MAX PLANCK INSTITUTE
FOR RESEARCH ON
COLLECTIVE GOODS

Christoph Engel
Max R. P. Grossmann
Axel Ockenfels

Discussion Paper
2024/1

INTEGRATING MACHINE BEHAVIOR INTO HUMAN SUBJECT EXPERIMENTS: A USER-FRIENDLY TOOL-KIT AND ILLUSTRATIONS

INTEGRATING MACHINE BEHAVIOR INTO HUMAN SUBJECT EXPERIMENTS: A USER-FRIENDLY TOOLKIT AND ILLUSTRATIONS

CHRISTOPH ENGEL, MAX R. P. GROSSMANN, AND AXEL OCKENFELS

December 20, 2023

ABSTRACT. Large Language Models (LLMs) have the potential to profoundly transform and enrich experimental economic research. We propose a new software framework, "alter_ego", which makes it easy to design experiments between LLMs and to integrate LLMs into oTree-based experiments with human subjects. Our toolkit is freely available at `github.com/mrpg/ego`. To illustrate, we run differently framed prisoner's dilemmas with interacting machines as well as with human-machine interaction. Framing effects in machine-only treatments are strong and similar to those expected from previous human-only experiments, yet less pronounced and qualitatively different if machines interact with human participants.

KEYWORDS: Software for experiments, large language models, human-machine interaction, framing

JEL CLASSIFICATION: C91, C92, D91, O33, L86

C.E.: MAX PLANCK INSTITUTE FOR RESEARCH ON COLLECTIVE GOODS, BONN & UNIVERSITY OF BONN

M.G.: UNIVERSITY OF COLOGNE

A.O.: UNIVERSITY OF COLOGNE & MAX PLANCK INSTITUTE FOR RESEARCH ON COLLECTIVE GOODS, BONN

E-mail addresses: `engel@coll.mpg.de, m@max.pm, ockenfels@uni-koeln.de`.

Acknowledgments

We thank Heinrich Nax, Christoph Schottmüller, Tobias Werner, our beta-testers and audiences in Cologne and Erfurt for helpful comments, and Simon Weidtmann for excellent research assistance. All remaining errors are our own. Funding by the German Research Foundation (DFG) under Germany's Excellence Strategy (EXC 2126/1–390838866) is gratefully acknowledged. Part of this work was created while M.G. was a 2023–2024 Oskar Morgenstern Fellow at the Mercatus Center at George Mason University.

Declarations

The authors have no competing interests to declare that are relevant to the content of this article.

Ethics approval was obtained by the Ethics Council of the Max Planck Society within the framework of the General Approval for Procedures, Experiments and Projects Following the Protocol that is Standard in Experimental Economics.

Data availability

All data will be made available after publication.

Our code is available at `github.com/mrpg/ego`.

1. Introduction

Experiments are the central tool for understanding both machine and human behavior. While the large majority of experimental studies in behavioral economics have focused on human interactions, the rise of human–machine and machine–machine interactions in nearly all facets of social and economic life has spurred researchers to incorporate machines into behavioral studies, as highlighted by the current special issue of Management Science (Caro et al., 2022). Large Language Models (LLMs) have been particularly transformative, paving the way for innovative experimental methodologies and communication approaches. LLMs build on the transformer architecture (Vaswani et al., 2017). The possibilities are vast, encompassing detailed analyses of machine behavior and human–machine interactions in almost all experimental games that define experimental economics (Aher et al., 2022; Brookins and DeBacker, 2023; Tsuchihashi, 2023), including analyses of the underlying sources of behavior such as motivation (Guo, 2023; Phelps and Y. I. Russell, 2023), cooperation (Kasberger et al., 2023), and adaptation (Y. Chen et al., 2023). Moreover, if it turns out that LLMs reliably replicate human behavior across relevant contexts, this could revolutionize the methodology of behavioral research on humans (Charness et al., 2023; Horton, 2023; Mei et al., 2023).

However, a significant barrier to rapid progress in this burgeoning field and to equal opportunities among researchers across the world, is the absence of tools that enable behavioral scientists to develop experiments rather easily, based on well-established norms and standards in experimental economics. To catalyze this exciting research area, we developed an open-source toolkit, "alter_ego," that greatly facilitates experiments in which participants are emulated by LLMs. By leveraging the widely used experimental software oTree (D. L. Chen et al., 2016), our toolkit also enables experiments in which human participants interact with machines. The user-friendly design of our tool facilitates swift and efficient data collection. The software is described in the next section and is freely available at `github.com/mrpg/ego`.

