Submitted to *Management Science*
manuscript

Building Socially Intelligent AI Systems: Evidence from the Trust Game using Artificial Agents with Deep Learning

Jason Xianghua Wu
University of New South Wales, School of Information Systems and Technology Management

Diana Yan Wu
School of Global Innovation and Leadership, Lucas College and Graduate School of Business, San Jose State University, One Washington Square, Business Tower 450, San Jose, CA.

Kay Yut Chen
Department of Information Systems and Operations Management, University of Texas at Arlington, Arlington, TX 76013.

Lei Hua
Soules College of Business, University of Texas at Tyler, Tyler, Texas 75799.

The trust game, a simple two-player economic exchange, has been extensively used as experimental measures for trust and trustworthiness of individuals. We construct deep neural network-based artificial intelligence (AI) agents to participate a series of experiments based upon the trust game. These artificial agents are trained by playing with one another repeatedly without any prior knowledge, assumption or data regarding human behaviors. We find that, under certain conditions, AI agents produce actions that are qualitatively similar to decisions of human subjects reported in the trust game literature. Factors that influence the emergence and levels of cooperation by artificial agents in the game are further explored. This study offers evidence that AI agents can develop trusting and cooperative behaviors purely from an interactive trial-and-error learning process. It constitutes a first step to build multi-agent based decision support systems in which interacting artificial agents are capable of leveraging social intelligence to achieve better outcomes collectively.

Key words: artificial intelligence; deep Q-network; interactive learning; trust; trustworthiness; social intelligence; decision support system

Introduction

There has been rapid development of artificial intelligence (AI) research and applications. Some focus on constructing "superhuman" AIs that are capable to defeat human professionals in increasingly complex games such as chess (Campbell et al. 2002), Go (Silver et al. 2016, 2017), and

Texas hold'em Poker (Brown and Sandholm 2018, Schmid et al. 2021). Different forms of self-play, where an artificial agent trains against copies or variations of itself, are applied to improve AI performance in these games. Others incorporate AIs into decision support systems (DSS) to help simulate human intelligence, optimize and automate decision-making activities in different fields such as cybersecurity, finance, healthcare, transportation, marketing and supply chain management (Gupta et al. 2021). According to recent estimates by Balakrishnan et al. (2020), the growth in businesses that are planning or implementing some form of AI technologies is at an astounding rate. As AI becomes more autonomous and prevalent in everyday life and enterprise settings, new questions and management challenges arise that require deeper understanding of AI behaviors, especially within a social context (Berente et al. 2021).

We report a series of experiments with multiple artificial agents playing the "trust game" (also known as the investment game) introduced in Berg et al. (1995). We use a value-based deep reinforcement learning method, Deep-Q-network (DQN) proposed by DeepMind (Mnih et al. 2015), to build the artificial agents. For convenience of discussion, we refer to them as AI agents or DQN agents interchangeably in the study. Two types of AI agents, one as the trustor and the other as the trustee, are created and trained by interacting with one another repeatedly in the game. No prior knowledge, assumption or training dataset regarding any human behavior is used in constructing these agents. We are interested in exploring the possibilities and conditions for AI to mimic social behaviors of humans, and more specifically, for them to behave as if they would trust and be trustworthy in the trust game.

Trust and trustworthiness are economic primitives that impact individual behaviors (Berg et al. 1995), organizational performance (Jeffries and Reed 2000, Dirks and Ferrin 2001), social and business relationships (Rempel et al. 1985, Ring and Van de Ven 1994), and efficiency of markets and channels (Bolton et al. 2004, 2013, Beer et al. 2018). Their characteristics, expressions, and implications have been studied in many disciplines including biology, psychology, sociology, economics and management. As evidenced by prior empirical research, determinants of trust can be biological, such as hormones, genes or brain structures (Kosfeld et al. 2005, Fehr et al. 2005, Riedl and Javor 2012); they can also be environmental, such as individual experience, cultures and institutions (Croson and Buchan 1999, Gächter et al. 2004, Engle-Warnick and Slonim 2004).

We discover that DQN agents can discover humanlike trust/trustworthiness through an interactive trial-and-error learning process without any human intervention. Under certain conditions, aggregate levels of cooperation that the AI agents attained in the trust game are close to those by human subjects reported in the experimental literature. We further identify requirements, with respect to training protocols, history of past actions and incentive for future rewards, for DQN agents to develop cooperative behaviors in the trust game. The study offers insights on how AI can

be built with capabilities needed to solve problems of cooperation, which is fundamental to business and economic decision making regarding, for example, new product and technology development, collaborative forecasting and logistic planning in supply chains. It constitutes an important first step for developing DSS in which multiple artificial agents need to interact frequently and be "socially intelligent" to effectively achieve cooperative outcomes.

Trust Game

To establish a clear and quantifiable measure of trust and trustworthiness, we follow the standard behavioral economics approach to conduct experiments using the trust game (Berg et al. 1995). It is a non-zero-sum game in which two players send money back and forth sequentially: Player 1 (i.e., the trustor) is given a sum of money (an endowment) and decides how much to send to the other player, knowing that the amount sent will be tripled; Player 2 (i.e., the trustee) then decides how much to send back, which the trustor has no control of. Trust herein is measured by the *amount sent*, and trustworthiness is measured by the *amount returned*.

From a rational choice perspective that assumes self-interest, the unique Nash equilibrium of the single-shot trust game is that the trustee should not return any money, and the trustor should therefore never send any money. Unlike the Prisoner's Dilemma in which two players move simultaneously and can both defect, the trust game captures the one-sided incentive problems (e.g., in e-Commerce or credit markets), where only the second mover wants to exploit. It is therefore a more promising environment to achieve a social norm of cooperation (Duffy et al. 2013). Indeed, in the one-shot experiments of Berg et al. (1995) and numerous follow-up studies (see Johnson and Mislin (2011) for a comprehensive review), human subjects are found to send and return significantly positive amounts albeit rather large variations across individuals. These results help demonstrate that trust and trustworthiness can allow for mutual gains to be realized without enforcement in human society (Alós-Ferrer and Farolfi 2019).

The literature on repeated trust games is relatively sparse. There are many equilibrium strategies (Engle-Warnick and Slonim 2006a, Xie and Lee 2012), and the binary-choice trust game is thus introduced to reduce the action space: for a predetermined split of payoffs, the trustor plays either *Don't Send* or *Send*; the trustee chooses either to *Return* or *Keep*. Some studies in this literature focus on the finitely repeated game that has a known ending point to investigate reputation effect (Bohnet and Huck 2004, Huck et al. 2012, Abraham et al. 2016, Attanasi et al. 2019); and others have subjects playing the game for an indefinite number of periods, which is stochastically determined by a continuation probability, to examine evolution of behaviors (Engle-Warnick and Slonim 2004, 2006b, Duffy et al. 2013).

In this study, we are interested in understanding how AI agents learn and behave given either one-shot or repeated interactions. Experiments are therefore designed to approximate the one-shot and the repeated trust game accordingly. It should be noted that, while trust in humans may involve a "psychological state comprising of the intention to accept vulnerability" (Rousseau et al. 1998), such intentions or perceptions cannot be observed or measured directly. Past research has attempted to correlate surveyed attitudes toward trust (Glaeser et al. 2000), or brain signals (King-Casas et al. 2005, Riedl et al. 2014) with results from the trust game. Consistent with the experimental measure extensively used in the literature, we evaluate AI behaviors exclusively by direct observations, i.e., the amount sent for trust and the amount returned for trustworthiness by artificial agents in the game.

Deep Q-network (DQN) Artificial Agents

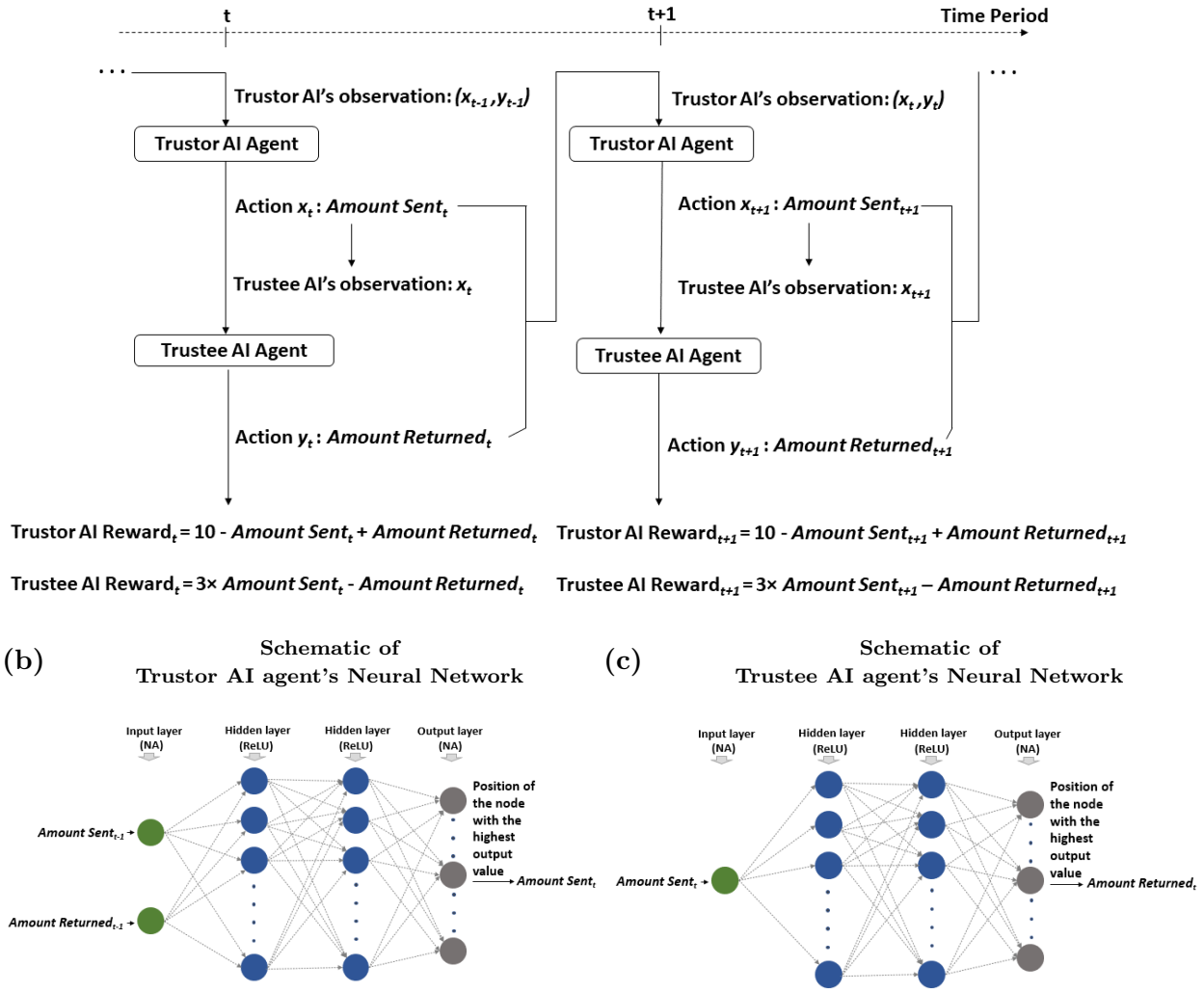
Reinforcement learning (RL), one of the basic machine learning paradigms, provides a framework of how an intelligent agent learns to optimize actions through interactions with the environment in order to maximize the expected cumulative reward (Sutton et al. 1998). For example, Q-learning, a RL algorithm, is used to construct autonomous interacting agents for modeling evolving economics systems in the computational economics literature (Tesfatsion 2006). It was also applied to study cooperation under Iterated Prisoner Dilemma, but with limited success (Sandholm and Crites 1996). In this study, we step further to employ an approach that combines RL paradigm with the power of deep neural networks, i.e., Deep reinforcement learning (DRL, see reviews by Shrestha and Mahmood (2019), Wang et al. (2020)). DRL has obtained striking success in challenging tasks such as AlphaGo (Silver et al. 2016).

