CESifo WORKING PAPERS

11124
2024

May 2024

Algorithmic Cooperation

Bernhard Kasberger, Simon Martin, Hans-Theo Normann, Tobias Werner

CESifo

Impressum:

CESifo Working Papers
ISSN 2364-1428 (electronic version)
Publisher and distributor: Munich Society for the Promotion of Economic Research - CESifo GmbH
The international platform of Ludwigs-Maximilians University's Center for Economic Studies and the ifo Institute
Poschingerstr. 5, 81679 Munich, Germany
Telephone +49 (0)89 2180-2740, Telefax +49 (0)89 2180-17845, email office@cesifo.de
Editor: Clemens Fuest
https://www.cesifo.org/en/wp
An electronic version of the paper may be downloaded
· from the SSRN website: www.SSRN.com
· from the RePEc website: www.RePEc.org
· from the CESifo website: https://www.cesifo.org/en/wp

CESifo *Working Paper No. 11124*

Algorithmic Cooperation

Abstract

Algorithms play an increasingly important role in economic situations. These situations are often strategic, where the artificial intelligence may or may not be cooperative. We study the determinants and forms of algorithmic cooperation in the infinitely repeated prisoner's dilemma. We run a sequence of computational experiments, accompanied by additional repeated prisoner's dilemma games played by humans in the lab. We find that the same factors that increase human cooperation largely also determine the cooperation rates of algorithms. However, algorithms tend to play different strategies than humans. Algorithms cooperate less than humans when cooperation is very risky or not incentive-compatible.

JEL-Codes: C720, C730, C920, D830.

Keywords: artificial intelligence, cooperation, large language models, Q-learning, repeated prisoner's dilemma.

Bernhard Kasberger
Department of Economics
University of Konstanz / Germany
bernhard.kasberger@uni.kn

Simon Martin
Düsseldorf Institute for Competition Economics (DICE), Heinrich-Heine-University
Düsseldorf / Germany
simon.martin@dice.hhu.de

Hans-Theo Normann
Düsseldorf Institute for Competition Economics (DICE), Heinrich-Heine-University
Düsseldorf / Germany
normann@hhu.de

Tobias Werner
Center for Humans and Machines at the Max Planck Institute for Human Development
Berlin / Germany
werner@mpib-berlin.mpg.de

May 14 2024
We are grateful to Maria Bigoni, Joe Harrington, and Itzhak Rasooly for valuable comments. Seminar participants and conference audiences at the following venues gave useful feedback on the paper: University of Düsseldorf, University of Linz, University of Münster, University of Potsdam, University of Salzburg, University of Vienna, Max Planck Institute for Research on Collective Goods, 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).

1 Introduction

Cooperation often increases the welfare of humans and other species, but 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 in 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 being played. The vast experimental literature (see our literature review below) has addressed the determinants, forms, and levels of cooperation for human players.

We study how self-learning algorithms play the repeated prisoner's dilemma. Specifically, 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 (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 self-learning algorithms and humans in social dilemmas. First, we examine whether the determinants that shape human cooperation also influence algorithmic cooperation. Second, we ask which strategies the algorithms adopt and contrast them with those of humans. Finally, we compare the levels of cooperation between humans and algorithms and ask which factors contribute to the differences.

Understanding the behavior of self-learning algorithms is essential (Rahwan et al., 2019). After all, algorithms advise humans or decide on their behalf more and more often. For example, algorithms may autonomously drive cars, adjust financial portfolios, detect fraud, or set prices, among other applications. Some autonomous algorithms operate in strategic situations and interact repeatedly with other self-learning agents. This can occur in coordination problems; for example, in choosing traffic routes, or

in warfare (Jensen et al., 2020). Other strategic situations present the AI with the possibility of cooperating in social dilemmas, e.g., in team production or computation offloading (Kuang et al., 2021), or in markets to the detriment of the consumers (Harrington, 2018, Miklós-Thal and Tucker, 2019, Calvano et al., 2020b, Ezrachi and Stucke, 2020, Harrington, 2022, Martin and Rasch, 2022). Either way, it is important to understand how algorithms interact with each other and their potential impact on society.

As a methodological step forward in this direction, we apply the strategy frequency estimation method (SFEM), developed for the analysis of human data (Dal Bó and Fréchette, 2011), to the algorithms' decisions. The algorithms' behavior often appears as a black box, and knowledge of how algorithms work and how to predict their behavior is important. A key challenge for interpreting algorithmic behavior is that the number and complexity of the strategies grows in the algorithm's complexity. The SFEM works around this issue by estimating the frequency of each strategy from a pre-specified set of candidate strategies (e.g., grim trigger, tit-for-tat, etc.). The result is a representation of strategies that is both understandable to humans and comparable to the strategies adopted by humans. We assess the estimates of the SFEM and find that it performs accurately in our setting. This finding suggests that the SFEM can also be fruitfully applied to studying algorithmic behavior in other strategic settings.

Our experimental design is as follows. We analyze how a Q-learning algorithm plays various repeated prisoner's dilemma games. Q-learning (Watkins, 1989, Watkins and Dayan, 1992) is a form of reinforcement learning, widely studied in economics (Calvano et al., 2020a, Johnson et al., 2023, Klein, 2021, Abada and Lambin, 2023, Asker et al., 2024) and forms the basis of several more sophisticated algorithms used in the field. We have three main treatment variables. First, we vary the reward from mutual cooperation across three levels. The discount factor is our second treatment variable, which we set at four different values. Our third treatment variable is the algorithm's memory, which is hard-coded in Q-learning. We consider algorithms with memory one, two, and three. Lastly, we study how cooperation and the distribution of learned strategies depend on the treatment variables and the algorithm's learning and exploration rate. We do not view these hyperparameters as classic treatment variables because

they lack an economic interpretation. As our objective is to compare the algorithms' to human behavior, we run additional laboratory experiments to collect data for parameter constellations that have been unexplored up to now. These results are of independent interest.

