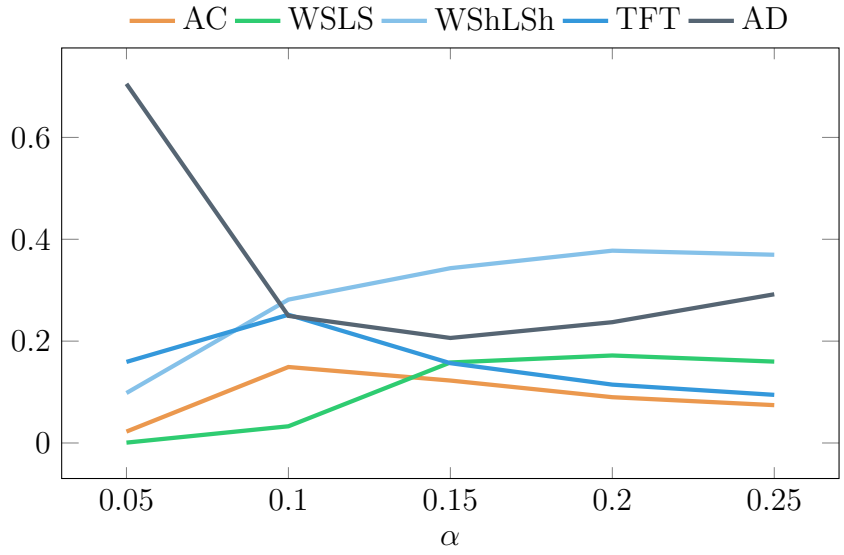
Result 5. *The algorithms' cooperative strategies AllC, WSLS, TFT, and WShLSh increase in δ and R. Exploration increases the share of WSLS and decreases the share of WShLSh. The learning rate α has a muted effect on the strategy choice.*

5.4 Win-shift-lose-shift

The widespread use of certain strategies that do not seem very attractive from a theoretical perspective may be surprising. For instance, an algorithm that plays WShLSh cooperates if and only if both players *defected*



(a) Frequently learned strategies of algorithms as a function of exploration.



(b) Frequently learned strategies of algorithms as a function of the learning rate.

Figure 4: Algorithms' strategies as a function of exploration and the learning rate

Note: Both figures show the averages for $\delta \geq 0.9$ and $R \geq 40$ and $k = 1$. In figure (a), $\alpha = 0.15$. In figure (b), $\nu = 20$. The x-axis in figure (a) is on a log-scale.

in the previous round. When paired with another player who also plays WShLSh, this results in a (CC, DD) cycle, alternating between mutual cooperation and mutual defection. Suppose players are in state DD; why do they still cooperate in the next round? Clearly, they could gain considerably by deviating in the next round, receiving a payoff of 50 instead of R, and also returning to DD again in two rounds.

The intuition behind WShLSh is as follows: Suppose that the Q-matrices of both players are currently such that D is played in all four states in the learning phase. Thus the initially relevant state is DD, with associated Q-values of subsequent cooperation and defection, respectively: $Q(DD, C)$ and $Q(DD, D)$. Since both players defect in state DD, each player continues to receive 25 in that state. Depending on how $Q(DD, C)$ was initialized and the payoff structure, $Q(DD, D)$ may eventually fall below $Q(DD, C)$, in which case the player begins switching to C in state DD. If this switch occurs around the same time for the other player, both players cooperate in state DD and keep getting positive feedback (payoff R) by doing so, which reinforces this action. When exploration eventually stops, both players have 'learned' that cooperation is the optimal action in state DD, resulting in WShLSh. Playing D and C in the DD state updates the Q-values $Q(DD, D)$ and $Q(DD, C)$ towards $50 + \delta Q(DC, D)$ and $R + \delta Q(CC, D)$ (depending on α and δ), respectively. If the latter is higher than the former, then the WShLSh strategy profile survives exploration. This occurs when both δ and R are high, and if $Q(DC, D)$ is sufficiently low relative to $Q(CC, D)$. Such a situation arises if $Q(DC, D)$ reflects a value stemming more from mutual defection rather than from alternating between DD and CC, as $Q(CC, D)$ does. Since there is noise both in the initialization of the Q-matrix and in the learning process, the above argument describes a possibility and not a deterministic convergence result. Bertrand et al. (2023) establish the generic possibility for convergence to cooperative strategies such as WShLSh (*lose-switch* in their terminology) and WSLS (*Pavlov* in their terminology) of Q-learning algorithms in the prisoner's dilemma. Their result also builds on the initialization of the Q-matrix and follows a similar intuition to the one described above.

Since the initial Q-matrix and exploration might explain the occurrence and prevalence of the non-subgame-perfect strategy WShLSh, we

investigate its robustness with respect to these inputs. In the first robustness exercise, Figure S.2 in the online appendix reports the strategies that emerge when the exploration probability is at least 5% in each round. The frequency of WShLSh is lower at the expense of AllC and WSLS. This suggests that exploration plays a role for the prevalence of WShLSh: More exploration implies that the off-path Q-value $Q(DC, D)$ more accurately reflects the continuation payoffs given the opponent's current strategy (derived from their Q-matrix). However, more exploration does not eliminate the strategy. The second robustness check varies the initialization of the Q-matrix. Here we use the same initialization as in Calvano et al. (2020a). That is, the initial Q-values are set to the discounted long-run payoff that would accrue to a player if the opponent were to be randomized uniformly across the two possible actions. In comparison to our main initialization, this leads to higher initial values and thus a more optimistic initial estimate of the value associated with each state-action combination. Additionally, we investigate four more initializations of the Q-matrix, including relatively optimistic and pessimistic ones. Qualitatively, the same set of strategy frequencies appears as with the main initialization (see Section S.4 in the online appendix). We conclude that while exploration contributes to the frequency of WShLSh, the initialization is less crucial.

5.5 Strategies with higher memory

We now briefly describe how the above results change when we allow for higher hard-coded memory. The effect of memory on cooperation is ex ante unclear. Also, the manual classification is no longer practical when $k > 1$. The complexity of the state space, and thus the set of all memory-k strategies, grows exponentially with k. To circumvent this problem, we use the strategy frequency estimation, used for the analysis of human data, and described in the online appendix. In that appendix, we also demonstrate the validity of this method by showing that it yields estimates consistent with the manual classification. In the online appendix, we show two main results when $k > 1$.

Result 6. *With memory two or three, the most frequent strategies learned by the algorithms are AllD and those in the TFT and WShLSh families.*

Notably, grim trigger, one of the most common human strategies, is almost entirely absent. However, this finding is specific to the partner-matching protocol; under stranger matching, grim trigger becomes the dominant cooperative strategy (see Section 7).

Result 7. *Under partner matching, algorithms hardly ever learn grim trigger strategies.*

5.6 Further determinants of strategies

In online appendix S.2, we investigate how the determinants of cooperation influence strategy choice. To this end, we classify the strategies into the categories "cooperative," "lenient," and "forgiving" (Fudenberg et al., 2012). We find that the occurrence of all three classes reacts positively to increases in the discount factor δ and reward parameter R (likely due to moving away from AllD). Memory increases the average cooperation rates mostly through increasing lenient and forgiving strategies. The learning rate α and the exploration parameter ν generally have no significant impact on the strategy class; the only exception is that a higher ν leads to more cooperative strategies.

In online appendix S.2, we also investigate how the economic and algorithmic parameters influence some properties of the limiting strategy profiles. We find that the cycle length increases in δ, R, and memory k. Hence, as average cooperation rates increase (due to higher δ–R), the increase is not due to on-path mutual cooperation but more complex behavior. This claim finds additional support in the finding that the fraction of states on the equilibrium path where both players play the same actions decreases in δ and k.

6 Determinants of algorithmic cooperation

We now investigate more systematically which factors influence algorithms to cooperate. Online appendix S.3 provides a detailed analysis of the determinants of cooperation. We summarize the main findings here. We distinguish between economic parameters and concepts and the algorithm's hyperparameters. Overall, our findings confirm that the determinants sug-

gested by economic theory and experimental economics also apply to Q-learning agents.

The experimental literature has suggested several determinants of cooperation (Dal Bó and Fréchette, 2018). First and foremost, the discount factor δ and the reward parameter R make cooperation more attractive and should therefore raise cooperation. The positive impact of these game parameters on cooperation can be inferred from Figure 2a, as summarized in Result 1. Table 3 reports OLS regressions with the cooperation rate (as in Figure 2a) for our baseline parameterization as the dependent variable. Regression (1) confirms the descriptive results: We see a substantial and highly significant effect of δ and R.

Table 3: Determinants of average cooperation

	(1)	(2)	(3)	(4)
δ	88.14^{***}	88.14^{***}	96.40^{***}	96.40^{***}
	(0.31)	(0.31)	(0.18)	(0.17)
R	2.04^{***}	2.04^{***}	1.95^{***}	1.95^{***}
	(0.01)	(0.01)	(0.00)	(0.00)
α		12.13^{***}		-8.38^{***}
		(0.77)		(0.43)
ν		0.01^{***}		0.01^{***}
		(0.00)		(0.00)
$k = 2$			1.16^{***}	1.16^{***}
			(0.08)	(0.07)
$k = 3$			0.24^{**}	0.24^{**}
			(0.08)	(0.07)
Constant	-130.25^{***}	-134.59^{***}	-133.02^{***}	-135.29^{***}
	(0.42)	(0.43)	(0.24)	(0.25)
Sample	$k = 1$ only	$k = 1$ only	all k	all k
Obs.	300000	300000	900000	900000

An observation is the cooperation rate, computed over 1,000 periods after convergence. The factorial design contains 900 $(k, \delta, R, \alpha, \nu)$ combinations, which take 3, 4, 3, 5, and 5 values, respectively. There are 1,000 simulations for each parameter combination, leading to 300,000 memory-one simulations and 900,000 observations across all memory levels. Standard errors in parentheses.
* $p < 0.05$, ** $p < 0.01$, *** $p < 0.001$

The game-theoretic literature cautions that rational players will cooperate only if cooperative equilibria exist. Hence, if grim trigger forms a subgame perfect Nash equilibrium, then cooperation is expected to be higher. Formally, this requires the discount factor to be sufficiently high, which is the case in all treatments except $\delta = 0.5$ and $R = 32$. Since al-

gorithms do not cooperate in this treatment but there are non-cooperative outcomes in other treatments, we conclude that the existence of SPNE is necessary but not sufficient for cooperation, as is the case for humans.

The literature has also suggested that cooperation rates depend on grim trigger being the risk dominant equilibrium and on the size of the basin of attraction of AllD (sizeBAD). Table S.16 in online appendix S.3 states for which δ–R GT is risk dominant and the respective sizeBAD. Risk dominance is also positively associated with cooperation, while sizeBAD is negatively associated. This finding was also observed by Boczoń et al. (2024a).

Besides these economic parameters, we expect the learning and exploration parameters to affect the cooperation rates of the algorithms. We analyze the role of these parameters in regression (2) of Table 3. Across all parameter specifications in the $k = 1$ sample, the effects of α and ν are positive and significant. In the "all k" sample, ν remains positive whereas α becomes negative significant. While cooperation often increases with α and ν, the effect of these parameters is not uniform for given game parameters. For example, with $k = 1$, $\delta = 0.90$, and $R = 40$, the average cooperation is about 40% for $\nu = 20$, but only about 20% for $\nu = 1,000$. For $\nu = 20$ and the same δ–R pair, average cooperation drops to around 20% as α increases from 0.15 to 0.25. Considering the entire dataset and controlling for α and ν does not change the previous insight that cooperation increases in R and δ. Despite the apparent non-linear interactions of k and the hyperparameters, we conclude with the following result.

Result 8. *Average cooperation across all δ–R parameters of memory-one algorithms increases in α and ν. The impact of α and ν is not uniform and depends on the game parameters.*

The regressions (3) and (4) in Table 3 additionally include memory as a control and use data from all computational experiments, that is, exploiting variation in δ, R, k, α, and ν. The average effect of k in regressions (3) and (4) is positive significant. Table S.3 in the online appendix further distinguishes the cooperation rates by k. There we see that the memory length has an ambiguous influence on cooperation in general. Cooperation rates at $k = 1$ often seem higher than those for $k = 2$ and $k = 3$, but

this is not the case throughout. We attribute this to nonlinear interaction effects between k on the one hand and ν and α on the other. In any case, memory k appears to be a second-order factor. Its effect on cooperation is dominated by the impact of δ and R.

Table 4: Difference human vs. algorithmic cooperation rates, $\nu = 20$ and $\nu = 1000$

$\nu = 20$	k	δ	$R = 32$	$R = 40$	$R = 48$
	1	0.50	8.15***	20.82***	42.38***
		0.75	18.54***	59.28***	37.08***
		0.90	27.45***	40.62	22.86
		0.95	13.85	27.16	27.80*
	2	0.50	8.15***	20.82***	44.85***
		0.75	18.54***	57.65***	44.82***
		0.90	25.58***	38.05**	30.29*
		0.95	18.15	37.22***	39.14***
	3	0.50	8.15***	20.82***	44.13***
		0.75	18.54***	59.12***	46.13***
		0.90	31.88***	58.47***	36.75***
		0.95	21.55**	41.28***	41.71***
$\nu = 1000$	k	δ	$R = 32$	$R = 40$	$R = 48$
	1	0.50	8.15***	20.82***	45.13***
		0.75	18.54***	59.63***	41.15***
		0.90	36.42***	60.23***	7.46
		0.95	44.02***	−2.19***	−3.51***
	2	0.50	8.15***	20.82***	45.13***
		0.75	18.54***	59.63***	65.25***
		0.90	36.42***	62.97***	−10.90***
		0.95	43.60***	−14.04***	−12.16***
	3	0.50	8.15***	20.82***	45.13***
		0.75	18.54***	59.63***	68.15***
		0.90	36.42***	62.23***	−4.82***
		0.95	32.33***	−8.39***	−12.13***

Note: This table depicts the difference between cooperation rates of humans and algorithms, as well as the significance of a two-sided Mann-Whitney-U-test. For the Mann-Whitney-U-test, an observation is one simulation for algorithms (1,000 per each parameterization), and a matching group for human experiments (between 6 and 11 per parameterization).* $p < 0.05$, ** $p < 0.01$, *** $p < 0.001$

Table 4 compares for all $k \in \{1, 2, 3\}$ the cooperation rates of humans and algorithms for $\nu = 20$ and $\nu = 1,000$. Focusing on $\nu = 20$, the

most noticeable observation is that humans cooperate significantly more for nearly all δ–R pairs when $k > 1$; the difference is not significant for four δ–R combinations when $k = 1$. For $\nu = 1,000$, higher memory reinforces the finding that algorithmic cooperation increases in ν (compared to the benchmark $\nu = 20$) for the highest δ–R combinations but decreases in ν for low and intermediate δ and R. It follows that higher memory does not qualitatively change the insights from the memory-one case.

7 Robustness check: Alternative matching protocols

Our main specification follows the standard approach in the reinforcement learning literature: Two algorithms play a single, continuous supergame throughout the training phase. We chose this design to directly compare human learning with how algorithms are typically trained in strategic environments like pricing. Because the algorithms interact continuously, they visit the initial state exactly once. Given that the training phase spans hundreds of thousands of periods, any potential path dependence from this initial state is negligible, making this setup effectively without loss of generality. In human laboratory experiments, however, participants are usually rematched across multiple supergames within a larger matching group. To assess whether our findings are sensitive to these procedural differences, we investigate two alternative matching protocols. With these alternative protocols, we align the algorithmic learning environment more closely with human laboratory setups, even though this diverges from standard algorithmic training practices.

In the first variation, two algorithms repeatedly play supergames with each other (partner rematching). The Q-matrix now includes an explicitly modeled initial state, and the length of each supergame is determined by the discount factor δ, as in human experiments. The significance of this variation lies in the importance of the initial state for a strategy. In expectation, the initial state is visited every $\frac{1}{1-\delta}$ rounds, which can have implications for the strategies that are learned.

In the second variation, we implement stranger rematching with a matching group of ten agents. This mimics the protocol in our human laboratory experiments. After each supergame, agents are randomly rematched within

the group.[19]

Figure 5 presents the cooperation rates for both variations. Panel (a) shows that cooperation rates under partner rematching (group size 2) are similar to our main specification. The non-monotonicity in δ for $R = 48$ persists. Panel (b) shows that stranger rematching with a group size of 10 leads to substantially lower cooperation rates, staying below 10% unless $\delta = 0.95$ and $R \geq 40$. Both findings strengthen our key result that humans tend to cooperate more than algorithms.

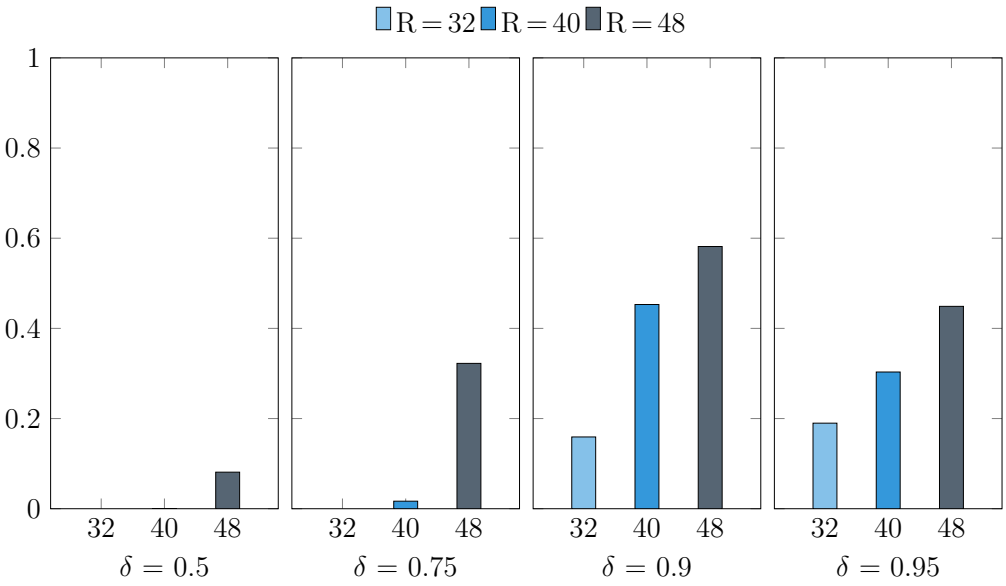
Figure 6 shows the strategy frequencies under both matching variations. Under partner rematching (Panel (a)), the main drivers of cooperation remain WShLSh and WSLS, as in our baseline specification. Thus, outcomes are driven by the same strategies when rematching occurs across supergames within a pair.

Under stranger rematching with groups of 10 (Panel (b)), strategies change substantially. WShLSh disappears entirely. Whenever cooperation arises, it is mostly supported by strategies from the grim trigger family. While grim trigger is also observed among humans, it typically coexists with other cooperative strategies such as tit-for-tat, which algorithms in this setup rarely learn. Thus, as WShLSh and WSLS vanish and grim trigger becomes dominant, algorithmic behavior remains distinctly different from human behavior, though in a different way than under partner matching.

These results carry two implications. First, the gap in cooperation between humans and algorithms is robust. Algorithmic cooperation rates are lower under both alternative matching protocols, reinforcing the finding that humans tend to cooperate more. Second, the specific strategies that sustain algorithmic cooperation are sensitive to the matching environment. Under partner matching, WShLSh and WSLS dominate, while under stranger matching in larger groups, grim trigger becomes the primary cooperative strategy. This qualifies Result 7, which states that algorithms rarely learn grim trigger via partner matching. The broader conclusion that algorithms cooperate differently from humans, however, remains ro-

[19]We focus on the same convergence criteria. However, the larger matching group naturally lengthens the training phase because there is more randomness in the environment as more agents learn. Furthermore, all ten agents must converge to a stable strategy, rather than just two. As such, differences in outcomes may also be attributed to the longer training, a possibility we leave for future research.

(a) Matching group size 2



(b) Matching group size 10

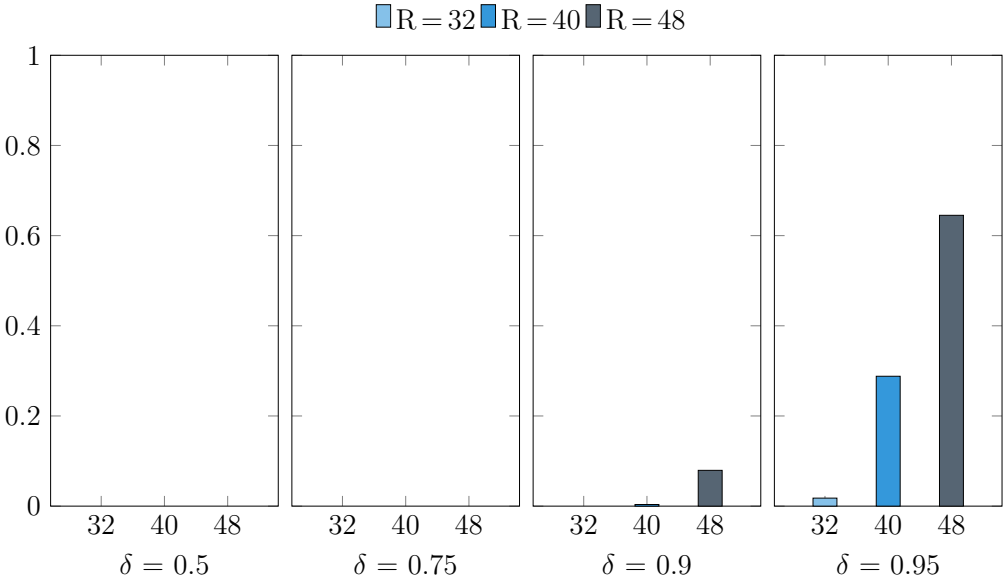


Figure 5: Cooperation rates by δ–R treatment, using different matching schemes across supergames.

Notes: The figure reports the algorithmic cooperation rates for the baseline parameters $k = 1$, $\alpha = 0.15$, $\nu = 20$, computed over 1,000 periods after convergence. An observation is one simulation for algorithms (1,000 for each parameterization).

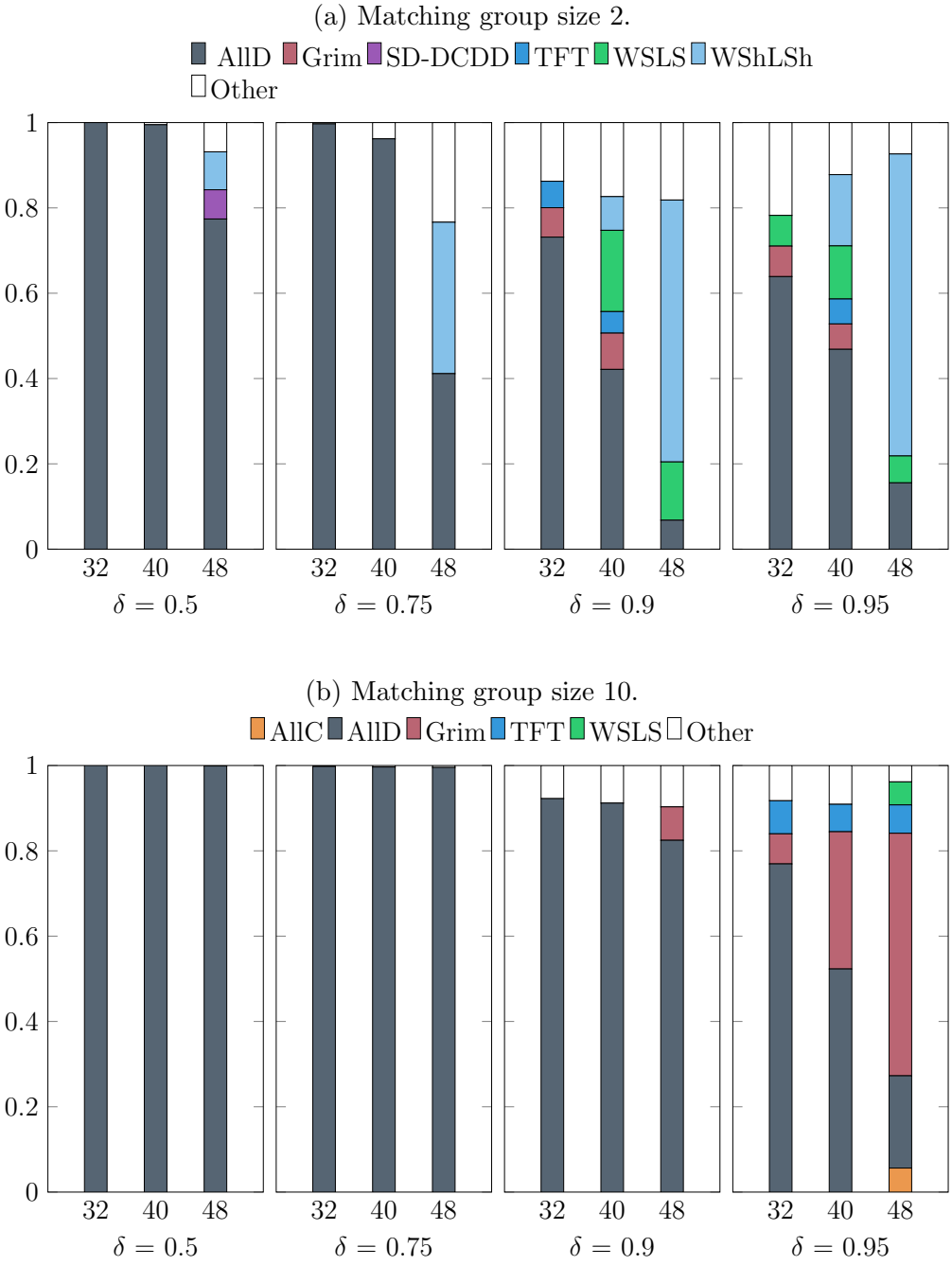


Figure 6: Strategy frequency of algorithmic data by δ-R treatment, using different matching schemes across supergames.

Note: The figure reports the manual classification for $k = 1$ for the baseline parameters $\alpha = 0.15$, $\nu = 20$. Details on the strategy definitions are provided in the online appendix. The additional strategy SD-DCDD begins with defection and otherwise defects, except that it cooperates following a round in which it defected while the opponent cooperated.

bust across all matching protocols, as the strategies algorithms adopt differ from the mix of TFT and GT strategies that characterizes human play.

We note that any matching protocol may be relevant depending on the application. Partner matching within a single supergame is standard practice for studying reinforcement learning algorithms in economics and can approximate the behavior of pricing algorithms that repeatedly interact with one another. Stranger rematching across supergames is the standard in experimental economics with human participants, and it is relevant for settings where agents interact with changing counterparts. Our main specification compares how algorithmic and human learning are typically studied in their respective fields, while recognizing that the matching protocol matters for the strategies that emerge.

8 Conclusion

Comprehensive knowledge of how algorithms work is essential as artificial intelligence is increasingly used in strategic situations (Rahwan et al., 2019). Our work aims to improve the understanding of algorithmic cooperation by comparing the behavior of Q-learning algorithms with that of human participants in the infinitely repeated prisoner's dilemma.

We summarize the established findings of our analysis. First, regarding cooperation levels: humans and algorithms are rational in that their behavior adjusts to the economic incentives. As cooperation becomes more attractive (due to a higher reward or discount factor), average cooperation tends to increase. Humans tend to be more cooperative, in particular when cooperation is relatively hard to sustain. Algorithms achieve higher cooperation rates when they explore extensively and the cooperation reward is very high. Both appear rational in that they cooperate little (or not at all) when it is not incentive compatible to do so. While algorithms can sometimes be tuned to cooperate more than humans, we cannot identify a universal set of parameters that leads to higher cooperation rates across all prisoner's dilemma variants.

Second, regarding strategies: Our Q-learning algorithms tend to play different strategies than humans. Both often choose strategies that are not incentive compatible in the sense of not forming a subgame perfect Nash

equilibrium. Under partner matching, algorithmic cooperation rates are mainly sustained through the cycling Win-Shift-Lose-Shift strategy, while human players frequently use tit-for-tat, grim trigger, or a related strategy. However, Q-learning may appear more rational because it is more likely to forgive past defections by reinitiating cooperation. Under stranger matching with larger groups, the strategies shift substantially: Win-Shift-Lose-Shift disappears and grim trigger becomes dominant among algorithms, while cooperation rates decline further. This sensitivity to the matching protocol indicates that the specific form of algorithmic cooperation depends on design choices that go beyond the game parameters. Overall, the artificial intelligence studied in this paper does not systematically outperform humans.

Several questions remain open. Our analysis establishes that tabular Q-learning under the specifications studied here—including variation in memory, exploration, and learning rates—does not generate cooperation rates comparable to those of humans across all environments. Whether this reflects an intrinsic property of tabular Q-learning or whether alternative algorithmic design choices (e.g., different state representations, learning rules, or function approximation) could close the gap remains to be explored. Investigating the role of exogenous noise in the stability of strategies, as studied theoretically for finite populations by Zhang (2018), is another promising direction: small perturbations could test which of the strategies identified here are robust and which are artifacts of the deterministic limit. Future work could also explore multiplayer settings, richer action spaces, and the interaction between algorithmic and human players.

Q-learning is an important building block of more sophisticated algorithms and widely studied in economics, but there are also other algorithms. As a robustness check on the algorithm's type, a former version of this paper also reported on the play of two Large Language Models (LLMs) in the repeated prisoner's dilemma (Kasberger et al., 2023).[20] We found the cooperation rates *not* to be responsive to changes in the economic environment as formalized by the discount factor δ and the reward parameter R. Hence, not all algorithms respond to the environment as humans and Q-learning algorithms do and as suggested by economic theory.

[20]We analyzed the *gpt-3.5-turbo-0301* model from OpenAI.

Our research can also address questions about collusion between firms. While broadly informative, these results should be read with the caveat that a two-action dilemma may not fully capture the strategic richness of oligopoly environments with larger action spaces. Nevertheless, there are two core policy issues to which our work seems relevant. First, it is essential for antitrust policy to know what market conditions are conducive to collusion by reinforcement learning algorithms. Our results indicate that, when it comes to the determinants of cooperation, there are no major differences compared to human decision-makers. A second important policy question is how to detect collusion by reinforcement learning algorithms (Calvano et al., 2020b). Here the strategy frequency estimation method, developed by Dal Bó and Fréchette (2011) and described in the online appendix, may also enhance our understanding of algorithmic collusion by providing an easy-to-interpret and theory-driven description of the algorithm's strategy. Indeed, we find evidence for retaliation and matching strategies, which are thought to be indicative of collusion in oligopoly.

References

Abada, Ibrahim, Xavier Lambin, and Nikolay Tchakarov, "Collusion by Mistake: Does Algorithmic Sophistication Drive Supra-Competitive Profits?," *Available at SSRN 4099361*, 2022.

Agranov, Marina and Pietro Ortoleva, "Revealed Preferences for Randomization: An Overview," *AEA Papers and Proceedings*, May 2022, *112*, 426–30.

Asker, John, Chaim Fershtman, and Ariel Pakes, "The impact of artificial intelligence design on pricing," *Journal of Economics & Management Strategy*, 2024, *33* (2), 276–304.

Axelrod, Robert, *The Evolution of Cooperation*, Basic Books, 1984.

Banchio, Martino and Giacomo Mantegazza, "Artificial intelligence and spontaneous collusion," *Working paper*, 2022.

Barfuss, Wolfram and Janusz Meylahn, "Intrinsic fluctuations of reinforcement learning promote cooperation," *arXiv preprint arXiv:2209.01013*, 2022.

Bertrand, Quentin, Juan Duque, Emilio Calvano, and Gauthier Gidel, "Q-learners Can Provably Collude in the Iterated Prisoner's Dilemma," *arXiv preprint arXiv:2312.08484*, 2023.

Bigoni, Maria, Marco Casari, Andrzej Skrzypacz, and Giancarlo Spagnolo, "Time Horizon and Cooperation in Continuous Time," *Econometrica*, 2015, *83* (2), 587–616.

Blonski, Matthias, Peter Ockenfels, and Giancarlo Spagnolo, "Equilibrium Selection in the Repeated Prisoner's Dilemma: Axiomatic Approach and Experimental Evidence," *American Economic Journal: Microeconomics*, 2011, *3* (3), 164–192.

Bock, Olaf, Ingmar Baetge, and Andreas Nicklisch, "hroot: Hamburg Registration and Organization Online Tool," *European Economic Review*, 2014, *71*, 117–120.

Boczoń, Marta, Emanuel Vespa, and Alistair Wilson, "Lab to algorithms: Predicting AIs with humans, and vice versa," *Working paper*, 2024.

_ , _ , Taylor Weidman, and Alistair Wilson, "Testing Models of Strategic Uncertainty: Equilibrium Selection in Repeated Games," *Journal of the European Economic Association*, 07 2024.

Calvano, Emilio, Giacomo Calzolari, Vincenzo Denicolò, and Sergio Pastorello, "Artificial Intelligence, Algorithmic Pricing, and Collusion," *American Economic Review*, 2020, *110* (10), 3267–3297.

_ , _ , _ , Joseph E Harrington, and Sergio Pastorello, "Protecting Consumers from Collusive Prices due to AI," *Science*, 2020, *370* (6520), 1040–1042.

Crandall, Jacob W. and Michael A. Goodrich, "Learning to Compete, Coordinate, and Cooperate in Repeated Games Using Reinforcement Learning," *Machine Learning*, 2011, *82*, 281–314.

_ , **Mayada Oudah, Fatimah Ishowo-Oloko, Sherief Abdallah, Jean-François Bonnefon, Manuel Cebrian, Azim Shariff, Michael A. Goodrich, and Iyad Rahwan**, "Cooperating with Machines," *Nature Communications*, 2018, *9* (1), 233.

D'agostino, Ralph B, Albert Belanger, and Ralph B D'Agostino Jr, "A Suggestion for Using Powerful and Informative Tests of Normality," *The American Statistician*, 1990, *44* (4), 316–321.

Dal Bó, Pedro, "Cooperation under the Shadow of the Future: Experimental Evidence from Infinitely Repeated Games," *American Economic Review*, 2005, *95* (5), 1591–1604.

_ **and Guillaume R. Fréchette**, "The Evolution of Cooperation in Infinitely Repeated Games: Experimental Evidence," *American Economic Review*, February 2011, *101* (1), 411–29.

_ **and** _ , "On the Determinants of Cooperation in Infinitely Repeated Games: A Survey," *Journal of Economic Literature*, March 2018, *56* (1), 60–114.

_ **and** _ , "Strategy Choice in the Infinitely Repeated Prisoner's Dilemma," *American Economic Review*, 2019, *109* (11), 3929–52.

Dolgopolov, Arthur, "Reinforcement learning in a prisoner's dilemma," *Games and Economic Behavior*, 2024, *144*, 84–103.

Fudenberg, D., D.G. Rand, and A. Dreber, "Slow to Anger and Fast to Forgive: Cooperation in an Uncertain World," *American Economic Review*, 2012, *102* (2), 720–749.

Ghidoni, Riccardo and Sigrid Suetens, "The Effect of Sequentiality on Cooperation in Repeated Games," *American Economic Journal: Microeconomics*, 2022, *14* (4), 58–77.

Hanaki, Nobuyuki, Rajiv Sethi, Ido Erev, and Alexander Peterhansl, "Learning strategies," *Journal of Economic Behavior & Organization*, 2005, *56* (4), 523–542.

Harrington, Joseph E, "The Effect of Outsourcing Pricing Algorithms on Market Competition," *Management Science*, 2022, *68* (9), 6889–6906.

Hettich, Matthias, "Algorithmic Collusion: Insights from Deep Learning," *Available at SSRN 3785966*, 2021.

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

Johnson, Justin P., Andrew Rhodes, and Matthijs R. Wildenbeest, "Platform Design When Sellers Use Pricing Algorithms," *Econometrica*, 2023, *91* (5), 1841–1879.

Kartal, Melis and Wieland Müller, "A New Approach to the Analysis of Cooperation Under the Shadow of the Future: Theory and Experimental Evidence," *Available at SSRN 3222964*, 2021.

Kasberger, Bernhard, Simon Martin, Hans-Theo Normann, and Tobias Werner, "Algorithmic Cooperation," *Available at SSRN 4389647*, July 2023.

Klein, Timo, "Autonomous Algorithmic Collusion: Q-learning under Sequential Pricing," *The RAND Journal of Economics*, 2021, *52* (3), 538–558.

Normann, Hans-Theo and Martin Sternberg, "Human-algorithm Interaction: Algorithmic Pricing in Hybrid Laboratory Markets," *European Economic Review*, 2023, *152*, 104347.

Nowak, Martin and Karl Sigmund, "A Strategy of Win-stay, Lose-shift that Outperforms Tit-for-tat in the Prisoner's Dilemma Game," *Nature*, 1993, *364* (6432), 56–58.

Rahwan, Iyad, Manuel Cebrian, Nick Obradovich, Josh Bongard, Jean-François Bonnefon, Cynthia Breazeal, Jacob W. Crandall, Nicholas A. Christakis, Iain D. Couzin, Matthew O. Jackson et al., "Machine Behaviour," *Nature*, 2019, *568* (7753), 477–486.

Romero, Julian and Yaroslav Rosokha, "Constructing Strategies in the Indefinitely Repeated Prisoner's Dilemma Game," *European Economic Review*, 2018, *104*, 185–219.

_ **and** _ , "A Model of Adaptive Reinforcement Learning," *Available at SSRN 3350711*, 2019.