We use the Deep-Q-network (DQN) method, an epoch-making value based DRL algorithm (Mnih et al. 2015), to construct the artificial agents. Two types of DQN agents, the trustor AI and the trustee AI, are built and interacting in an environment with the same parameter settings as the trust game introduced by Berg et al. (1995). Fig.1a illustrates the timeline for their learning process. Fig.1b and Fig.1c present schematic of how the neural networks are structured for the trustor and the trustee AIs accordingly. In a time period t, the trustor AI receives \$10 of endowment, observes the amount sent and the amount returned in the previous period (x_{t-1}, y_{t-1}), and then decides the amount to send to the trustee AI in the current period (x_t). The trustee AI observes such an amount being tripled, and then decides the amount to return to the trustor AI (y_t). Reward to the artificial agent in the period t ($reward_t$) is characterized as: $10 - x_t + y_t$ for the trustor AI and $3 \times x_t - y_t$ for the trustee AI, respectively.

To capture the potential impact of future reward on AI's actions, we use the recursive formulation of the Bellman Equation to define the objective (action-value) functions of the trustor AI and the

trustee AI (see Section S1 in supplementary materials for details). This formulation is theoretically identical to the sum of an infinite stream of discounted per-period rewards (Bertsekas et al. 1995). The discount rate (γ) ranges between 0 and 1, with 0 implying that an agent completely ignores the future and only learns about actions that produce an immediate reward. As a starting point, we train all artificial agents with $\gamma = 0.75$. It is in a similar range to the average discount rate estimated from individuals by field studies (Harrison et al. 2002, Warner and Pleeter 2001) and to those used in experimental studies (Engle-Warnick and Slonim 2006b, Duffy et al. 2013). In subsequent experiments, we manipulate the discount rate to examine its impact systematically. A full description of the DQN algorithm is provided in the supplementary materials (see Section S1 and S2).

Figure 1 Reinforcement Learning Process and Neural Network Structure



Notes. (a) Timeline for how artificial agents interact under the trust game through a sequence of observations, actions and rewards. (b) and (c) Neural network architectures for the trustor and trustee AI, respectively. It has been shown that credit assignment path (CAP) of depth two can be a universal approximator to emulate any function (Sugiyama 2019). Given that the environment examined is relatively simple, we choose a deep neural network with depth three, and use only dense layers to approximate the action-value functions. Inputs to the neural network are the observations of the respective type of AIs. The input layer is followed by two fully connected density layers with ReLU activation function (i.e.,$\max(0, x)$). The action taken by the AI agent corresponds to the position of the output unit which has the highest output value (i.e., the estimated action value). Thus, the number of neurons in the output layer equals to the number of actions the AI agent can take. More architectural details can be found in Table S2 of the supplementary materials.

Training of Artificial Agents and Experimental Design

We build multiple DQN agents independently, 20 as the trustor and 20 as the trustee, to participate a series of experiments designed with three research objectives: 1) to determine whether or

not AI agents can develop "human-like" social behaviors, 2) to identify conditions for trust and trustworthiness to emerge in these agents without human intervention, and 3) to explore factors that would influence their levels of cooperation in the trust game.

All of our experiments consist of a training stage that lasts for 1,000,000 periods and a playing stage that lasts for 10,000 periods. Consistent with human-subject experiments, choices of artificial agents in the game are restricted to be only integers. In each period of the training stage, one trustor AI interacts with one trustee AI according to the process illustrated by Fig.1a. We initialize the training stage with 200 periods where all agents choose random actions. Thereafter, training of the neural networks starts. Using a back-propagation algorithm, artificial agents adjust parameters of their neural networks to reduce mean-squared errors in the Bellman equation every two periods. After one million periods, update of the neural networks stops, and the playing stage begins.

It should be noted that the training of our DQN agents does not involve any human intervention. No prior knowledge or pre-structured training data (on desired outputs) is used in this process. All data fed into the training process purely come from their own interactive playing of the trust game by the AI agents. More details of the training process can be found in supplementary materials (see Section S3).

In the first set of experiments, we use two different matching protocols, fixed partner versus random stranger, to control how the two types of agents interact during the training stage and the playing stage. Under "partner training", pairings between the trustor and the trustee AIs are randomly assigned at the start of the training stage and then are fixed for one million periods. Under "stranger training", 20 trustor and 20 trustee AIs are randomly re-matched in every period of the training stage. Similarly, in the playing stage, the artificial agents can stay with the same partners with whom they have trained together, or continue with random re-matching to play for another ten thousand periods. As a result, we have a 2 (partner or stranger matching) by 2 (in training or playing stage) factorial design.

In the next sets of experiments, we further explore other requirements, with respect to inputs and rewards in constructing the artificial agents, for cooperative behaviors to emerge in the game. In particular, we manipulate the availability and the length of *memory*, i.e., the amount sent and the amount returned observed by an artificial agent in the past, and the discount rate as means to control how much they care about the future. Lastly, we conduct additional experiments to provide robustness checks for the main results by varying structures of the neural network and parameters controlling the learning process of AI.

Each experiment in the study consists of 20 trustor AIs and 20 trustee AIs. They are not aware of how many periods that the training stage or the playing stage lasts, nor are they provided with any information to identify one another. All DQN agents are reset before participating in a different training protocol.

Results

We focus on actions of artificial agents, i.e., the amount sent and the amount returned, in the playing stage after the neural networks stop updating. In each experiment, we have a total of 200,000 observations (20 agents × 10,000 periods) for the trustor (trustee) AI. No significant time trend is found in any treatment. In the subsequent statistical analysis, we treat each agent as an independent sample by averaging observations from all 10,000 playing periods. The normality check of the data suggests nonparametric statistical tests be used. We apply the Wilcoxon rank-sum tests for comparisons between experiments, and one sample Wilcoxon signed-rank tests to test the null hypothesis of no trust/trustworthiness.

Table 1 provides summary statistics of our first set of experiments under the 2 (partner vs. stranger) by 2 (training vs. playing) design. For the measure of trust, in addition to the amount sent, we report the *transfer rate* which is calculated as the percentage of endowment given away by the trustor AI. For trustworthiness, we also calculate the *return rate* which is the amount returned by the trustee AI as a proportion of the total amount available to return (conditional on positive amount sent).

Table 1 Summary Statistics for the 2x2 factorial design of AI-agent experiments

Average Over 10,000 Playing Periods	Trustor AI Agents (N=20)		Trustee AI Agents (N=20)	
	Amount Sent	Transfer Rate: $\frac{\text{Amount Sent}}{10}$	Amount Returned	Return Rate: $\frac{\text{Amount Returned}}{3\times\text{Amount Sent}}$
Partner training, Partner playing	5.45 (2.54)	54.50% (25.44%)	6.20 (2.98)	39.77% (15.42%)
Partner training, Stranger playing	3.78 (2.10)	37.84% (20.99%)	3.25 (0.62)	41.12% (46.14%)
Cochard et al. (2004) Repeated Game (N=16)	7.47 (NA)	74.70% (NA)	NA (NA)	56.14% (NA)
Cochard et al. (2004) One-shot Game (N=20)	5.00 (NA)	50.00% (NA)	NA (NA)	38.21% (NA)
Berg et al. (1995) One-shot Game (N=28)	5.36 (3.53)	53.57% (35.29%)	6.46 (6.19)	37.08% (24.83%)

Notes. $\gamma = 0.75$ in all experiments. The amount sent and the amount returned are averaged across 20 AI agents from their 10,000 playing periods. Standard deviations across AI agents are shown in the parentheses. All numbers are rounded to the second decimal place. In the two experiments with "stranger training", the amount sent and the amount returned are both 0.00 as the observations include very few periods of positive amounts. Results of Berg et al. (1995) shown in the table are from their "social history" treatment, where subjects are provided with information on past decisions from a control group.

Result 1: DQN agents can develop trust and trustworthiness purely from an interactive trial-and-error learning process.

We first observe that, under two experiments in which AI agents are trained as fixed partners, both the amount sent and the amount returned are significantly positive (two-sided p-values < 0.01 by the Wilcoxon signed-rank test); they are also statistically different from random draws of a uniform distribution over the respective possible amounts (p-values < 0.01 by the randomization test, see details in supplementary materials Session S4). Under the two experiments with stranger training, however, no amount is sent or returned by any agent. These results imply that, under appropriate learning environment, trust and trustworthiness can arise naturally in DQN agents.

Henceforth, we focus on the two experiments in which AI agents exhibit positive social behaviors given partner training, and compare them with human-subject experiments in the trust game literature. We use two studies, Cochard et al. (2004) and Berg et al. (1995), that have game parameters calibrated the same as ours for the comparison. We include the statistics reported in these two papers in Table 1. Cochard et al. (2004) investigate a finitely repeated trust game in which matched player pairs interact repeatedly, and they compare the results with those from the one-shot game. Berg et al. (1995) study only the one-shot game with manipulations on the availability of history information. In our study, when the artificial agents play as fixed partners without knowing the endpoint, it corresponds to an infinitely repeated game; when they play as randomly re-matched strangers, it approximates the one-shot game.

Result 1a: Aggregate levels of cooperation that AI agents achieve in the trust game after partner training are close to those found in human-subject experiments.

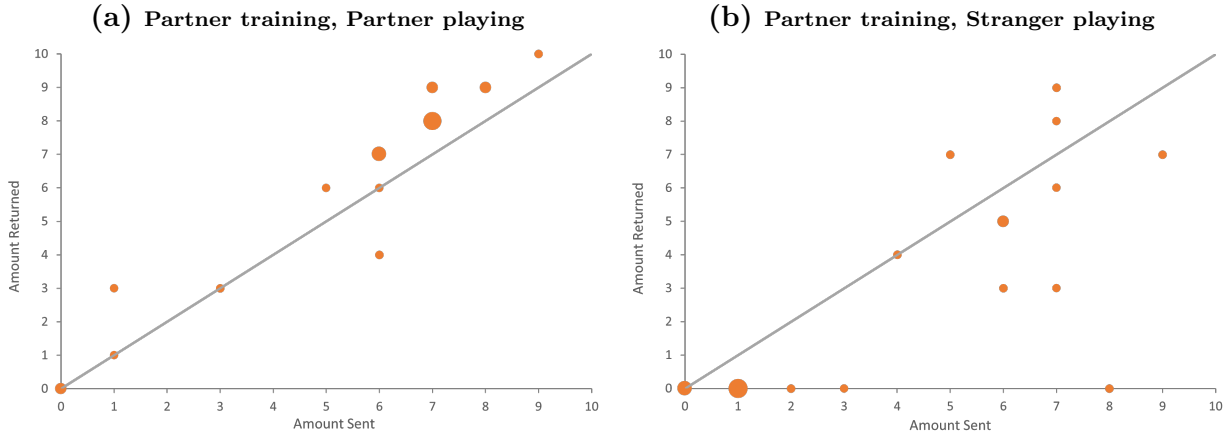
Cochard et al. (2004) find that the trustor sends more, and the trustee returns more in the finitely repeated trust game than in the one-shot game statistically. Similarly, we also observe that both the amount sent and the amount returned are significantly higher when AI agents play as fixed partners than as random strangers (two-sided p-values < 0.01 by the Wilcoxon rank-sum test). Levels of trust and trustworthiness that AI agents exhibit in these two experiments appear to be, by and large, lower than measures of human subjects in the corresponding games of Cochard et al. (2004). However, we cannot perform formal statistical comparisons due to lack of data access.

Berg et al. (1995) report detailed experimental data from one-shot games that we can use to compare with the stranger playing experiment directly. Aggregate levels of the amount sent and the amount returned by AI agents again seem to be lower than those from the respective human counterparts, but not statistically so (two-sided p-values > 0.10 by the Wilcoxon rank-sum test). We conduct additional analysis using the permutation test to evaluate the differences in standard deviations between our AI agents and human subjects in Berg et al. (1995). Results show that variances of decisions across AI agents, especially in the amount returned, are smaller (given the number of permutation of 100,000, two-sided p-values are 0.11 and < 0.01 for the amount sent and the amount returned, correspondingly).

Another commonly observed behavior in the trust game is reciprocity, as evidenced by both Cochard et al. (2004) and Berg et al. (1995). It implies that the amount returned is positively related to the amount sent. we compute the Spearman's rank correlation coefficient using data from Berg et al. (1995), which is 0.78 (with a p-value < 0.01). We also find highly significant positive correlations between the amount sent by the trustor AI and the amount returned by trustee AI: 0.83 for the stranger playing experiment and 0.96 for the partner playing experiment (both p-values < 0.01).

Johnson and Mislin (2011) survey 162 replications of the trust game involving more than 23,000 participants. An average of 50% of transfer rate and 37% of return rate result from this meta-analysis. In our study, the trustor AI transfers about 55% (38%) and the trustee AI returns about 40% (41%) when playing with partners (strangers). We acknowledge that settings under AI-agent experiments do not permit an exact comparison with human-subject experiments. Nevertheless, the analysis above offers evidence that DQN agents can develop trusting and cooperative behaviors that are qualitatively similar to humans in the trust game. It is also important to point out that, with biological and demographic differences being removed completely from AI agents, their behaviors still show a certain degree of heterogeneity. As an illustration, we plot observations from the last period of the playing stage, with partner playing in Fig.2a and stranger playing in Fig.2b. It can be seen that not all of the AI agents achieve cooperation in the end of either experiment.

Figure 2 Amount Sent and Returned in the Last Playing Period

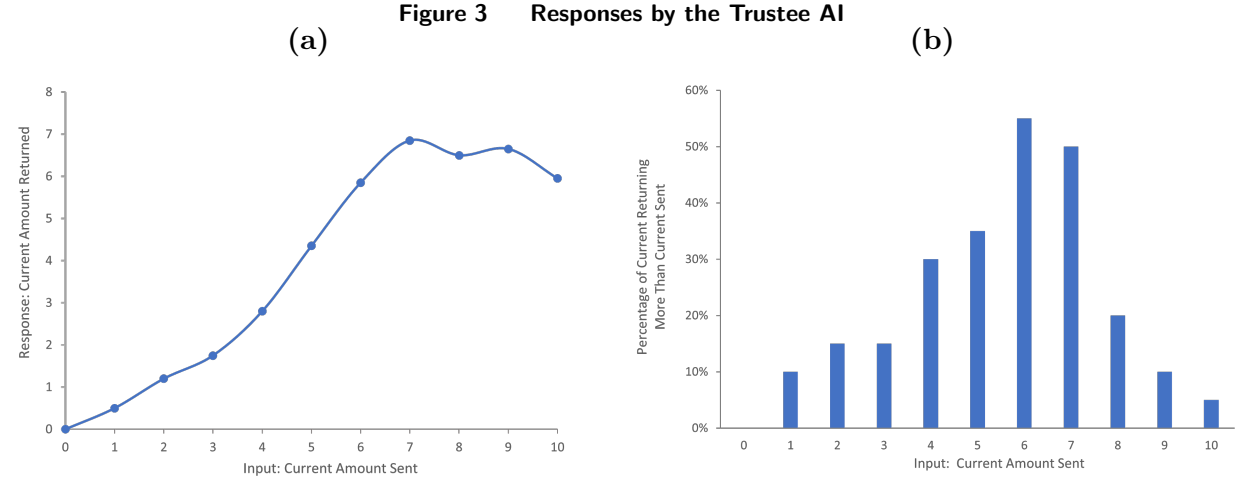


Notes. Both scatter plots are weighted by the number of observations to account for duplicates. The solid line represents when the amount of sent equals to the amount returned.

As a deeper probe into artificial agents' learning process for cooperative behaviors, we analyze their response functions, which become static after training of the neural networks stops. For the trustee AI, we feed each agent all possible inputs, i.e., the amount sent in the current period ranging from 0 to 10 (x_t), to obtain the corresponding outputs, i.e., the amount it returns in the current

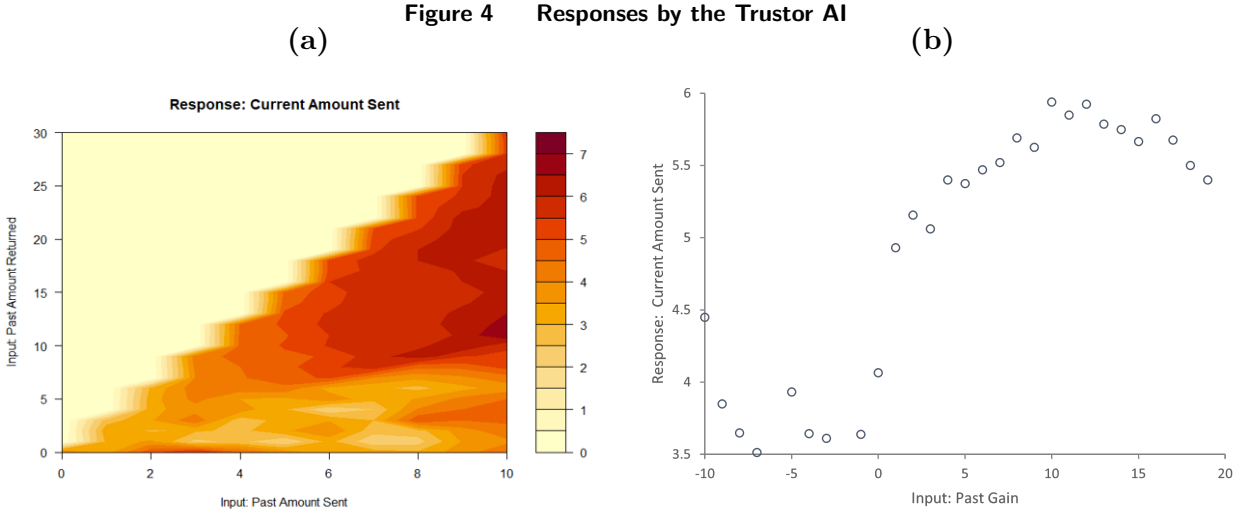
period (y_t). The resulting response function for the trustee AI, averaged across all 20 agents, is graphed in Fig.3a. In addition, we plot in Fig.3b, the percentage of trustee AIs returning an amount more than what is sent given each possible input. This allows us to check how often the trustor AIs can actually benefit from trusting.

Figure 3 Responses by the Trustee AI



Notes. (a) The response function is averaged across 20 trustee AIs given each possible input of x_t in the range from 0 to 10. (b) We calculate the percentage of trustee AIs who return more than the amount sent given each possible input.

Figure 4 Responses by the Trustor AI



Notes. (a) The response function is averaged across 20 trustor AIs given all feasible input combinations (x_{t-1}, y_{t-1}) where $y_{t-1} \leq 3x_{t-1}$. (b) It is averaged across all 20 trustor AIs as well as all inputs of (x_{t-1}, y_{t-1}) that result in the same past gain.

For the trustor AI, calculation of the response function is more complicated as it depends on the amount sent and the amount returned in the previous period (x_{t-1}, y_{t-1}). We input all feasible combinations in the game, i.e., (x_{t-1}, y_{t-1}) where $y_{t-1} \leq 3x_{t-1}$, to obtain responses from each trustor AI agent on the amount sent in the current period (x_t). We use a heat map to showcase the average response function of all trustor AIs in Fig.4a. Higher amount sent is associated with darker coloring. Recall that the reward to the trustor AI is directly related to its gain or loss in

a period. In Fig.4b, we plot the average response by all trustor AIs as a function of its *past gain*, which is calculated as $(y_{t-1} - x_{t-1})$, i.e., the difference between the past amount returned and the past amount sent.

Result 1b: An analysis of response functions shows that DQN agents are able to sustain cooperation in a way akin to the trigger strategy.

For the trustee AI, we observe from Fig.3a that, for much of the input range, it tends to return more on average as the trustor AI sends more in the current period. This is in accordance with the positive correlations found in our experiments as well as the reciprocal behavior reported in human-subject experiments. More interestingly, in Fig.3b, we see that the likelihood for the trustee AI to return more than the amount sent peaks at the trustor AI sending an amount of 6, more than half of the endowment (10). For the trustor AI, as shown in Fig.4a, a higher past amount sent coupled with a higher past amount returned in general induces the agent to send more in the current period. In other words, a former cooperative relationship helps reinforce the trusting behavior of the trustor AI. Moreover, visual inspection of Fig.4b suggests a threshold strategy: when the trustor AI experiences a negative past gain, its amount sent in the current period stays relatively flat yet still positive on average; when its past gain exceeds the threshold of zero, the trustor AI is then triggered to raise its sending to be more than 5. Note that the highest average amount that the trustor AIs send is roughly around 6, which coincides with the point where the trustee AIs are most likely to return more than what is sent, benefiting the trustor AI.