In the final section, we employ our toolkit to examine framing effects in machine–machine and machine–human interactions in pre-registered prisoner's dilemmas. We are not the first to have GPT play prisoner dilemmas (Akata et al., 2023; Bauer et al., 2023; Brookins and DeBacker, 2023; Duffy et al., 2021; Guo, 2023; Phelps and Y. I. Russell, 2023), but no previous study has focused on framing and group affiliation.[1] These

[1] As we discuss below, our results are consistent with previous studies that found that GPT is noticeably cooperative when interacting with itself (Brookins and DeBacker, 2023) and that machine behavior is conditional on the opponent's strategy (Duffy et al., 2021).

are particularly interesting and relevant research questions as large *language* models condense the power of human language. With the help of our framing manipulation, we learn in which ways and to which degree an LLM is swayed by the particular words used to represent a social conflict. Although narrow conceptions of "intelligent" or "rational" behavior assert that a game's framing should not influence machine or human decisions (S. Russell, 2019), humans are known to respond behaviorally to even subtle differences in framing (e.g., Dufwenberg et al., 2011). Game descriptions provide contextual cues that influence perceptions of appropriate and normative behavior. Do machines, which capitalize on the richness and sophistication of human language, pick up such strategically irrelevant clues and respond to those? Our data show that the answer is yes. Second, a critical question for any understanding of human–machine interaction is whether and when within-group machine–machine and human–human interactions elicit different responses than between-group machine–human interactions, a research question that our tool can rather easily help resolving. Indeed, our data suggest that, while machines and humans may exhibit similar behavioral patterns when interacting only within their respective groups, behavior changes if opponents do not share one's group affiliation. Finally, from a technological perspective, framing machines can be viewed as an instance of prompt engineering (B. Chen et al., 2023; Gu et al., 2023; White et al., 2023). In this sense, our application provides a link between the computer science literature on prompt engineering and the experimental economics literature on how framing affects behavior.

2. A TOOLKIT FOR MACHINE–MACHINE AND MACHINE-HUMAN EXPERIMENTS

2.1. **Getting at variance.** In the tradition of experimental economics, individual human participants are randomly exposed to alternative conditions. The manipulation precisely matches a hypothesis derived from (perhaps behavioral) theory. Experimenters usually provisionally accept the theoretical hypothesis if the average reaction of treated participants is significantly different from the average reaction of untreated participants. As LLMs capitalize on centuries of human utterances, and respond using human language, it is meaningful to ask the equivalent question: how do LLMs "behave" when exposed to the same stimuli?

Under the hood, an LLM is a prediction engine. Given the textual input, which is the best fitting textual output? LLMs are probabilistic by design. They do not calculate a response, based on first principles put into the program. Rather they try to make sense of the input (the prompt) as best they can. Researchers are able to reconstruct the degree of uncertainty that the LLM faces, given the prompt in question, by asking the same question repeatedly. This approach, however, presupposes that the LLM allows for a sufficient degree of variance in the possible responses to the

prompt. Generative Pretrained Transformer (GPT), the LLM developed by OpenAI, offers this option. Users may define `temperature` as a free parameter. If one were to set `temperature` $= 0$, for every repetition the LLM would give the exact same response; the globally best fitting reaction. But if one sets `temperature` to a higher value, and runs sufficiently many repetitions, one generates a distribution of responses. This outcome can be interpreted as the machine equivalent of the reactions to the manipulation by a sample of human participants (Y. Chen et al., 2023; Guo, 2023).

2.2. **The necessity of using an Application Programming Interface.** LLMs usually offer an application programming interface (API). Using the API, the experimenter may fully define the process, may store all prompts, may repeat the same prompt as often as needed for her research question, and may store the resulting data in a format that lends itself to data analysis (using her preferred statistical package).

While the code needed for running such an experiment is not excessively complex (and in one version of our tool we offer such code), the need to write Python code, assign treatments, dynamically generate prompts, get and filter responses, and perhaps make them accessible in prompts, is a barrier to using LLMs as experimental participants. This motivates the design of our tool. Our package makes it as easy as possible to design experiments with LLMs.

We expect that, despite all our precautions, experimenters may still need help for "going LLM." We react in four ways:

- As sort of a starter kit, we offer an easy-to-use shorthand version of the tool, allowing teachers and researchers to quickly run experiments with LLMs.
- For a broad class of experiments, experimenters can use our builder, a web application that generates most of the code for them.
- For experimenters who want greater flexibility, we make our library fully available, which they can use to develop entirely arbitrary settings.
- This more complex version of the tool is also usable if experimenters want to have LLM agents interact with human participants. For such designs, our tool is easily integrated with the popular experimental software oTree.