Regarding the determinants of cooperation, we find that the same factors that increase human cooperation largely also determine algorithmic cooperation rates: A higher reward from cooperation and a higher weight on future payoffs facilitate algorithmic cooperation. The length of the memory of the agent has an ambiguous influence, and we find that many algorithms do not fully exploit the memory, as most learned strategies are memory-one. 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, Blonski and Spagnolo, 2015). We confirm that algorithmic cooperation emerges only if there are cooperative equilibria and that cooperation increases as it becomes risk-dominant.

Our main finding is that humans and 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. While we find that algorithms also play tit-for-tat, they hardly ever select the grim trigger. 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 last rounds.

Our third object of interest is the level of cooperation. Here we find no unambiguous answer as to whether algorithms outperform humans. While this is sometimes the case, we also find that algorithms often cooperate less than humans. In particular, algorithms never cooperate for low discount factors and low 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.

In an extension, we repeat the experiments with ChatGPT, a Large Language Model (LLM), as the players to study the robustness of our findings to the algorithmic class. LLMs are not designed to learn optimal behavior in a particular environment but are trained on vast human-generated

data. As such, they are readily available, and humans increasingly interact with them for various tasks (see online appendix S.4 for references). The algorithm's propensity to cooperate is similar to the one of humans for medium discount rates and reward parameters. However, strikingly, the determinants that shape cooperation among humans and Q-learning algorithms do not play a significant role for ChatGPT. ChatGPT mainly adopts strategies with memory up to one and chooses always cooperate, tit-for-tat, grim, and win-stay-lose-shift.

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) and Murnighan and Roth (1983). 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.[1] 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 as in Dal Bó and Fréchette (2018).

The second related literature is the one on self-learning algorithms in economics, which has so far largely focused on cooperation in the sense of (socially undesirable) anti-competitive collusion in oligopoly games.[2] 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

[1]Both the theoretical and experimental literature have also recognized the importance of monitoring (Green and Porter, 1984, Harrington and Skrzypacz, 2011, Aoyagi et al., 2019), communication (Fonseca and Normann, 2012, Freitag et al., 2021) and beliefs (Aoyagi et al., 2022, Gill and Rosokha, 2024) for the sustainability of cooperation. Although enabling "communication" among algorithms might be of interest in its own right, we do not follow that avenue in this paper.

[2]There is also a literature that studies pricing algorithms in the field. Chen et al. (2016) provide an early empirical analysis of algorithmic pricing on Amazon Marketplace. Assad et al. (2023) analyze the impact of algorithms in the German retail gasoline market. Brown and MacKay (2023) show that pricing algorithms have important effects in the allergy medications industry. Finally, Wieting and Sapi (2021) analyze algorithmic pricing with data from the online marketplace *Bol.com*.

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 (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 have a full $3 \times 4 \times 3$ treatment design and find non-linear interaction effects.

Schaefer (2022) and Boczoń et al. (2023) also inquire into 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. (2023) test equilibrium selection focusing on the size of the basin of attraction of always defect and the effect of strategic uncertainty. 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.

The computer science literature that studies artificial intelligence in strategic situations often prioritizes algorithmic performance in various games (Crandall and Goodrich, 2011, Lerer and Peysakhovich, 2017, Crandall et al., 2018, Dafoe et al., 2020) or explores complex, video-game-like settings (Hughes et al., 2018, Agapiou et al., 2022). Unlike these studies, we analyze a fundamental reinforcement learning algorithm, and apply tools from game theory and experimental economics to describe its behav-

[3]While in our setting the algorithm itself is the experimental subject of interest, other forms of reinforcement learning have been used to describe human learning, starting with early contributions by Thorndike (1898), Bush and Mosteller (1955), and Siegel (1961). In comparison to Q-learning, most of these learning models are myopic in the sense that they have no memory of the history of the game. As such, they often have limited foresight of the consequences of their current action on future payoffs (Waltman and Kaymak, 2008). For applications that use these simpler learning algorithms to explain human behavior in multiplayer strategic games, see Roth and Erev (1995), Erev and Roth (1998), and Romero and Rosokha (2019).

ior. 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). 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 contrast, we compare human-human with machine-machine interaction.

2 Economic environment and hypotheses

2.1 Basic setup

We study the infinitely repeated 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). Table 1 shows two payoff matrices of the stage game: the normalized payoffs and the payoffs we implement in our experiments. 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 1, g then stands for the payoff the player gains when defecting (instead of cooperating) while the other player cooperates, and $-\ell$ represents the payoff loss when cooperating (instead of defecting) while the other player defects. Naturally, both g and ℓ are positive. In our experiments, we implement the payoffs in Table 1b and vary the reward parameter R from 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 re-

Table 1: Stage game payoffs

(a) Normalized payoffs

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

(b) Payoffs in the experiment

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

peated 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 the state CD. With memory one and two actions, 16 (pure) strategies are possible, if one ignores the initial state (see Table S.1 in the online appendix). 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 to play C for any behavior of the previous round and is never a 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 a SPNE. A strategy with punishment that potentially forms a SPNE is 'grim trigger'

(GT): A player starts by cooperating but defects whenever any player has deviated 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 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 a 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 (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 either 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 The self-learning algorithm

We study how Q-learning algorithms play the repeated prisoner's dilemma in computational experiments. Q-learning is a popular reinforcement learning algorithm designed to solve Markov decision processes (Watkins, 1989, Watkins and Dayan, 1992). Recent work by Calvano et al. (2020a) and Klein (2021) has shown the potential of Q-learning algorithms in strategic economic situations (largely with memory one).

An advantage of Q-learning is that it is tractable and interpretable. Both properties are important for identifying the strategies of the algorithm and for comparing them with the strategies of human players. Tractability in the sense of hard-coding low memory levels guarantees sufficiently simple strategies that can be interpreted by the researchers.[4] Moreover, the set of potential strategies remains manageable (its size grows exponentially in memory length k) and similar to the strategies that human players have

[4]More sophisticated reinforcement learning algorithms may not lead to stationary and low-memory strategies. Note that the principles of Q-learning are at the core of most of the more advanced (deep) reinforcement learning algorithms used in the field. In reinforcement learning, Q-learning is fundamental because it provides the basic structure around which many sophisticated algorithms are built. Deep Q-Networks (DQN) and Double Q-Learning are sophisticated algorithms that extend its simple but effective value estimation process.

been found to use. If humans and algorithms choose from essentially the same set of strategies, we keep "all else fixed" and can focus on the choice differences. Interpretability in the sense of direct observation of the learned strategies is important for evaluating the strategy frequency estimation method.

For ease of exposition, we now describe Q-learning for memory-one strategies and relegate more details on Q-learning to the online appendix. The decision-making process of a Q-learning player is represented by a Q-matrix. The dimension of this Q-matrix depends on the player's memory, i.e., how many past periods the player considers for the decision in the given period and the number of possible actions. For strategies 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 is given by the row-wise maximizers.

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 form of the Q-matrix. 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 $\pi(s, a)$, where $\pi(s, a) \in \{0, 1, -\ell, 1 + g\}$, which is naturally dependent on the player's action a and the other player's action. The player uses the payoff feedback in round t to update the guess of the long-run payoff of

choosing action a in state s according to

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

The new value is a convex combination of the old value, and the current stage-game payoff π plus the best possible guessed 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 is given by the players' period-t actions. Note that each player updates only a single cell in each period.

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 $\beta > 0$. Note that the updating procedure in Q-learning also crucially depends on the discount factor δ, which we vary across treatments. While δ is given by 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.[5] Another important parameter is ν, which is implied by α, β, and k; it denotes the expected number of times a cell in the Q-matrix is being 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. The online appendix contains the formula of ν. We discuss our Q-learning implementation in Section 3.2.

2.3 Experimental insights and our hypotheses

We draw upon the experimental literature to form our hypotheses about the determinants and forms of algorithmic cooperating. Starting with the

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

determinants, the experimental literature has identified several factors that shape human cooperation (Dal Bó and Fréchette, 2018, Embrey et al., 2018, Mengel, 2018). We consider the following four factors where we conjecture that these are also relevant for algorithmic cooperation.[6]

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.

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

Note that Calvano et al. (2020a) find a non-monotone relation between δ and collusion (cooperation) in their repeated differentiated Bertrand game. The relation 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 (as in the following concepts).

Second, cooperation rates tend to be higher in experiments with humans when cooperation can be supported in a 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 a SPNE equilibrium and 0 otherwise. Formally (GT, GT) is a 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.

[6]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.

Hypothesis 2. *A necessary but not sufficient condition for self-learning algorithms with $k > 0$ to cooperate is that grim trigger forms a 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, 2021, 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, $p \cdot (1+g) + (1-p) \cdot 0$. The expected payoff of GT is (weakly) larger than that of AllD if and only if

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

If GT does not form a 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 (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 $\delta \geq \delta^{RD}$.*

There are also hypotheses that relate to the specific Q-learning algorithms and that have no human counterpart. Based on Calvano et al. (2020a, 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.*

In contrast to humans, memory is hard-coded in Q-learning algorithms. The effect of memory on cooperation is ex ante unclear. On the one hand, cooperation can increase in memory as higher memory allows more sophisticated punishment strategies. For example, it may be that a single period of punishment, as in WSLS, may not deter deviations while two periods of punishment do. On the other hand, cooperation may decrease in memory due to the increased state space and potentially longer cycles.[7] The possibility of longer cycles may come with fewer rounds in which players cooperate.

Exploratory Question 1. *Does cooperation among self-learning algorithms increase or decrease in memory?*

The next question relates to the forms of cooperation. How do algorithms cooperate on path and how do they punish deviations off path? 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).

[7]The cycle length is defined as the number of rounds until an initial state is reached again.

Exploratory Question 2. *Which strategies do algorithms learn? How do the strategies depend on the game parameters (δ and R), on the learning parameters α and ν, and on memory k?*

The final question relates to the levels of cooperation. Humans are able to sustain cooperation in lab experiments (Dal Bó and Fréchette, 2018) and self-learning algorithms learn to cooperate (collude) in pricing games (Calvano et al., 2020a). It is thus natural to compare the levels of cooperation.

Exploratory Question 3. *When are algorithms more or less cooperative than humans?*

3 The Experiments

We now describe our treatment variables, the numerical implementation of the self-learning algorithm, and the human-subject experiments.

3.1 Treatment design

There are two main motivations for our experimental design. On the one hand, we want to find the determinants, forms, and levels of algorithmic cooperation. On the other hand, we wish to compare these to the human counterparts. Hence, we chose parameters for which experimental data was available and conducted additional experiments with human subjects.

Table 2: Experiments

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

We study a $3 \times 4 \times 3$ 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), Kartal and Müller (2021) and Romero and Rosokha (2018). The variants with $\delta = 0.95$ are particularly relevant to compare with the parametrization 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 from the algorithmic simulations.

Table 2 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.

3.2 The algorithmic Q-learning experiments

In our AI-based experiment, we distinguish for each run (parameterization) the training stage and the playing stage. 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 10^9 rounds.[8] In the subsequent playing stage, the algorithms' initial actions are the optimal actions in the round of convergence. After that, they play according to

[8]The necessity for such a tight convergence criterion arises in the context of $k = 3$ and $\nu = 1000$, which features a large state space and substantial initial exploration that slows convergence times. In order to allow comparability across parametrizations, we use the same convergence criterion throughout.

the learned strategies. 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 along that path. As such, our results are best understood as an algorithm that was trained in isolation and deployed to the 'real world' only after thorough preparation.

Following our experimental design, the hard-coded memory is at most three.[9] 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.

In our main specification, we let $\alpha = 0.15$, and we compute $\beta(k)$ such that we have $\nu = 20$ for each k.[10] We explore the robustness of our results with respect to α (i.e., $\alpha \in \{0.05, 0.1, 0.15, 0.2, 0.25\}$) and ν (i.e., $\nu \in \{4, 20, 100, 450, 1000\}$). For each parametrization, we repeat 1000 runs with a different random seed. Throughout all simulations, we use a random draw from the unit interval as the initial values of the Q-matrix.

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 in Table 2, ample lab data exist already (see Table S.2 in the online appendix for a complete list of experimental data from other studies that we use in this paper).

[9] 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). See Dawid et al. (2023) for the role of "experience replay" in deep Q-learning. Anyhow, 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.

[10] 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 experiments took place at the DICElab of the University of Duesseldorf and the PLEx at the University of Potsdam between December 2022 and May 2023. Subjects were recruited from the lab's subject pool 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 would be 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 experimetns as registered. We had a total of 240 participants. Participants earned 22.54 euros on average.

4 Determinants of cooperation

Figure 1 shows the average cooperation rate of the algorithms for each δ–R treatment, averaged across all k and calculated over 1000 periods after the algorithms converge. As expected, 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. We will return to this point when we examine learned strategies. Cooperation is far from dominant, let alone perfect: Even for high realizations of the δ–R parameters, cooperation rates do not exceed 60%. Despite the non-monotonicity in δ for high values of R, we take the following result from Figure 1, which is consistent with Hypothesis 1.

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

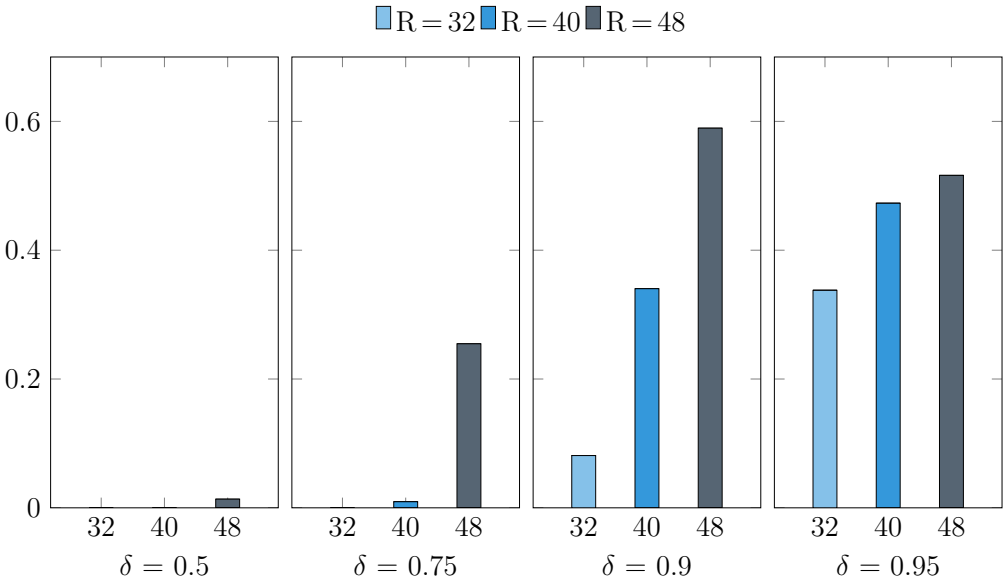

Figure 1: Cooperation rates of algorithms by δ–R treatment.
Note: The figure reports the cooperation rates averaged across all k for the baseline parameters $\alpha = 0.15$, $\nu = 20$. The numerical values are available in Table S.3 in the online appendix.

To investigate the role of memory and the other parameters on cooperation, we run several regressions with the cooperation rate (as in Figure 1) for our baseline parameterization ($\alpha = 0.15$, $\nu = 20$) as the dependent variable and the variables from the hypotheses section as regressors, see Table 3. Regression (1) confirms the descriptive results above. We see a substantial and highly significant effect of δ and R. The regression also includes memory as a control. The average effect of k is negatively 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. 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. Finally, we note that the unexpected drop of cooperation for $R = 48$ and when moving from $\delta = 0.9$ to $\delta = 0.95$ is indeed visible for all $k \in \{1, 2, 3\}$. We answer the Exploratory Question 1 as follows.

Result 2. *The effect of memory on cooperation of self-learning algorithms*

Table 3: 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, * $p < 0.05$, ** $p < 0.01$, *** $p < 0.001$

is ambiguous in general and negative on average.

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 2 shows for which treatments this condition is met. From the cooperation rates in Figure 1, we note three points regarding δ^{SPNE}. First, there is no cooperation in treatment ($\delta = 0.5$, $R = 32$) where GT is not a SPNE. Second, in all treatments with significant levels of cooperation, GT does form a 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 3. *For self-learning algorithms, cooperation occurs only when grim trigger forms a SPNE.*

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 2 shows that for $\delta = 0.5$, GT is risk dominant only when $R = 48$. Looking at Figure 1 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 1 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 3. 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 3, we also find that cooperation increases in $\delta - \delta^{RD}$, which is an intuitive measure of how risk dominant cooperation is.

Result 4. *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 3. The sign of the estimated coefficient is negative, as expected by Hypothesis 3.

Result 5. *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 4, where we report the same set of regressions as in Table 3 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. This provides

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

Result 6. *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 a 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.

We conclude this section by noting that the same determinants influence human and algorithmic cooperation rates. In the next section, we delve deeper into *how* algorithms learn to play the repeated prisoner's dilemma.

5 Forms of cooperation

We now analyze the strategies that the algorithms learn to play. These strategies tell us how the algorithms cooperate and how cooperation is sustained through punishment. One advantage of Q-learning is that the algorithm's strategy can be inferred directly from the Q-matrix. While this is true in principle, 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.

Table 4: 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, * $p < 0.05$, ** $p < 0.01$, *** $p < 0.001$

5.1 Estimating the strategies

We use the Strategy Frequency Estimation Method (SFEM) to estimate the distribution of the limit strategies of the algorithms. The 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).

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. 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 loglikelihood 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. Furthermore, we add an additional memory-one strategy to the estimation procedure which we found when manually classifying the strategies. This strategy prescribes to defect unless both player defected in the previous period. We call this strategy win-shift-lose-shift (WShLSh) and discuss it further below. 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 version 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.[11] If players do no not make mistakes, it is impossible to distinguish between certain strategies. For example, suppose 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 two algorithms 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. To separate strategies off-path, the deviating player deviates in a total of three randomly selected periods.[12] 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.

Crucially, compared to many human-player experiments, we can verify that the SFEM yields correct estimates. The reason is that, for each algorithm, we directly observe the Q-matrix, which provides the complete mapping from states to actions. We assess the SFEM for memory-one ($k = 1$) algorithms, where only 16 strategies are possible (see Table S.1 in the online appendix). Table S.5 in the online appendix 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. To keep the method of estimating the proportions of the strategies the same for the

[11] Furthermore, 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.

[12] The random round in which the algorithm deviates is drawn from a Poisson distribution with $\lambda = 20$. 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 $\lambda = 1$.

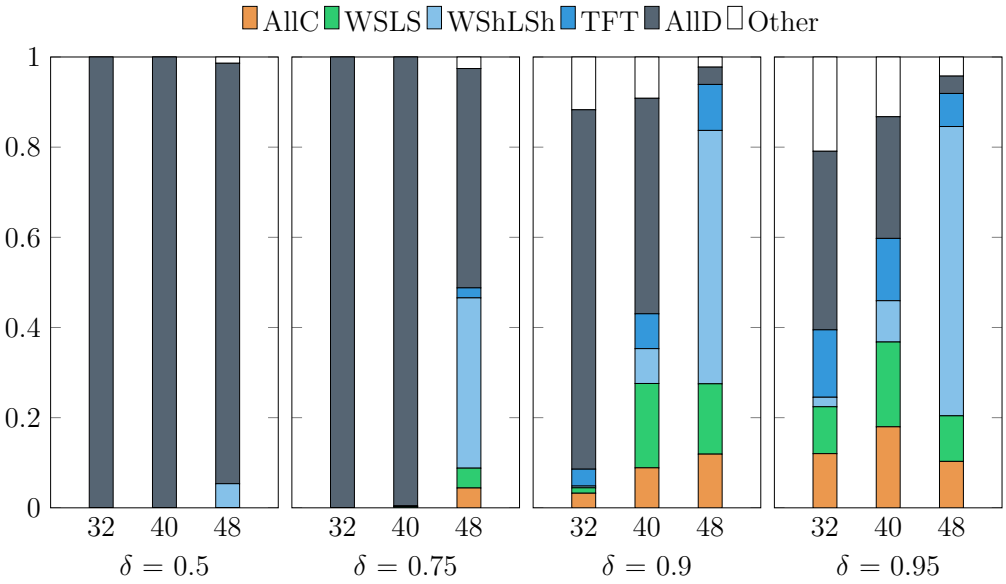

Figure 2: Strategy frequency estimation of algorithmic data by δ-R treatment.

Note: The figure reports the estimates for $k = 1$ for the baseline parameters $\alpha = 0.15$, $\nu = 20$. The numerical values are available in Table S.5 in the online appendix.

human and the algorithmic experiments with different memory lengths, we focus on the SFEM throughout the paper.

5.2 Algorithmic strategies

We first focus on memory-one algorithms ($k = 1$), where technically only memory-one strategies are feasible.[13] Figure 2 shows the results of the SFEM for $k = 1$. Consistent with low cooperation rates (Figure 1), 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 ($\delta = 0.90$, $R = 40$). WShLSh is most common for $R = 48$, and it is even the modal strategy for $\delta \geq 0.9$ and $R = 48$. It is learned so often that the average cooperation rate actually decreases in R. Note that WShLSh never forms a symmetric SPNE. Nevertheless,

[13] We nevertheless use all 25 strategies in the set of possible strategies of the SFEM to keep the analysis comparable to $k > 1$. Online appendix S.3 provides a robustness check.

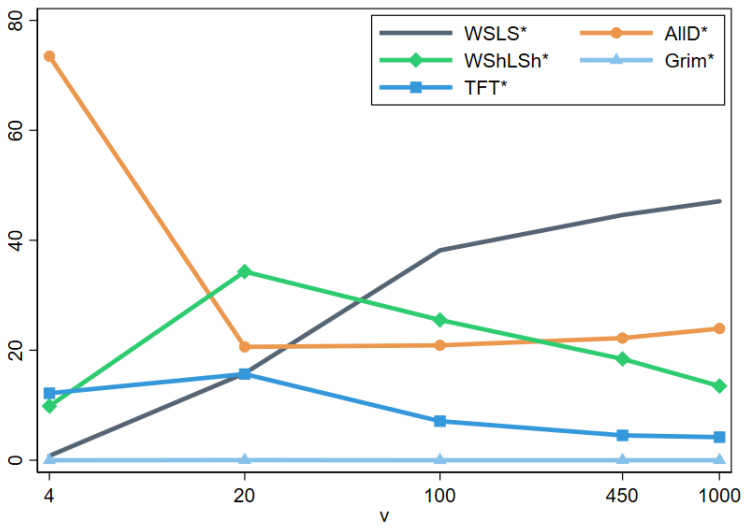

Figure 3: Exploration and the frequency of strategies.

Note: The figure reports the estimates for $\delta \geq 0.9$, $R \geq 40$, $k = 1$, and $\alpha = 0.15$. 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, and 2TF2T. The AllD* family includes AllD, DTFT, DTF2T, and DTF3T. The x-axis is on a log-scale.

the algorithms learn the strategy for large realizations of R. We discuss WShLSh in detail below.

Result 7. *With memory one, the most frequently learned strategies by the algorithms are AllD, WSLS, TFT, and WShLSh.*

Next, we analyze the dependency of the strategies on the learning parameter ν; a higher ν implies more exploration. We combine the various variants of WSLS, WShLSh, TFT, and Grim, into "families" of strategies (as described in Figure 3). Figure 3 shows the dependency of the most frequent families of strategies on the learning parameter ν. The figure reports pooled means of $\delta \in \{0.90, 0.95\}$ and $R \in \{40, 48\}$. A first observation is that the prevalence of AllD drops initially in ν but reaches a constant level of around 22%. Second, WSLS increases monotonically in ν and becomes the modal strategy for $\nu \geq 100$. Third, the WShLSh family is always among the top three strategies in terms of frequency but falls in ν for ν sufficiently high. Lastly, the TFT family accounts for rather consistently between 5 and 15% of the data.

We continue the SFEM when $k > 1$. Table S.4 in the online appendix

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. Figures S.1 and S.2 in online appendix S.2 summarize the strategy estimation comparable to Figure 2. With higher memory, we see that the share of AllC and WSLS decreases while the share of the TFT family increases.

Result 8. *With memory two or three, the most frequently adopted strategies by the algorithms are AllD and those in the TFT and WShLSh families.*

Table S.4 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 9. *Algorithms hardly ever learn grim trigger strategies.*

In online appendix S.3, 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.

The widespread use of certain strategies that do not seem very attractive from a theoretical perspective may be surprising. For instance, for $k = 1$, an algorithm that plays WShLSh cooperates if and only if both players *defected* in the previous round (i.e., the exact opposite of Grim). 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. 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. 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.

In online appendix S.3, we 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.

Table 5: SFEM for human data, TFT and GT aggregated as families of strategies.

Treatment	AllC	AllD	TFT*	GT*
($\delta = 0.90$, $R = 40$)	2.9	5.0	53.3	29.9
($\delta = 0.90$, $R = 48$)	8.1	0.0	69.1	16.6
($\delta = 0.95$, $R = 40$)	0.0	3.3	83.0	12.0
($\delta = 0.95$, $R = 48$)	0.0	3.3	44.4	48.9

5.3 Human strategies

How do the strategies implemented by humans differ from those chosen by the algorithms? In their meta study, Dal Bó and Fréchette (2018) find that, across experiments, humans tend to adopt AllD, TFT and GT.[14] While algorithms learn AllD for low δ, they rarely learn GT. In contrast, algorithms play WSLS and WShLSh, which are rarely observed in human populations.

Table 5 shows the results of the SFEM for the new laboratory experiments we conduct with high discount factors. The table aggregates the TFT and GT families of strategies (see Table S.7 in the online appendix for the full set of strategies.) We note that the TFT and GT strategy families strongly dominate among humans. Together, they account for 83% to 95% of the strategy estimates. This exceeds the share of TFT and GT in the most cooperative games in Dal Bó and Fréchette (2018). Given the high cooperation rates, it is not surprising that AllD plays only a minor role. It may be more surprising that also AllC captures only a minor share. It appears that human subjects learn not to cooperate unconditionally, despite the high cooperation rates.

6 Levels of cooperation

We finally ask about the differences in cooperation rates between humans and algorithms. Figure 4 shows the cooperation rates across all rounds

[14]Recently, 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.

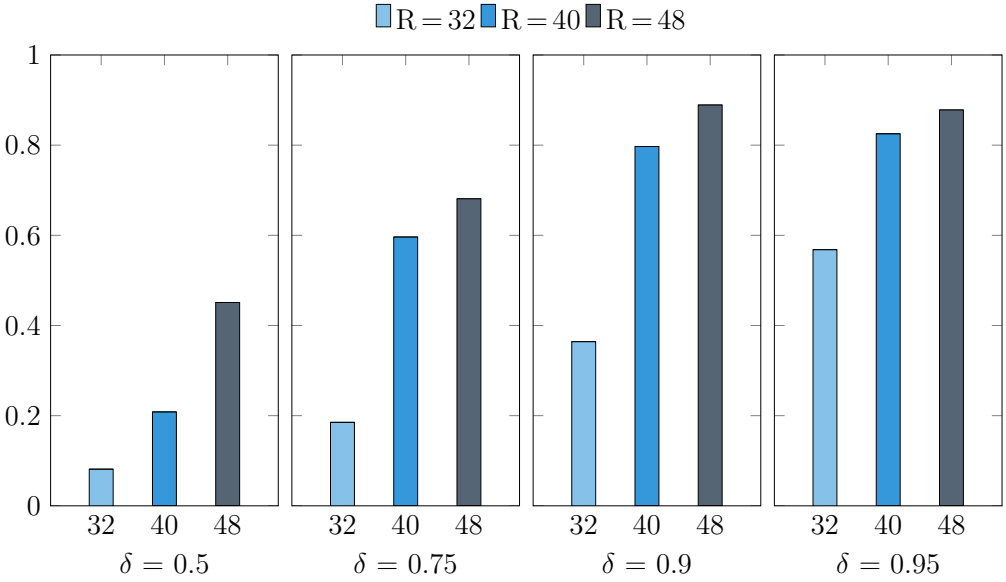

Figure 4: Cooperation rates of humans by δ–R treatment.
Note: The numerical values are available in Table S.6 in the online appendix.

and supergames from the laboratory experiments.[15] The figure includes previous experiments with all $\delta \leq 0.75$ or $R = 32$ treatments, and our experiments with $R \geq 40$ and $\delta \geq 0.90$. Human cooperation is surprisingly high for these δ–R realizations.[16]

The comparison with algorithmic data is non-trivial, since the learning parameters α and ν determine the cooperation level, as do the hard-coded memory length k. Therefore, we compare the human data to two parameterizations. First, our baseline parameterization as summarized in Figure 1. Second, as result 6 suggests that algorithmic cooperation increases in ν, the highest ν in our computational experiments, $\nu = 1,000$ (keeping $\alpha = 0.15$).

For the baseline parameterization ($\alpha = 0.15$ and $\nu = 20$), Table 6 shows the difference between the human and the algorithmic cooperation rates and tests the significance with a two-sided Mann-Whitney-U-test. First, humans cooperate more in all treatments, although the difference is not always statistically significant. A second striking insight is that humans

[15] 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.16.

[16] We note for the humans the same non-monotonicity in the cooperation rate for $\delta = 0.95$ as R increases from 40 to 48.

cooperate to some extent where the algorithm entirely fails to choose C, namely when $\delta = 0.5$ and in treatment ($\delta = 0.75$, $R = 32$). This is true for all levels of memory k. 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 2.75%. We see this as suggestive evidence that humans try to establish cooperation even in environments where it is hard to sustain cooperation. Third, the differences in the cooperation rates are high also for high δ–R treatments. Depending on memory k, the difference may or may not be statistically significant.

Having said that, the conclusion that humans cooperate more than the algorithm is not generally tenable. For the second parametrization ($\nu = 1,000$), Table 6 shows that with the higher ν, algorithms cooperate more on average for high δ–R realizations. 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 3.

Result 10. *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 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 self-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 self-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 contrast, reinforcement learning algorithms must start from scratch and

Table 6: 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	19.05	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	23.35^{*}	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	26.75^{***}	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	49.22^{***}	-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	48.81^{***}	-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	37.53^{***}	-8.39^{***}	-12.13^{***}

Note: This table shows the difference between cooperation rates of humans and algorithms, as well as the significance level of a two-sided Mann-Whitney-U-test. * $p < 0.05$, ** $p < 0.01$, *** $p < 0.001$

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.

7 Cooperation among Large Language Models

As a robustness check, we now report on a computational experiment with an algorithm other than Q-learning. In the new experiment, two Large Language Models (LLMs) play the repeated prisoner's dilemma against each other. As the LLM, we analyze the *gpt-3.5-turbo-0301* model from OpenAI. The details are in online appendix S.4.

We again organize the algorithm's behavior through the lens of the determinants, forms, and levels of cooperation. Our first finding is that the LLM's cooperation rates are *not* responsive to changes in the economic environment as formalized by the discount factor δ and the reward parameter R. The level of cooperation is around 75-80%. Using the SFEM, we find that LLMs mainly adopt cooperative strategies like AllC, TFT, GT, and WSLS.

We take the following two insights away. First, not all algorithms respond to the environment as humans (and Q-learning algorithms) do and as economic theory suggests. Second, LLMs do not match human behavior in strategic situations despite being trained on a large corpus of human-generated text, often with the explicit goal of mimicking human behavior (OpenAI, 2022). In our case, ChatGPT and humans differ in the determinants of cooperation, the adopted strategies, and largely also in the levels of cooperation: ChatGPT cooperates significantly more than humans for low δ–R combinations but statistically indistinguishably for high δ–R pairs.

The failure to respond to the economic environment suggests that LLMs are not up to the task of maximizing long-run payoffs under strategic uncertainty. This is despite our efforts to make the key parameters salient in

our instructions to the algorithm.

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.

In a series of computational experiments on the repeated prisoner's dilemma, we find that the same factors that influence human cooperation also apply to Q-learning algorithms. However, algorithms tend to play different strategies than humans. On the one hand, Q-learning may appear more rational because it is more likely to forgive past defections by reinitiating cooperation. On the other hand, Q-learning frequently converges to strategies that are never part of an equilibrium. While algorithms can be tuned to cooperate more than humans, no universal set of parameters leads to higher cooperation rates across all prisoner's dilemma variants. In particular, Q-learning algorithms tend to cooperate less than humans in environments where cooperation is relatively hard to sustain. Investigating the theoretical drivers for this ambiguity appears to be a fruitful area for future research. Overall, the artificial intelligence studied in this paper does not systematically outperform humans.

Methodologically, we demonstrate that the tools that game-theoretic and experimental research have developed for analyzing human behavior can be fruitfully applied to open the black box of algorithmic behavior. Game-theoretic concepts such as risk dominance (Harsanyi and Selten, 1988) and the size of the basin of 'always defect' (Dal Bó and Fréchette, 2011) explain not only human but also algorithmic cooperation rates. Moreover, the strategy frequency estimation method (Dal Bó and Fréchette, 2011) can approximate the strategies complex algorithms learn. We expect the SFEM to also work well in other settings.

There are also questions about collusion between firms that our research can address. These may need to be taken with a grain of salt, as a two-action dilemma may not fit oligopoly setups with richer action sets. 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 self-learning algorithms. Our results suggest that there are no major differences from human decision-makers. A second important policy question is how to detect collusion by self-learning algorithms (Calvano et al., 2020b). Here the SFEM 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 and Xavier Lambin, "Artificial intelligence: Can seemingly collusive outcomes be avoided?," *Management Science*, 2023.

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

Agapiou, John P, Alexander Sasha Vezhnevets, Edgar A Duéñez-Guzmán, Jayd Matyas, Yiran Mao, Peter Sunehag, Raphael Köster, Udari Madhushani, Kavya Kopparapu, Ramona Comanescu et al., "Melting Pot 2.0," *arXiv preprint arXiv:2211.13746*, 2022.

Aoyagi, Masaki, Guillaume R Fréchette, and Sevgi Yuksel, "Beliefs in repeated games," *ISER DP*, 2022, (1119RR).

_ , V Bhaskar, and Guillaume R Fréchette, "The impact of monitoring in infinitely repeated games: Perfect, public, and private," *American Economic Journal: Microeconomics*, 2019, *11* (1), 1–43.

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.

Assad, Stephanie, Robert Clark, Daniel Ershov, and Lei Xu, "Algorithmic Pricing and Competition: Empirical Evidence from the Ger-

man Retail Gasoline Market," *Journal of Political Economy*, 2023. Forthcoming.

Banchio, Martino and Giacomo Mantegazza, "Adaptive Algorithms and Collusion via Coupling," *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 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.

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, Taylor Weidman, and Alistair Wilson, "Testing Models of Strategic Uncertainty: Equilibrium Selection in Repeated Games," Working paper 2023.

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

Brown, Zach Y. and Alexander MacKay, "Competition in Pricing Algorithms," *American Economic Journal: Microeconomics*, May 2023, *15* (2), 109–156.

Bush, Robert R and Frederick Mosteller, *Stochastic models for learning.*, John Wiley & Sons, Inc., 1955.

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.

Chen, Le, Alan Mislove, and Christo Wilson, "An Empirical Analysis of Algorithmic Pricing on Amazon Marketplace," in "Proceedings of the 25th International Conference on World Wide Web" 2016, pp. 1339–1349.

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.

Dafoe, Allan, Edward Hughes, Yoram Bachrach, Tantum Collins, Kevin R McKee, Joel Z. Leibo, Kate Larson, and Thore Graepel, "Open Problems in Cooperative AI," *arXiv preprint arXiv:2012.08630*, 2020.

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.

Dawid, Herbert, Philipp Harting, and Michael Neugart, "Implications of Algorithmic Wage Setting on Online Labor Platforms: A Simulation-based Analysis," *Working paper*, 2023.

Dolgopolov, Arthur, "Reinforcement Learning in a Prisoner's Dilemma," *Working paper*, 2021.

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.

Erev, Ido and Alvin E Roth, "Predicting how people play games: Reinforcement learning in experimental games with unique, mixed strategy equilibria," *American economic review*, 1998, pp. 848–881.

Ezrachi, Ariel and Maurice E. Stucke, "Sustainable and Unchallenged Algorithmic Tacit Collusion," *Northwestern Journal of Technology and Intellectual Property*, 2020, *17*, 217–260.

Fonseca, Miguel A and Hans-Theo Normann, "Explicit vs. tacit collusion—The impact of communication in oligopoly experiments," *European Economic Review*, 2012, *56* (8), 1759–1772.

Freitag, Andreas, Catherine Roux, and Christian Thöni, "Communication and market sharing: an experiment on the exchange of soft and hard information," *International Economic Review*, 2021, *62* (1), 175–198.

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.

Gill, David and Yaroslav Rosokha, "Beliefs, learning, and personality in the indefinitely repeated prisoner's dilemma," *American Economic Journal: Microeconomics*, 2024. Forthcoming.

Green, Edward J and Robert H Porter, "Noncooperative collusion under imperfect price information," *Econometrica*, 1984, *52*, 87–100.

Harrington, Joseph E, "Developing Competition Law for Collusion by Autonomous Artificial Agents," *Journal of Competition Law & Economics*, 2018, *14*, 331–363.

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

_ **and Andrzej Skrzypacz**, "Private monitoring and communication in cartels: Explaining recent collusive practices," *American Economic Review*, 2011, *101* (6), 2425–2449.

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

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

Hughes, Edward, Joel Z Leibo, Matthew Phillips, Karl Tuyls, Edgar Dueñez-Guzman, Antonio García Castañeda, Iain Dunning, Tina Zhu, Kevin McKee, Raphael Koster et al., "Inequity Aversion Improves Cooperation in Intertemporal Social Dilemmas," *Advances in Neural Information Processing Systems*, 2018, *31*.

Jensen, Benjamin M., Christopher Whyte, and Scott Cuomo, "Algorithms at War: the Promise, Peril, and Limits of Artificial Intelligence," *International Studies Review*, 2020, *22* (3), 526–550.

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.

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

Kuang, Zhufang, Zhihao Ma, Zhe Li, and Xiaoheng Deng, "Cooperative Computation Offloading and Resource Allocation for Delay Minimization in Mobile Edge Computing," *Journal of Systems Architecture*, 2021, *118*, 102167.

Lerer, Adam and Alexander Peysakhovich, "Maintaining Cooperation in Complex Social Dilemmas using Deep Reinforcement Learning," *arXiv preprint arXiv:1707.01068*, 2017.

Martin, Simon and Alexander Rasch, "Collusion by Algorithm: The role of Unobserved Actions," 2022.

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

Miklós-Thal, Jeanine and Catherine Tucker, "Collusion by Algorithm: Does Better Demand Prediction Facilitate Coordination between Sellers?," *Management Science*, 2019, *65* (4), 1552–1561.

Murnighan, J. Keith and Alvin E. Roth, "Expecting Continued Play in Prisoner's Dilemma Games: A Test of Several Models," *Journal of Conflict Resolution*, 1983, *27* (2), 279–300.

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.

OpenAI, "Introducing ChatGPT," https://openai.com/blog/chatgpt 2022.

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.

Siegel, Sidney, "Decision making and learning under varying conditions of reinforcement.," *Annals of the New York Academy of Sciences*, 1961.

Thorndike, Edward L, "Animal intelligence: An experimental study of the associative processes in animals.," *The Psychological Review: Monograph Supplements*, 1898, *2* (4), i.

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, "Learning from Delayed Rewards." PhD dissertation, King's College, Cambridge 1989.

_ **and Peter Dayan**, "Q-Learning," *Machine Learning*, 1992, *8*, 279–292.

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

Wieting, Marcel and Geza Sapi, "Algorithms in the Marketplace: An Empirical Analysis of Automated Pricing in E-Commerce," *Available at SSRN 3945137*, 2021.