_ **and** _ , "Mixed Strategies in the Indefinitely Repeated Prisoner's Dilemma," *Econometrica*, 2023, *91* (6), 2295–2331.

Roth, Alvin E and Ido Erev, "Learning in extensive-form games: Experimental data and simple dynamic models in the intermediate term," *Games and Economic Behavior*, 1995, *8* (1), 164–212.

Roth, Alvin E. and J.Keith Murnighan, "Equilibrium Behavior and Repeated Play of the Prisoner's Dilemma," *Journal of Mathematical Psychology*, 1978, *17* (2), 189–198.

Sandholm, Tuomas W and Robert H Crites, "Multiagent Reinforcement Learning in the Iterated Prisoner's Dilemma," *Biosystems*, 1996, *37* (1-2), 147–166.

Schaefer, Maximilian, "On the Emergence of Cooperation in the Repeated Prisoner's Dilemma," *arXiv preprint arXiv:2211.15331*, 2022.

Waltman, Ludo and Uzay Kaymak, "Q-learning Agents in a Cournot Oligopoly Model," *Journal of Economic Dynamics and Control*, 2008, *32* (10), 3275–3293.

Watkins, Christopher John Cornish Hellaby and Peter Dayan, "Q-Learning," *Machine Learning*, 1992, *8*, 279–292.

Werner, Tobias, "Algorithmic and Human Collusion," *Available at SSRN 3960738*, 2022.

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

Online Appendix for Algorithmic Cooperation: A Comparison with Human Play in the Infinitely Repeated Prisoner's Dilemma

Bernhard Kasberger* Simon Martin† Hans-Theo Normann‡
Tobias Werner§

June 2026

Contents

S.1 Additional tables 3
S.2 Further analyses of strategies 6
S.2.1 Strategy frequency estimation 6
S.2.2 Algorithmic strategies with higher memory 9
S.2.3 Comparison of human and memory-one algorithmic strategies . . . 13
S.2.4 Impact of hyperparameters on strategy choice 14
S.2.5 Algorithmic strategies with memory one 16
S.2.6 Classifying the algorithms' strategies 16
S.2.7 Properties of the limiting strategy profile 18
S.3 Determinants of algorithmic cooperation 20
S.3.1 Hypotheses 20
S.3.2 Results 23

*bernhard.kasberger@jku.at. Department of Economics, Johannes Kepler University Linz and Austrian Institute of Economic Research
†simon.martin@univie.ac.at. Department of Economics, University of Vienna; also affiliated with CEPR and CESifo
‡normann@hhu.de. Düsseldorf Institute for Competition Economics (DICE), HHU
§tobias.felix.werner@soton.ac.uk. Department of Economics, University of Southampton

S.4 Robustness checks 26
S.4.1 More exploration: Lower bound on ε 26
S.4.2 Different initialization of Q-matrix (Calvano et al.) 27
S.4.3 Different initialization of Q-matrix (Pessimistic) 29
S.4.4 Different initialization of Q-matrix (Midpoint) 30
S.4.5 Different initialization of Q-matrix (Random) 31
S.4.6 Different initialization of Q-matrix (Optimistic) 32
S.4.7 Heatmaps 34
S.5 Q-Learning 37
S.6 Human experiment 40
S.7 Instructions in the human-player experiment 42
S.7.1 Original version 42
S.7.2 English translation 44

S.1 Additional tables

Table S.1: List of $k = 1$ strategies for manual classification

ID	Label	DD (00)	DC (01)	CD (10)	CC (11)	Remark
1	AllC	C	C	C	C	
2		C	C	C	D	
3		C	C	D	C	
4	CDAlt	C	C	D	D	
5		C	D	C	C	
6		C	D	C	D	
7	WSLS	C	D	D	C	
8	WShLSh	C	D	D	D	
9		D	C	C	C	Conv. State = (DD)
10		D	C	C	D	
11	TFT	D	C	D	C	
12		D	C	D	D	
13		D	D	C	C	
14		D	D	C	D	
15	Grim	D	D	D	C	Conv. State = (CC)
16	AllD	D	D	D	D	
17	AllD'	D	D	D	C	Conv. State $\neq (CC)$
18	AllC'	D	C	C	C	Conv. State $\neq (DD)$

Note: The table describes the 18 memory-one strategies used in the manual strategy classification. These strategies are the 16 memory-one strategies when ignoring the initial state and two strategies (AllD' and AllC') that take the initial state into account.

Table S.2: Data sources for experiments with human subjects

δ	$R = 32$	$R = 40$	$R = 48$
0.5	DBF2011/19, GS2022 (9)	DBF2011/19, GS2022 (6)	DBF2011/19, GS2022 (11)
0.75	DBF2011/19, GS2022 (11)	DBF2011/19, GS2022 (6)	DBF2011/19, GS2022 (11)
0.90	DBF2019 (7)	KMNW2024 (6)	KMNW2024 (6)
0.95	DBF2019, RR2018 (8)	KMNW2024 (6)	KMNW2024 (6)

Note: This table lists the studies with human subjects whose data we used for our comparisons (DBF2011: Dal Bó and Fréchette (2011), DBF2019: Dal Bó and Fréchette (2019), GS2022: Ghidoni and Suetens (2022), RR2018: Romero and Rosokha (2018), KMNW2024: this paper). In parentheses, we indicate the number of independent observations for each treatment. A single independent observation corresponds to a matching group.

Table S.3: Average cooperation by treatment, $\nu = 20$

k	δ	$R = 32$	$R = 40$	$R = 48$
1	0.50	0.00 (0.00)***	0.00 (0.00)***	2.75 (11.62)***
	0.75	0.00 (0.00)***	0.35 (5.69)***	31.07 (33.10)***
	0.90	8.97 (26.76)***	39.12 (45.27)	66.05 (27.21)
	0.95	37.77 (45.82)	55.35 (46.63)	60.03 (26.42)*
2	0.50	0.00 (0.00)***	0.00 (0.00)***	0.29 (3.22)***
	0.75	0.00 (0.00)***	1.98 (13.45)***	23.33 (34.37)***
	0.90	10.84 (22.54)***	41.69 (38.35)**	58.63 (28.52)*
	0.95	33.47 (25.23)	45.29 (24.06)***	48.70 (18.05)***
3	0.50	0.00 (0.00)***	0.00 (0.00)***	1.01 (9.56)***
	0.75	0.00 (0.00)***	0.51 (6.80)***	22.02 (36.99)***
	0.90	4.54 (13.66)***	21.26 (26.40)***	52.17 (22.21)***
	0.95	30.07 (20.37)**	41.23 (19.43)***	46.13 (14.27)***

Note: This table shows the average cooperation rates by algorithms and the significance of a two-sided Mann-Whitney-U-test relative to human cooperation (see Table S.4). Standard deviations in parentheses. * $p < 0.05$, ** $p < 0.01$, *** $p < 0.001$

Table S.4: Average cooperation by humans

δ	$R = 32$	$R = 40$	$R = 48$
0.50	8.15 (4.76)	20.82 (6.83)	45.13 (18.39)
0.75	18.54 (9.28)	59.63 (15.71)	68.15 (14.41)
0.90	36.42 (14.10)	79.73 (5.78)	88.91 (7.38)
0.95	51.62 (8.51)	82.51 (7.30)	87.84 (13.22)

Note: This table depicts average cooperation rates by humans. Standard deviations in parentheses.

Table S.5: Distribution of strategies used by humans

δ	R	AllC	AllD	TFT	TF2T	TF3T	2TFT	2TF2T	Grim	Grim2	Grim3	σ
0.90	40	2.9	5.0	24.9**	17.9	3.9	6.4	0.2	9.2	6.9	13.8*	0.96
	48	8.1	0.0	31.0**	21.0	0.0	5.6	11.5	0.0	0.0	16.6	0.98
0.95	40	0.0	3.3	37.0***	11.0	10.4	5.8	18.8	3.0	0.0	9.0	0.98
	48	0.0	3.3	23.6	20.8	0.0	0.0	0.0	6.8	13.8	28.3*	0.99

Note: Strategies that never exceed 5% are omitted. * $p < 0.05$, ** $p < 0.01$, *** $p < 0.001$

S.2 Further analyses of strategies

S.2.1 Strategy frequency estimation

Strategy frequency estimation (SFEM) was developed to analyze human decision data by Dal Bó and Fréchette (2011) and has since then been widely used for the estimation of the strategies that humans use in the repeated prisoner's dilemma (see, for instance, Fudenberg et al., 2012, Bigoni et al., 2015, Romero and Rosokha, 2018, Dal Bó and Fréchette, 2019), including the data underlying Figure 3(b) in the main part of the paper.

For a given set of strategies, the SFEM assumes that player i chooses strategy s^l, $l = 1, \ldots, L$, with probability ϕ^l in a given supergame. In each period of the supergame, the player either plays according to strategy s^l, or makes a random mistake. We denote the probability of following the strategy and not making a mistake by $\sigma \in (1/2, 1)$, which is a parameter to be estimated.[1] The probability that a player plays according to the strategy s^l is then given by $P_i(s^l) = \prod_t \sigma^{I_{t,i}}(1-\sigma)^{1-I_{t,i}}$, where $I_{t,i}$ is an indicator variable that is equal to one if the player's action corresponds to the action prescribed by strategy s^l and is zero otherwise. Summing over all players in the game leads to the log-likelihood function $\mathscr{L} = \sum_i \ln(\sum_l \phi^l P_i(s^l))$. We maximize $\mathscr{L}$ to estimate $\{\phi^l\}_{l=1}^L$, the frequency with which the predefined strategies are played in the population.

We include the 20 strategies of Fudenberg et al. (2012) into our set of predefined strategies. These include classic memory-one strategies such as tit-for-tat, grim trigger, win-stay-lose-shift, as well as strategies that require a longer memory length such as lenient grim trigger strategies or win-stay-lose-shift with two punishment periods. As above, we also add win-shift-lose-shift (WShLSh). Our set of strategies consists of 25 strategies, the remaining being a suspicious version of WShLSh with defection in the first round, win-stay-lose-shift with three periods of punishment, and memory two and three versions of WShLSh. In the memory-two version of WShLSh, the player cooperates if and only if both players defected in the previous two periods. The memory-three variant works analogously.

Identification in SFEM relies on the assumption that players make mistakes in the form of the random deviation described by σ.[2] If players do not make mistakes, it is impossible to distinguish between certain strategies. For example, suppose

[1] Allowing for random mistakes is frequently quite successful in rationalizing the behaviour of experimental subjects, see e.g. Kirman et al. (2023).

[2] While model extensions exist (see, for instance, Breitmoser, 2015), SFEM assumes that players can use pure strategies only. Focusing on pure strategies is without loss of generality in our setup, as the algorithm cannot learn mixed strategies.

that one player plays AllC while the other player plays TFT. When matched with each other, the observed actions are observationally equivalent, yet the underlying strategies differ. Upon convergence, however, the algorithms play according to their limit strategy and no longer deviate from this strategy in the form of random errors. To identify ϕ^l, we therefore need to induce random noise into the environment. We start from the convergence state. The same two algorithms that played the training phase now play according to their limit strategy for 50 rounds. In a randomly selected round, one of the algorithms deviates from the action dictated by its limit strategy.[3] To separate strategies off-path, the deviating player deviates in a total of three randomly selected periods.[4] The recorded actions after this deviation create noise in the environment, which allows us to identify the strategies using SFEM. We use this approach for 1,000 independent simulation runs for each environment and algorithmic parameterization. Furthermore, for each simulation run, we induce the random deviation separately, that is, we only consider the actions of the player who did not deviate.

Before estimating the frequency of strategies for algorithms with higher memory, we demonstrate the validity of the SFEM by showing that it yields correct estimates for $k = 1$. As mentioned, the Q-matrix provides the complete mapping from states to actions. Table S.6 shows the differences between the SFEM and the manual classification. We find only few and minor differences, which demonstrates that the SFEM can indeed be applied to algorithmic decision-making in strategic situations.

[3]These random deviations are used for estimating strategies only, but not for computing cooperation rates.

[4]The random round in which the algorithm deviates is drawn from a Poisson distribution with arrival rate $\lambda = 20$. Conditional on the first draw, a second Poisson distribution with $\lambda = 1$ determines the second round in which the player deviates. The third round is again determined by a Poisson draw with $\lambda = 1$.

Table S.6: Manual vs. SFEM

δ	R	Method	AllC	CDAlt	WSLS	WShLSh	TFT	Grim	AllD	Other	σ	LL	MSE
0.50	32	Manual	0.0	0.0	0.0	0.0	0.0	0.0	100.0	0.0			
		SFEM	0.0	0.0	0.0	0.0	0.0	0.0	100.0	0.0	1.00	0.0	0.0
	40	Manual	0.0	0.0	0.0	0.0	0.0	0.0	99.9	0.1			
		SFEM	0.0	0.0	0.0	0.0	0.0	0.0	100.0	0.0	1.00	0.0	0.0
	48	Manual	0.1	0.0	0.0	5.3	0.0	0.0	90.6	4.0			
		SFEM	0.0	0.0	0.0	5.3	0.0	0.0	93.3	1.4	1.00	−0.6	1.7
0.75	32	Manual	0.0	0.0	0.0	0.0	0.0	0.0	100.0	0.0			
		SFEM	0.0	0.0	0.0	0.0	0.0	0.0	100.0	0.0	1.00	0.0	0.0
	40	Manual	0.0	0.0	0.3	0.1	0.0	0.0	98.8	0.8			
		SFEM	0.0	0.0	0.3	0.1	0.0	0.0	99.6	0.0	1.00	−0.1	0.2
	48	Manual	2.2	0.1	4.4	37.8	1.1	0.0	47.2	7.2			
		SFEM	4.4	1.8	4.4	37.8	2.2	0.0	48.6	0.8	0.98	−5.4	6.5
0.90	32	Manual	1.7	1.0	1.2	0.1	5.3	0.0	67.5	23.2			
		SFEM	3.2	1.0	1.2	0.4	3.7	0.0	79.7	10.7	0.99	−3.5	38.8
	40	Manual	6.1	1.0	18.4	7.2	6.8	0.0	43.1	17.4			
		SFEM	8.9	1.5	18.7	7.7	7.7	0.0	47.8	7.7	0.99	−5.2	15.7
	48	Manual	10.6	0.0	15.3	56.2	8.1	0.0	3.3	6.5			
		SFEM	11.9	1.2	15.6	56.2	10.2	0.0	3.8	1.0	0.99	−5.4	4.7
0.95	32	Manual	9.0	2.3	10.3	2.2	17.1	0.0	33.6	25.5			
		SFEM	12.0	1.4	10.4	2.1	14.9	0.0	39.6	19.5	0.99	−4.8	10.9
	40	Manual	13.3	0.4	18.4	8.9	18.5	0.0	23.4	17.1			
		SFEM	18.0	0.9	18.8	9.2	13.8	0.0	27.0	12.4	0.99	−4.9	9.9
	48	Manual	7.1	0.2	9.9	64.2	6.0	0.0	3.1	9.5			
		SFEM	10.3	1.3	10.1	64.2	7.3	0.0	3.8	2.9	0.99	−5.3	7.2

S.2.2 Algorithmic strategies with higher memory

One advantage of Q-learning is that the algorithm's strategy can be inferred directly from the Q-matrix, as we did in the above manual classification. While this is easily done for $k = 1$, the complexity of the state space and hence the set of all memory-k strategies grows exponentially in k. Analyzing and classifying the strategies becomes a daunting task as the number of strategies that differ only in inessential off-path states grows in k. We circumvent the complexity problem by estimating the proportions of the strategies from a fixed set of potential strategies.

Result 6. *With memory two or three, the most frequent strategies learned by the algorithms are AllD and those in the TFT and WShLSh families.*

Table S.7: Strategies used for the estimation

Strategy Family	Included Strategies
Grim Trigger Strategies	Grim, Grim2, Grim3, DGrim2, DGrim3
Tit-for-Tat Strategies	Tit-for-Tat, Suspicious Tit-for-Tat, Tit-for-Two-Tats, Tit-for-Three-Tats, Two-Tits-for-One-Tat, Two-Tits-for-Two-Tats, Suspicious Tit-for-Two-Tats, Suspicious Tit-for-Three-Tats
Win-Stay-Lose-Shift	Win-Stay-Lose-Shift, WSLS (memory 2), WSLS (memory 3)
Win-Shift-Lose-Shift	WShLSh, Suspicious WShLSh, WShLSh (memory 2), WShLSh (memory 3)
Single Strategies	
	Always Cooperate
	Always Defect
	False cooperator (C-to-Always Defect)
	DC Alternator
	T2

Note: This table lists all 25 strategies used in the strategy frequency estimation. Strategies are grouped into families in the paper if they have similar behavioural logic. Ungrouped strategies are listed individually. For full definitions and implementation details, see Fudenberg et al. (2012) and the main text.

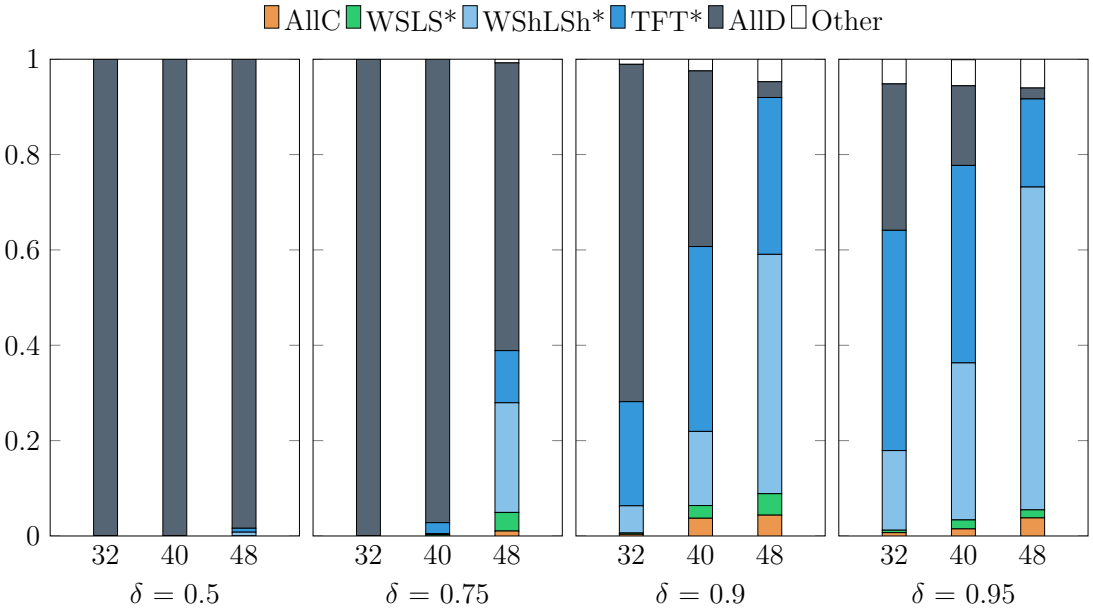
We now ask which strategies the algorithms learn when they have higher memory. Table S.8 reports how the distribution of learned strategies depends on the memory length k. Now that memory-two and -three strategies are feasible for the algorithm, we indeed observe higher-memory strategies. In particular, an increase in k goes along with higher-memory versions of WShLSh, which typically co-exist

with lower-memory versions. For high (δ, R)-pairs, the occurrence of the WShLSh-family is roughly constant in k, but their decomposition changes. Moreover, for $k = 2$, there is now a non-negligible share of TF2T. With $k = 3$, we also observe 2TF2T. Similarly, the memory-one strategy DCAlt becomes relevant. Figure S.1 summarizes the strategy estimation comparable to Figure 3 in the main text. With higher memory, we see that the share of AllC and WSLS decreases while the share of the TFT family increases. To account for higher memory, we collect the strategies in families; e.g., the WSLS family has three members, namely WSLS with memory one, two, and three, respectively. Table S.7 provides the details.

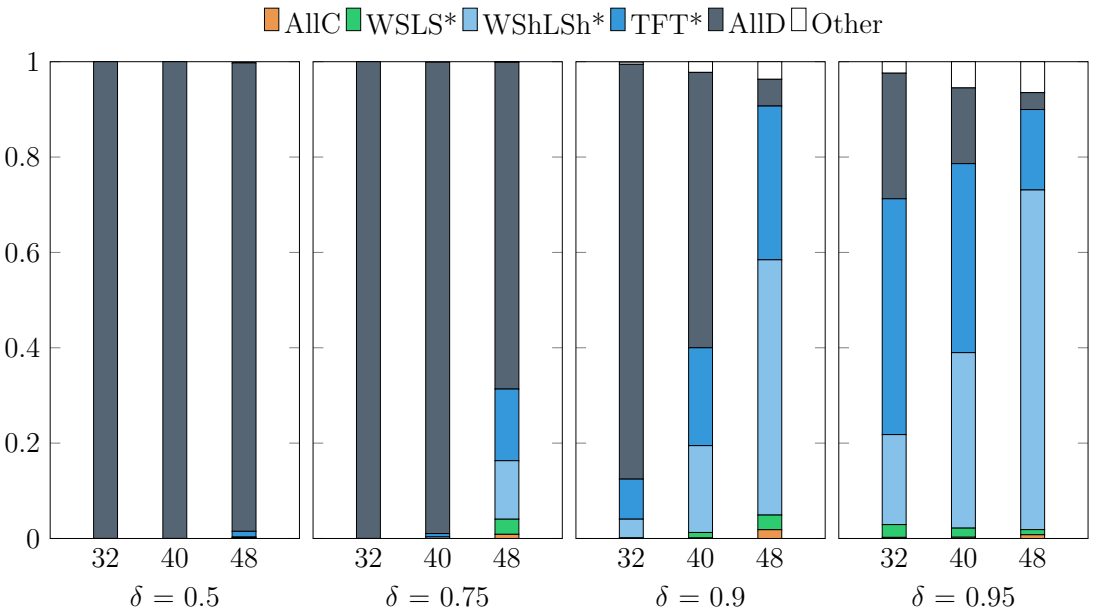
Table S.8 also shows that algorithms hardly ever adopt strategies from the grim trigger family. GT is never played for most parameter combinations. Its highest estimated share is 0.1%.

Result 7. *Under partner matching, algorithms hardly ever learn grim trigger strategies.*

This finding is specific to the partner-matching protocol. As we show in Section 7 in the main text, grim trigger becomes the dominant cooperative strategy when algorithms are rematched across supergames in larger matching groups.



(a) $k = 2$.



(b) $k = 3$.

Figure S.1: Strategy frequency estimation of algorithmic data by δ-R treatment.

Note: The figure reports SFEM results for the baseline parameters $\alpha = 0.15$, $\nu = 20$. The WSLS* family includes WSLS with memory 1, 2, and 3. The WShLSh* family includes WShLSh with memory 1, 2, and 3 and suspicious WShLSh. The TFT* family includes TFT, TF2T, TF3T, 2TFT, 2TF2T, and the suspicious variants. See Table S.7 for details.

Table S.8: Distribution of strategies by k and treatment

k	δ	R	AllC	AllD	TFT	DTFT	TF2T	2TFT	2TF2T	WSLS	DCAlt	WShLSh	2WShLSh	3WShLSh	σ
1	0.50	32	0.0	100.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.00
		40	0.0	100.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.00
		48	0.0	93.3***	0.0	1.3***	0.0	0.0	0.0	0.0	0.0	5.3***	0.0	0.0	1.00
	0.75	32	0.0	100.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.00
		40	0.0	99.6***	0.0	0.0	0.0	0.0	0.0	0.3*	0.0	0.1***	0.0	0.0	1.00
		48	4.4***	48.6***	2.2***	0.8**	0.0	0.0	0.0	4.4***	1.8***	37.8***	0.0	0.0	0.98
	0.90	32	3.2***	79.7***	3.7***	9.9***	0.2	0.0	0.0	1.2***	1.0***	0.4***	0.0	0.0	0.99
		40	8.9***	47.8***	7.7***	7.4***	0.0	0.0	0.0	18.7***	1.5***	7.7***	0.0	0.0	0.99
		48	11.9***	3.8***	10.2***	1.0**	0.0	0.0	0.0	15.6***	1.2***	56.2***	0.0	0.0	0.99
	0.95	32	12.0***	39.6***	14.9***	19.0***	0.0	0.0	0.0	10.4***	1.4***	2.1***	0.0	0.0	0.99
		40	18.0***	27.0***	13.8***	12.2***	0.0	0.0	0.0	18.8***	0.9**	9.2***	0.0	0.0	0.99
		48	10.3***	3.8***	7.3***	2.9***	0.0	0.0	0.0	10.1***	1.3***	64.2***	0.0	0.0	0.99
2	0.50	32	0.0	100.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.00
		40	0.0	100.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.00
		48	0.0	98.4***	0.0	0.8**	0.0	0.0	0.0	0.0	0.0	0.0	0.8**	0.0	1.00
	0.75	32	0.0	99.9***	0.0	0.0	0.0	0.1	0.0	0.0	0.0	0.0	0.0	0.0	1.00
		40	0.0	97.2***	0.9***	0.2	0.2	0.7*	0.0	0.1	0.0	0.2***	0.0	0.0	1.00
		48	1.0**	60.4***	5.4***	1.1**	2.1***	1.0**	0.4	1.1**	0.8**	4.0***	18.4***	0.0	0.97
	0.90	32	0.4*	70.8***	4.2***	9.0***	1.6***	3.3***	2.0***	0.3	1.1**	4.2***	1.4***	0.0	0.95
		40	3.7***	36.8***	15.2***	6.2***	9.1***	4.4***	2.0**	0.8*	2.5***	10.3***	4.0***	0.0	0.91
		48	4.4***	3.3***	15.0***	2.7***	8.9***	3.3***	1.7**	2.1***	4.7***	20.0***	30.0***	0.0	0.92
	0.95	32	0.7**	30.7***	14.1***	17.2***	4.3***	2.2***	4.4***	0.2	5.2***	12.4***	3.5***	0.0	0.89
		40	1.5***	16.7***	14.6***	10.2***	8.4***	1.7***	1.4**	0.8**	5.5***	27.1***	5.2***	0.0	0.85
		48	3.8***	2.3***	6.4***	2.4***	5.8***	1.4**	0.0	0.6*	6.0***	26.3***	41.0***	0.0	0.88
3	0.50	32	0.0	100.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.00
		40	0.0	100.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.00
		48	0.0	98.3***	0.1	0.3	0.0	0.6*	0.0	0.1	0.0	0.0	0.0	0.1	1.00
	0.75	32	0.0	100.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.00
		40	0.0	98.9***	0.0	0.0	0.0	0.5*	0.2	0.0	0.0	0.1***	0.0	0.0	1.00
		48	0.9*	68.5***	3.5***	0.9**	1.1**	5.3***	3.2***	1.0**	0.1	3.4***	2.4***	5.2***	0.96
	0.90	32	0.0	87.0***	2.2***	4.1***	0.1	0.1	1.6***	0.0	0.4*	1.5***	1.6***	0.7**	0.97
		40	0.1	57.8***	7.6***	5.7***	2.0***	2.2***	1.9***	0.5*	2.2***	8.3***	7.1***	2.3***	0.91
		48	1.8**	5.6***	11.7***	4.7***	4.9***	4.3***	3.7***	1.2**	3.7***	24.0***	14.3***	13.5***	0.83
	0.95	32	0.2	26.3***	18.7***	24.2***	0.4*	0.6*	5.2***	1.5***	2.4***	8.3***	7.2***	2.6***	0.92
		40	0.3	15.9***	17.2***	13.3***	0.8*	3.0***	3.0***	0.2	5.5***	20.2***	10.4***	4.8***	0.84
		48	0.8**	3.5***	5.4***	5.6***	2.5***	2.3***	0.4	0.8*	6.5***	36.8***	16.2***	17.0***	0.84

Note: Strategies that never exceed 5% are omitted. * $p < 0.05$, ** $p < 0.01$, *** $p < 0.001$

S.2.3 Comparison of human and memory-one algorithmic strategies

The SFEM was also used to estimate the strategies humans play. Figure 3(b) shows the strategy frequency estimates for the data from our human-subject experiments with $\delta \geq 0.90$ and $R \geq 40$. For other treatments, the figure reports estimates from papers that use different underlying strategy sets, which explains the frequency of "other" strategies.

We now investigate whether the previously identified differences in the strategies used by $k = 1$ algorithms and humans (both based on the SFEM) are statistically significant.[5] To this end, we proceed as follows. First, we estimate the strategy frequency for humans using both the data collected by us and by previous studies, restricting the set of strategies to those that have at least a 0.05 proportion within a given treatment. We then estimate the strategy frequency for the Q-learning algorithms under both the restricted and unrestricted sets of strategies. We store the goodness-of-fit estimates as well as the log-likelihood values. In the third step, we use a log-likelihood ratio test to assess whether restricting the set of strategies to those used by humans results in a loss of information.

The results are shown in Table S.9. For $\delta = 0.5$ and low values of R, we cannot reject the hypothesis that algorithmic strategies lie within the set of human strategies. This makes intuitive sense, as algorithms always play AllD in these treatments, and humans also frequently use AllD for these parameterizations of the game. However, as we increase δ and R, we see that the goodness-of-fit parameter σ is lower in the constrained model compared to the unconstrained model. Moreover, the log-likelihood value is larger for the unconstrained model than for the constrained model (as must be true for nested models), with the likelihood ratio close to zero.

The log-likelihood ratio tests reject the hypothesis that the algorithms' strategies can be fully captured by the set of strategies used by humans, with p-values consistently approaching zero for $\delta \geq 0.75$ and $R \geq 40$. This suggests that restricting the set of strategies to those used by humans results in a loss of information. The strategies that algorithms use are not necessarily well approximated by the set of human strategies for these game parameterizations. This is consistent with the observation that algorithms often use WSLS and WShLSh strategies that humans never use.

[5] We thank the editor for suggesting this test.

Table S.9: Log-likelihood Ratio Test

δ	R	σ		LL		$\frac{L_C}{L_{UC}}$	p-value
		Unconstr.	Constr.	Unconstr.	Constr.		
0.50	32	1.00	1.00	0.0	0.0	1	1.00
0.50	40	1.00	1.00	0.0	0.0	1	1.00
0.50	48	1.00	0.97	-634.8	-6165.3	< 0.001	0.00
0.75	32	1.00	1.00	0.0	0.0	1	1.00
0.75	40	1.00	1.00	-60.5	-324.1	< 0.001	0.00
0.75	48	0.98	0.80	-5354.8	-25685.8	< 0.001	0.00
0.90	32	0.99	0.99	-3527.9	-4157.3	< 0.001	0.00
0.90	40	0.99	0.93	-5233.8	-13082.1	< 0.001	0.00
0.90	48	0.99	0.69	-5437.3	-31322.0	< 0.001	0.00
0.95	32	0.99	0.97	-4821.8	-7834.2	< 0.001	0.00
0.95	40	0.99	0.92	-4947.5	-15283.1	< 0.001	0.00
0.95	48	0.99	0.65	-5301.6	-32688.4	< 0.001	0.00

Note: δ represents the discount factor, and R is the reward parameter. σ is the goodness-of-fit measure from the frequency estimations, with higher values indicating a better fit. LL refers to the log-likelihood of the model, and L_C/L_{UC} is the likelihood ratio comparing the constrained (C) and unconstrained (UC) models. The p-value corresponds to the log-likelihood ratio test, assessing whether restricting the AI strategies to human strategies results in a worse model fit.

S.2.4 Impact of hyperparameters on strategy choice

To further investigate the determinants of strategies played by algorithms, we conduct a multinomial logit regression. We consider the impact of the game and hyperparameters on the (families of) strategies AllC, WSLS, WShLSh, and TFT. The remaining strategies are pooled together as "Other" and used as the baseline group for the multinomial logit. Importantly, AllD belongs to the baseline group. For each parameter configuration, we compute the log odds ratio $\Delta_i = \log(f_i/f_0)$, where f_0 denotes the fraction of strategies classified as "Other", and f_i the fraction of strategies classified as i.

Table S.10 reports the regression estimates for memory-one algorithms and manually classified strategy frequencies; we regress Δ_i on a constant, the game parameters R and δ, and the hyperparameters α and ν, where we log-transform ν to account for the exceptionally large differences between the values of ν. *Ceteris paribus*, all of the non-other strategies are less prevalent than "Other." This is intuitively consistent with algorithms always defecting for these parameter values. In contrast, the prevalence of all the main cooperative strategies increases in R and

δ. The learning rate α has no significant impact on the frequency of any strategy. The exploration parameter ν, however, has a negative impact on all strategies except WSLS. To elaborate on the effect of ν, recall that the cooperative rates depend in a non-linear way on the game parameters (δ, R) and the exploration parameter ν; cooperation increases in ν for sufficiently high δ and R but decreases otherwise. Hence, cooperative strategies such as WSLS decrease in ν when $\delta < 0.9$ or $R < 40$ (at the expense of AllD). Otherwise, WSLS increases in ν. The effect of ν on the other cooperative strategies is negative. The parameter estimate of WSLS is therefore the compound effect (negative for low (δ, R) and positive for high (δ, R)).

Table S.10: Determinants of strategies (manual SFEM, multinomial logit, $k = 1$ only)

	Const.	R	δ	α	$\log(\nu)$
AllC	-27.35*** (1.29)	0.39*** (0.02)	17.38*** (0.90)	0.98 (2.22)	-0.28*** (0.08)
WSLS	-29.34*** (1.29)	0.34*** (0.02)	21.65*** (0.90)	1.18 (2.22)	0.01 (0.08)
WShLSh	-27.31*** (1.29)	0.47*** (0.02)	16.71*** (0.90)	3.40 (2.22)	-0.80*** (0.08)
TFT	-22.34*** (1.29)	0.32*** (0.02)	17.14*** (0.90)	-1.50 (2.22)	-0.63*** (0.08)

To study the entire data and the effect of memory k, we also compute the log odds ratio Δ_i for all numerical treatments and then regress Δ_i on the covariates R, δ, α, $log(\nu)$ and k, and a constant. The results are shown in Table S.11. Again, *ceteris paribus*, all of these strategies are less prevalent than "Other", and also less so as k increases. The negative effect of memory might be due to the number of "Other" strategies growing exponentially in k. In contrast, the prevalence of "Other" decreases in R and δ. The effect of α and ν differs across strategies: WSLS and WShLSh increase in α (again, relative to "Other"), whereas AllC and TFT decrease; however, none of these effects is statistically significant. The effect of ν is negative for WShLSh but positive for the other strategies.

Table S.11: Determinants of strategies (multinomial logit)

	Const.	R	δ	α	$\log(\nu)$	k
AllC	-8.36*** (0.37)	0.07*** (0.01)	2.83*** (0.25)	-0.17 (0.61)	0.06** (0.02)	-0.53*** (0.05)
WSLS	-10.53*** (0.37)	0.09*** (0.01)	4.33*** (0.25)	0.45 (0.61)	0.22*** (0.02)	-0.62*** (0.05)
WShLSh	-10.73*** (0.37)	0.12*** (0.01)	4.70*** (0.25)	0.06 (0.61)	-0.05* (0.02)	-0.41*** (0.05)
TFT	-11.26*** (0.37)	0.08*** (0.01)	5.39*** (0.25)	-0.11 (0.61)	0.11*** (0.02)	-0.08 (0.05)

S.2.5 Algorithmic strategies with memory one

As a robustness check, we now estimate the strategy frequencies within the set of memory-one strategies for memory-one algorithms. An algorithm with memory length one ($k = 1$) cannot learn a strategy that requires higher memory. In our main SFEM specification, this possibility is not ruled out and hence might arise due to noise introduced in the SFEM procedure.

The memory-one strategies we now investigate are the following: AllC, AllD, CtoD, TFT, DFFT, Grim, WSLS, DCAlt, and WShLSh. The results of the adjusted SFEM are shown in Table S.12. Comparing with the baseline results from Table S.8, we find that the SFEM is very robust to our adjustments. Even when strategies with memory two were enabled, SFEM very rarely classified observed strategies as such, and hence omitting them is inconsequential. Our interpretation of these results is that our main specification and the following results are robust to the adjustments made in this robustness check.

Table S.12: Distribution of strategies by experiment for $k = 1$, memory one strategies only.

δ	R	AllC	AllD	TFT	DTFT	WSLS	WShLSh	σ
0.50	32	0.0	100.0	0.0	0.0	0.0	0.0	1.00
	40	0.0	100.0	0.0	0.0	0.0	0.0	1.00
	48	0.1	93.3***	0.0	1.3***	0.0	5.3***	1.00
0.75	32	0.0	100.0	0.0	0.0	0.0	0.0	1.00
	40	0.0	99.6***	0.0	0.0	0.3*	0.1***	1.00
	48	4.4***	48.6***	2.2***	0.8**	4.4***	37.8***	0.98
0.90	32	3.4***	79.7***	3.7***	10.3***	1.2***	0.6***	0.99
	40	9.1***	47.8***	7.7***	7.4***	18.7***	7.8***	0.99
	48	11.9***	3.8***	10.2***	1.0**	15.6***	56.2***	0.99
0.95	32	12.1***	39.6***	14.9***	19.4***	10.4***	2.1***	0.99
	40	18.1***	27.0***	13.8***	12.2***	18.8***	9.1***	0.99
	48	10.3***	3.8***	7.3***	2.9***	10.1***	64.2***	0.99

S.2.6 Classifying the algorithms' strategies

To understand the influence of the different parameters on the adopted strategies, we classify the strategies into the categories proposed by Fudenberg et al. (2012). All strategies except AllD and DTFT are classified as *cooperative*. The set of

lenient strategies includes TF2T, TF3T, 2TF2T, Grim2 and Grim3. The *forgiving* strategies are TFT, TF2T, TF3T, 2TFT, 2TF2T. Note that this classification is not mutually exclusive, e.g., TFT is both *cooperative* and *forgiving*.

In Table S.13, we analyze the incidence of strategy categories for all α-ν combinations. Recall that average cooperation was found to increase in δ and R. All three strategy classes increase in δ and R. This may be due to AllD becoming less likely as these parameters increase. The learning rate α has no significant impact on the strategy class, which may be surprising because the average effect of α on average cooperation is negative.

The analysis reveals interesting differences of the effect of memory length k. Although higher memory length increases average cooperation, we see in Table S.13 that this effect is *not* uniformly driven by cooperative, lenient, and forgiving strategies. Indeed, there is no statistically significant effect of k for cooperative strategies. The effect of k is only statistically significant for lenient and forgiving strategies. Therefore, we conclude that memory increases the average cooperation rates mostly through an increase of lenient and forgiving strategies.

Table S.13: Determinants of strategy categories

	(1) Cooperative	(2) Lenient	(3) Forgiving
δ	119.06***	8.49***	33.07***
	(4.61)	(0.56)	(1.77)
R	2.46***	0.13***	0.37***
	(0.12)	(0.02)	(0.05)
$k = 2$	1.44	3.13***	6.30***
	(1.98)	(0.24)	(0.76)
$k = 3$	2.02	2.01***	6.60***
	(1.98)	(0.24)	(0.76)
α	−10.66	−1.47	−4.89
	(11.41)	(1.39)	(4.37)
ν	0.00*	−0.00	0.00
	(0.00)	(0.00)	(0.00)
Constant	−166.02***	−11.42***	−37.38***
	(6.52)	(0.79)	(2.50)
N	900	900	900

Standard errors in parentheses, * $p < 0.05$, ** $p < 0.01$, *** $p < 0.001$

S.2.7 Properties of the limiting strategy profile

Instead of analyzing limiting strategies in isolation, we investigate in Table S.14 the joint distribution of strategies. Specifically, we investigate the equilibrium cycle length of the strategies at convergence and the fraction of states on the equilibrium path where both players play the same actions. We do this both for our main (α, ν) specification (columns (1) and (3)) and for all specifications (columns (2) and (4)).

We find that δ, R, and the memory length k not only positively influence average cooperation rates but also result in strategy profiles that involve more nodes on the equilibrium path (columns (1) and (2)). Thus, higher cooperation rates are not driven by increased usage of simple strategies such as AllC, but rather by more involved strategies like WShLSh that are more cooperative on average. The influence of the learning and exploration parameters α and ν is negative—more exploration reduces the cycle length. For algorithms with higher memory length k, it is more likely that algorithms converge to a state that involves at least partial cooperation, increasing the average cooperation.

Table S.14: Cycle length and same actions

	(1) Cycle length	(2) Cycle length	(3) Frac. same a	(4) Frac. same a
δ	2.88^{***}	1.24^{***}	-48.38^{***}	-21.86^{***}
	(0.03)	(0.00)	(0.65)	(0.10)
R	0.04^{***}	0.02^{***}	0.20^{***}	0.13^{***}
	(0.00)	(0.00)	(0.02)	(0.00)
$k = 2$	0.51^{***}	0.13^{***}	-9.31^{***}	-2.16^{***}
	(0.01)	(0.00)	(0.28)	(0.04)
$k = 3$	0.76^{***}	0.31^{***}	-12.03^{***}	-4.67^{***}
	(0.01)	(0.00)	(0.28)	(0.04)
α		-0.49^{***}		14.67^{***}
		(0.01)		(0.24)
ν		-0.00^{***}		0.01^{***}
		(0.00)		(0.00)
Constant	-2.47^{***}	-0.32^{***}	126.83^{***}	105.88^{***}
	(0.04)	(0.01)	(0.88)	(0.14)
Mean	1.62	1.28	90.36	95.67
Subsample	($\alpha = 0.15$, $\nu = 20$)	All	($\alpha = 0.15$, $\nu = 20$)	All
N	36000	899975	36000	899975

Standard errors in parentheses, * $p < 0.05$, ** $p < 0.01$, *** $p < 0.001$

This claim finds additional support when we examine the fraction of states on the equilibrium path where both players play the same actions (columns (3) and (4)). This fraction is negatively affected by δ and k. This again implies that higher

cooperation rates are driven by a higher propensity for players to play opposing actions in certain states.

This idea only holds in the non-stationary setting of competing against another player whose action may also change over time, i.e., before convergence. This could not happen in a stationary setting where a Q-learning agent always learns the best response. Thus, there is an interesting analogy to the "meeting of minds" concept of collusion in competition policy. A Q-learning agent in isolation behaves individually optimal. When paired with one another, the outcome may closely resemble the coordinated outcome, despite the lack of explicit communication.[6]

[6] A similar argument also explains why mutual cooperation can arise in memoryless strategies, despite the fact that cooperation is clearly dominated.

Table S.15: The prisoner's dilemma with normalized payoffs

	C	D
C	$1, 1$	$-\ell,\ 1+g$
D	$1+g,\ -\ell$	$0,\ 0$

S.3 Determinants of algorithmic cooperation

We draw upon the experimental literature to form our hypotheses about the determinants of algorithmic cooperation. The experimental literature has identified several factors that shape human cooperation (Dal Bó and Fréchette, 2018, Embrey et al., 2018, Mengel, 2018). We consider four factors where we conjecture that these are also relevant for algorithmic cooperation.[7] Except for the first hypothesis, which is also stated in the main text, we develop the theoretical predictions with normalized payoffs where mutual cooperation leads to a payoff of 1 and mutual defection to a payoff of 0. In Table S.15, g then stands for the payoff the player gains when defecting (instead of cooperating) while the other player cooperates, and ℓ represents the payoff loss when cooperating (instead of defecting) while the other player defects. Naturally, both g and ℓ are positive.

S.3.1 Hypotheses

The experimental literature has shown that cooperation among humans can be expected to increase in the discount factor and the reward parameter (Dal Bó and Fréchette, 2018). A higher discount factor δ increases the probability of future interactions and makes cooperation more attractive compared to short-run gains from defection. A larger reward payoff generally makes cooperation more attractive. Our first hypothesis is stated in the main text: We expect cooperation among self-learning algorithms to increase in R and δ.

Second, cooperation rates tend to be higher in experiments with humans when cooperation can be supported in an SPNE (Dal Bó and Fréchette, 2011, 2018). The condition is formalized through a binary variable that takes the value 1 when the payoff parameters (δ, g, ℓ) are such that GT forms an SPNE and 0 otherwise.

[7]There are also other factors that influence human cooperation rates. For example, in lab experiments, an important determinant of average cooperation is the level of cooperation in period one (Breitmoser, 2015, Dal Bó and Fréchette, 2018). Whereas this allows for a parsimonious restriction of the analysis to the first period, there is no comparable counterpart in self-learning algorithms.

Formally (GT, GT) is an SPNE if

$$1+\delta+\delta^2+\delta^3+\ldots \geq 1+g+\delta\cdot 0+\delta^2\cdot 0+\delta^3\cdot 0+\ldots \Leftrightarrow \delta \geq \frac{g}{1+g} \equiv \delta^{\text{SPNE}},$$

that is, if the discount factor is above the critical value δ^{SPNE}. The mere fact that cooperation is part of an equilibrium does not guarantee cooperation in lab experiments; the discount factor being sufficiently large is more of a necessary condition for cooperation than a sufficient one (Dal Bó and Fréchette, 2018). We conjecture that this also holds for algorithms.

Hypothesis 2. *A necessary but not sufficient condition for self-learning algorithms with $k > 0$ to cooperate is that grim trigger forms an SPNE.*

Hypothesis 2 does not claim that Q-learning results in strategies that are always subgame perfect. Moreover, we know that Q-learning can lead to cooperative outcomes even in the absence of memory (Asker et al., 2024, Dolgopolov, 2024, Banchio and Mantegazza, 2022), so subgame perfection cannot play a role in that case. We hypothesize that a necessary condition for cooperation to emerge with $k > 0$ is that the discount factor is high enough for the grim trigger strategy to be subgame perfect.

The third determinant of cooperation in lab experiments is the size of the basin of attraction of always defect, "sizeBAD" (Dal Bó and Fréchette, 2011, 2018), which is a measure for how robust cooperation is to strategic uncertainty. To define the basin of attraction, consider a hypothetical coordination game in which the players choose between the repeated-game strategies GT and AllD. In this game, the players believe that the opponent plays GT with probability p and AllD with probability $1-p$. The basin of attraction of AllD is then defined as the maximum p that makes AllD the best response. We use $\underline{p}$ to denote sizeBAD. To find the formula for $\underline{p}$, compare the expected payoff from playing GT, $p/(1-\delta) + (1-p)\cdot(-\ell)$, to the expected payoff from AllD, $p\cdot(1+g)+(1-p)\cdot 0$. If GT forms an SPNE, the expected payoff of GT is (weakly) larger than that of AllD if and only if

$$p \geq \frac{(1-\delta)\ell}{1-(1-\delta)(1+g-\ell)} \equiv \underline{p}. \tag{1}$$

However, if GT does not form an SPNE, set $\underline{p}$ equal to 1.

Hypothesis 3. *Algorithmic cooperation decreases in sizeBAD.*

A related fourth determinant of cooperation is Risk Dominance (Blonski et al., 2011, Blonski and Spagnolo, 2015). Specifically, cooperation is found to be higher

in the infinitely repeated prisoner's dilemma if in the hypothetical coordination game consisting of AllD and GT, the cooperative GT equilibrium is risk dominant (RD). This is the case if the discount factor is sufficiently high. To find the minimum discount factor for risk dominance, assume that both strategies are equally likely and substitute $p = 1/2$ in equation 1 (Harsanyi and Selten, 1988). This leads to the critical discount factor

$$\delta \geq \frac{g+\ell}{1+g+\ell} \equiv \delta^{\text{RD}},$$

as in Blonski et al. (2011, Proposition 2, page 175).

Hypothesis 4. *Algorithmic cooperation is higher when cooperation is risk dominant, i.e., when* $\delta \geq \delta^{RD}$.

Table S.16 summarizes the first two dimensions of our treatments and provides the theoretical predictions. The table entry for each variant shows whether GT can be supported as SPNE, and whether the specification satisfies the Risk Dominance (RD) criterion. For GT, the thresholds for δ are 0.72, 0.40 and 0.08 for $R = 32$, 40, and 48, respectively. For RD, the thresholds δ^{RD} are 0.82, 0.61 and 0.39 for $R = 32$, 40, and 48, respectively. As seen above, these are potentially important determinants of cooperation. The table also reports the size of the basin of attraction of AllD.

Table S.16: Experiments

	$R = 32$	$R = 40$	$R = 48$
$\delta = 0.50$	No criterion met	GT	GT, RD
	$\underline{p} = 1.000$	$\underline{p} = 0.722$	$\underline{p} = 0.383$
$\delta = 0.75$	GT	GT, RD	GT, RD
	$\underline{p} = 0.813$	$\underline{p} = 0.271$	$\underline{p} = 0.163$
$\delta = 0.90$	GT, RD	GT, RD	GT, RD
	$\underline{p} = 0.224$	$\underline{p} = 0.094$	$\underline{p} = 0.060$
$\delta = 0.95$	GT, RD	GT, RD	GT, RD
	$\underline{p} = 0.102$	$\underline{p} = 0.045$	$\underline{p} = 0.029$

There are also hypotheses that relate to the specific Q-learning algorithms and that have no human counterpart. Based on Calvano et al. (2020, Figure 1), we conjecture that cooperation decreases in α and β. As ν decreases in β, we expect cooperation to increase in ν.

Hypothesis 5. *The level of cooperation among self-learning algorithms decreases in α and increases in ν.*

S.3.2 Results

We now ask how cooperation rates are affected when cooperative equilibria exist. We formalize this using a binary variable that takes the value of one if the discount factor δ exceeds δ^{SPNE}, and zero otherwise. Table S.16 shows for which treatments this condition is met. From the cooperation rates in Figure 2a, we note three points regarding δ^{SPNE}. First, there is no cooperation in treatment ($\delta = 0.5$, $R = 32$) where GT is not an SPNE. Second, in all treatments with significant levels of cooperation, GT does form an SPNE. Third, the fact that GT is an equilibrium is not sufficient for cooperation. Indeed, for $\delta = 0.5$ and $R = 40$ there is virtually no cooperation, and there is very little cooperation in ($\delta = 0.5$, $R = 48$) and ($\delta = 0.75$, $R = 32$). This is despite GT being an equilibrium in these cases. We conclude with the following statement, which also has a human counterpart.

Result 9. *A necessary but not sufficient condition for self-learning algorithms to cooperate is that grim trigger forms an SPNE.*

Table S.17: Determinants of average cooperation, $\alpha = 0.15$, $\nu = 20$

	(1)	(2)	(3)	(4)
δ	94.76***			
	(0.77)			
R	1.49***			
	(0.02)			
$k = 2$	−3.10***	−3.10***	−3.10***	−3.10***
	(0.33)	(0.38)	(0.35)	(0.33)
$k = 3$	−6.88***	−6.88***	−6.88***	−6.88***
	(0.33)	(0.38)	(0.35)	(0.33)
GT		10.47***		
		(0.61)		
RD		20.91***		
		(0.34)		
$\underline{p}$			−52.35***	
			(0.45)	
$\delta - \delta^{RD}$				76.07***
				(0.56)
Constant	−108.01***	3.33***	42.15***	12.21***
	(1.04)	(0.59)	(0.29)	(0.25)
N	36000	36000	36000	36000

Standard errors in parentheses, * $p < 0.05$, ** $p < 0.01$, *** $p < 0.001$

The next potential determinant of cooperation is risk dominance (Blonski et al., 2011, Blonski and Spagnolo, 2015). We expect cooperation to be higher when $\delta \geq \delta^{\text{RD}}$. For example, Table S.16 shows that for $\delta = 0.5$, GT is risk dominant only when $R = 48$. Looking at Figure 2a and $\delta = 0.5$, while cooperation does indeed increase as R increases from 40 to 48, the gain in cooperation is very modest (from 0 to 2.75%). Nevertheless, Figure 2a suggests a positive influence of the RD criterion on cooperation.

To systematically examine the influence of RD and GT on cooperation, we drop δ and R as regressors and instead analyze whether a treatment satisfied the condition for GT or RD in regression (2) of Table S.17. We find that cooperation is indeed higher when there are cooperative equilibria and, in addition, when the equilibrium is risk dominant. We take this as evidence in favor of Hypothesis 4, where an analogous statement also holds for human players. In regression (4) in Table S.17, we also find that cooperation increases in $\delta - \delta^{RD}$, which is an intuitive measure of how risk dominant cooperation is.

Result 10. *Algorithms cooperate more when cooperation is risk dominant.*

The final determinant of cooperation that is motivated by human cooperation is the size of the basin of attraction of always defect. A smaller basin of attraction can be interpreted in the sense that cooperative strategies are more robust to the uncertainty surrounding the other player's strategy (Dal Bó and Fréchette, 2011). We investigate the role of $\underline{p}$ in regression (3) in Table S.17. The sign of the estimated coefficient is negative, as expected by Hypothesis 3.

Result 11. *Algorithmic cooperation decreases in sizeBAD.*

In addition to the factors that determine human cooperation, we expect the learning parameters to affect the cooperation rates of the algorithms. In the rest of this section, we examine all data, not just the baseline parameters with $\alpha = 0.15$ and $\nu = 20$. We analyze the role of the learning parameters in Table S.18, where we report the same set of regressions as in Table S.17 but now for all data and with the additional controls α and ν.

Across all parameter specifications, the effect of α is negative and highly significant, whereas the effect of ν is positive and significant. However, while the average cooperation decreases in α and increases in ν, the impact of these learning parameters is ambiguous for given game parameters. For example, with memory one, $\delta = 0.90$ and $R = 40$, average cooperation is around 40% for $\nu = 20$ but only around 20% for $\nu = 1000$. For $\nu = 20$, and the same δ–R pair, average cooperation

Table S.18: Determinants of average cooperation, all (α, ν)

	(1)	(2)	(3)	(4)
δ	96.40***			
	(0.17)			
R	1.95***			
	(0.00)			
$k = 2$	1.16***	1.16***	1.16***	1.16***
	(0.07)	(0.09)	(0.08)	(0.08)
$k = 3$	0.24**	0.24**	0.24**	0.24**
	(0.07)	(0.09)	(0.08)	(0.08)
α	−8.38***	−8.38***	−8.38***	−8.38***
	(0.43)	(0.49)	(0.47)	(0.43)
ν	0.01***	0.01***	0.01***	0.01***
	(0.00)	(0.00)	(0.00)	(0.00)
GT		4.58***		
		(0.13)		
RD		27.46***		
		(0.08)		
$\underline{p}$			−51.96***	
			(0.10)	
$\delta - \delta^{RD}$				85.48***
				(0.13)
Constant	−135.29***	−2.73***	34.38***	2.97***
	(0.25)	(0.15)	(0.10)	(0.09)
N	900000	900000	900000	900000

Standard errors in parentheses, * $p < 0.05$, ** $p < 0.01$, *** $p < 0.001$

drops to around 16% as α is decreased from 0.15 to 0.05. Thus, there is no clear support for Hypothesis 5.

Result 12. *Average cooperation across all δ-R-k parameters decreases in α and increases in ν. For a given game, the impact of α and ν is ambiguous.*

Looking at the entire data and controlling for α and ν does not change most of the previous insights. Cooperation still increases in R and δ, is higher when GT is an SPNE, and risk dominance and sizeBAD have the expected influence on cooperation. The only exception is the influence of memory on cooperation. Here, for the general set of α and ν, the average effect of k is positive significant.

S.4 Robustness checks

S.4.1 More exploration: Lower bound on ε

As a robustness check, we consider an alternative specification in which the exponential decay of ε_t is bounded from below, specifically at $\varepsilon_t \geq 0.05$. This ensures that throughout the process, a minimal amount of exploration remains, arguably more closely resembling the behavior of humans. The lower bound on exploration renders the algorithms' convergence, as defined earlier, as very unlikely. Therefore, the "learning period" of the algorithms runs until the maximum number of iterations is reached. Otherwise, we use the same baseline parameters. Note that for $k = 1$ shown in the figures and tables below, using $\beta = 2.31 \times 10^{-5}$ implies $\nu = 20$ in the baseline.

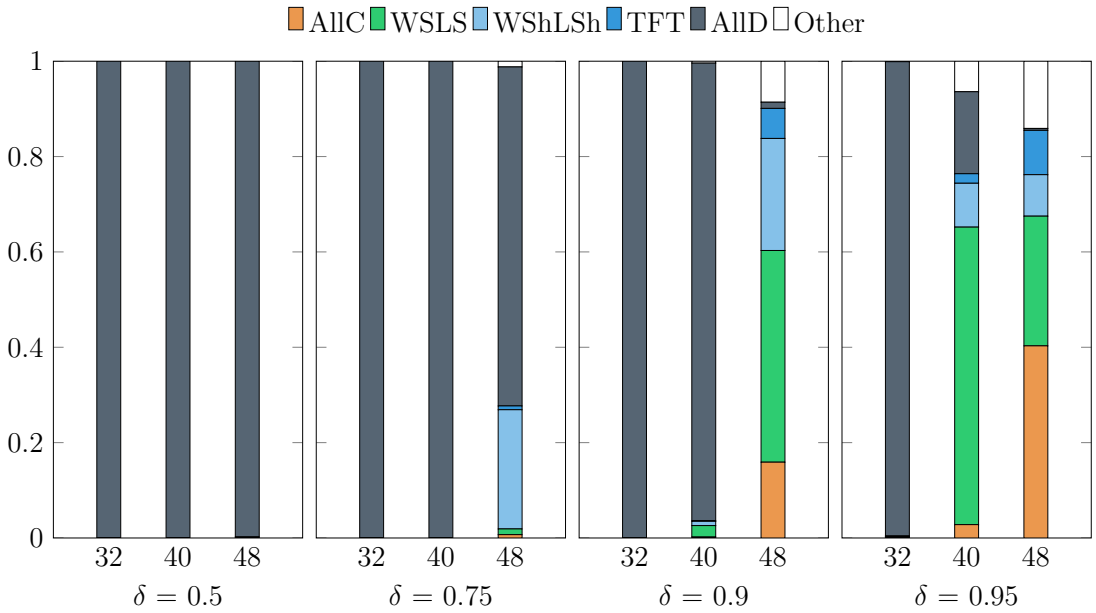
Table S.19: ε lower bound: Average cooperation by treatment, $\beta = 2.31 \times 10^{-5}$

k	δ	$R = 32$	$R = 40$	$R = 48$
1	0.50	0.00^{***}	0.00^{***}	0.10^{***}
	0.75	0.00^{***}	0.00^{***}	15.63^{***}
	0.90	0.00^{***}	3.28^{***}	84.29^{*}
	0.95	0.40^{***}	74.13	91.96^{***}

Note: This table depicts average cooperation rates by algorithms, as well as the significance of a two-sided Mann-Whitney-U-test relative to human cooperation (see Table S.4). * $p < 0.05$, ** $p < 0.01$, *** $p < 0.001$

Table S.19 reports the average cooperation rates in the playing stage. The cooperation rates are consistent with the finding that a higher ν (which also implies more exploration) *reduces* average cooperation unless both δ and R are very high. For these parameter values in the bottom-right corner ($R = 48$, $\delta = 0.95$), cooperation is *higher* than without the lower bound. The table also shows that the cooperation rates differ significantly from those of humans. Humans cooperate more unless δ and R are very high.

Figure S.2 depicts the strategy choice of the algorithms, using the manual classification. Consistent with the lower cooperation rates for relatively low values of δ and R, AllD is the most dominant strategy across the treatments. However, WSLS and AllC become more popular for the highest value combinations of these game parameters. The rise of AllC may seem surprising, because AllC does not



Note: The figure reports the manual classification for $k = 1$ for the baseline parameters $\alpha = 0.15$, $\beta = 2.31 \times 10^{-5}$.

Figure S.2: Lower bound on ε: Strategy choices of algorithms

form an SPNE. WSLS and AllC occur at the expense of WShLSh and TFT. WShLSh does not fully disappear. For cooperative δ-R combinations, it is still one of the four most frequent strategies.

S.4.2 Different initialization of Q-matrix (Calvano et al.)

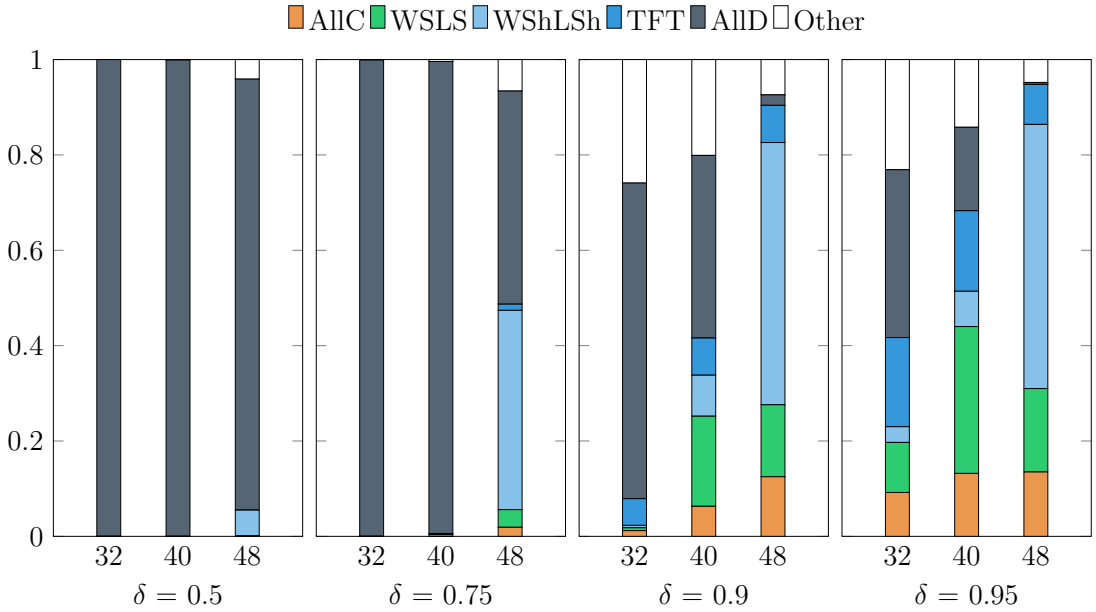
As a second robustness check, we run a separate set of simulations in which we initialize the Q-matrix as in Calvano et al. (2020) to investigate its influence on cooperation rates and strategy choice. The initial Q-matrix is based on the future discounted rewards that a player would expect to earn if their opponent just chose actions randomly. Compared to our main initialization method, this yields higher initial Q-values. We keep $\alpha = 0.15$ and $\nu = 20$ but vary memory k.

Table S.20: Different initial Q-matrix: Average cooperation by treatment, $\nu = 20$

k	δ	$R = 32$	$R = 40$	$R = 48$
1	0.50	0.00***	0.00***	2.80***
	0.75	0.00***	0.50***	31.47***
	0.90	7.97***	41.83	68.30
	0.95	37.20	66.45	70.45

Note: This table depicts average cooperation rates by algorithms, as well as the significance of a two-sided Mann-Whitney-U-test relative to human cooperation (see Table S.4). * $p < 0.05$, ** $p < 0.01$, *** $p < 0.001$

Table S.20 shows the average cooperation rates for the different initialization of the Q-matrix. The main comparison between humans and algorithms also holds under the alternative initial Q-matrix: Humans are more cooperative when cooperation is relatively hard to sustain (low δ and R). However, the levels in Table S.20 are somewhat higher than under the baseline initialization in Table S.3.



Note: The figure reports the manual classification for $k = 1$ for the baseline parameters $\alpha = 0.15$, $\nu = 20$.

Figure S.3: Different Q-matrix initialization (Calvano et al.): Strategy choices of algorithms

Figure S.3 illustrates for the different initialization the most frequently learned strategies when $k = 1$. The strategies are manually classified. The picture is

qualitatively similar to the corresponding figure in the main text. The algorithms largely adopt the same types of strategies for both initializations.

S.4.3 Different initialization of Q-matrix (Pessimistic)

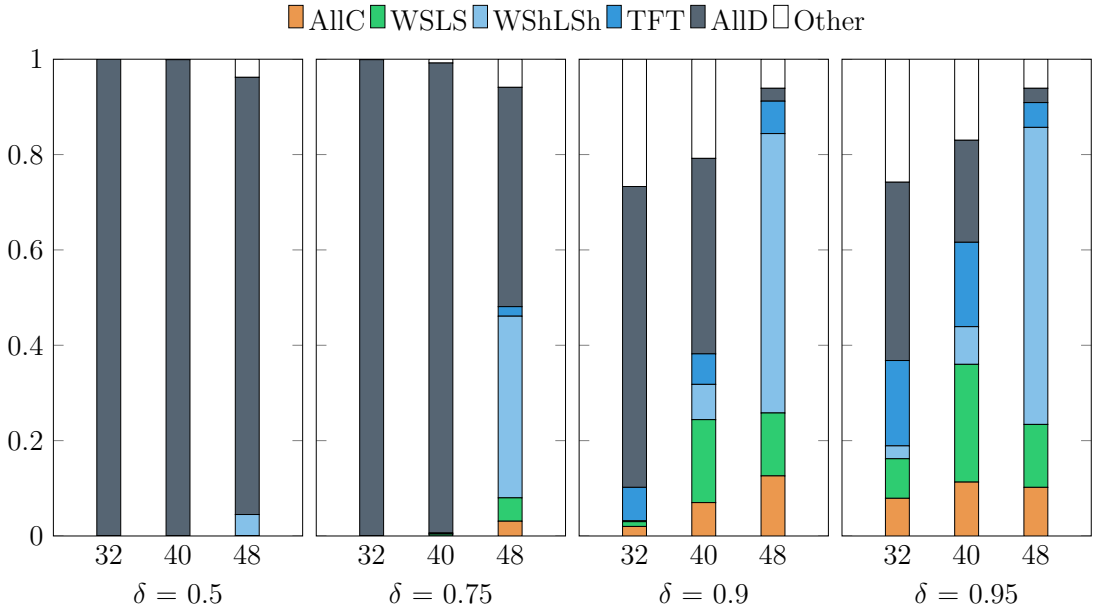
As a robustness check, we repeated our simulation experiments with a "pessimistic" initialization of the Q-matrix, specifically with $12/(1-\delta)$ in all cells. This initialization corresponds to the net present value of playing C forever, while your opponent plays D forever, resulting in the worst possible payoff of 12 in every period.

The results are shown in Table S.21 and Figure S.4. As these results show, our main insights are robust to these changes in the initialization of the Q-matrix.

Table S.21: Average cooperation by treatment, $\nu = 20$

k	δ	$R = 32$	$R = 40$	$R = 48$
1	0.50	0.00^{***}	0.00^{***}	2.30^{***}
	0.75	0.00^{***}	0.50^{***}	32.62^{***}
	0.90	10.46^{***}	39.85	66.15
	0.95	34.52	58.71	63.05^{*}

Note: This table depicts average cooperation rates by algorithms, as well as the significance of a two-sided Mann-Whitney-U-test relative to human cooperation (see Table S.4). * $p < 0.05$, ** $p < 0.01$, *** $p < 0.001$



Note: The figure reports the manual classification for $k = 1$ for the baseline parameters $\alpha = 0.15$, $\nu = 20$.

Figure S.4: Different Q-matrix initialization (Pessimistic): Strategy choices of algorithms

S.4.4 Different initialization of Q-matrix (Midpoint)

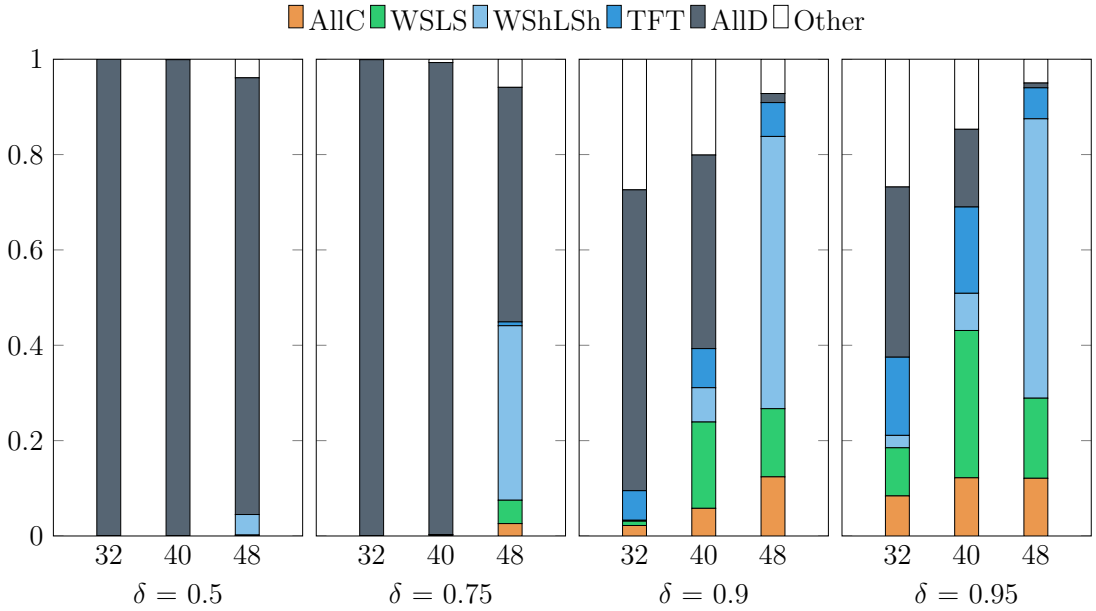
As a robustness check, we repeated our simulation experiments with a "midpoint" initialization of the Q-matrix, specifically with $(12 + 50)/(2(1 - \delta))$ in all cells. This initialization corresponds to the net present value of a constant payoff equal to 31; the average of (C, D) and (D, C).

The results are shown in Table S.22 and Figure S.5. As these results show, our main insights are robust to these changes in the initialization of the Q-matrix.

Table S.22: Average cooperation by treatment, $\nu = 20$

k	δ	$R = 32$	$R = 40$	$R = 48$
1	0.50	0.00***	0.00***	2.35***
	0.75	0.00***	0.25***	30.12***
	0.90	10.17***	41.28	67.27
	0.95	37.02	67.35	67.80

Note: This table depicts average cooperation rates by algorithms, as well as the significance of a two-sided Mann-Whitney-U-test relative to human cooperation (see Table S.4). * $p < 0.05$, ** $p < 0.01$, *** $p < 0.001$



Note: The figure reports the manual classification for $k = 1$ for the baseline parameters $\alpha = 0.15$, $\nu = 20$.

Figure S.5: Different Q-matrix initialization (Midpoint): Strategy choices of algorithms

S.4.5 Different initialization of Q-matrix (Random)

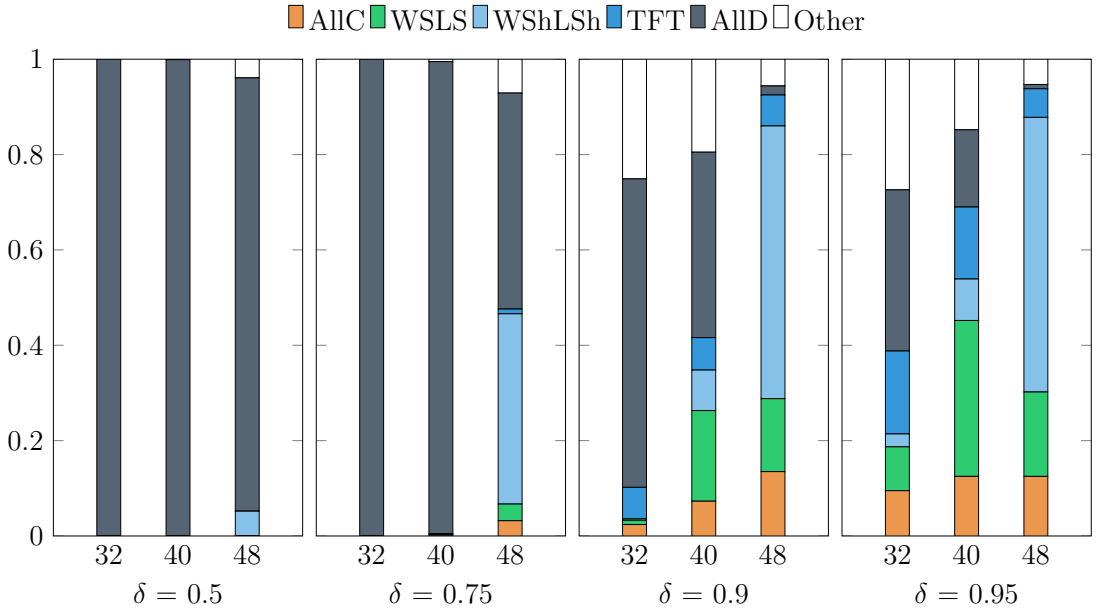
As a robustness check, we repeated our simulation experiments with a different "random" initialization of the Q-matrix, specifically with uniformly random values on $[12/(1-\delta), 50/(1-\delta)]$ in each cell. This initialization corresponds to a (uniform) random draw from the set of feasible net present values, which is the interval with extreme points given by the payoffs of (C, D) and (D, C).

The results are shown in Table S.23 and Figure S.6. As these results show, our main insights are robust to these changes in the initialization of the Q-matrix.

Table S.23: Average cooperation by treatment, $\nu = 20$

k	δ	$R = 32$	$R = 40$	$R = 48$
1	0.50	0.00***	0.00***	2.65***
	0.75	0.00***	0.40***	31.63***
	0.90	11.28***	42.66	67.75
	0.95	37.08	68.83	68.70

Note: This table depicts average cooperation rates by algorithms, as well as the significance of a two-sided Mann-Whitney-U-test relative to human cooperation (see Table S.4). * $p < 0.05$, ** $p < 0.01$, *** $p < 0.001$



Note: The figure reports the manual classification for $k = 1$ for the baseline parameters $\alpha = 0.15$, $\nu = 20$.

Figure S.6: Different Q-matrix initialization (Random): Strategy choices of algorithms

S.4.6 Different initialization of Q-matrix (Optimistic)

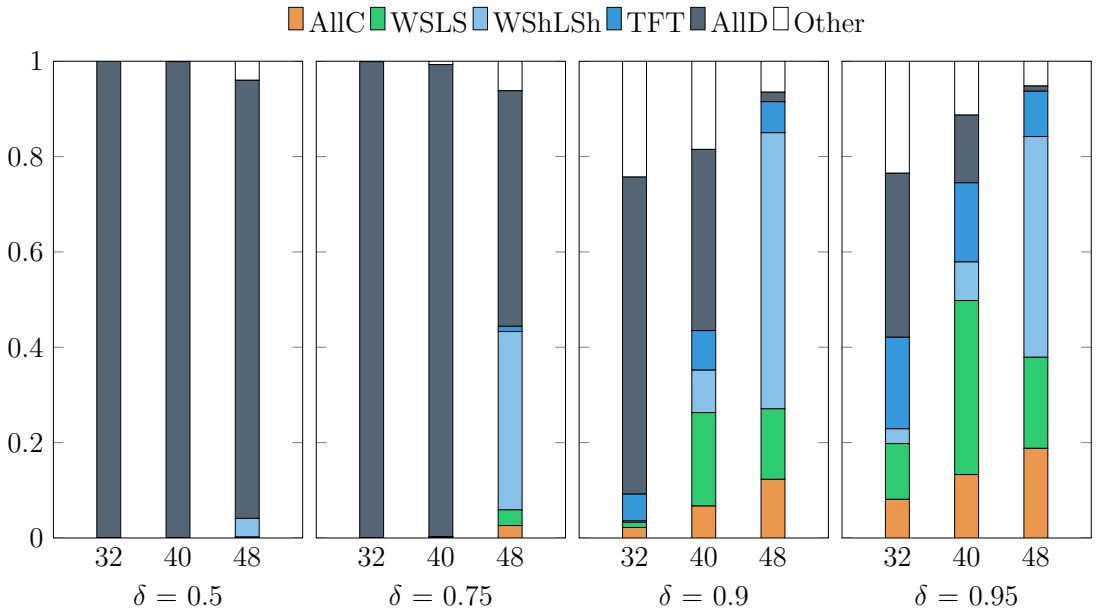
As a robustness check, we repeated our simulation experiments with an "optimistic" initialization of the Q-matrix, specifically with $50/(1-\delta)$ in all cells. This initialization corresponds to the net present value of playing D forever, while your opponent plays C forever, resulting in the best possible payoff of 50 in every period.

The results are shown in Table S.24 and Figure S.7. As these results show, our main insights are robust to these changes in the initialization of the Q-matrix.

Table S.24: Average cooperation by treatment, $\nu = 20$

k	δ	$R = 32$	$R = 40$	$R = 48$
1	0.50	0.00***	0.00***	2.15***
	0.75	0.00***	0.35***	29.57***
	0.90	9.40***	43.22	67.05
	0.95	37.44	70.85	74.68

Note: This table depicts average cooperation rates by algorithms, as well as the significance of a two-sided Mann-Whitney-U-test relative to human cooperation (see Table S.4). * $p < 0.05$, ** $p < 0.01$, *** $p < 0.001$



Note: The figure reports the manual classification for $k = 1$ for the baseline parameters $\alpha = 0.15$, $\nu = 20$.

Figure S.7: Different Q-matrix initialization (Optimistic): Strategy choices of algorithms

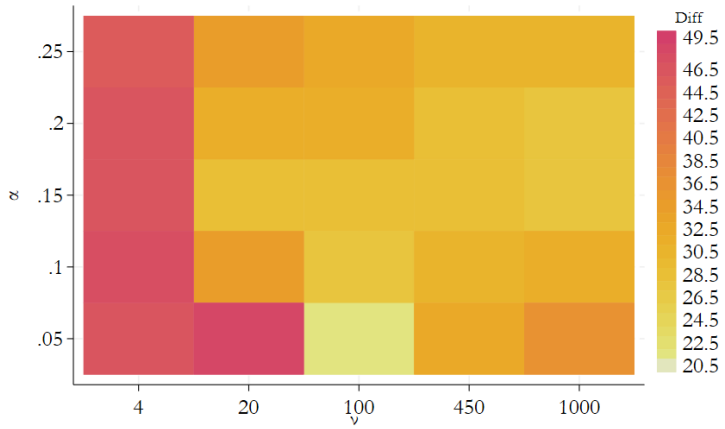
S.4.7 Heatmaps

In order to investigate robustness of our main results, we present heatmaps in (α, ν) space (for $\alpha \in \{0.05, 0.10, 0.15, 0.20, 0.25\}$ and $\nu \in \{4, 20, 100, 450, 1000\}$) and for each k. For each set of hyperparameters, we have 12 parametrizations of the game, namely $R \in \{32, 40, 48\}$ and $\delta \in \{0.50, 0.75, 0.90, 0.95\}$.

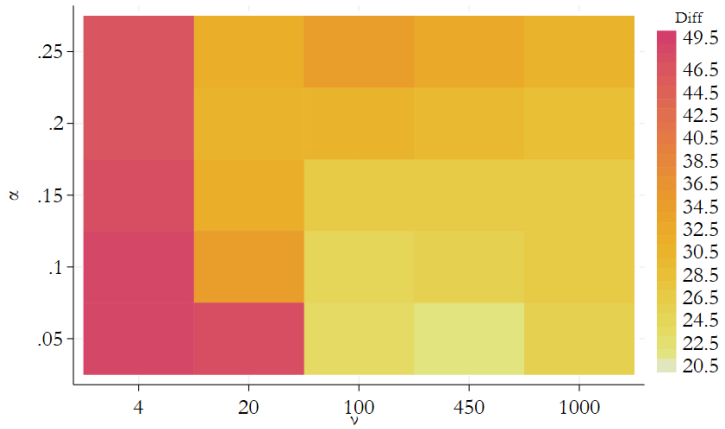
For each set of hyperparameters, we compute the difference between human and algorithmic cooperation rates, averaged over the 12 parametrizations. The average differences are in the range between 20 and 50 percentage points (and hence strictly positive) for all 25 (α, ν)-combinations and for all $k \in \{1, 2, 3\}$. The associated heatmaps are shown in Figure S.8.

Additionally, we compute the fraction of parametrizations for which average human cooperation rates are higher than those of algorithms. These fractions exceed 75% for all hyperparameters and k. The associated heatmaps are shown in Figure S.9.

(a) $k = 1$



(b) $k = 2$



(c) $k = 3$

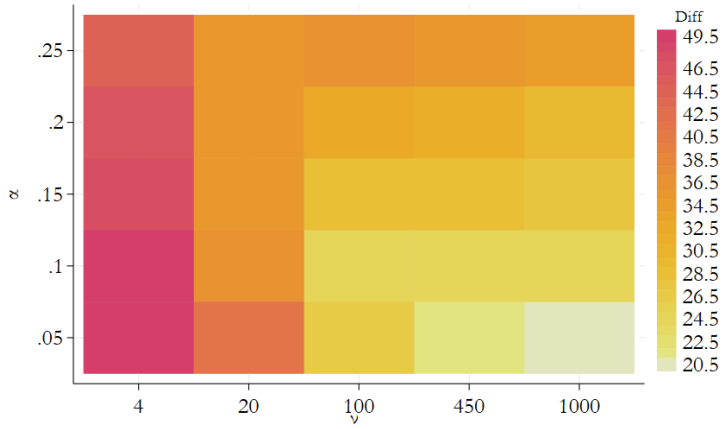
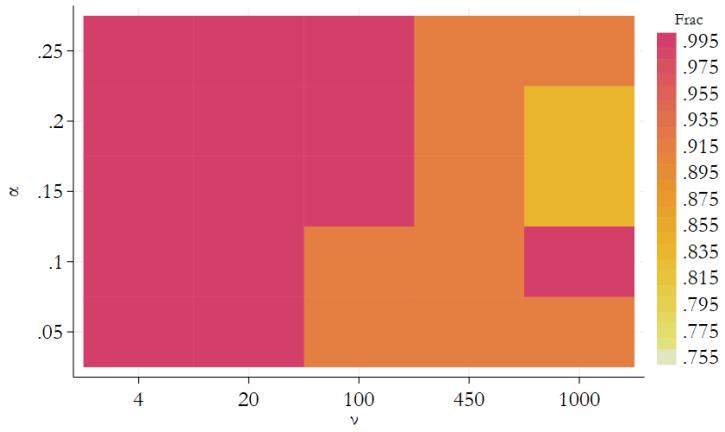


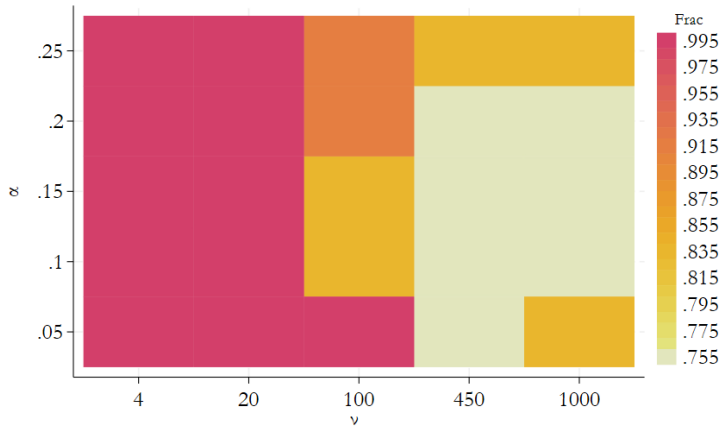
Figure S.8: Heatmaps with average differences between human and algorithmic cooperation rates

Note: These figures show the difference of human and algorithmic cooperation rates, averaged over the 12 parametrizations of δ and R.

(a) $k = 1$



(b) $k = 2$



(c) $k = 3$

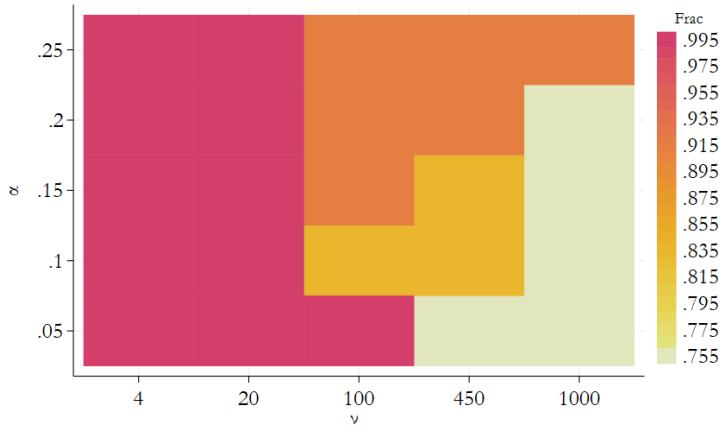


Figure S.9: Heatmaps with fractions where human cooperation rates are higher than algorithmic cooperation rates

Note: These figures show the fraction of parametrizations of δ and R where human cooperation rates are higher than algorithmic cooperation rates.

S.5 Q-Learning

Let S and A denote the sets of states and actions, respectively. Q-learning agents aim to maximize the sum of discounted payoffs π_t, where the per-period payoff π_t depends on state $s_t \in S$ and the chosen action profile, including the algorithm's $a_t \in A$. The Bellman equation expresses the optimization problem:

$$V(s_t) = \max_{a_t}\{\mathbb{E}[\pi_t|s_t, a_t] + \delta\mathbb{E}[V(s_{t+1})|s_t, a_t]\} \tag{2}$$

with $\delta \in [0, 1)$ being the discount factor. Importantly, the Q-learning agent does not have prior knowledge about the environment it interacts with. Hence, it does not know which action is optimal in the current state. In Q-learning, the Bellman equation is rewritten as the Q-function

$$Q(s_t, a_t) = \mathbb{E}[\pi_t|s_t, a_t] + \delta\mathbb{E}[\max_a Q(s_{t+1}, a)|s_t, a_t]. \tag{3}$$

The Q-function can be expressed as an $|S| \times |A|$ matrix when A and S are finite. Each cell of this Q-matrix then represents the approximation of the expected discounted stream of rewards for a specific state-action combination.

In each period t the Q-learning agent picks an action a_t and updates the Q-matrix as a weighted average between the past approximation of the state-action value and the newly learned information:

$$Q_{t+1}(s_t, a_t) = (1 - \alpha)Q_t(s_t, a_t) + \alpha(\pi_t + \delta \max_a Q_t(s_{t+1}, a)) \tag{4}$$

The parameter $\alpha \in [0, 1)$ is the learning rate. It governs the weight which is given to newly arriving information. The agent does not learn α but it is a hyperparameter that is set by the researcher or the developer of the algorithm.

Exploration and exploitation. In each period, the agent has to decide which action to choose conditional on observing the current state. Given the current Q-matrix, the agent should pick the profit-maximizing action $a_t^* = \arg\max_a Q(s_t, a)$. However, the Q-matrix is an approximation and might not reflect the true environment. Hence, the algorithm should also explore the performance of different actions. For instance, some actions might be underexplored in the given period and yield a higher reward than the current approximation might suggest. Thus, the agent faces a trade-off between exploring the action space and exploiting the information it already has about the environment. To circumvent the trade-off,

the ε-greedy algorithm is used in Q-learning.[8] With probability $1-\varepsilon_t$ the agent picks the greedy action a_t^*. Conversely, to explore the environment, the agent picks a random action from the action set with probability ε_t. We let ε_t decay over time and define it as $\varepsilon_t = e^{-\beta t}$ for some small $\beta > 0$. At the beginning of the agents' training, the Q-matrix is still uninformative as it has not explored the state-action space sufficiently. Since t is small, ε_t is close to 1 and the agent picks random actions frequently to explore the environment. As the training of the agent progresses, the Q-matrix becomes more informative and it chooses the greedy action more often. In the limit, the agent picks a_t^* only as $\lim_{t\to\infty} \varepsilon_t = 0$.

Q-learning in the Prisoner's Dilemma. In the Prisoner's Dilemma, the agent can choose between cooperation (C) and defection (D). Thus, the action set is given by $A = \{C, D\}$. We define $m = |A| = 2$ as its cardinality. The reward signal π_t is simply given by the payoff for the agent in the given round.

Following Calvano et al. (2020), we consider state representations where the agents condition their current action in period t on the set of actions from the previous k periods. We denote $\mathbf{a_k}$ as the vector of actions by both agents in period k. The state space is then $S = \{\mathbf{a_1}, \mathbf{a_2}, ..., \mathbf{a_k}\}$. Note that the cardinality of S grows exponentially in k as $|S| = m^{nk} = 2^{2k}$.

The cardinality of S interacts with the exploration parameter β. Holding β fixed, the probability that a cell is visited by chance through exploration is smaller in larger state spaces. To address this, we choose $\beta(k)$ to keep ν constant for all k, where ν is the expected value of a cell in the Q-matrix being explored purely by randomness. This is the only way to make reasonable comparisons across different k's, since otherwise the effective propensity to explore would differ across k's. The parameter ν is given by

$$\nu = \frac{(m-1)^n}{m^{kn+n+1}\left(1-e^{-\beta(n+1)}\right)} = \frac{1}{2^{2k+3}\left(1-e^{-3\beta}\right)}; \tag{5}$$

(see Calvano et al. (2020), Section III.A).[9]

The precise values we use in our simulations (baseline and robustness checks) are shown in Table S.25, with associated β values given ν and k shown in Table S.26.

[8] There are other exploration algorithms like Boltzmann exploration. Calvano et al. (2020) show that it yields similar results in market environments; thus, we focus on the ε-greedy algorithm.

[9] The formula reported in Calvano et al. (2020), footnote 20, erroneously contains $kn(n+1)$ in the denominator instead.

Table S.25: Q-learning parameters and values

Para.	Description	Baseline value	Robustness check(s)
α	Learning rate	0.15	0.05, 0.10, 0.20, 0.25
β	Exploration parameter; implied by ν and k	5.21×10^{-4}	See Table S.26 and Section S.4.1
ν	Expected times Q-matrix cell is randomly explored	20	4, 100, 450, 1000
k	Memory length	1	2,3
	Q-matrix initialization	$6/(1-\delta) + U[0,1]$	Calvano et al.: Opp. random Pessimistic: $12/(1-\delta)$ Midpoint: $(12+50)/(2(1-\delta))$ Random: $U[12/(1-\delta), 50/(1-\delta)]$ Optimistic: $50/(1-\delta)$

Table S.26: Parameter configurations: $\beta(k)$ given ν

ν	k		
	1	2	3
4	2.61×10^{-3}	6.52×10^{-4}	1.63×10^{-4}
20	5.21×10^{-4}	1.30×10^{-4}	3.26×10^{-5}
100	1.04×10^{-4}	2.60×10^{-5}	6.51×10^{-6}
450	2.31×10^{-5}	5.79×10^{-6}	1.45×10^{-6}
1000	1.04×10^{-5}	2.60×10^{-6}	6.51×10^{-7}

S.6 Human experiment

In this appendix, we state the hypotheses that we pre-registered for our experiment with human subjects. The comparison with algorithms and data from other human experiments is in the main text, and the instructions for our experiment are in Section S.7.

The experimental literature has shown that cooperation among humans can be expected to increase in the discount factor and the reward parameter (Dal Bó and Fréchette, 2018). Our first hypothesis extends this insight to the parameter region in the δ-R space that has hitherto been unexplored. Specifically, we conduct our own experiments in a 2×2 factorial design $(R = 40, 48) \times (\delta = 0.9, 0.95)$.

Hypothesis 6 (Humans 1)**.** *The cooperation rate among humans increases in R and δ.*

We briefly comment on the results in isolation, and discuss the wider implications and results in the main text. In Table S.27, we investigate the determinants of cooperation for our experiments. We examine both (1) a probit model and (2) a linear probability model (LPM), where we cluster standard errors at the matching group level. Both δ and R have a positive effect on cooperation. The effect of R is weakly statistically significant. The lack of statistical power is due to the fact that cooperation is very high in all of our treatments, and moreover the treatments are relatively "close" (e.g., a continuation probability of 0.95 instead of 0.9 may not make that big a difference for human subjects). However, cooperation rates from our experiments tend to be well above those found in existing experiments, lending support to our initial hypothesis.

Table S.27: Determinants of cooperation, our human experiments

	(1) Probit	(2) LPM
δ	0.01	0.01
	(3.21)	(0.72)
R	0.04∗	0.01∗
	(0.02)	(0.00)
Constant	−0.67	0.44
	(3.50)	(0.77)
N	43950	43950

Standard errors in parentheses
* $p < 0.10$, ** $p < 0.05$, *** $p < 0.010$

As a robustness check, we investigate the levels of human cooperation in our own experiments (where $\delta \in \{0.9, 0.95\}$ and $R \in \{40, 48\}$) for varying subsets of supergames. In Table S.28, we consider all supergames (1), the first supergame only (2), the last three quarters of rounds (3), and the last supergame (4).

Table S.28: Average cooperation by humans by supergame

	(1) All SG		(2) First SG		(3) Last 3/4 Rnds		(4) Last SG	
δ	$R = 40$	$R = 48$	$R = 40$	$R = 48$	$R = 40$	$R = 48$	$R = 40$	$R = 48$
0.90	0.80	0.89	0.73	0.73	0.84	0.96	0.86	0.99
0.95	0.83	0.88	0.69	0.68	0.88	0.94	0.93	0.93

Note: This table depicts average cooperation rates by humans.

S.7 Instructions in the human-player experiment

We now provide the experimental instructions for humans for the treatment with $\delta = 0.95$ and $R = 48$. The instructions differ across treatments only in these two parameters. The original instructions were in German and based on the instructions by Dal Bó and Fréchette (2011). We also provide an English translation.

S.7.1 Original version

Willkommen:

> Willkommen zu unserem Experiment!
>
> Sie nehmen gleich an einem Experiment zum Thema Entscheidungsfindung teil und werden am Ende des Experiments privat für Ihre Teilnahme bezahlt. Was Sie verdienen, hängt zum Teil von Ihren Entscheidungen, zum Teil von den Entscheidungen anderer und zum Teil vom Zufall ab. Für dieses Experiment gibt es generell 4€ Teilnahmeprämie, plus den Verdienst, der aus den Entscheidungen resultiert.
>
> Das gesamte Experiment wird über Computer abgewickelt, und die gesamte Interaktion zwischen Ihnen wird über die Computer stattfinden.
>
> Bitte schalten Sie jetzt Ihre Mobiltelefone aus. Bitte sprechen Sie nicht und versuchen Sie nicht, während der Sitzung mit anderen Teilnehmern zu kommunizieren.
>
> Wir beginnen mit einer kurzen Einführungsphase. Während der Einweisung erhalten Sie eine Beschreibung der Hauptmerkmale des Experiments und es wird Ihnen gezeigt, wie Sie die Computer benutzen. Wenn Sie während dieser Zeit Fragen haben, heben Sie die Hand und Ihre Frage wird so beantwortet, dass jeder sie hören kann.

Anleitung:

1. In diesem Experiment werden Sie aufgefordert, in mehreren Runden Entscheidungen zu treffen. Sie werden nach dem Zufallsprinzip mit einer anderen Person für eine Reihe von Runden gepaart. Jede Folge von Runden wird als Spiel bezeichnet.
2. Die Länge eines Spiels wird zufällig bestimmt. Nach jeder Runde besteht eine 95-prozentige Wahrscheinlichkeit, dass das Spiel für

mindestens eine weitere Runde fortgesetzt wird. Das ist so, als würden wir nach jeder Runde einen zwanzigseiten Würfel werfen und bei "1" bis "19" weitermachen und bei "20" aufhören. Wenn Sie z. B. in Runde 2 sind, beträgt die Wahrscheinlichkeit, dass es eine dritte Runde gibt, 95%, und wenn Sie in Runde 9 sind, beträgt die Wahrscheinlichkeit, dass es eine weitere Runde gibt, ebenfalls 95%.

3. Sobald ein Spiel endet, werden Sie zufällig mit einer anderen Person für ein neues Spiel gepaart.

4. Es werden insgesamt maximal 15 Spiele gespielt. Das erste Spiel, das nach 120 Minuten Spielzeit beendet ist, markiert das Ende des Experiments.

5. Die Auswahlmöglichkeiten und die Auszahlungen in jeder Runde sind wie folgt:

	Entscheidung des anderen Spielers	
Ihre Entscheidung	A	B
A	48, 48	12, 50
B	50, 12	25, 25

   Der erste Eintrag in jeder Zelle steht für Ihre Auszahlung, während der zweite Eintrag die Auszahlung der Person darstellt, mit der Sie zusammengeführt werden.

   - Sie wählen A und der/die andere wählt A, dann erhalten Sie beide 48.
   - Sie wählen A und der/die andere wählt B, dann erhalten Sie 12 und der/die andere 50.
   - Sie wählen B und der/die andere wählt A, dann erhalten Sie 50 und der/die andere 12.
   - Sie wählen B und der/die andere wählt B, dann erhalten Sie beide 25.

6. Am Ende des Experiments wird ein Spiel zufällig für die Auszahlung ausgelost. Von den in diesem Spiel erzielten Punkten erhalten Sie 1 € für 40 erzielte Punkte. Hinzukommt die oben erwähnte Zahlung von 4€.

7. Bevor wir beginnen, möchten wir Sie daran erinnern: Die Dauer eines Spiels wird nach dem Zufallsprinzip bestimmt. Nach jeder

Runde besteht eine 95-prozentige Wahrscheinlichkeit, dass das Spiel für mindestens eine weitere Runde fortgesetzt wird. Sie spielen während des gesamten Spiels (über alle Runden) mit der gleichen Person. Bei einem neuen Spiel werden Sie zufällig mit einer anderen Person für ein neues Spiel gepaart. Für Ihre Auszahlung wird am Ende ein Spiel zufällig ausgelost.

Gibt es noch Fragen?

Quiz

1. Wie viele Punkte verdienen Sie in einer Runde, wenn Sie und der andere Teilnehmer Option A wählen?
2. Wie viele Punkte verdienen Sie in einer Runde, wenn sowohl Sie als auch der andere Teilnehmer Option B wählen?
3. Wie viele Punkte verdienen Sie in einer Runde, wenn Sie Option A wählen und der andere Teilnehmer Option B wählt?
4. Wie viele Punkte verdienen Sie in einer Runde, wenn Sie Option B wählen und der andere Teilnehmer Option A wählt?
5. Nehmen wir an, Sie befinden sich in der vierten Runde eines Spiels. Wie hoch ist die Wahrscheinlichkeit, dass das Spiel nach dieser Runde beendet wird?

S.7.2 English translation

Welcome

Welcome to our experiment!

You are about to participate in an experiment on decision-making, and at the end of the experiment you will be paid privately for your participation. What you earn depends partly on your decisions, partly on the decisions of others, and partly on chance. There is generally 4€ participation pay for this experiment, plus the earnings that result from the decisions.

The entire experiment will be conducted through computers, and all interaction between you will be through the computers.

Please turn off your cell phones now. Please do not speak or attempt to communicate with other participants during the session.

We will begin with a brief introductory period. During the briefing, you will be given a description of the main features of the experiment and shown how to use the computers. If you have any questions during this time, raise your hand and your question will be answered so that everyone can hear it.

Instruction

1. In this experiment, you will be asked to make choices in several rounds. You will be randomly paired with another person for a series of rounds. Each sequence of rounds is called a game.
2. The length of a game is determined randomly. After each round, there is a 95 percent chance that the game will continue for at least one more round. This is like rolling a twenty-sided die after each round and continuing at "1" through "19" and stopping at "20". For example, if you are in round 2, the probability that there will be a third round is 95%, and if you are in round 9, the probability that there will be another round is also 95%.
3. Once a game ends, you will be randomly paired with another person for a new game.
4. There will be a maximum of 15 games played in total. The first game to finish after 120 minutes of play marks the end of the experiment.
5. The choices and payouts in each round are as follows:

	Other player's choice	
Your decision	A	B
A	48, 48	12, 50
B	50, 12	25, 25

The first entry in each cell represents your payoff, while the second entry represents the payoff of the person you are matched with.

- You choose A and the other person(s) chooses A, then you both get 48.

- You choose A and the other chooses B, then you get 12 and the other 50.
- You choose B and the other chooses A, then you get 50 and the other 12.
- You choose B and the other chooses B, then you both get 25.

6. At the end of the experiment, one game is randomly drawn for payoff. From the points scored in this game, you will receive 1 € for 40 points scored. In addition, there is the above-mentioned payment of 4€.
7. Before we begin, we would like to remind you: the duration of a game is determined randomly. After each round, there is a 95% probability that the game will continue for at least one more round. You will play with the same person during the entire game (over all rounds). When a new game is played, you will be randomly paired with another person for a new game. For your payout, a game is randomly drawn at the end.

Are there any questions?

Quiz

1. How many points do you earn in a round if you and the other participant choose Option A?
2. How many points do you earn in a round if both you and the other participant choose Option B?
3. How many points do you earn in a round if you choose Option A and the other participant chooses Option B?
4. How many points do you earn in a round if you choose Option B and the other participant chooses Option A?
5. Assuming you are in the fourth round of a game, what is the probability that the game will end after this round?

References

Asker, John, Chaim Fershtman, and Ariel Pakes, "The impact of artificial intelligence design on pricing," *Journal of Economics & Management Strategy*, 2024, *33* (2), 276–304.

Banchio, Martino and Giacomo Mantegazza, "Artificial intelligence and spontaneous collusion," *Working paper*, 2022.

Bigoni, Maria, Marco Casari, Andrzej Skrzypacz, and Giancarlo Spagnolo, "Time Horizon and Cooperation in Continuous Time," *Econometrica*, 2015, *83* (2), 587–616.

Blonski, Matthias and Giancarlo Spagnolo, "Prisoners' other Dilemma," *International Journal of Game Theory*, 2015, *44*, 61–81.

_ , Peter Ockenfels, and Giancarlo Spagnolo, "Equilibrium Selection in the Repeated Prisoner's Dilemma: Axiomatic Approach and Experimental Evidence," *American Economic Journal: Microeconomics*, 2011, *3* (3), 164–192.

Breitmoser, Yves, "Cooperation, but no Reciprocity: Individual Strategies in the Repeated Prisoner's Dilemma," *American Economic Review*, 2015, *105* (9), 2882–2910.

Calvano, Emilio, Giacomo Calzolari, Vincenzo Denicolò, and Sergio Pastorello, "Artificial Intelligence, Algorithmic Pricing, and Collusion," *American Economic Review*, 2020, *110* (10), 3267–3297.

Dal Bó, Pedro and Guillaume R. Fréchette, "The Evolution of Cooperation in Infinitely Repeated Games: Experimental Evidence," *American Economic Review*, February 2011, *101* (1), 411–29.

_ and _ , "On the Determinants of Cooperation in Infinitely Repeated Games: A Survey," *Journal of Economic Literature*, March 2018, *56* (1), 60–114.

_ and _ , "Strategy Choice in the Infinitely Repeated Prisoner's Dilemma," *American Economic Review*, 2019, *109* (11), 3929–52.

Dolgopolov, Arthur, "Reinforcement learning in a prisoner's dilemma," *Games and Economic Behavior*, 2024, *144*, 84–103.

Embrey, Matthew, Guillaume R. Fréchette, and Sevgi Yuksel, "Cooperation in the Finitely Repeated Prisoner's Dilemma," *The Quarterly Journal of Economics*, 2018, *133* (1), 509–551.

Fudenberg, D., D.G. Rand, and A. Dreber, "Slow to Anger and Fast to Forgive: Cooperation in an Uncertain World," *American Economic Review*, 2012, *102* (2), 720–749.

Ghidoni, Riccardo and Sigrid Suetens, "The Effect of Sequentiality on Cooperation in Repeated Games," *American Economic Journal: Microeconomics*, 2022, *14* (4), 58–77.

Harsanyi, John C. and Reinhard Selten, *A General Theory of Equilibrium Selection in Games*, MIT Press, 1988.

Kirman, Alan, François Laisney, and Paul Pezanis-Christou, "Relaxing the symmetry assumption in participation games: a specification test for cluster-heterogeneity," *Experimental Economics*, 2023, *26* (4), 850–878.

Mengel, Friederike, "Risk and Temptation: A Meta-study on Prisoner's Dilemma Games," *The Economic Journal*, 2018, *128* (616), 3182–3209.

Romero, Julian and Yaroslav Rosokha, "Constructing Strategies in the Indefinitely Repeated Prisoner's Dilemma Game," *European Economic Review*, 2018, *104*, 185–219.

PREVIOUS DISCUSSION PAPERS

437 Kasberger, Bernhard, Martin, Simon, Normann, Hans-Theo and Werner, Tobias, Algorithmic Cooperation: A Comparison with Human Play in the Infinitely Repeated Prisoner's Dilemma, June 2026.
Forthcoming in: Games and Economic Behavior.

436 Moeller, Jakob, Halla, Martin and Thomas, Tobias, Pre-Booked Vaccination Appointments as a Nudge: Evidence from a Nationwide Intervention, June 2026.

435 Magin, Jana Anjali, Neyer, Ulrike and Stevens, Alexandra, Central Bank Digital Currency as a New Means of Payment: An Experimental Approach, March 2026.

434 Gregor, Leonard and Haucap, Justus, Energy Markets at War: The Effect of the Russian Invasion of Ukraine on Refinery Margins, March 2026.

433 Bertermann, Alexander and Schildberg-Hörisch, Hannah, Parental Mental Health and the Economic Preferences of the Next Generation, January 2026.

432 Haucap, Justus, Karacuka, Mehmet and Inke, Hakan, An Empirical Inquiry into Cartel Overcharges and Cartel Fines Including an Assessment of the EU's Guidelines on Cartel Fines and Damages, November 2025.

431 Gregor, Leonard and Haucap, Justus, The Rise of Refinery Margins: The Case of Energy Tax Cut in Germany, October 2025.

430 Duso, Tomaso, Harrington, Joseph E. Jr., Kreuzberg, Carl and Sapi, Geza, Public Communication and Collusion: New Screening Tools for Competition Authorities, October 2025.

429 Benndorf, Volker, Kübler, Dorothea and Normann, Hans-Theo, Information Unraveling and Limited Depth of Reasoning, September 2025.
Forthcoming in: Games and Economic Behavior.

428 Martinez-Martinez, Ismael and Normann, Hans-Theo, Cooperation in Multiplayer Dilemmas, September 2025.
Forthcoming in: Journal of Economic Behavior and Organization.

427 Neyer, Ulrike, Stempel, Daniel and Stevens, Alexandra, Subsistence Consumption and Inflation Heterogeneity: Implications for Monetary Policy Transmission in a HANK Model, September 2025.

426 Dibiasi, Andreas and Erhardt, Katharina, Young Firms under Pressure: Heterogeneous Investment Responses to a Trade Shock, August 2015.

425 Dertwinkel-Kalt, Markus and Wey, Christian, Comparing the Effects of Subsidies on Target Goods, July 2025.

424 Dertwinkel-Kalt, Markus and Wey, Christian, The Economics of Fleet-Wide Emission Targets and Pooling in the EU Car Market, July, 2025.

423 Chowdhury, Shyamal, Puente-Beccar, Manuela, Schildberg-Hörisch, Hannah, Schneider, Sebastian O. and Sutter, Matthias, Spatial Patterns in the Formation of Economic Preferences, July 2025.

422 Breitkopf, Laura, Chowdhury, Shyamal, Kamhöfer, Daniel A., Schildberg-Hörisch, Hannah and Sutter, Matthias, The Right Timing Matters: Sensitive Periods in the Formation of Socio-Emotional Skills, June 2025.

421 Mori, Tomoya and Wrona, Jens, Culture, Tastes, and Market Integration: Testing the Localized Tastes Hypothesis, May 2025.

420 Marchal, Léa, Ourens, Guzmán and Sabbadini, Giulia, When Immigrants Meet Exporters: A Reassessment of the Migrant-Native Wage Gap, May 2025.

419 Kesler, Reinhold, Skiera, Bernd, Kraft, Lennart and Koschella, Tim, Existence, Antecedents and Consequences of Non-Compliance in Mobile App Markets, April 2025.

418 Oschmann, Sebastian, Vertical Market Structure Matters: The Case of a Horizontal Retail Merger in the German Gasoline Market, March 2025.

417 Breitkopf, Laura, Chowdhury, Shyamal, Priyam, Shambhavi, Schildberg-Hörisch, Hannah and Sutter, Matthias, Nurturing the Future: How Positive Parenting is Related to Children's Skills and Well-being, October 2024.

416 Rademacher, Philip, Forecasting Recessions in Germany with Machine Learning, September 2024.
Published in Journal of Business Cycle Research, November 2025.

415 Erhardt, Katharina and Gupta, Apoorva, Go Wide or Go Deep: Margins of New Trade Flows, August 2024.

414 Gupta, Apoorva and Stiebale, Joel, Gains from Patent Protection: Innovation, Market Power and Cost Savings in India, May 2024.

413 Cobb-Clark, Deborah A., Dahmann, Sarah C., Kamhöfer, Daniel A., and Schildberg-Hörisch, Hannah, Schooling and Self-Control, March 2024.

412 Huelden, Tobias, Jascisens, Vitalijs, Roemheld, Lars and Werner, Tobias, Human-Machine Interactions in Pricing: Evidence from Two Large-Scale Field Experiments, March 2024.

411 Hermes, Henning, Lergetporer, Philipp, Mierisch, Fabian, Schwerdt, Guido and Wiederhold, Simon, Does Information about Inequality and Discrimination in Early Child Care Affect Policy Preferences? January 2024.

410 Hunold, Matthias and Werner, Tobias, Algorithmic Price Recommendations and Collusion: Experimental Evidence, December 2023.

409 Herzog, Sabrina, Schildberg-Hörisch, Hannah, Trieu, Chi and Willrodt, Jana, Who is in Favor of Affirmative Action? Representative Evidence from an Experiment and a Survey, November 2023.

408 Stadelmann, David, Thomas, Tobias and Zakharov, Nikita, Too Hot to Play it Cool? Temperature and Media Bias, November 2023.

407 Dertwinkel-Kalt, Markus and Wey, Christian, Why “Energy Price Brakes” Encourage Moral Hazard, Raise Energy Prices, and Reinforce Energy Savings, September 2023.
Forthcoming in: RAND Journal of Economics.

406 Bertermann, Alexander, Kamhöfer, Daniel A. and Schildberg-Hörisch, Hannah, More Education Does Make You Happier – Unless You Are Unemployed, September 2023.

405 Trieu, Chi, Who’s Who: How Uncertainty About the Favored Group Affects Outcomes of Affirmative Action, August 2023.
Published in: Journal of the Economic Science Association, 9 (2023), pp. 252-292.

404 Fischer, Kai, Martin, Simon and Schmidt-Dengler, Philipp, The Heterogeneous Effects of Entry on Prices, July 2023.

403 García-Vega, María, Gupta, Apoorva and Kneller, Richard, Is Acquisition-FDI During an Economic Crisis Detrimental for Domestic Innovation?, July 2023.

402 Haucap, Justus and Stiebale, Joel, Non-price Effects of Mergers and Acquisitions, July 2023.

401 Simion, Ștefania and Sulka, Tomasz, Multidimensional Cognitive Ability, Intermediate Channels, and Financial Outcomes, May 2023.

400 Bartling, Björn, Cappelen, Alexander W., Hermes, Henning, Skivenes, Marit and Tungodden, Bertil, Free to Fail? Paternalistic Preferences in the United States, May 2023.

399 Kandelhardt, Johannes, Flexible Estimation of Random Coefficient Logit Models of Differentiated Product Demand, May 2023.

398 Hermes, Henning, Lergetporer, Philipp, Mierisch, Fabian, Peter, Frauke and Wiederhold, Simon, Discrimination on the Child Care Market: A Nationwide Field Experiment, April 2023.

397 Schmal, W. Benedikt, Haucap, Justus and Knoke, Leon, The Role of Gender and Coauthors in Academic Publication Behavior, March 2023.
Published in: Research Policy, 52 (2023), 104874.

396 Magin, Jana Anjali, Neyer, Ulrike and Stempel, Daniel, The Macroeconomic Effects of Different CBDC Regimes in an Economy with a Heterogeneous Household Sector, March 2023.

395 Dertwinkel-Kalt, Markus and Wey, Christian, Resale Price Maintenance in a Successive Monopoly Model, February 2023.
Forthcoming in: Journal of Industrial Economics.

Older discussion papers can be found online at:
http://ideas.repec.org/s/zbw/dicedp.html

Heinrich-Heine-Universität Düsseldorf

Düsseldorfer Institut für Wettbewerbsökonomie (DICE)

Universitätsstraße 1, 40225 Düsseldorf

ISSN 2190-992X (online)
ISBN 978-3-86304-436-7