Combining evidence from responses of both types of AI agents, it is intuitive that a trustor AI would consider a trustee AI "trustworthy" if it had benefited from trusting in the past; a trustee AI, on the other hand, will be more likely to be "trustworthy" if the trustor AI sends more than 5, an equal split of the endowment between two agents. Therefore, the higher levels of trust (5.45) and trustworthiness (6.20) found in AI agents under the partner training and partner playing experiment can be sustainable for long-term cooperation due to such positive reinforcement. The above observations about artificial agents' responses in the trust game are noisy and may not constitute a formal equilibrium argument. However they are, at least in spirit, akin to a trigger strategy based Nash equilibrium under the Folk Theorem of Repeated Games.

In the first set of experiments, the trustor AIs always observe the amount sent and the amount returned in the previous period as input, and training as fixed partners enable artificial agents to achieve cooperation. Our next set of experiments aims at further understanding the required inputs, and particularly the role of memory in the learning process of artificial agents, for them to become cooperative in the trust game. More specifically, we test the effect of memory by replacing the trustor AI's inputs of (x_{t-1}, y_{t-1}) with two random integers drawn from the respective possible ranges. We again look into the two experiments where AI agents develop positive social behaviors

after partner training, and extend each of them to a 2 (training or playing) by 2 (with or without memory) design. Summary statistics on the amount sent and the amount returned under corresponding treatments are shown in Table 2. For the convenience of discussion, we refer to the case where the trustor AI trains and plays as fixed partners always with its memory as the Baseline.

Table 2 Effect of the Trustor AI's Memory given Partner Training

		Training Stage: Memory	Training Stage: No Memory
Playing Stage: Memory	Partner Playing	Baseline: 5.45, 6.20 (2.54, 2.98)	0.00, 0.00 (0.00, 0.00)
	Stranger Playing	3.78, 3.25 (2.10, 0.62)	0.00, 0.00 (0.00, 0.00)
Playing Stage: No Memory	Partner Playing	4.23, 4.05 (1.78, 2.00)	0.00, 0.00 (0.00, 0.00)
	Stranger Playing	4.23, 3.40 (1.77, 0.72)	0.00, 0.00 (0.00, 0.00)

Notes. $\gamma = 0.75$ and $N = 20$ in all experiments. The amount sent by trustor AIs is followed by the amount returned by trustee AIs. Both are averages over 10,000 periods in the playing stage. Corresponding standard deviations are in parentheses. Inputs to trustor AIs with Memory are (x_{t-1}, y_{t-1}). For No Memory, x_{t-1} is replaced by an integer randomly drawn from 0 to 10; y_{t-1} is replaced by an integer randomly drawn from 0 to 30.

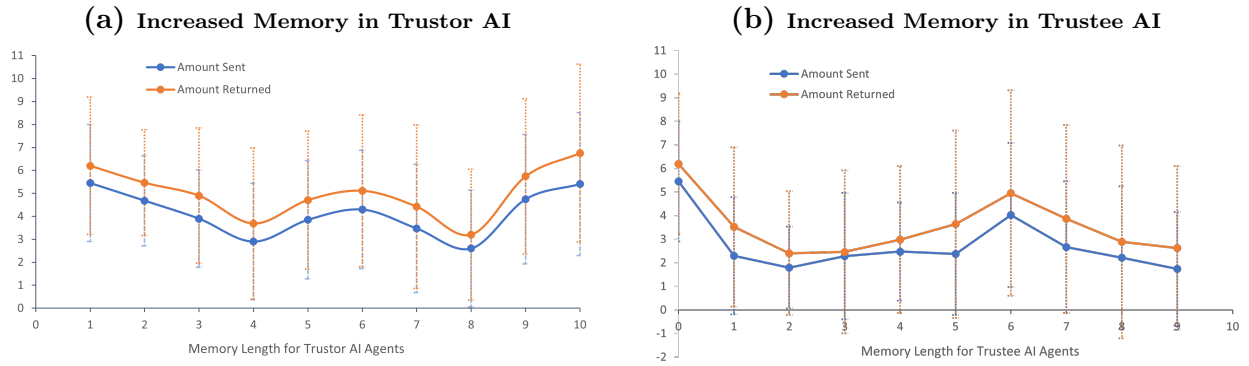
Result 2: Under partner training, memory of past experience by the trustor AI is required for DQN agents to establish trusting and cooperative behaviors.

From Table 2, we first observe that, without the trustor AI being able to observe memory in the training stage, no trust or trustworthiness would ever occur. Even when memory becomes available to the trustor AI later in the playing stage, both the amount sent and the amount returned remain at 0. As a further analysis, we conduct another sets of experiments where fixed pairs are trained with the trustee AI being able to observe (x_{t-1}, y_{t-1}) in addition to x_t, but the trustor AI having no memory. No cooperation is found there. We therefore conclude that memory of past experience by the trustor AI is required for DQN agents to develop human-like trusting behaviors and cooperation in the game through interactive learning as fixed partners.

When artificial agents train with the trustor AI having memory but plays without it, either as partners or strangers, the amount sent and the amount returned are always significantly positive (two-sided p-values < 0.01 by Wilcoxon signed-rank tests). So trust and trustworthiness persist as long as AI agents train with memory. Under partner playing, removing memory from the trustor AI has a significant impact: compared with the Baseline, the amount sent is reduced to 4.23 (two-sided p-value $= 0.07$ by Wilcoxon rank-sum test) and the amount returned is reduced to 4.05 (two-sided

p-value = 0.01 by Wilcoxon rank-sum test). Under stranger playing, whether or not the trustor AI has memory does not influence levels of trust or trustworthiness significantly. With memory, recall that AI agents achieve higher levels of cooperation when they play as partners than as strangers. Without memory, however, this difference disappears. The trustor AIs would still trust yet at a lower level as if they were always playing with a random stranger every period.

Figure 5 Impact of Past: Length of Memory



Notes. The amount sent and the amount returned plotted are averages across fixed pairs of artificial agents ($N = 20$) over the 10,000 playing periods. We include error bars for the standard deviations. (a) Consistent with the Baseline, which is plotted at length = 1, the trustee AI only observes x_t in these experiments. Length of memory in the trustor AI starts from 1 (i.e., history from the last period only) and goes up to 10 (i.e., history from the past 10 periods). (b) Consistent with the Baseline, which is plotted at length = 0, the trustor AI only observes (x_{t-1}, y_{t-1}) in these experiments. Length of memory in the trustee AI starts from 0 (i.e., observation from the current period only) and goes up to 9 (i.e., history from the past 9 periods).

Result 2a: Longer memory in DQN agents does not necessarily increase their levels of cooperation in the trust game.

The above findings reveal effects of memory, due to its availability in the training or playing stage, on the development of social behaviors by AI agents in the trust game. A follow-up question is that whether or not longer memory will help AI agents further improve their cooperation. To address it, another set of experiments is conducted in which we manipulate the length of history that is observable to the trustor AI and the trustee AI independently. All other game settings are kept the same as the Baseline where the trustor AI observes (x_{t-1}, y_{t-1}) and the trustee AI observes x_t in training and playing all the time. Fig.5a shows how the amount sent and the amount returned vary when memory in the trustor AI increases from its memory in the last period to the last ten periods. Likewise, Fig.5b illustrates the corresponding results for the trustee AI, where inputs to the agent increase from the amount sent in the current period only (i.e., no memory at all) to its memory from the last nine periods.

From the above graphs, we do not find support that longer memory, either in the trustor AI or the trustee AI, leads to monotonic improvement of cooperation in comparison with the Baseline.

To verify this, we perform the polynomial regression analysis with model selection, and estimates for the nonlinear terms of Fig.5a and Fig.5b are found to be highly significant (p-values < 0.01, see Session S5 in supplementary materials for details). We speculate that the artificial agents may be influenced by the increasing complexity in processing input information associated with longer memory.

In experiments presented so far, all of our DQN agents are trained with a discount rate of 0.75. Next, we examine the effect of incentives for future rewards on their levels of trust and trustworthiness in the game. Experiments are conducted in which discount rate γ is varied between 0 and 1 for each type of AI agent independently. All other game settings again are controlled to be the same as the Baseline. Results for the trustor AI and the trustee AI are shown in Fig.6a and Fig.6b, accordingly.

Result 3: Higher incentives for future rewards, controlled by the discount rate in DQN agents, generally increase their levels of trust and trustworthiness in the game; the discount rate in the trustee AI plays a dominant role in the emergence of cooperative behaviors.

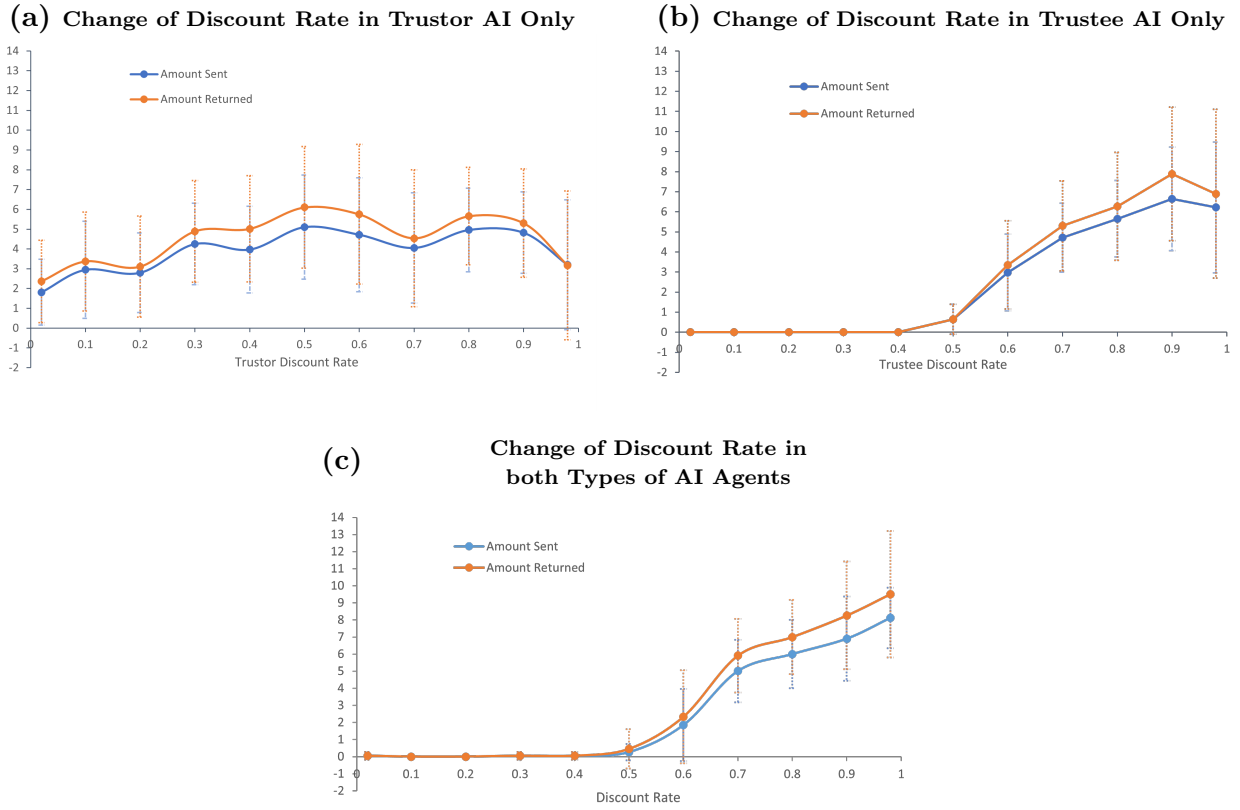
It is not surprising to see that, with more weights on the future rewards built into the artificial agents, both the amount sent and the amount returned increase in general. But the curves also appear to be nonlinear (see Section S5 in supplementary materials for formal tests using polynomial regressions). For the trustor AI (given the trustee AI's discount rate fixed at 0.75), as Fig.6a displays, levels of trust and trustworthiness stay positive over the entire range of analysis even when γ is close to 0. For the trustee AI (given the trustor AI's discount rate fixed at 0.75), more interestingly as Fig.6b shows, trust and trustworthiness become almost 0 once the discount rate drops below 0.5, i.e., when the future reward is weighted less than half of the immediate reward.

These results seem to imply that the discount rate in trustee AI might be more influential. To test this, we conduct experiments to vary the discount rate jointly for both types of AI agents. Results from these experiments are plotted in Fig.6c, whose shape is similar to Fig.6b. And again, it suggests a threshold of 0.5 for trust and trustworthiness to become significant. When the trustor AI and the trustee AI have the same discount rate, the amount sent and the amount returned appear monotonically increasing with γ. We thus speculate that the nonlinear curves in Fig.6a and Fig.6b might be due to the mismatch between two types of AI agents with different discount rates.

To summarize, memory for past actions and incentives for future rewards both affect levels of trust and trustworthiness of AI agents at present. Our results indicate that memory is essential to

the trustor AI, whereas the discount rate (exceeding a threshold) is critical to the trustee AI for them to become socially intelligent in the trust game.

Figure 6 Impact of Future: Value of Discount Rate



Notes. The amount sent and the amount returned plotted are averages across fixed pairs of artificial agents (N = 20) over the 10,000 playing periods. We include error bars for the standard deviations. The lowest discount rate used in these experiments is 0.02, and the highest is 0.98. (a) The discount rate of the trustor AI is varied while that of the trustee AI is fixed at 0.75. (b) The discount rate of the trustee AI is varied while that of the trustor AI is fixed at 0.75. (c) The discount rate is varied for both types of AI agents simultaneously.

Robustness Check

The learning process of the artificial agents are controlled by a set of parameters, known as *hyper-parameters*. They are tunable and may affect how well the neural network trains (Elgeldawi et al. 2021). The descriptions and values of hyperparameters used in the study are provided in Table S3 in the supplementary materials. As one of the first few research that employs DQN agents to conduct economic experiments, owing to the high computational cost, we did not perform a formal search in selecting values of these hyperparameters (Bergstra and Bengio 2012). We conduct robustness checks for our calibrations as the final analysis. The main results from the Baseline are duplicated in most cases without statistical differences. For a complete description and summary of results from the robustness check, refer to Section S6 in the supplementary materials.

Conclusion and Discussions

In this study, we construct multiple DQN agents and train them to play a non-zero sum game in which agents' interests are not fully aligned. Positive social behaviors of trust and trustworthiness are required and found in human subjects to generate mutually beneficial outcomes in this game. We discover that deep neural network-based artificial agents can establish humanlike trusting and trustworthy behaviors by learning interactively as fixed partners. The AI agents, unlike humans, are not subject to any influence from biological or demographic differences. The aggregate levels of cooperation they achieve in the trust game, however, are close to those of human subjects.

Using a series of experiments, we explore and identify requirements for trust and trustworthiness to naturally arise in these artificial agents. While training as fixed partners, the trustor AI has to "memorize" its past experience; whereas the trustee AI needs to "care" about future rewards to at least some degree instead of being entirely myopic. The resulting levels of cooperation in the trust game depend nonlinearly on the length of memory and the weight on future rewards built into DQN agents. These findings are eerily similar to our understanding of trust and trustworthiness in humans from existing literatures. Studies in behavioral economics, psychology and sociology have shown that a stable family environment is conducive to develop trusting relationships (Bowlby 1969, Erikson 1993, Bernath and Feshbach 1995); someone's past experience or history effects have been known as an influential factor to trust building (Bohnet and Croson 2004, Bolton et al. 2004, King-Casas et al. 2005, Charness et al. 2011); and the consideration of future is regarded as a key driver of trust and trustworthiness to foster long-term cooperation (Engle-Warnick and Slonim 2004, 2006b, Mahajna et al. 2008).

Unlike much recent AI research that concentrates on improving individual intelligence of artificial agents, this study attempts to understand how to construct AI agents with social intelligence through a trial-and-error learning process naturally. It is a first step to create a socially intelligent ecosystem in which multiple or even groups of artificial agents interact frequently and are capable to achieve better outcomes collectively - by going beyond self-interested optimizations. One of the near-term application scenarios can be a smart transportation and logistic management system where different self-driving vehicles, e.g., cars, trucks or drones controlled by AIs, coordinate and cooperate with route planning for improved traffic. Our study is in line with the recent call for interdisciplinary research to build "a science of cooperative AI" by Dafoe et al. (2021). They argue that AI agents need social understanding and intelligence to help humans manage cooperation challenges such as breakdowns in supply chains, humanitarian operations, climate change and pandemic preparedness.

Broader adoption and acceptance of AI systems require better understanding of how they work to ensure that such systems align with values of the human society that they are designed for.

To instill human values in AI, many recent studies use human data to train artificial agents (e.g., Koster et al. (2022)). We propose a different approach that does not require human interventions, as the use of training dataset can be biased (see a survey by Ntoutsi et al. (2020)). We argue that training AI agents to play games that require social interactions and contrasting them with human decision makers could help deepen our knowledge of AI behaviors in different social contexts. Moreover, since social behaviors of AI agents can be endogenously determined through interactive learning, it may also provide a new tool for us to explore learning behaviors in response to the need for cooperation under specific decision making scenarios. Our study is merely scratching the surface of this direction of research on how artificial intelligence and behavioral economics may interact to influence decision making activities (Camerer 2018).

While we provide some robustness check of the main findings using DQN agents, future studies may consider to test if these results can be generalized to artificial agents that are constructed or trained differently. The trust game we focus on represents one-sided incentive problems between two parties. Future extensions could study interactions of DQN agents under other games that involve cooperation with two-sided incentives (e.g., the Prisoner's Dilemma), more than two players (e.g., the public goods game), asymmetric information (e.g., the forecast-sharing game by Özer et al. (2011), or behavioral uncertainties (e.g., the beer game by Croson et al. (2014)). Another natural step is to investigate whether or not other types of social behaviors such as altruism, fairness or group bias would emerge from similarly conducted AI-agent experiments. While AI with deep learning is different from the human brain in many substantial aspects, both are built upon nonlinear and densely connected networks with learning capabilities. Perhaps, findings from future AI-agent experiments can also help shed light on the origin and evolution of some human behaviors.

References

Abraham M, Grimm V, Neeß C, Seebauer M (2016) Reputation formation in economic transactions. *Journal of Economic Behavior & Organization* 121:1–14.

Alós-Ferrer C, Farolfi F (2019) Trust games and beyond. *Frontiers in Neuroscience* 13:887.

Attanasi G, Battigalli P, Manzoni E, Nagel R (2019) Belief-dependent preferences and reputation: Experimental analysis of a repeated trust game. *Journal of Economic Behavior & Organization* 167:341–360.

Balakrishnan T, Chui M, Hall B, Henke N (2020) The state of ai in 2020. *McKinsey Global Institute* .

Beer R, Ahn HS, Leider S (2018) Can trustworthiness in a supply chain be signaled? *Management science* 64(9):3974–3994.

Berente N, Gu B, Recker J, Santhanam R (2021) Managing artificial intelligence. *MIS quarterly* 45(3):1433–1450.

Berg J, Dickhaut J, McCabe K (1995) Trust, reciprocity, and social history. *Games and economic behavior* 10(1):122–142.

Bergstra J, Bengio Y (2012) Random search for hyper-parameter optimization. *Journal of machine learning research* 13(2).

Bernath MS, Feshbach ND (1995) Children's trust: Theory, assessment, development, and research directions. *Applied and Preventive Psychology* 4(1):1–19.

Bertsekas DP, Bertsekas DP, Bertsekas DP, Bertsekas DP (1995) *Dynamic programming and optimal control*, volume 1 (Athena scientific Belmont, MA).

Bohnet I, Croson R (2004) Trust and trustworthiness. *Journal of Economic Behavior and Organization* 4(55):443–445.

Bohnet I, Huck S (2004) Repetition and reputation: Implications for trust and trustworthiness when institutions change. *American economic review* 94(2):362–366.

Bolton G, Greiner B, Ockenfels A (2013) Engineering trust: reciprocity in the production of reputation information. *Management science* 59(2):265–285.

Bolton GE, Katok E, Ockenfels A (2004) How effective are electronic reputation mechanisms? an experimental investigation. *Management science* 50(11):1587–1602.

Bowlby J (1969) Attachment and loss. vol 1: Attachment, vol 2: Separation, vol 3: Loss. *London: Hogarth Press* 1973:1980.

Brown N, Sandholm T (2018) Superhuman ai for heads-up no-limit poker: Libratus beats top professionals. *Science* 359(6374):418–424.

Camerer CF (2018) Artificial intelligence and behavioral economics. *The Economics of Artificial Intelligence: An Agenda*, 587–608 (University of Chicago Press).

Campbell M, Hoane Jr AJ, Hsu Fh (2002) Deep blue. *Artificial intelligence* 134(1-2):57–83.

Charness G, Du N, Yang CL (2011) Trust and trustworthiness reputations in an investment game. *Games and economic behavior* 72(2):361–375.

Cochard F, Van PN, Willinger M (2004) Trusting behavior in a repeated investment game. *Journal of Economic Behavior & Organization* 55(1):31–44.

Croson R, Buchan N (1999) Gender and culture: International experimental evidence from trust games. *American Economic Review* 89(2):386–391.

Croson R, Donohue K, Katok E, Sterman J (2014) Order stability in supply chains: Coordination risk and the role of coordination stock. *Production and Operations Management* 23(2):176–196.

Dafoe A, Bachrach Y, Hadfield G, Horvitz E, Larson K, Graepel T (2021) Cooperative ai: machines must learn to find common ground. *Nature* 593(7857):33–36.

Dirks KT, Ferrin DL (2001) The role of trust in organizational settings. *Organization science* 12(4):450–467.

Duffy J, Xie H, Lee YJ (2013) Social norms, information, and trust among strangers: Theory and evidence. *Economic theory* 52(2):669–708.

Elgeldawi E, Sayed A, Galal AR, Zaki AM (2021) Hyperparameter tuning for machine learning algorithms used for arabic sentiment analysis. *Informatics*, volume 8, 79 (MDPI).

Engle-Warnick J, Slonim RL (2004) The evolution of strategies in a repeated trust game. *Journal of Economic Behavior & Organization* 55(4):553–573.

Engle-Warnick J, Slonim RL (2006a) Inferring repeated-game strategies from actions: evidence from trust game experiments. *Economic theory* 28(3):603–632.

Engle-Warnick J, Slonim RL (2006b) Learning to trust in indefinitely repeated games. *Games and Economic Behavior* 54(1):95–114.

Erikson EH (1993) *Childhood and society* (WW Norton & Company).

Fehr E, Fischbacher U, Kosfeld M (2005) Neuroeconomic foundations of trust and social preferences: initial evidence. *American Economic Review* 95(2):346–351.

Gächter S, Herrmann B, Thöni C (2004) Trust, voluntary cooperation, and socio-economic background: survey and experimental evidence. *Journal of Economic Behavior & Organization* 55(4):505–531.

Glaeser EL, Laibson DI, Scheinkman JA, Soutter CL (2000) Measuring trust. *The quarterly journal of economics* 115(3):811–846.

Gupta S, Modgil S, Bhattacharyya S, Bose I (2021) Artificial intelligence for decision support systems in the field of operations research: review and future scope of research. *Annals of Operations Research* 1–60.

Harrison GW, Lau MI, Williams MB (2002) Estimating individual discount rates in denmark: A field experiment. *American economic review* 92(5):1606–1617.

Huck S, Lünser GK, Tyran JR (2012) Competition fosters trust. *Games and Economic Behavior* 76(1):195–209.

Jeffries FL, Reed R (2000) Trust and adaptation in relational contracting. *Academy of management review* 25(4):873–882.

Johnson ND, Mislin AA (2011) Trust games: A meta-analysis. *Journal of Economic Psychology* 32(5):865–889.

King-Casas B, Tomlin D, Anen C, Camerer CF, Quartz SR, Montague PR (2005) Getting to know you: reputation and trust in a two-person economic exchange. *Science* 308(5718):78–83.

Kosfeld M, Heinrichs M, Zak PJ, Fischbacher U, Fehr E (2005) Oxytocin increases trust in humans. *Nature* 435(7042):673–676.

Koster R, Balaguer J, Tacchetti A, Weinstein A, Zhu T, Hauser O, Williams D, Campbell-Gillingham L, Thacker P, Botvinick M, et al. (2022) Human-centered mechanism design with democratic ai. *arXiv preprint arXiv:2201.11441* .

Mahajna A, Benzion U, Bogaire R, Shavit T (2008) Subjective discount rates among israeli arabs and israeli jews. *The Journal of Socio-Economics* 37(6):2513–2522.

Mnih V, Kavukcuoglu K, Silver D, Rusu AA, Veness J, Bellemare MG, Graves A, Riedmiller M, Fidjeland AK, Ostrovski G, et al. (2015) Human-level control through deep reinforcement learning. *Nature* 518(7540):529–533.

Ntoutsi E, Fafalios P, Gadiraju U, Iosifidis V, Nejdl W, Vidal ME, Ruggieri S, Turini F, Papadopoulos S, Krasanakis E, et al. (2020) Bias in data-driven artificial intelligence systems—an introductory survey. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* 10(3):e1356.

Özer Ö, Zheng Y, Chen KY (2011) Trust in forecast information sharing. *Management Science* 57(6):1111–1137.

Rempel JK, Holmes JG, Zanna MP (1985) Trust in close relationships. *Journal of personality and social psychology* 49(1):95.

Riedl R, Javor A (2012) The biology of trust: Integrating evidence from genetics, endocrinology, and functional brain imaging. *Journal of Neuroscience, Psychology, and Economics* 5(2):63.

Riedl R, Mohr PN, Kenning PH, Davis FD, Heekeren HR (2014) Trusting humans and avatars: A brain imaging study based on evolution theory. *Journal of Management Information Systems* 30(4):83–114.

Ring PS, Van de Ven AH (1994) Developmental processes of cooperative interorganizational relationships. *Academy of management review* 19(1):90–118.

Rousseau DM, Sitkin SB, Burt RS, Camerer C (1998) Not so different after all: A cross-discipline view of trust. *Academy of management review* 23(3):393–404.

Sandholm TW, Crites RH (1996) Multiagent reinforcement learning in the iterated prisoner's dilemma. *Biosystems* 37(1-2):147–166.

Schmid M, Moravcik M, Burch N, Kadlec R, Davidson J, Waugh K, Bard N, Timbers F, Lanctot M, Holland Z, et al. (2021) Player of games. *arXiv preprint arXiv:2112.03178* .

Shrestha A, Mahmood A (2019) Review of deep learning algorithms and architectures. *IEEE Access* 7:53040–53065.

Silver D, Huang A, Maddison CJ, Guez A, Sifre L, Van Den Driessche G, Schrittwieser J, Antonoglou I, Panneershelvam V, Lanctot M, et al. (2016) Mastering the game of go with deep neural networks and tree search. *nature* 529(7587):484.

Silver D, Schrittwieser J, Simonyan K, Antonoglou I, Huang A, Guez A, Hubert T, Baker L, Lai M, Bolton A, et al. (2017) Mastering the game of go without human knowledge. *nature* 550(7676):354–359.

Sugiyama S (2019) *Human Behavior and Another Kind in Consciousness: Emerging Research and Opportunities: Emerging Research and Opportunities*. Advances in Human and Social Aspects of Technology (2328-1316) (IGI Global), ISBN 9781522582182, URL https://books.google.com/books?id=9CqQDwAAQBAJ.

Sutton RS, Barto AG, et al. (1998) *Introduction to reinforcement learning*, volume 135 (MIT press Cambridge).

Tesfatsion L (2006) Agent-based computational economics: A constructive approach to economic theory. *Handbook of computational economics* 2:831–880.

Wang Hn, Liu N, Zhang Yy, Feng Dw, Huang F, Li Ds, Zhang Ym (2020) Deep reinforcement learning: a survey. *Frontiers of Information Technology & Electronic Engineering* 1–19.

Warner JT, Pleeter S (2001) The personal discount rate: Evidence from military downsizing programs. *American Economic Review* 91(1):33–53.

Xie H, Lee YJ (2012) Social norms and trust among strangers. *Games and Economic Behavior* 76(2):548–555.

Acknowledgements

We thank Chaochao Yan (Google), Xin Miao (Amazon), Xiantong Zhen (United Image) and Andy Wang (Apple), for helping set up GPU facilities and their suggestions. The paper benefits from the discussions with Jennifer Zhang (UTA) and Jeff Hou (NCKU). Kay-Yut Chen has a potential research conflict of interest due to a financial interest with companies Hewlett-Packard Enterprise, Boostr and DecisionNext. A management plan has been created to preserve objectivity in research in accordance with UTA policy.

Supplementary Materials

S1. Objective Function Formulation: Bellman Equation

We created two types of artificial agents, referred to as the trustor AI and trustee AI, to play the respective roles in the trust game. In this section, we describe the corresponding objective functions that each type of AI is trained to optimize, and the associated temporal structures.

Let x_t be the amount sent and y_t be the amount returned in period t. The reward in period t for the trustor AI is $R - x_t + y_t$, where R is the initial endowment of the trustor. The reward in period t for the trustee AI is $\alpha x_t - y_t$, where α is the multiplier for the amount sent. In all experiments of this study, we have $R = 10$ and $\alpha = 3$, which are the same as Berg et al. (1995).

To capture the potential impact of future rewards, we use the recursive formulation of the Bellman Equation (Bertsekas et al. 1995) to define the objectives of the artificial agents. This formulation is theoretically identical to the sum of an infinite stream of discounted per-period rewards. Let the objective functions, also known as the action-value functions, of the trustor AI and the trustee AI be $Q^*_{trustor}$ and $Q^*_{trustee}$ respectively. In each period, a pair of artificial agents interacts through observing information, taking actions and receiving rewards. The following table summarizes these components.

Table S1 Observation, Action and Reward for Artificial Agents

	Trustor AI	**Trustee AI**
Observations (in period t)	(x_{t-1}, y_{t-1}) x_{t-1}: Amount sent by the trustor AI itself in period $t-1$ y_{t-1}: Amount returned by the trustee AI matched in period $t-1$	x_t x_t: Amount sent by the trustor AI matched in period t
Action (in period t)	x_t	y_t
Reward (in period t)	$R - x_t + y_t$	$\alpha \times x_t - y_t$

Hence, the respective action-value functions for the trustor AI and the trustee AI agents are defined as:

$$Q^*_{trustor}((x_{t-1},y_{t-1}),x_t)=\mathbf{E}\big\{R-x_t+y_t+\gamma\cdot\mathbf{max}_{x_{t+1}}Q^*_{trustor}((x_t,y_t),x_{t+1})|(x_{t-1},y_{t-1}),x_t\big\},$$

and

$$Q^*_{trustee}(x_t,y_t)=\mathbf{E}\big\{\alpha x_t-y_t+\gamma\cdot\mathbf{max}_{y_{t+1}}Q^*_{trustee}(x_{t+1},y_{t+1})|x_t,y_t\big\},$$

where γ is the discount rate.

S2. Deep Q-network (DQN)

We apply deep-Q network method (Mnih et al. 2015) to build the artificial agents. Specifically, the trustor AI uses a deep neural network $Q_{trustor}((x_{t-1},y_{t-1}),x_t,\theta_{trustor})$ to approximate the optimal action-value function, i.e., $Q_{trustor}((x_{t-1},y_{t-1}),x_t,\theta_{trustor})\approx Q^*_{trustor}((x_{t-1},y_{t-1}),x_t)$, where $\theta_{trustor}$ is the weights of the trustor AI's neural network.The optimal target value $R-x_t+y_t+\gamma\cdot\mathbf{max}_{x_{t+1}}Q^*_{trustor}((x_t,y_t),x_{t+1})$ is approximated by target values $T_{trustor}=R-x_t+y_t+\gamma\cdot\mathbf{max}_{x_{t+1}}Q_{trustor}((x_t,y_t),x_{t+1},\theta^{-1}_{trustor})$, where the deep neural network $Q_{trustor}((x_t,y_t),x_{t+1},\theta^{-1}_{trustor})$ has exactly the same neural network structure as
$Q_{trustor}((x_{t-1},y_{t-1}),x_t,\theta_{trustor})$ and its parameters $\theta^{-1}_{trustor}$ are copied from $\theta_{trustor}$ every C training iterations (see the hyper-parameter values we used in Table S3). At training iteration k, the parameter $\theta_{trustor}$ of the neural network
$Q_{trustor}((x_{t-1},y_{t-1}),x_t,\theta_{trustor})$ is adjusted to minimize the mean-squared error in the Bellman equation. Following the method (Mnih et al. 2015), we define the loss function based on a mean-squared error:

$$\begin{aligned}L(\theta_{trustor})=&\mathbf{E}_{x_{t-1},y_{t-1},x_t,y_t}\big(R-x_t+y_t+\gamma\cdot\mathbf{max}_{x_{t+1}}Q_{trustor}((x_t,y_t),x_{t+1},\theta^{-1}_{trustor})-\\&Q_{trustor}((x_{t-1},y_{t-1}),x_t,\theta_{trustor})\big)^2,\end{aligned}$$

where parameters set $\theta^{-1}_{trustor}$ is updated to equal to $\theta_{trustor}$ only every C training iterations and thus fixed when we calculate the loss $L(\theta_{trustor})$ in training iteration k. Once we have this well-defined loss function, we can use gradient descent method to train the neural network, i.e., to update $\theta_{trustor}$.

Similarly, for the trustee AI agents, we use a deep neural network $Q_{trustee}(x_t,y_t,\theta_{trustee})$ to approximate the objective function, i.e., $Q_{trustee}(x_t,y_t,\theta_{trustee})\approx Q^*_{trustee}(x_t,y_t)$, where $\theta_{trustee}$ is the weights for the trustee neural network. On the other hand, the optimal target value $\alpha x_t-y_t+\gamma\cdot\mathbf{max}_{y_{t+1}}Q^*_{trustee}(x_{t+1},y_{t+1})$ is approximated by target values $T_{trustee}=\alpha x_t-y_t+\gamma\cdot\mathbf{max}_{y_{t+1}}Q_{trustee}(x_{t+1},y_{t+1},\theta^{-1}_{trustee})$, where the deep neural network $Q_{trustee}(x_{t+1},y_{t+1},\theta^{-1}_{trustee})$ has exactly the same neural network structure as $Q_{trustee}(x_t,y_t,\theta_{trustee})$ and its parameters $\theta^{-1}_{trustee}$ are copied from $\theta_{trustee}$ every C training iterations. At training iteration k, we aim to adjust the parameter $\theta_{trustee}$ in neural network $Q_{trustee}(x_t,y_t,\theta_{trustee})$ to minimize the mean-squared error in the

Bellman equation. Following the method in (Mnih et al. 2015), we define the loss function based on a mean-squared error:

$$L(\theta_{trustee}) = \mathbf{E}_{x_t, y_t, x_{t+1}} \left(\alpha x_t - y_t + \gamma \cdot \mathbf{max}_{y_{t+1}} Q_{trustee}(x_{t+1}, y_{t+1}, \theta^{-1}_{trustee}) - Q_{trustee}(x_t, y_t, \theta_{trustee}) \right)^2,$$

where parameters set $\theta^{-1}_{trustee}$ is updated to equal to $\theta_{trustee}$ only every C training iterations and thus fixed when we calculate the loss $L(\theta_{trustee})$ at training iteration k. Once we have this well-defined loss function, we can use gradient descent method to train this deep neural network, i.e., to update $\theta_{trustee}$.

The architecture of the trustor AI and trustee AI neural network, illustrated schematically in Fig. 1(b), is shown in Table S2 with full details. Structures of both types of AI agents' neural networks are exactly the same except the number of neural nodes in the input layer and output layer. In the first set of experiments (reported in Table 1), the trustor AI observes both the amount sent and the amount returned in the past period, and therefore the input layer in its neural network has two nodes. The trustee AI only observes the amount sent by the trustor AI in the current period, and thus its input layer has one node. The input layer is followed by two fully-connected layers with the "ReLU" activation functions. The number of neural nodes in these two hidden layers are 800 and 1000, respectively. The output layer is a fully-connected linear layer with no activation function, and the position of each single output unit corresponds to a valid action. Since the trustor AI can take any integer decision from $\{0, 1, 2, ..., 10\}$, its neural network has 11 nodes in the output layer. The trustee AI can take any integer decision from $\{0, 1, 2, ..., 29, 30\}$, and thus its neural network has 31 nodes in the output layer.

Table S2 Structural Details of the Neural Network

Layer	**Activation Function**	**Number of Nodes in the Neural Network**	
		Trustor AI	Trustee AI
Input layer	NA	2	1
Hidden layer 1	ReLU $\max(0, x)$	800	800
Hidden layer 2	ReLU $\max(0, x)$	1000	1000
Output layer	NA	11	31

Notes. In experiments where we manipulate the length of memory for each type of AI agents (as shown in Figure 5), the number of neural nodes in the input layer varies accordingly. In addition, in experiments where we do robustness check for the number of neural network nodes in the two hidden layers (as shown in Table S8), the two values will be doubled or halved simultaneously.

S3. Training details

We created twenty trustor and twenty trustee AIs, each with a unique and independent neural network. Pairs of artificial agents are configured to be either " fixed" or "random" according to the experimental treatment. There is an *initialization phase* of 200 periods in the training stage. In the first period of the initialization phase, both types of artificial agents take random actions. From the second time period and onward, the trustor AI can observe actions of its own and the matched trustee AI from the previous period. The trustor AI takes an action based on an ε-greedy policy: (1) with a probability of $1-\varepsilon$, the trustor AI selects the action which corresponds to the position of the neural node with the highest output of its updated neural network (i.e., the estimated action value); (2) or with a probability of ε, the trustor AI selects a random action from a discrete uniform distribution over $\{0,1,2,...,10\}$. ε decreases with training iterations at a diminishing rate, i.e., $\varepsilon_t = e^{-\phi \Delta t}$, where ε_t represents the probability at which the AI agent chooses random action in period t, ϕ is the decayed rate, and Δt is the number of training iterations accumulated up to period t. In addition, we use a small probability as the lower bound of ε (see all hyper-parameter values in Table S3).

After the trustor AI takes an action in time period t, the trustee AI can observe it, and then take an action based on the ε-greedy policy as the trustor AI does. Rewards to the artificial agents in period t follow the standard trust game (Berg et al. 1995). For the trustor AI's reward in period t, $reward_t = R - x_t + y_t$. For the trustee AI's reward in period t, $reward_t = \alpha \times x_t - y_t$, where x_t is amount sent by the trustor AI in period t and y_t is amount returned by the trustee AI in period t. Training of the artificial agents starts after the initialization phase. The neural networks are updated every two periods with the deep-Q learning algorithm (Mnih et al. 2015) (its full structure is shown in the DQN Algorithm below). Specifically, we use two key techniques in this algorithm:

1) Experience replay

For each time period $t \geq 1$ we can store trustor AI agents' action experience tuple $e^t_{trustor} = \big((x_{t-1}, y_{t-1}), x_t, R - x_t + y_t, (x_t, y_t)\big)$ in a data set $D^t_{trustor} = \{e^1_{trustor}, e^2_{trustor}, ..., e^t_{trustor}\}$, where x_{t-1} is the amount sent by trustor AI agent itself in the previous time period $t-1$, y_{t-1} is the amount returned by trustee AI agent with who the trustor AI agent matched in the previous time period $t-1$, (x_{t-1}, y_{t-1}) is the trustor AI agent's observation in time period t, x_t is trustor AI agent's action taken in time period t, $R - x_t + y_t$ is the trustor AI agent's reward in time period t, (x_t, y_t) is the trustor AI agent's observation in the next time period $t+1$.

Similarly, for the trustee AI agents, we store their action experience tuple $e^t_{trustee} = \big(x_t, y_t, \alpha x_t - y_t, x_{t+1}\big)$ in a data set $D^t_{trustee} = \{e^1_{trustee}, e^2_{trustee}, ..., e^t_{trustee}\}$, where x_t is the amount sent by trustor AI agent with whom the trustee AI agent matched in the current time period t (x_t is also the trustee AI agent's observation in the current time period t), y_t is the amount returned by trustee

AI agent itself in the current time period t (y_t is also the action taken by the trustee AI agent in the current time period t), x_{t+1} is the amount sent by trustor AI agent with whom the trustee AI agent matched in the next time period $t+1$ (x_{t+1} is also the trustee AI agent's observation in the next time period $t+1$).

The data set $D^t_{trustor}$ and $D^t_{trustee}$ are pooled into a replay memory $D_{trustor}$ and $D_{trustee}$ with capacity K (new data samples will gradually replace old samples into the replay memory when its capacity is full). At each training iteration, we randomly draw a mini batch of samples stored in the replay memory. Note that all experiences are uniformly distributed in the data set, i.e., $e_{trustor} \sim U(D_{trustor})$ and $e_{trustee} \sim U(D_{trustee})$. This technique can increase data efficiency, reduce updating variance and smooth out the learning process (Mnih et al. 2015).

2) Fixed Q-targets network

We denote the neural network $Q_{trustor}((x_{t-1}, y_{t-1}), x_t, \theta_{trustor})$ and $Q_{trustee}(x_t, y_t, \theta_{trustee})$ in the loss functions as $Q_{trustor}$ and $Q_{trustee}$, which is updated by performing gradient descent in each training iteration. For the target neural network $Q_{trustor}((x_t, y_t), x_{t+1}, \theta^{-1}_{trustor})$ and $Q_{trustee}(x_{t+1}, y_{t+1}, \theta^{-1}_{trustee})$ in the loss functions, we denote them by $\hat{Q}_{trustor}$ and $\hat{Q}_{trustee}$ respectively. They have exactly the same neural network structure as $Q_{trustor}$ and $Q_{trustee}$. We copy all the weights of $Q_{trustor}$ and $Q_{trustee}$ to $\hat{Q}_{trustor}$ and $\hat{Q}_{trustee}$ every C training iterations, and then use $\hat{Q}_{trustor}$ and $\hat{Q}_{trustee}$ to generate the Q-learning targets $T_{trustor}$ and $T_{trustee}$ for the next C training iterations. This technique can further improve the training stability of the agents' neural networks (Mnih et al. 2015).

DQN Algorithm

For $t = 1$,T **Do**

{

For each pair of the trustor and trustee AI agents (20 pairs in total) **Do**

{

Initialize neural networks $Q_{trustor}$, $\hat{Q}_{trustor}$,$Q_{trustee}$, $\hat{Q}_{trustee}$;

Initialize replay memory $D_{trustor}$ and $D_{trustee}$ to capacity K;

If time period t **equals to one**

{ Assign x^t and y^t as random actions; }

Else {

Trustor AI agent i selects a random action x^t with probability ε_t ;

otherwise choose action $x^t = \mathbf{argmax_x} Q_{trustor_i}((x^{t-1}, y^{t-1}), x, \theta_{trustor})$;

Trustee AI agent j selects a random action y^t with probability ε_t ;

otherwise choose action $y^t = \mathbf{argmax_y} Q_{trustee_j}(x^t, y, \theta_{trustee})$;

If time period $t > 200$ **and** t **is even**

{ Sample mini-batch of transitions $e^k_{trustor} = \left((x^{k-1}, y^{k-1}), x^k, R - x^k + y^k, (x^k, y^k)\right)$ from $D_{trustor}$;

Calculate target value $T^k_{trustor} = R - x^k + y^k + \gamma \cdot \mathbf{max}_x \hat{Q}_{trustor}((x^k, y^k), x, \theta^{-1}_{trustor})$;

Perform a gradient descent on loss $\left(T^k_{trustor} - Q_{trustor}((x^{k-1}, y^{k-1}), x^k, \theta_{trustor})\right)^2$ with respect to weight $\theta_{trustor}$;

Every C training iterations reset $\hat{Q}_{trustor} = Q_{trustor}$;

Sample mini-batch of transitions $e^k_{trustee} = \left(x^k, y^k, \alpha x^k - y^k, x^{k+1}\right)$ from $D_{trustee}$;

Calculate target value $T^k_{trustee_j} = \alpha x^k - y^k + \gamma \cdot \mathbf{max}_y \hat{Q}_{trustee}(x^{k+1}, y, \theta^{-1}_{trustee})$;

Perform a gradient descent on loss $\left(T^k_{trustee} - Q_{trustee}(x^k, y^k, \theta_{trustee})\right)^2$ with respect to weight $\theta_{trustee}$;

Every C training iterations reset $\hat{Q}_{trustee} = Q_{trustee}$;

}

Store transition $e^t_{trustor}$ in $D_{trustor}$ (Store transition $e^{t-1}_{trustor}$ in $D_{trustor}$ in "random training" treatments);

Store transition $e^t_{trustee}$ in $D_{trustee}$ (Store transition $e^{t-1}_{trustee}$ in $D_{trustee}$ in "random training" treatments);

}

}

End For

}

End For

Descriptions of the hyperparameters and their values used in the study are listed in Table S3.

Table S3 Hyperparameters

Hyperparameter	Value used	Description
Time periods in initialization phase	200	In the first 200 periods of the training stage, the neural networks are not trained.
Time periods in the training stage	1000000	Total number of periods in the training stage
Time periods in the playing stage	10000	Total number of periods in the playing stage
Training frequency	2	The neural networks are updated every two periods after the initialization phase in the training stage.
Learning rate	0.0016	The learning rate used by the RMSprop optimizer
Initial exploration	1	Initial value of ε in ε-greedy policy.
Final exploration	0.00001	Final value of ε in ε-greedy policy.
Decayed rate	0.0001	ε in ε-greedy policy is exponentially decayed by this rate with training iterations.
Target network updating frequency (C)	3000	The target network weights are updated every 3000 training iterations.
Replay memory size	300000	Stochastic gradient descent (SGD) samples update from this number of most recent combinations of game information.
Mini-batch size	200	The number of training cases over which the SGD update is computed.

Notes. In experiments where we do robustness check for each hyper-parameter (as shown in Table S8), the value of each hyper-parameter varies accordingly.

S4. Randomization Test

Similar to Berg et al. (1995), we performed a randomization test for the null hypothesis that actions of the artificial agent are randomly drawn from some uniform distribution. For a trustor AI agent i ($i = 1, 2, 3, ..., 20$), we randomly draw a sample with 10,000 observations (which equal to the number of playing periods) from the discrete uniform distribution over the amounts $\{0, 1, 2, ..., 10\}$. We denote the sample as s_i and the frequency of each amount $m \in \{0, 1, 2, ..., 10\}$ in this sample as f_m^i. We measure the variance of the sample s_i as: $v(s_i) = \sum_{m=1}^{11}(f_m^i - \frac{N}{11})^2$, where $N = 10,000$. Given a trustor AI agent's actual decisions in the playing stage denoted d_i (which also include 10,000 observations), we have $v(d_i)$. We then can calculate the probability of $v(s_i) \geq v(d_i)$, which is the p-value of the randomization test, based upon 100,000 times of the random sampling. For each trustee AI, the same test procedure is repeated with a discrete uniform distribution over the

amounts $\{0, 1, 2, ..., 30\}$. All p-values from the above randomization tests are smaller than 0.01. We thus reject the null hypothesis that the artificial agent takes random actions.

S5. Polynomial Regressions on the Length of Memory and the Discount Rate

To formally test whether the effect of length of memory and the discount rate on the amount sent and the amount returned is linear or not, we conduct polynomial regression analysis and perform model selections. The final models chosen by Akaike information criterion (AIC) and Bayesian information criterion (BIC) are shown below. In the regressions, the amount sent and the amount returned are averages across fixed pairs of artificial agents ($N = 20$) over the 10,000 playing periods.

Table S4 **Trustor AI**

	Amount Sent	Amount Returned
Length of Memory	(1)	(2)
Linear Term	-1.11***	-1.20***
Square Term	0.10***	0.11***
Number of observations	200	200
Adjusted R^2	0.06	0.05
AIC	961.54	1043.55
BIC	971.43	1053.45

Notes. This table reports the estimated coefficients. Significance at $^{*}p < 0.1$; $^{**}p < 0.05$; $^{***}p < 0.01$.

Table S5 **Trustee AI**

	Amount Sent	Amount Returned
Length of Memory	(1)	(2)
Linear Term	-2.60***	-2.95***
Square Term	0.66***	0.75***
Cubic Term	-0.05***	-0.05***
Number of observations	200	200
Adjusted R^2	0.11	0.07
AIC	951.20	1079.61
BIC	964.39	1092.80

Notes. This table reports the estimated coefficients. Significance at $^{*}p < 0.1$; $^{**}p < 0.05$; $^{***}p < 0.01$.

Table S6 **Trustor AI**

	Amount Sent	Amount Returned
Discount Rate	(1)	(2)
Linear Term	9.95***	12.74***
Square Term	-7.98***	-10.92***
Number of observations	220	220
Adjusted R^2	0.11	0.11
AIC	1015.51	1096.08
BIC	1025.69	1106.26

Notes. This table reports the estimated coefficients. Significance at $^{*}p < 0.1$; $^{**}p < 0.05$; $^{***}p < 0.01$.

Table S7 **Trustee AI**

	Amount Sent	Amount Returned
Discount Rate	(1)	(2)
Linear Term	-15.81***	-18.27***
Square Term	48.14***	55.17***
Cubic Term	-26.45***	-30.24***
Number of observations	220	220
Adjusted R^2	0.72	0.67
AIC	847.60	957.27
BIC	861.18	970.84

Notes. This table reports the estimated coefficients. Significance at $^{*}p < 0.1$; $^{**}p < 0.05$; $^{***}p < 0.01$.

S6. Robustness Check Results

Table S8 Robustness Check

	Original Value		Averaged over 10000 playing periods: Amount Sent	Averaged over 10000 playing periods: Amount Returned	Two-sided p-value from Wilcoxon test: Amount Sent	Two-sided p-value from Wilcoxon test: Amount Returned
Periods in the training stage	1000000	Half	4.31 (1.55)	5.20 (1.79)	0.03	0.08
		Double	4.83 (3.05)	5.34 (3.69)	0.60	0.51
Periods in the playing stage	10000	Half	5.45 (2.54)	6.20 (2.98)	0.93	0.98
		Double	5.44 (2.54)	6.19 (2.98)	0.92	0.98
Time periods in initialization phase	200	Half	5.78 (2.37)	6.75 (2.65)	0.91	0.67
		Double	5.67 (2.33)	6.37 (2.74)	0.83	0.98
Training frequency	2	Half	5.95 (3.23)	7.05 (4.04)	0.44	0.24
		Double	5.36 (1.88)	6.04 (2.04)	0.42	0.63
Final exploration	0.00001	Half	5.09 (2.29)	6.14 (2.72)	0.29	0.58
		Double	5.96 (2.52)	6.99 (2.80)	0.79	0.74
Decayed rate	0.0001	Half	4.46 (1.72)	5.45 (2.77)	0.06	0.24
		Double	4.95 (2.84)	5.75 (3.61)	0.31	0.48
Replay memory size	300000	Half	5.54 (2.57)	6.67 (2.94)	0.97	0.73
		Double	5.25 (0.77)	6.14 (0.98)	0.12	0.28
Mini-batch size	200	Half	6.17 (1.89)	7.15 (2.31)	0.53	0.37
		Double	5.58 (1.90)	6.90 (2.45)	0.69	0.92
Learning rate	0.0016	Half	4.36 (2.42)	5.19 (2.83)	0.07	0.13
		Double	6.07 (2.13)	6.52 (3.14)	0.36	0.62
Target network updating frequency	3000	Half	6.17 (1.97)	7.42 (2.45)	0.46	0.32
		Double	4.57 (2.29)	4.93 (3.16)	0.18	0.13
Number of neural network nodes	800 & 1000 for the 1st & 2nd hidden layer	Half	5.37 (2.53)	6.17 (3.84)	0.92	0.98
		Double	5.67 (1.86)	6.47 (2.12)	0.72	0.85

Notes. The amount sent and the amount returned are averaged across 20 AI agents from their 10,000 playing periods. Standard deviations across AI agents are shown in the parentheses.

For each hyperparameter, we rerun the Baseline experiment with its original value being doubled or halved while keeping all other parameters unchanged. We also conduct similar experiments to check the number of neural nodes used in constructing the hidden layers of neural networks (see Table S2 for details). Statistics on the amount sent and the amount returned in these experiments are reported in Table S8. Under the Baseline, the average amount sent is 5.45 and the average amount returned is 6.20 (more details can be found in Table 2 of the manuscript). The two-sided

p-values from Wilcoxon rank-sum test for comparisons with the Baseline are also shown in Table S8.

We note a few points from the robustness check here. First, our main conclusion is robust to all parameters tested, except for the number of periods in the training stage. If we reduce the one million training periods by half, the amount sent and the amount returned are different from the Baseline (two-sided p-values = 0.03 and 0.08, respectively by Wilconxon rank-sum tests); however, there is no significant difference when the training periods are doubled. This suggests that our original choice of training periods is long enough for behaviors of AI agents to stabilize. The amount sent also appears sensitive to the reduced decayed rate and the reduced learning rate (two-sided p-values = 0.06 and 0.07, respectively by Wilconxon rank-sum tests). Another point to mention is about reducing the number of neural nodes used in constructing the DQN agents. It does not change the result but cuts down significant amount of machine time. Future studies for similar tasks may consider to use less number of neural nodes to improve computation efficiency.