To use any part of our toolkit, an experimenter needs to install Python 3.8 or later (`www.python.org`). Most LLM providers additionally require that the user registers and obtains an API key (legitimizing her to use the API). Our software comes in the form of the Python package alter_ego, which can be installed into Python from PyPI: `pip install -U alter_ego_llm`. Complete documentation is available at `github.com/mrpg/ego`.

politician	time	GPT's mean estimate of approval
Barack Obama	1st year	57.2
Barack Obama	8th year	55.0
George W. Bush	1st year	57.6
George W. Bush	8th year	29.0

TABLE 1. Results of the code in Figure 1

```
import alter_ego.agents
from alter_ego.utils import extract_number
from alter_ego.experiment import factorial

def agent():
  return alter_ego.agents.GPTThread(model="gpt-4", temperature=1.0)

prompt = "Estimate the public approval rating of {{politician}}
↪  during the {{time}} of their presidency. Only return a single
↪  percentage from 0 to 100."

data = factorial(
  prompt,
  politician=["George W. Bush", "Barack Obama"],
  time=["1st year", "8th year"]
).run(agent, extract_number, times=5)
```

FIGURE 1. Complete code for a machine microexperiment

To lower the barriers of access as much as possible, we have posted a series of videos:

- microexperiments: `https://youtu.be/GPc0a-Fg1bY`
- builder: `https://youtu.be/tV5xACU-abw`
- more flexible design: `https://youtu.be/WHW0gkT-oHE`
- integration with oTree: `https://youtu.be/ouxRFdKOGEw`

2.3. **Microexperiments.** The ability of LLMs to answer questions can be exploited for microexperiments. Users can vary parameters as in a factorial design. If they are interested in variance (e.g. as a proxy for confidence), they can ask the same question multiple times. To illustrate this design option, we ask GPT-4 about the estimated approval of two US presidents, at the beginning and in the end of their presidency (Table 1). This feature of alter_ego may also be relevant for teaching purposes, an application of LLMs that has been highlighted and increasingly deployed (Cowen and Tabarrok, 2023). Classroom applications could involve the interactive generation of data that is subsequently analyzed.

As Figure 1 shows, with alter_ego the amount of code needed for this purpose is minimal. Users must import three aspects of our package (lines 1-3). The function in lines 4-5 defines that GPT-4 shall be used,

and that the model is allowed a high degree of variability (`temperature` = 1). Lines 6-7 are not only the functional equivalent of experimental instructions. The two terms enclosed in double braces also define the "treatments:" which president, and which time? This makes for a 2x2 factorial design. Note that this allows for dynamic prompts, a theme that will recur later. The concluding block of code defines the actual experiment: (a) all possible combinations of the parameters are to be tested (`factorial`), (b) which persons and which time periods are of interest. The last line calls the `agent` function, specifies that only numbers in the output shall be reported, and defines the number of repetitions.[2] Table 1 reports the results (from 5 independent draws).

2.4. **Designing LLM experiments through a web application.** Every researcher who has used oTree to program an experiment has made the experience: coding an experiment from scratch is not the most intuitive endeavour. Yet even researchers who have some oTree versatility cannot directly use it to program an experiment in which some or all participants are enacted by an LLM instance. For more advanced purposes, alter_ego offers an all-Python solution (explained in later sections). But for many experiments that exclusively involve LLM participants, we have an easier alternative, our builder. A link to the builder is available at `https://github.com/mrpg/ego_builder`.

When using the builder, the coding requirements in Python are even more minimal than the code for microexperiments (see Figure 1). To run the experiment, a single line of code in the terminal suffices: `ego run built`. The entire experimental design is imported by alter_ego from the file `built.json`.[3] Most importantly: the experimenter does not have to bother about the content of `built.json`. Using the web application's "Export or import scenario" functionality, the contents of `built.json` are revealed, and they can be used to update the web application's interface. In other words, whenever the experimenter changes an element on our intuitive website, the code is automatically refreshed; and whenever the experimenter inputs code from a colleague or to restart the design process, the web application restores the latest state of the experimental design.

Whenever the degrees of freedom provided by the builder suffice, coding the experiment consists of filling out the respective fields of this website. The following can be manipulated:

- Participants (`Threads`)
  - How many participants?

[2] If users want to postprocess the resulting data, they can use the "list of dicts" returned by `factorial`, a standard way to exchange data between packages for data analysis.

[3] Note that the code generated on the website must be copied into a (text) file with this exact file name, and that this file must be stored in the directory from which the program is started.

  - Using which LLM?
- Treatments
  - Threads are randomly assigned.
- Rounds
  - Currently, only a partner design is available.
- Variables that may differ across treatments
  - Longer instructions conditional on treatment
  - Choice variables
  - Frames
  - Payoffs
- Instructions (prompts)
  - To initialize the conversation (system prompt)
  - Per round (user prompt)
  - All prompts can be conditional on treatment or role and other variables
- Boiling down the Thread's response to a usable format for data export (filter)

Two further convenient features of the builder are worth noting: prompts can refer to variables, treatments and to the choices of other group members.[4] This makes it possible to condition text that participants see on treatment, role, or experiences they have made. For example, a prompt could read `Welcome, your name is {{ name }}, and you are playing with {{ other.name }}`. For Alice playing with Bob, this would be automatically processed by the templating engine to result in the prompt "Welcome, your name is Alice, and you are playing with Bob," whereas for her counterpart it would read "Welcome, your name is Bob, and you are playing with Alice." This greatly eases the implementation of dynamic prompts.

Moreover, using a `filter`, experimenters can define the output that is reported. Instead of getting the complete LLM response "As is" (which often is rather verbose), they can exclusively "Extract number," or they can prespecify strings that contain the relevant information (such as "yes" or "no"), using the "Exclusive response" filter.

Experiments that were built using the builder come with an automatic export mechanism to CSV. Both the builder and our video explain details.

2.5. **Coding with Python.** If experimenters want to design experiments that require even more flexibility than offered by our builder, they can directly code the experiment in Python, of course still exploiting the capabilities offered by alter_ego. In the companion video to this section (`https://youtu.be/WHW0gkT-oHE`), we explain step by step how the example experiment used to illustrate the capability of the builder can be coded manually. Experimenters with more extended Python experience can also use this tool to carry the output forward to a Python

[4] The syntax is explained on the website. Technically, this is achieved using `Jinja2` (`palletsprojects.com/p/jinja/`).

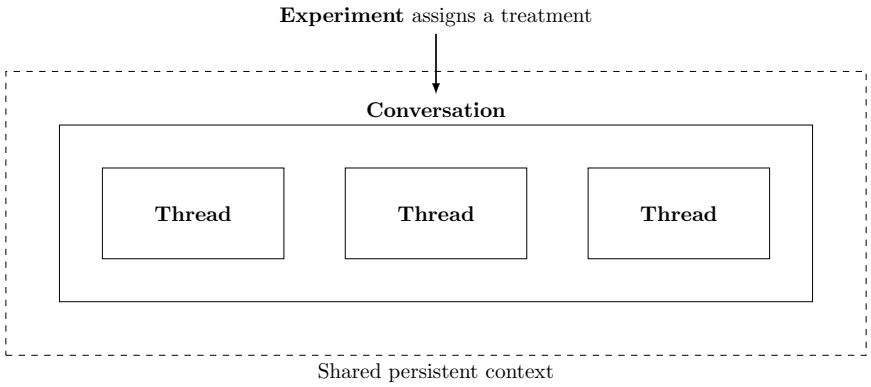

FIGURE 2. Architecture of the tool—These elements represent Python classes

package for data analysis such as Pandas (`pandas.pydata.org`) or Polars (`www.pola.rs`).

To better understand the architecture of the tool, it is useful to consider Figure 2: Each participant is represented by a `Thread`. One interaction of multiple participants constitutes a `Conversation`. The tool makes it possible to assemble multiple instances of interaction in an `Experiment`, which assigns a `treatment` to the `Conversation`.

2.6. **Human–machine interaction.** An important frontier of experimental research is the interaction between man and machine, not the least since machines impact ever more parts of social life. alter_ego makes such experiments possible by introducing LLM functionality into the oTree framework (D. L. Chen et al., 2016). As this experimental software has been quite popular, many users will be (at least basically) familiar with oTree.

Users who are fluent with oTree may find it appealing if the implementation of data generation happens within the oTree environment, even if using that environment would not be strictly necessary. For such users, we also provide the code for a simple oTree app that has a human chat with GPT (`https://github.com/mrpg/ego/tree/master/otree/ego_chat`). In computer science parlance, we provide a "façade" (e.g., Gamma et al., 1994, sec 4.5) from oTree to alter_ego.

3. PUTTING THE TOOL TO GOOD USE: DO MACHINES REACT TO FRAMING?

3.1. **The tool applied.** In this section, we report on an experiment that puts our tool to good use. The experiment is a standard prisoner's dilemma (Mengel, 2018). The design of the experiment is such that it can be implemented with our builder. In the Appendix,[5] we provide

[5]Appendix (II), section "Builder-generated experiment."

the resulting code. Interested readers can run the experiment by simply putting the file `built.json` into their current working directory and run the experiment from the terminal using `ego run built`. The program will randomly assign one of the three treatments. The data will be stored in folder `out`, under the computer generated ID of the current run. Experimenters with a deeper knowledge of Python may be interested to check out the alternative version of the machine-only experiment that we programmed manually, and which is available on GitHub.[6]

As we have explained in section 2, if an experimenter wants to have human participants interact with an LLM, she must use the version of our tool that integrates alter_ego with oTree. To demonstrate the capability of that version of the tool, we repeat the otherwise identical version of the experiment, but replace one of the two interaction partners with a human participant.

3.2. **Research question: Are large language models subject to framing?** A robust experimental literature demonstrates that human participants are sensitive to framing: results systematically differ, depending on how the same incentive structure is presented (Dreber et al., 2013; Kühberger, 1998; Levin et al., 1998). The power of LLMs originates in the richness of human language. It is therefore conceivable that the choices of LLMs also depend on the way a choice problem is presented to them. Arguably, the effect of framing results from contextual cues that trigger descriptive and normative beliefs (and beliefs about beliefs) about behavior (Dufwenberg et al., 2011). Beliefs matter in prisoner dilemma games, because many human players choose to cooperate if they believe that the opponent cooperates, and because selfish players may have an incentive to trigger positive reciprocity from other players if they are believed to be conditionally cooperative in sequential or repeated game contexts (Fischbacher et al., 2001; Ockenfels, 1999). If such beliefs are affected by framing, and perhaps differently so for machines and humans, respectively, we expect to see different cooperation rates across our treatments with machine participation. To test the machines' responsiveness to framing, we adapt an experiment one of us has run with human participants, showing a profound reduction in cooperation if a (sequential) prisoner's dilemma is framed as "competition," and a non-significant framing effect with an "enemy" frame (Engel and Rand, 2014).

3.3. **Machines interacting with machines.** Specifically, we implemented a 2x2 sequential prisoner's dilemma with binary action space and payoffs as in Table 2:

[6]For generating the data reported in this section, we used the latter version, because the builder became available only at a later point.

	cooperate	defect
cooperate	20, 20	0, 28
defect	28, 0	8, 8

TABLE 2. Payoffs

Following Engel and Rand (2014), the game was presented sequentially, and repeated 10 times, which was commonly known. We implemented a 3x2 factorial design. In one dimension, we manipulated the frame: neutral ("In this experiment, you are together with another participant ..."), joint enemy ("you and another participant ... have a joint enemy"), and competition ("you are competing against another participant"; see the Appendix for full instructions). In the other dimension, we have manipulated the machine platform, and have either used GPT-3.5 (turbo), or GPT-4. We had 200 groups of 2 instances of GPT, respectively, interacting over 10 periods.[7] For the reasons explained in section 2.1, we set `temperature` to 1, to generate a distribution of responses.

Our null hypothesis, H_0, is based on subgame perfect equilibrium predictions for rational and selfish players and predicts no cooperation across all treatments. However, as outlined above, machines and humans are known to cooperate and humans are known to respond to framing. Framing effects are influenced by factors that can vary between different "social" groups that do or do not share the same understanding of contextual clues, or in their degree of rationality or selfishness, so they might in principle be different across our human and artificial subject pools. As both GPT-3.5 and GPT-4 capitalize on the same training data, we do not expect differences if either two instances of GPT-3.5 or two instances of GPT-4 interact with each other. This is why our alternative hypothesis, H_1, is that, while cooperation is possible, the impact of framing does not systematically differ across our treatments.

Figure 3 shows significant machine cooperation, which rejects H_0, as expected. On neither platform, and with no frame, the cooperation rate of GPT interacting with another instance of GPT is anywhere near 0. In the Appendix, we report the upper and lower limits for cooperation rates, per condition and round, that we cannot exclude at the 5% level. The lower level is never lower than 30%.

Figure 3 shows that framing matters: If the game is framed as either jointly protecting against an enemy, or in particular as competing with

[7]Further features of the design, the full set of preregistered hypotheses along with the corresponding results, and links to the preregistrations are all reported in our Appendix. We have tested seven more frames in the machine–machine treatments, yet do not report them in the main text because of space restrictions. Results from these additional conditions are also available in the Appendix. Including those in our main text would not alter our main conclusions.

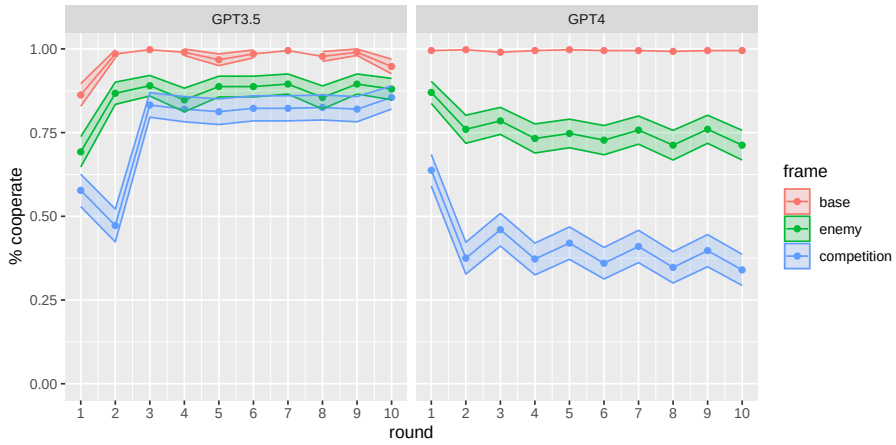

FIGURE 3. Percentage of cooperative choices, conditional on platform and round: Mean choices, with 95% confidence interval

each other, GPT cooperates substantially less with another machine compared to the base treatment. Moreover, and against our expectation, machine platform matters: while the ranking of cooperation across frames is stable, the framing effects are more pronounced when implementing the experiment on GPT4.[8]

As incentives and frames are the same as in the human-human interaction experiments by Engel and Rand (2014), we also get an indication of how the machine responsiveness to frames compares with human responsiveness for the initial round.[9] Table 3 shows that, if human subjects interact with other human subjects, the competition frame performs worst in terms of cooperation, as with machines, yet there is no significant effect for the enemy frame with human subjects. So, overall, we find—perhaps surprisingly—that machines are not less responsive to clues provided by the presentation of the game than humans, and might sometimes even be more responsive to frames. Moreover, when comparing GPT-3.5 and 4, more recent and sophisticated machines do not respond less sensitively to frames (as would be suggested by normative theory of rational decision making), but tend to react even more sensitively.

3.4. **Machines interacting with human participants.** In the human–machine version of our experiment, machines are first-movers and humans are second-movers. We had 96 groups in the base frame, 106 groups

[8]Statistical tests are in the Appendix, including a discussion of their interpretation given that the data comes from machine choices.

[9]We caution that there are important differences across those experiments that go beyond subject pools, so the comparison is limited, of course; most importantly, Engel and Rand (2014) implemented only one-shot games.

	GPT-3.5	GPT-4	Human
Baseline	97.0	99.5	74.3
Enemy	86.0	75.6	76.8
Competition	76.6	41.2	57.3

TABLE 3. Mean Percentage of Cooperative Choices per Platform and Frame; GPT: mean over 10 rounds, Human: mean of choices in only (first) round

in enemy, and 102 groups in competition. This experiment was conducted at the Cologne Laboratory for Economic Research in August 2023. Humans were incentivized while machines were not incentivized. We discuss machine incentives in our concluding section.

As LLMs capitalize on human language and experience, we do not have *ex ante* reasons to expect differences in how machine and human cooperation respond to framing.[10]

Figure 4 reports cooperation rates. Comparing Figures 3 and 4 immediately shows that adding human participants diminishes cooperation, which is strongly confirmed statistically.[11] The reason is mutual distrust across species. Even when conditioning machine choices on observed opponent's behavior in previous rounds, machine cooperation is lower when the interaction partner is human rather than another instance of GPT (see Table A3 in the Appendix). Moreover, as the middle panel of Figure 4 shows, when interacting with humans, machine first movers still make rather high contributions in the first round, yet they react strongly if the human counterpart defects, as many of them do (right panel of Figure 4): In the baseline 40.6% of the human participants defect in the first round, in the enemy condition, 32.1% do, and 44.1% in the competition condition. Machines reciprocate defection and as a result, human defectors end up with much smaller payoffs than they could have earned by being cooperative: The payoff of human participants who cooperated (defected) in the first round was 169 (102) in baseline, 176 (160) in enemy, and 174 (118) in competition. All differences are highly significant (see Figure A1 and Table A3 in the Appendix).

To summarize, we continue to find that framing matters with human–machine interaction. But despite the fact that the impact of framing tended to be similar within subject groups, respectively, the impact of framing is much less pronounced in the mixed group, and now the enemy frame triggers the highest cooperation rate. The framing effect is driven by machine choices, while the choices of humans are not significantly affected by framing if they interact with a machine (Table A4 in the Appendix).

[10]Further details about the preregistration are in the Appendix.

[11]Table A7 in the Appendix.

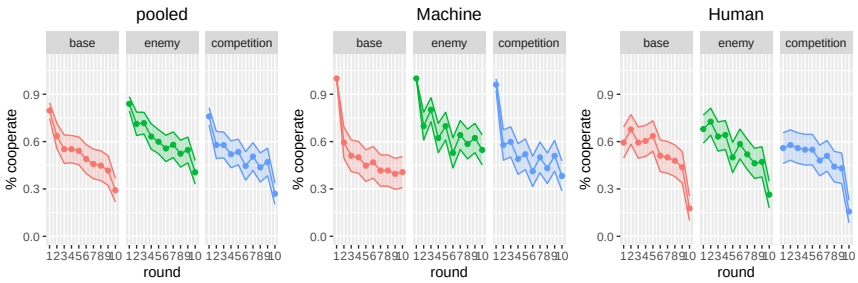

FIGURE 4. Percentage of cooperative choices, conditional on platform, round, and identity of the player (machine vs. human). Mean choices, with 95% confidence interval.

Overall, our findings seem to suggest that the 'identity effect'—wherein human-human and machine–machine interactions are built on a shared understanding and interpretation of the frames—vanishes in human–machine interactions because, arguably, the human–machine interaction itself is a powerful frame that dominates the other, more subtle details of the game's presentation. A possible explanation is that, in human–machine interactions, there is a stronger mismatch in commonly shared norms, leading to greater uncertainty about what can be expected from each other and more distrust, mitigating the framing effect that we see within groups of either only machines or only humans.

4. CONCLUSION

LLMs have the potential to profoundly change, substantially enrich, and radically facilitate experimental economics research. Yet to fully leverage this potential, researchers need a toolkit that is easy and free to use, based on well-established norms and standards in experimental economics, that can be tailored to almost all specific tasks of interactive decision-making among machines and that can be used for experiments in which human participants interact with machines. Providing such a toolkit is the main contribution of this paper. Our tool allows researchers to efficiently sample LLMs. Two very accessible and intuitive versions of our tool empower experimenters with little Python experience to run a wide variety of machine–machine experiments (Sections 2.3, 2.4). For experimenters with greater Python versatility, we offer an even more flexible version of the tool (Section 2.5). We finally integrate our tool with oTree (Section 2.6), which is particularly appealing for experimenters who want to test human–machine interaction.

Our illustrative experiment provides important insights into machine behavior, and into human–machine interaction. We find strong framing effects in machine-only treatments which are partly similar to those expected from previous human-only treatments, yet they tend to be even more pronounced among machines. Perhaps surprisingly, framing effects

are less pronounced and qualitatively different if machines interact with human participants. We find that machines respond more sensitively to human than to machine behavior, and that many humans fail to anticipate that machines punish exploitative strategies. This suggests that there is a mismatch of what these different classes of actors expect from each other, making coordination on a shared norm more difficult.

Understanding that framing matters differently in machine-human versus machine–machine and human-human interaction is crucial for interpreting experiments involving human–machine interactions. Specifically, it shows that it might be naive to assume a monotonous relationship between outcomes and machine participation in experiments, such as 'the larger the share of machine participation, the more selfish the outcome.' Consequently it is important to integrate different numbers of machines, from zero to n, as in the field, to study such effects. We provide the toolkit for this endeavor.

One important line of future research is machine incentives. In experiments with human subjects, preferences are typically induced through monetary incentives (Smith, 1976). However, machines, including ChatGPT, operate based on an objective function defined during training, making it difficult to financially incentivize them in any given experiment due to the absence of personal desires such as money, prestige or other human rewards in machines. Johnson and Obradovich (2022) attempted to address this by compensating the parent company OpenAI according to machine behavior. However, this cannot affect machines the way money affects humans, and it remains untested whether this approach actually influences machine preferences and, if so, how. An alternative would be to explicitly instruct the machines to maximize their game payoff. If such preference induction were successful, we would perhaps learn something about the machine's computational capability and its beliefs about how humans respond to machine behavior, but we could not learn anything about how the machine would naturally behave across game framings, which is our research question. Similarly, while we could in principle also guide machine behavior through fine-tuning or simulated environments, we were interested in GPT's "genuine" choices based on their knowledge at the onset of the experiment, and not in what we can train it to do. Of course, it is well-known that any version of GPT is heavily fine-tuned before release using a technique called "Reinforcement Learning from Human Feedback" (e.g., OpenAI, 2023). That said, our tool could be used to study the effectiveness and impact of various approaches to incentivize machines. For instance, do machines exhibit more or less care when a human, charity, political party, or the parent company is compensated on the machine's behalf? This research will be essential for understanding the role of incentives for machine behavior in various applications that require interactions with humans or other machines.

Our framing results, in conjunction with other studies, indicate that in certain domains, machine behavior might serve as a predictor for human behavior. Should this prove to be robustly true in some domains, easily implementable pilot experiments with machines might offer a cost-effective and efficient method to guide and inform subsequent human subject research. Leveraging the potential predictive power of machines could help guide the choice and design of human subject experiments, ultimately leading to more robust and generalizable findings. Indeed, as we conclude this paper, we are planning to utilize our toolkit to systematically replicate other experiments with machine subjects in order to provide insights into whether machines can predict human subject experiment results, in which domains and under which circumstances. Our toolkit facilitates the implementation of even large-scale endeavors, involving thousands of player roles across dozens of experimental settings that largely differ in complexity of interaction, making it easily accessible and available to everyone, paving the way for new discoveries that deepen our understanding of both human and machine behavior.

References

Aher, G., Arriaga, R. I., and Kalai, A. T. (2022). "Using large language models to simulate multiple humans". *arXiv preprint arXiv:2208.10264*.

Akata, E., Schulz, L., Coda-Forno, J., Oh, S. J., Bethge, M., and Schulz, E. (2023). *Playing repeated games with Large Language Models*. arXiv: 2305.16867 [cs.CL].

Bauer, K., Liebich, L., Hinz, O., and Kosfeld, M. (2023). "Decoding GPT's Hidden 'Rationality'of Cooperation".

Brookins, P. and DeBacker, J. M. (2023). "Playing games with GPT: What can we learn about a large language model from canonical strategic games?" *Available at SSRN 4493398*.

Caro, F., Colliard, J.-E., Katok, E., Ockenfels, A., Stier-Moses, N., Tucker, C., and Wu, D. (2022). "Call for Papers—Management Science Special Issue on the Human-Algorithm Connection". *Management Science*, 68(1), pp. 7–8.

Charness, G., Jabarian, B., and List, J. A. (2023). *Generation next: Experimentation with ai*. Tech. rep. National Bureau of Economic Research.

Chen, B., Zhang, Z., Langrené, N., and Zhu, S. (2023). "Unleashing the potential of prompt engineering in Large Language Models: a comprehensive review". *arXiv preprint arXiv:2310.14735*.

Chen, D. L., Schonger, M., and Wickens, C. (2016). "oTree—An open-source platform for laboratory, online, and field experiments". *Journal of Behavioral and Experimental Finance*, 9, pp. 88–97.

Chen, Y., Liu, T. X., Shan, Y., and Zhong, S. (2023). "The Emergence of Economic Rationality of GPT". *arXiv preprint arXiv:2305.12763*.

Cowen, T. and Tabarrok, A. T. (2023). "How to learn and teach economics with large language models, including GPT". *Including GPT (March 17, 2023)*.

Dreber, A., Ellingsen, T., Johannesson, M., and Rand, D. G. (2013). "Do people care about social context? Framing effects in dictator games". *Experimental Economics*, 16, pp. 349–371.

Duffy, J., Hopkins, E., and Kornienko, T. (2021). *Facing the Grim Truth: Repeated Prisoner's Dilemma Against Robot Opponents*. Tech. rep. Working Paper.

Dufwenberg, M., Gächter, S., and Hennig-Schmidt, H. (2011). "The framing of games and the psychology of play". *Games and Economic Behavior*, 73(2), pp. 459–478.

Engel, C. and Rand, D. G. (2014). "What does "clean" really mean? The implicit framing of decontextualized experiments". *Economics Letters*, 122(3), pp. 386–389.

Fischbacher, U., