Algorithmic Cooperation*

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

July 13, 2023

Abstract

Algorithms play an increasingly important role in economic situations. Often these situations are 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.

Keywords: Artificial Intelligence, Cooperation, Large language models, Q-learning, Repeated Prisoner's Dilemma

JEL Codes: C72, C73, C92, D83

*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 Linz, University of Düsseldorf, BECCLE (Bergen), University of Potsdam, BSE Summer Forum Workshop on Computational and Experimental Economics (Barcelona), University of Münster, and ESA World Meeting (Lyon).

†kasberger@dice.hhu.de. Düsseldorf Institute for Competition Economics (DICE), Heinrich-Heine-Universität Düsseldorf

‡simon.martin@dice.hhu.de. Düsseldorf Institute for Competition Economics (DICE), Heinrich-Heine-Universität Düsseldorf and CESifo Research Affiliate

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

¶werner@mpib-berlin.mpg.de. Center for Humans and Machines at the Max Planck Institute for Human Development, Berlin

1 Introduction

Cooperation 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. Indeed, the folk theorems (Friedman, 1971) prove that cooperation may emerge when the probability of such future encounters is sufficiently high.[1] 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 same 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

[1] The probability of future instances of the stage-game is linked to the discount factor. In laboratory experiments, the subjects' discount factor is induced via the probability of continuing the supergame.

situations present the AI with the possibility of cooperating in social dilemmas, where cooperation can be socially efficient, e.g., in team production or computation offloading (Kuang et al., 2021), or to the detriment of the consumers (Calvano et al., 2020b, Ezrachi and Stucke, 2020, Harrington, 2018, 2022). Either way, it is important to understand how algorithms interact with each other and their 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 behavior of algorithms 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., always defect, 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. It 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 Dyan, 1992) is a form of reinforcement learning, widely studied in economics (Calvano et al., 2020a, Johnson et al., 2023, Klein, 2021), and forms the basis for more sophisticated algorithms. We have three main treatment variables. First, adopting the parameters from the experimental literature, 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 that have previously been studied in the experimental and computational literature.[2] 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. Note that the strategies most frequently played by humans are of memory up to two (Dal Bó and Fréchette, 2018). Lastly, we study how cooperation depends on the algorithm's learning and exploration rate. We do not view these hyperparameters as classic

[2]The algorithmic simulations usually use a discount rate of 0.95 or higher (Lerer and Peysakhovich, 2017, Calvano et al., 2020a, Klein, 2021, Johnson et al., 2023). By contrast, the human lab experiments typically study relatively lower rates. Our research connects both research areas. We study low discount factors uncommon in the artificial intelligence literature and high discount rates hitherto not conducted with humans in the lab.

treatment variables as 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.

Our first major finding is 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 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 (Dal Bó and Fréchette, 2018). We confirm that algorithmic cooperation emerges only if there are cooperative equilibria and that cooperation increases as it becomes risk-dominant.

A significant difference between humans and algorithms lies in the strategies they adopt (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 our strategy frequency estimation suggests that algorithms also play tit-for-tat, they hardly ever select 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. LLM 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 Section 7 for references). The algorithm's propensity to cooperate is similar to the one of humans for medium discount rates and reward parameters. No-

tably, 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. Roth and Murnighan (1978) and Murnighan and Roth (1983) were the first to implement infinitely repeated prisoner's dilemma games in the lab by imposing a random move that determines the end of a supergame. Dal Bó (2005) first implemented several supergames, each indefinitely repeated. The meta-study of Dal Bó and Fréchette (2018) summarizes the subsequent literature on the determinants, forms, and levels of cooperation.[3] Throughout the paper, we draw upon the insights and methods of this literature to form hypotheses about algorithmic behavior and analyze observed behavior.

Axelrod (1984) provides an early computational study on the performance of strategies from a fixed set of strategies in the infinitely repeated prisoner's dilemma. In contrast, we use Q-learning to determine the strategies. The economics literature on self-learning algorithms has so far largely focused on cooperation in the sense of (socially undesirable) anti-competitive collusion in oligopoly games.[4] 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), which allows us to get a more complete understanding of on-path and off-path behavior of Q-learning agents. In particular, we use the strategy frequency estimation method (Dal Bó and Fréchette, 2011) to study the strategies algorithms adopt and how these strategies depend on the game parameters. Moreover, our setting allows us to draw upon a rich set of experimental

[3]Embrey et al. (2018) provide a similar analysis for finitely repeated games, as does Mengel (2018) for one-shot and finitely-repeated prisoner's dilemmas. Bigoni et al. (2015) compare repeated prisoner's dilemma games in continuous time with indefinite duration to those with finite length.

[4]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*.

studies to compare human with algorithmic behavior.

In simultaneous and independent work, 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 by varying the number of players in prisoner's dilemma experiments using humans. In an extension, they compare these findings with those of Q-learning agents. We systematically investigate the determinants of cooperation proposed by the experimental literature and also study the algorithm's memory as a determinant for cooperation. Banchio and Mantegazza (2022) provide theoretical insights into why independent Q-learning agents often learn symmetric strategies. Dolgopolov (2021) shows that Q-learning players may cooperate even with zero memory in the repeated prisoner's dilemma. Barfuss and Meylahn (2022) focus on the relevance of noise in sustaining cooperative outcomes for reinforcement learning algorithms.

The computer science literature also studies artificial intelligence in strategic situations (Dafoe et al., 2020). The focus is often on designing algorithms that achieve "better performance" than previous algorithms and on the technical mechanisms that are responsible for the algorithm's success in a broad set of games (Crandall and Goodrich, 2011, Lerer and Peysakhovich, 2017, Crandall et al., 2018). Other studies explore cooperation between algorithms in more complex, video-game-like settings going beyond classical game theoretical models (Hughes et al., 2018, Agapiou et al., 2022). Instead of designing algorithms that perform well in various environments, we analyze a fundamental reinforcement learning algorithm that has been studied elsewhere in economics. Our focus is on how methods from game theory and experimental economics can be used to describe algorithmic 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.

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

2 Economic environment and hypotheses

2.1 Basic setup

We study the infinitely repeated Prisoner's Dilemma (PD) with perfect monitoring. There are two players who repeatedly play the stage-game PD and discount future payoffs with the common discount factor δ. In the stage game, each player either cooperates (C) or defects (D). Hence, the set of stage-game actions is $\{C, D\}$ for each player. Table 1 shows two payoff matrices of the stage game: the normalized payoffs and the payoffs we implement in our experiments.[5] 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 strategies of the infinitely repeated PD that have finite memory. 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

[5] See Dal Bó and Fréchette (2018) for how to obtain the normalized payoffs.

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). 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.[6] The first is 'always defect' (AllD), which prescribes playing D, the strictly dominant action in the stage game, in any state. Both players playing AllD is for any discount factor δ a subgame perfect Nash equilibrium (SPNE) of the repeated PD. A similar strategy, 'always cooperate' (AllC), prescribes to play C for any behavior of the previous round but 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. The minimal memory to implement TFT is one. TFT is generically not subgame perfect in the repeated game. 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.[7] More forgiving than GT is the strategy 'win-stay, lose-shift' (WSLS) (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 is subgame perfect and, unlike TFT and GT, can correct erroneous defections.

We also consider memory-k, $k > 1$, strategies: A trigger strategy with two periods of punishment (T2), for example. There are also versions of TFT and WSLS with memory two, such as TF2T (play C unless the rival played D in either of the last two periods) or WSLS with two rounds of punishment. Similar extensions for memory three are possible (T3 or TF3T, say).

2.2 The self-learning algorithm

We study how Q-learning algorithms play the repeated Prisoner's Dilemma in a sequence of computational experiments that are described below. Q-learning is a popular reinforcement learning algorithm designed to solve Markov decision pro-

[6] Generally, when we say that a strategy has a certain memory, we mean the minimal memory needed to implement the strategy.

[7] Note that our definition of GT requires a minimal memory of only the previous round.

cesses (Watkins, 1989, Watkins and Dyan, 1992). We focus on this algorithm as its ideas are at the core of more advanced (deep) reinforcement learning algorithms that outperform humans at the board game of Go or Atari Video games (Mnih et al., 2013, Silver et al., 2016). Hence, Q-learning is an algorithm that has particular real-world relevance. Furthermore, Q-learning has the advantage of being tractable, and its results are at least partially interpretable, as we can directly observe the resulting strategies. Moreover, recent work by Calvano et al. (2020a) and Klein (2021) has shown the potential of Q-learning algorithms in strategic economic situations.

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 respective Q-matrices to choose actions and update their approximations of the long-run payoffs. For a given Q-matrix, the optimal strategy is just the row-wise maximizer.

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

$$Q_{t+1}(s, a) = (1 - \alpha)Q_t(s, a) + \alpha \left(\pi(s, a) + \delta \max_{a' \in \{C,D\}} Q_t(s', a') \right).$$

The new value is a convex combination of the old and the actual stage-game payoff π plus the best possible guessed long-run payoff in the next state. 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' chosen actions in period t. Note that each player updates only a single cell at each point in time.

Besides the learning rate α, a key parameter is the exploration probability ε_t. Following common practices 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. Another important parameter is ν, which is implied by α, β, and k, and 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 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.[8]

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

[8]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.

gains from defection. A larger reward payoff generally makes cooperation more attractive. Our first hypothesis relates to these determinants of cooperation.

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

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

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

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 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., 2023, 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). 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 it still optimal for a player to play AllD. We use $\underline{p}$ to denote sizeBAD. To find the formula for $\underline{p}$, compare the expected payoff from

playing GT

$$p \cdot \frac{1}{1-\delta} + (1-p) \cdot (-\ell),$$

to the expected payoff from AllD

$$p \cdot (1+g) + (1-p) \cdot 0.$$

The expected payoff from selecting GT is (weakly) larger than that of the AllD strategy 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. Dal Bó and Fréchette (2011) interpret sizeBAD as a measure for how robust cooperation is to strategic uncertainty. Dal Bó and Fréchette (2018) find that cooperation rates decrease in $\underline{p}$ across experiments. We hypothesize that self-learning algorithms also cooperate more when cooperation is more robust to strategic uncertainty.

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 PD if in the hypothetical coordination game consisting of AllD and GT, the cooperative strategy GT is risk dominant (RD). GT is risk dominant 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). We expect that risk dominance also plays a role for the cooperation rates of self-learning algorithms.

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. 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).

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 some experimental data was available and conducted additional experiments with human subjects ourselves.

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. Following Dal Bó and Fréchette (2011), the variation in the first two dimensions are the reward payoff of joint cooperation, R, and the discount factor δ. In the third dimension, we study the hard-coded variation in memory, $k \in \{1, 2, 3\}$ (this, of course, applies to the algorithmic experiments only). Specifically, we consider $R \in \{32, 40, 48\}$ and $\delta \in \{0.50, 0.75, 0.90, 0.95\}$, motivated by configurations also used in 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.[9] In the subsequent playing stage, the algorithms' initial actions are the optimal actions in the round of convergence. After that, they play according to the learned strategies.

Following our experimental design, the hard-coded memory is at most three.[10] 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.[11] 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 was identical to Dal Bó and Fréchette (2011), Romero and Rosokha (2018), Ghidoni and Suetens (2022), and Kartal and Müller

[9]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.

[10]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.

[11]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.

(2021). The instructions were likewise identical to these experiments. In order to compare algorithmic cooperation with humans 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, which have not yet been studied in the lab. For the other treatments in Table 2, ample lab data exist already (see 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 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/. We had a total of 240 participants. Participants earned 22.54 euros on average.

4 Determinants of cooperation

We now discuss the cooperation rates of the algorithms and their determinants. Figure 1 shows the average cooperation rate of the algorithms for each δ–R treatment, averaged over all k. 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.

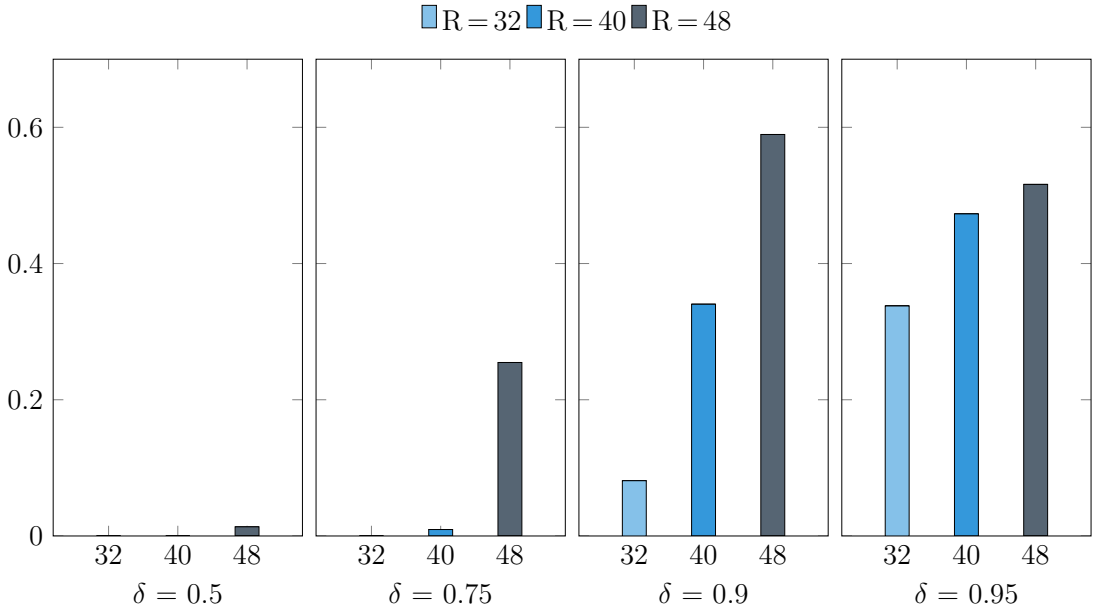

Figure 1: Cooperation rates of algorithms by δ–R treatment.

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 8 in the appendix.

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

To investigate the role of memory and the other parameters on cooperation, we run several regressions with the cooperation rate (as depicted in Figure 1) for our baseline parameterization ($\alpha = 0.15$, $\nu = 20$) as the dependent variable. The regressors are the variables used to explain human cooperation rates, and where we hypothesize that they also shape algorithmic cooperation rates.

Table 3 summarizes the analysis of the determinants of algorithmic cooperation. Focusing on regression (1), the regression 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 8 in the 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.

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, * $p < 0.05$, ** $p < 0.01$, *** $p < 0.001$

Result 2. *The effect of memory on cooperation of self-learning algorithms is ambiguous in general and negative on average.*

We now ask how cooperation rates are affected when cooperative equilibria exist. Recall that we formalize this test using a binary variable that takes the value of one if the discount factor δ exceeds δ^{SPNE}, and a value of zero otherwise. Table 2 shows for which treatments this condition is met. Going back to the average 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 applies to how humans play the repeated prisoner's dilemma.

Result 3. *A necessary but not sufficient condition for self-learning algorithms to cooperate is that 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, 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 on average 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.

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, * $p < 0.05$, ** $p < 0.01$, *** $p < 0.001$

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. Specifically, we use the state-of-the-art strategy frequency estimation method (SFEM). We introduce the method in Section 5.1, where we also explain how we apply it in our setting. In Section 5.2, we discuss the estimation results and the strategies that algorithms learn. Finally, in Section 5.3, we use SFEM to estimate the strategies of humans and examine how algorithmic and human strategies differ for the various environments we consider.

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, 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 (TFT), grim trigger (GT), win-stay-lose-shift (WSLS), 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.[12] 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.[13] 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.3 in the online appendix shows the differences between the SFEM and the "manual"

[12]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.

[13]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 $\lambda = 1$.

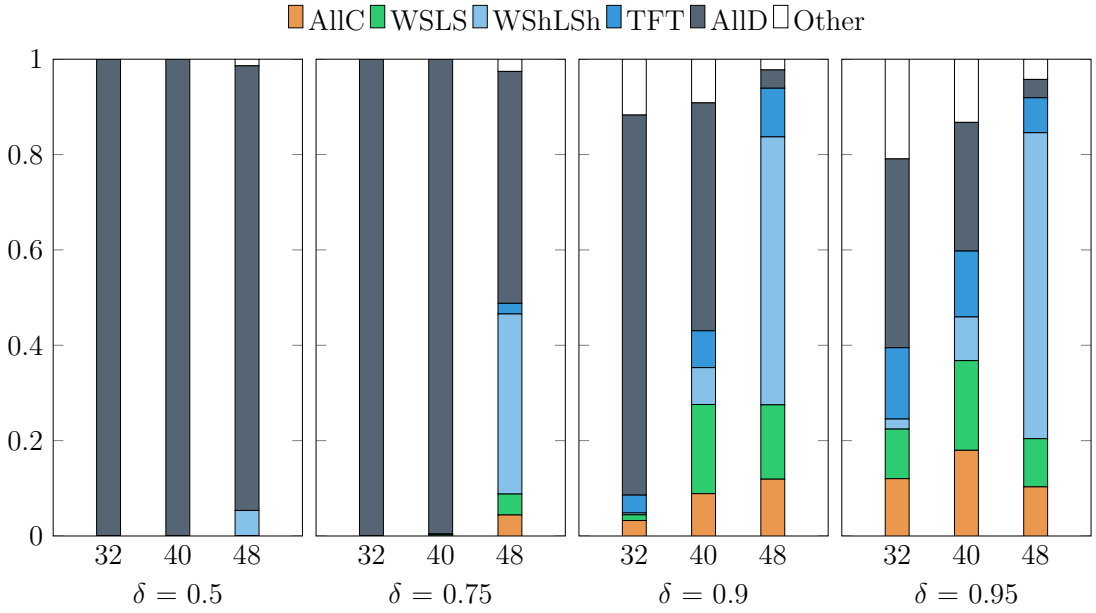

Figure 2: Strategy frequency estimation of algorithmic data by δ-R treatment.

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.3 in the online appendix.

classification. We find only few and minor differences, which demonstrates that the SFEM can indeed be applied to algorithmic decision-making in strategic situations. Importantly, the method could also be applied to completely different algorithms such as Large Language Models or algorithms that numerically approximate the Q-matrix. We highlight this by using the SFEM to analyze a Large Language Model in Section 7. To keep the method of estimating the proportions of the strategies the same for the 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.[14] 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

[14]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.2 provides a robustness check.

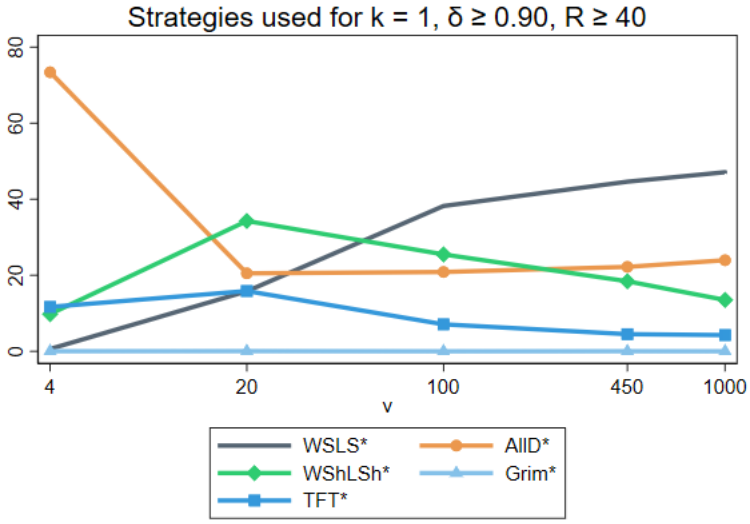

Figure 3: Exploration and the frequency of strategies.

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.

$\delta \geq 0.9$ 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, 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 with the SFEM when $k > 1$. Table 9 in the appendix shows the results. Now that memory-two and -three strategies are feasible for the algorithm,

we indeed observe a substantial share of TF2T, especially for $k = 2$ and some TF3T. With $k = 3$, also 2TF2T. Likewise, 2WSLS has a significant share for $k \in \{2, 3\}$. When we allow for memory $k > 1$ strategies, the memory-one strategy DTFT achieves a sizeable share, as does its memory-two counterpart DTF2T. Similarly, the memory-one strategy DCAlt becomes relevant. Figures 6 and 7 in the appendix 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 9 in the appendix also shows that algorithms hardly ever adopt strategies from the grim trigger family. For almost all parameter constellations, grim trigger is never played. If it is played, its share is estimated to be 0.1%.

Result 9. *Algorithms hardly ever learn grim trigger strategies.*

To understand the influence of the different parameters on the adopted strategies, we classify the strategies into a few simple categories following 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 exclusive, e.g., TFT is both *cooperative* and *forgiving*.

In Table 5, we analyze the incidence of strategy categories, using the same set of determinants as above. Recall that average cooperation was found to increase in δ, R and k. Zooming in on the set of strategies that drive these outcomes, we find that for δ and R it is a combination of more cooperative, more lenient, and more forgiving strategies (all of which become more likely as δ and R increase). For both δ and R, the effect is most pronounced for cooperative and forgiving strategies. Similarly, a higher learning rate α leads to lower cooperation; mostly, because forgiving and cooperative strategies become less likely. However, this analysis reveals an important difference for the effect of memory length k. Although higher memory length also increases average cooperation, we see in Table 5 that this effect is *not* uniformly driven by cooperative, lenient, and forgiving strategies. Indeed, there is no statistically significant effect of k for cooperative and lenient strategies. The effect of k is only statistically significant for forgiving strategies. Therefore, we conclude that the increasing emergence of cooperative and forgiving strategies leads to higher average cooperation as δ and R increase. When k increases, mostly the increase of forgiving strategies lead to higher average cooperation rates.

Table 5: 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

* $p < 0.05$, ** $p < 0.01$, *** $p < 0.001$

Regarding the specific strategies found, the widespread use of certain strategies that do not seem very appealing 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 $((C, C), (D, D))$ cycle, that is, alternating between mutual cooperation and mutual defection. Suppose players are in state (D, D), 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 (D, D) again in two rounds.

The intuition behind this is as follows. Suppose both players' Q-matrices are currently such that D is played in all four states. We focus on the state DD, with associated values $Q(DD, C)$ and $Q(DD, D)$ of subsequent cooperation and defection, respectively. Since both players defect in state (D, D), each player continues to receive 25 in that state. Depending on how $Q(DD, C)$ was initialized, $Q(DD, D)$ may eventually fall below $Q(DD, C)$, in which case the player begins switching to C in state (D, D). If this switch occurs around the same time for the other player, both players cooperate in state (D, D) and keep getting positive feedback (payoff R) by doing so, which reinforces this action. If exploration eventually stops, both players have 'learned' that cooperation is the optimal action in

state (D, D), resulting in WShLSh.

To explore this line of reasoning in more detail, we investigate the joint distribution of strategies in Table 10 in the appendix. Specifically, we investigate both the equilibrium cycle length of the strategies at convergence (columns (1) and (2)), and the fraction of states on the equilibrium path where both players play the same actions (columns (3) and (4)), separately for our main (α, ν) specification (columns (1) and (3)) as well as for all specifications (columns (2) and (4)). We find that exactly the same factors that positively influence cooperate rates (namely δ, R and higher memory length k), also result in outcomes that involve more nodes on the equilibrium path. 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. For algorithms with higher memory length k, it is also more likely that algorithms converge to a state that involves at least partial cooperation, increasing the average cooperation.

This claim finds additional support when we examine the fraction of states on the equilibrium path where both players play the same actions. This fraction is negatively affected by δ and k, i.e., the exact opposite effect of the average cooperation rates. This again implies that higher cooperation rates are driven by a higher propensity for players to play opposing actions in certain states.

Note that this idea only holds in the non-stationary setting of competing against another player whose action may also change over time. Obviously, 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.[15]

5.3 Human strategies

In their meta study, Dal Bó and Fréchette (2018) find that, across experiments, humans tend to adopt AllD, TFT and GT. While algorithms learn AllD for low δ, they rarely learn GT. In contrast, algorithms play WSLS and WShLSh, which are a strategies that are not often observed in human players.

Table 6 shows the results of the SFEM for the new laboratory experiments we conduct with high discount factors. The table aggregates the TFT and GT

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

Table 6: SFEM for human data, TFT and GT aggregated as families of stratgies.

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

families of strategies (see Table 12 in the 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. More surprising is 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

Now that we know that the determinants of human and algorithmic cooperation are largely the same, but that humans and algorithms tend to adopt different strategies, we finally ask about the differences in cooperation rates. Figure 4 shows the cooperation rates in human laboratory experiments (based on data from sources summarized in Table S.2 in the online appendix). The figure includes previous experiments with all $\delta \leq 0.75$ treatments and all $R = 32$ realizations. We ran the remaining cell variants where $R \geq 40$ 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. And second, as our analysis in Section 4 suggests that algorithmic cooperation increases in ν (result 6), 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 7 shows the difference between the human and the algorithmic cooperation rates and tests the

[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.

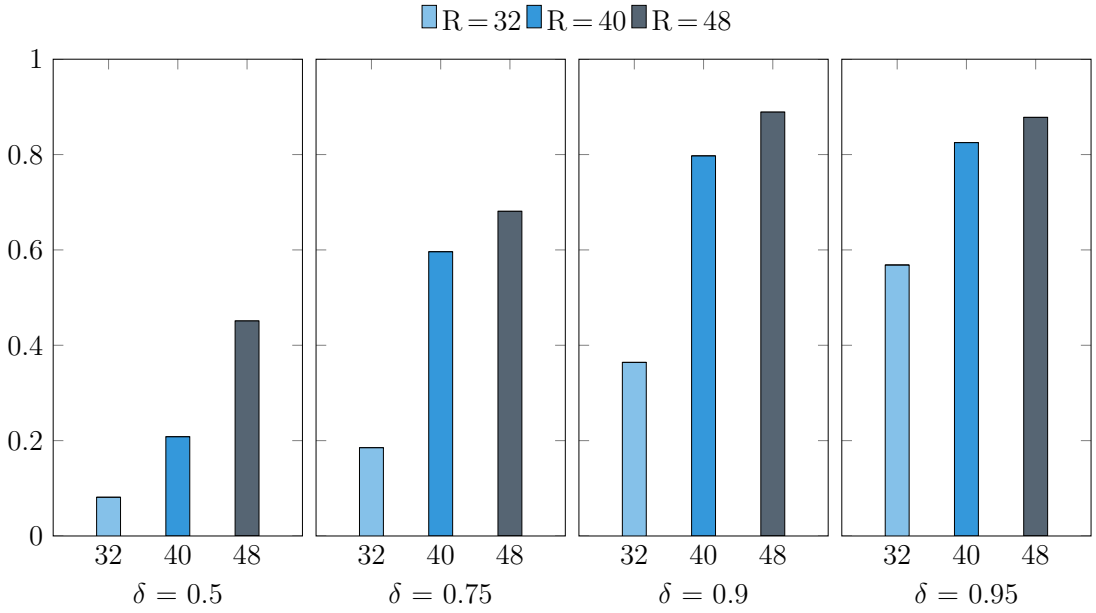

Figure 4: Cooperation rates of humans by δ–R treatment.

Note: The numerical values are available in Table 11 in the appendix.

significance with a two-sided Mann-Whitney-U-test. The first observation is that humans cooperate more in all treatments, although the difference is not always statistically significant. A second striking insight is that humans cooperate to some extent where the algorithm entirely fails to choose C, namely when $\delta = 0.5$ and in treatment ($\delta = 0.75$, $R = 32$). 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 ($\alpha = 0.15$, $\nu = 1,000$), Table 7 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 as follows.

Table 7: 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. * $p < 0.05$, ** $p < 0.01$, *** $p < 0.001$

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.*

We note that "learning to cooperate" means quite different things to humans and to algorithms. To begin with, humans do not have a parameter that determines the exploration of new strategies. Humans can learn within a supergame (where the discount factor largely influences the value of such experimentation) and across supergames. Algorithmic learning, on the other hand, is strongly influenced by the exploration parameter, an exogenous parameter. Second, "learning to cooperate" and "cooperation" itself are separate issues for self-learning algorithms (learning and playing phases). For humans, the data comprises both phases.

Where humans and self-learning algorithms differ strongly is in the learning phase. Humans seem to only need a small number of rounds or a few supergames to succeed. They are, however, proponents of a generally cooperative species. In contrast, reinforcement learning algorithms need to start from scratch and it takes them an enormous number of rounds to learn. These algorithms are backward-looking, while humans can be forward-looking. Humans can interpret each other, play deliberately, and infer the intentions of their opponents. It seems that these differences between humans and algorithms can explain some of our results; for example, why humans cooperate more when cooperation is relatively hard to sustain. While the discount factor and reward parameters affect the "learning to cooperate" of both humans and algorithms, the differences in the nature of learning lead to different outcomes when it comes to playing a supergame, such as the more forgiving nature of the strategies employed by the algorithm.

7 Cooperation among Large Language Models

The paper has so far focused on Q-learning algorithms. These reinforcement learning algorithms are relevant because they seek to maximize long-run discounted payoffs and produce a strategy in the game-theoretic sense. Moreover, they are the building block of more complex algorithms while still being relatively interpretable from an economic perspective. As such, they strike a balance between maintaining a high degree of external validity and providing an abstraction from more complex algorithms.

In this section, we turn our attention to a different class of algorithms that are more representative of those that humans interact with on a daily basis: Large

Language Models (LLMs). LLMs are trained on a large corpus of human-generated text, often with the explicit goal of mimicking human behavior (OpenAI, 2022, 2023). They can generate text that is often indistinguishable from text written by humans (Köbis and Mossink, 2021, Clark et al., 2021). Moreover, their practical utility has been established across a range of applications. These include assisting in creating text, enhancing web search capabilities, and serving as coding assistants (see Bubeck et al., 2023, for an overview). A popular LLM is ChatGPT, which offers a chat window to interact with the algorithm. As of June 2023, ChatGPT is claimed to have over 100m users,[17] and the tool's release created a large media echo.[18]

Recent papers by Horton (2023) and Grossmann et al. (2023) argue for the relevance of LLMs for research in the experimental social sciences. Due to the training process of the algorithms, they can condition their output on a wide range of human knowledge and are trained to respond in a way similar to human reasoning. As a result, they might be a valuable model of human behavior. Moreover, humans regularly use them as advisors, which may include strategic situations like price setting, negotiations, or everyday interactions with colleagues. Understanding the behavior of LLMs in (strategic) games can thus potentially be used to gain a better understanding of humans, but also to understand how advice from those models to humans might affect outcomes. Recent studies in economics and computer science use this idea to show parallels between the behavior of LLMs and humans in finitely repeated (Akata et al., 2023, Guo, 2023), sequential (Bauer et al., 2023) or one-shot games (Horton, 2023, Brookins and DeBacker, 2023).

Building on the experimental design from the previous sections, we consider the interaction between two LLMs in the infinitely repeated prisoner's dilemma. We focus on the *gpt-3.5-turbo-0301* model from OpenAI, which is, at the time of writing, one of the most advanced language models. For each treatment cell in Table 2, we simulate 250 independent conversations between two LLM agents. At the beginning of each simulation, we provide the agent with the instructions for the game. The instructions follow a style similar to those provided to humans and use current best practices for instructing LLMs, such as instructing the agent to think and plan carefully before making a final response (see, for instance, Wei et al., 2022). Those tactics, known as prompt engineering, usually improve the model performance and increase the likelihood that the agent understands the strategic

[17] https://www.demandsage.com/chatgpt-statistics/, last accessed June 23, 2023.
[18] See, for instance, nytimes.com, cnn.com, bbc.co.uk and economist.com.

nature of the game.[19] Our instructions are similar to those used by Guo (2023).

LLMs do not have an inherent objective when playing the game. To mimic the goal of humans and Q-learning algorithms in the experiment, we explain to the LLM that its objective is to maximize its total payoff over the entire experiment. We describe this to the agents in the initial instructions and when providing the agent intermediate feedback during the simulations.

We make it clear to the model that the game may end randomly after each round with a certain probability determined by δ. Similar to human experiments, the instructions are worded neutrally to avoid any direct association in the wording with the classic prisoner's dilemma framing. The model has complete knowledge of the payoff matrix of the game. The complete instructions are in Online Appendix S.7.

After correctly answering a series of control questions, both models choose an action for the current round. Afterward, we inform each agent of the current round's payoff, their current total payoff, their previous choice, and the other agent's choice. We repeat this procedure for ten rounds in each simulation.[20] In comparison to Q-learning algorithms, the LLMs are not limited in their memory but can observe the whole conversation and may thus condition their choices on the entire history of the game.

Besides the choice of the model itself, one of the most critical parameters influencing the behavior of LLM is the so-called "temperature" of the model. It determines the degree of randomness in the agent's answers. A higher temperature is often associated with more creative responses, while a lower temperature implies more deterministic behavior. Following other recent work considering LLMs in economic environments, we keep it at the default value of 1 to allow for some creativity in their responses (see, for example, Brand et al. 2023 or Guo 2023).

Figure 5 shows the average cooperation rate for each treatment for the LLMs. The language models are considerably more cooperative than both humans and Q-learning algorithms when the values of δ and R are small. The cooperation rates are similar to humans for high values of δ and R. Importantly, the level of cooperation seems to be largely independent of δ and R. While the cooperation rates for humans and classical reinforcement learning algorithms differ strongly for

[19] https://platform.openai.com/docs/guides/gpt-best-practices/give-gpts-time-to-think, last accessed June 23, 2023.

[20] Implementing the simulations with actual random stopping is not feasible, as conversations may become too long for the model to handle. Importantly, the LLMs do not know that the game will end after this fixed number of rounds but believe they are playing an infinitely repeated game.

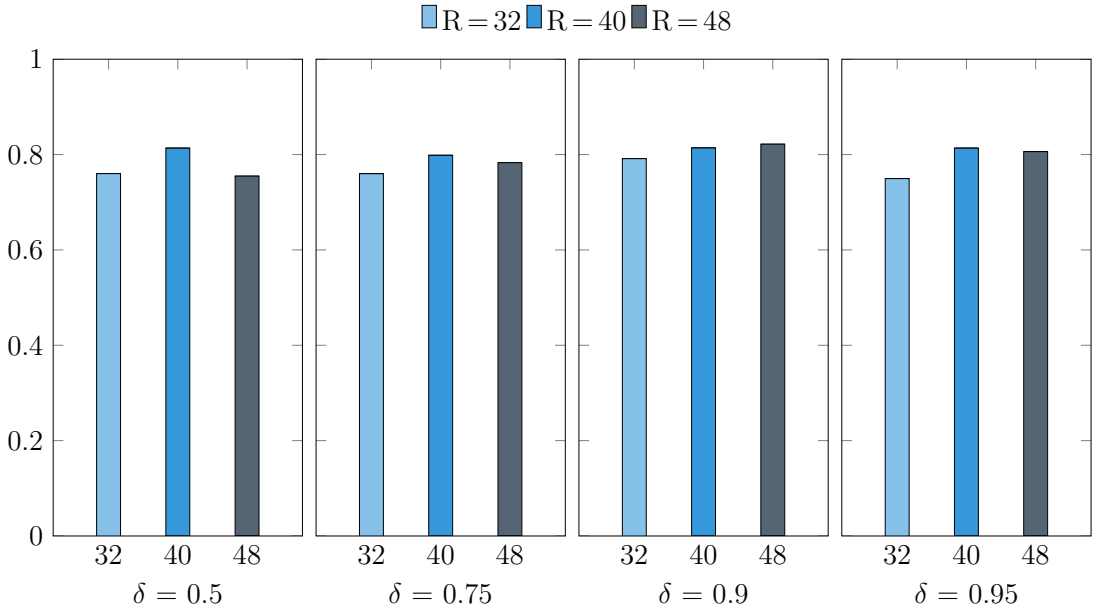

Figure 5: Cooperation rates of ChatGPT by δ–R treatment.

Note: The numerical values are available in Table S.4 in the online appendix

different parameterizations of the environment, they are seemingly irrelevant for the large language model considered here. This finding is particularly surprising given that LLMs have been trained on human text data, suggesting that their behavior in this domain would resemble that of humans. Moreover, in our prompts we remind ChatGPT of its objective and δ in each round.

Using the strategy frequency estimation method, we estimate the strategies that the LLMs use. The results are shown in Table 14 in the appendix. LLMs use mostly cooperative strategies like AllC and TFT. Additionally, a significant portion of the data can be attributed to GT and WSLS. It is worth noting that the usage of GT resembles human behavior, whereas the use of WSLS is more akin to the strategies employed by Q-learning algorithms.

Our additional results on the interaction between LLMs highlight that these models can be studied with indefinitely repeated games. Despite fundamental differences in the learning behavior, we observe parallels in the cooperation rates for specific parameterizations of the environment and in the strategy choices of these models. Notably, however, the behavior of these models is less influenced by the environmental parameters that are important determinants of human and algorithmic cooperation, such as the discount rate or the payoff from mutual cooperation. As tools like ChatGPT continue to proliferate in various domains, gaining a deeper understanding of their behavior will be increasingly crucial for understanding their impact on society.

8 Conclusion

Comprehensive knowledge of how algorithms work is essential (Rahwan et al., 2019) as artificial intelligence is increasingly used in strategic situations: Humans ask AI for advice, delegate their choices to the algorithm, and use the AI as a mediator in strategic situations involving other humans. Our work aims to improve the understanding of algorithmic cooperation.

In a series of computational experiments on the infinitely repeated prisoner's dilemma, we find that the same factors that increase human cooperation largely also determine algorithmic cooperation rates. While this is true for Q-learning agents, it is not true for Large Language Models such as ChatGPT, suggesting important differences between the type of algorithm used for advice in strategic situations. A second finding is that algorithms tend to play different strategies than humans. For example, Q-learning adopts strategies that try to restart cooperation after mutual defections more frequently than humans and LLMs. Another finding is that no decision-maker "class" cooperates uniformly more than other classes. However, Q-learning algorithms tend to cooperate less than humans in environments in which cooperation is relatively hard to sustain.

The attention that research on the behavior of artificial intelligence in strategic situations receives is probably due to the general dynamic development of the field of AI and its substantial future potential. Our results point to some of this potential (e.g., the sometimes spectacularly high cooperation rates), but they also highlight limitations, at least of the current state of the art. For example, the algorithm generally does not cooperate at higher levels than humans can achieve. While the learning and exploitation parameters can be fine-tuned to make the algorithm cooperate better than humans, we find that there is no set of parameters that universally improves cooperation across all the prisoner's dilemma variants we study: Parameters that improve cooperation in one game may reduce it in different games. Investigating the theoretical drivers for this ambiguity appears to be a fruitful area for future research. A similarly sobering conclusion concerns strategies. While the algorithm often plays more rationally than humans (e.g., more forgiving strategies), it can also converge to strategies that are never an equilibrium. Overall, current artificial intelligence does not seem to systematically outperform humans in environments prone to cooperation.

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.

Appendix

A Additional tables

Table 8: Average cooperation by treatment, $\nu = 20$

k	δ	$R = 32$	$R = 40$	$R = 48$
1	0.50	0.00^{***}	0.00^{***}	2.75^{***}
	0.75	0.00^{***}	0.35^{***}	31.07^{***}
	0.90	8.97^{***}	39.12	66.05
	0.95	37.77	55.35	60.03^{*}
2	0.50	0.00^{***}	0.00^{***}	0.29^{***}
	0.75	0.00^{***}	1.98^{***}	23.33^{***}
	0.90	10.84^{***}	41.69^{**}	58.63^{*}
	0.95	33.47^{*}	45.29^{***}	48.70^{***}
3	0.50	0.00^{***}	0.00^{***}	1.01^{***}
	0.75	0.00^{***}	0.51^{***}	22.02^{***}
	0.90	4.54^{***}	21.26^{***}	52.17^{***}
	0.95	30.07^{***}	41.23^{***}	46.13^{***}

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 11). * $p < 0.05$, ** $p < 0.01$, *** $p < 0.001$

Table 9: Distribution of strategies by k and experiment. Strategies which are never above 5% are omitted.

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

Table 10: 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
* $p < 0.05$, ** $p < 0.01$, *** $p < 0.001$

Table 11: Average cooperation by humans

δ	$R = 32$	$R = 40$	$R = 48$
0.50	8.15	20.82	45.13
0.75	18.54	59.63	68.15
0.90	36.42	79.73	88.91
0.95	56.82	82.51	87.84

Note: Average cooperation rates of humans, based on data from sources summarized in Table S.2

Table 12: Distribution of strategies used by humans. Strategies which are never above 5% are omitted.

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

Table 13: Difference human vs. ChatGPT cooperation rates

δ	$R = 32$	$R = 40$	$R = 48$
0.50	-68.19^{***}	-60.52^{***}	-30.35^{***}
0.75	-57.44^{***}	-20.21^{*}	-10.15^{*}
0.90	-42.72^{***}	-1.65^{*}	6.73^{*}
0.95	-18.14	1.17^{*}	7.22

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

Table 14: Distribution of strategies used by ChatGPT. Strategies which are never above 5% are omitted.

δ	R	AllC	AllD	TFT	2TFT	T2	Grim	WSLS	2WSLS	3WSLS	σ
0.50	32	20.1^{***}	5.3^{***}	20.8^{***}	12.7^{*}	2.4	4.5	14.6^{**}	4.2	4.1	0.95
	40	13.3^{**}	4.3^{***}	27.3^{***}	0.2	0.0	22.3^{*}	17.6^{***}	4.6	5.9	0.96
	48	14.8^{***}	4.6^{**}	22.0^{**}	0.0	3.5	20.8^{**}	17.0^{***}	6.0	0.7	0.94
0.75	32	8.0^{*}	6.3^{***}	16.0^{**}	0.0	3.8	18.6^{*}	10.8^{**}	7.3	20.5^{*}	0.94
	40	14.5^{**}	4.5^{***}	17.3^{**}	0.0	1.1	16.5^{*}	19.8^{***}	8.7	4.4	0.96
	48	12.9^{**}	4.5^{***}	18.0^{***}	2.4	2.8	13.1	23.9^{***}	5.2	9.9	0.96
0.90	32	7.8^{*}	5.2^{***}	18.9^{**}	6.6	0.0	23.3^{*}	17.8^{***}	10.6^{*}	0.0	0.96
	40	15.2^{**}	7.9^{***}	16.6^{**}	0.0	0.0	26.4^{***}	15.3^{**}	8.1	5.5	0.97
	48	18.8^{***}	4.4^{***}	27.1^{***}	2.3	8.4^{*}	28.1^{**}	3.5	4.4	0.0	0.97
0.95	32	16.5^{***}	7.7^{***}	22.4^{***}	0.0	5.3	17.9^{**}	5.8	0.0	9.9^{*}	0.96
	40	8.1^{*}	6.0^{***}	25.4^{***}	0.0	0.0	32.2^{***}	13.1^{**}	4.7	4.0	0.97
	48	15.6^{***}	8.3^{***}	12.7^{*}	0.0	10.6^{*}	27.1^{***}	18.9^{***}	0.0	2.9	0.97

B Additional figures

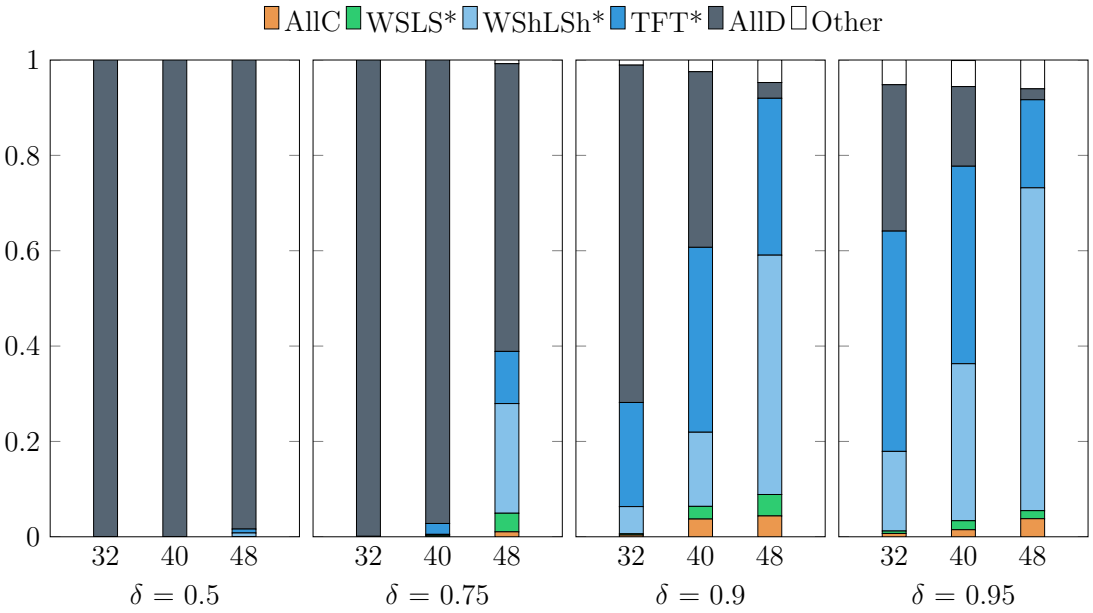

Figure 6: Strategy frequency estimation of algorithmic data by δ-R treatment and $k = 2$.

Note: The figure reports the cooperation rates averaged across for the baseline parameters $\alpha = 0.15$, $\nu = 20$.

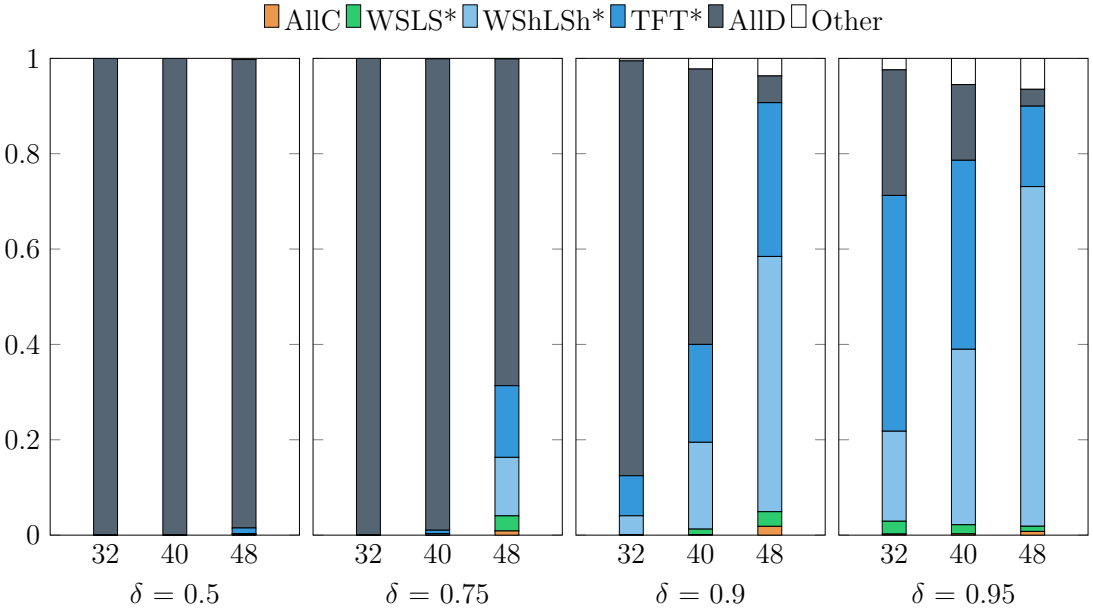

Figure 7: Strategy frequency estimation of algorithmic data by δ-R treatment and $k = 3$.

Note: The figure reports the cooperation rates averaged across for the baseline parameters $\alpha = 0.15$, $\nu = 20$.

References

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.

Akata, Elif, Lion Schulz, Julian Coda-Forno, Seong Joon Oh, Matthias Bethge, and Eric Schulz, "Playing repeated games with Large Language Models," *arXiv preprint arXiv:2305.16867*, 2023.

Asker, John, Chaim Fershtman, and Ariel Pakes, "The Impact of AI Design on Pricing," *Journal of Economics & Management Strategy*, 2023. Forthcoming.

Assad, Stephanie, Robert Clark, Daniel Ershov, and Lei Xu, "Algorithmic Pricing and Competition: Empirical Evidence from the German Retail Gasoline Market," *Journal of Political Economy*, 2023. Forthcoming.

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

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.

Bauer, Kevin, Oliver Hinz, Michael Kosfeld, and Lena Liebich, "Let's pl(AI): Evidence on the Implications of using LLMs as Surrogates in Human Decision-Making," 2023. Work in progress.

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.

Brand, James, Ayelet Israeli, and Donald Ngwe, "Using gpt for market research," *Available at SSRN 4395751*, 2023.

Breitmoser, Yves, "Cooperation, but no reciprocity: Individual strategies in the repeated prisoner's dilemma," *American Economic Review*, 2015, *105* (9), 2882–2910.

Brookins, Philip and Jason Matthew DeBacker, "Playing Games With GPT: What Can We Learn About a Large Language Model From Canonical Strategic Games?," *Available at SSRN 4493398*, 2023.

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

Bubeck, Sébastien, Varun Chandrasekaran, Ronen Eldan, Johannes Gehrke, Eric Horvitz, Ece Kamar, Peter Lee, Yin Tat Lee, Yuanzhi Li, Scott Lundberg et al., "Sparks of artificial general intelligence: Early experiments with gpt-4," *arXiv preprint arXiv:2303.12712*, 2023.

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.

Clark, Elizabeth, Tal August, Sofia Serrano, Nikita Haduong, Suchin Gururangan, and Noah A Smith, "All That's 'Human'Is Not Gold: Evaluating Human Evaluation of Generated Text," in "Proceedings of the 59th An-

nual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)" 2021, pp. 7282–7296.

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.

Ezrachi, Ariel and Maurice E. Stucke, "Sustainable and unchallenged algorithmic tacit collusion," *Northwestern Journal of Technology and Intellectual Property*, 2020, *17*, 217–260.

Friedman, James W., "A non-cooperative equilibrium for supergames," *Review of Economic Studies*, 1971, *38* (1), 1–12.

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.

Grossmann, Igor, Matthew Feinberg, Dawn C Parker, Nicholas A. Christakis, Philip E Tetlock, and William A Cunningham, "AI and the transformation of social science research," *Science*, 2023, *380* (6650), 1108–1109.

Guo, Fulin, "GPT Agents in Game Theory Experiments," *arXiv preprint arXiv:2305.05516*, 2023.

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.

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.

Horton, John J., "Large Language Models as Simulated Economic Agents: What Can We Learn from Homo Silicus?," *arXiv preprint arXiv:2301.07543*, 2023.

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. Forthcoming.

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.

Köbis, Nils and Luca D. Mossink, "Artificial intelligence versus Maya Angelou: Experimental evidence that people cannot differentiate AI-generated from human-written poetry," *Computers in human behavior*, 2021, *114*, 106553.

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.

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

Mnih, Volodymyr, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller, "Playing atari with deep reinforcement learning," *arXiv preprint arXiv:1312.5602*, 2013.

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.

_ , "GPT-4 Technical Report," 2023.

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.

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.

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

Silver, David, Aja Huang, Chris J Maddison, Arthur Guez, Laurent Sifre, George Van Den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot et al., "Mastering the game of Go with deep neural networks and tree search," *Nature*, 2016, *529* (7587), 484–489.

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 Dyan**, "Q-Learning," *Machine Learning*, 1992, *8*, 279–292.

Wei, Jason, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou et al., "Chain-of-thought prompting elicits reasoning in large language models," *Advances in Neural Information Processing Systems*, 2022, *35*, 24824–24837.

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.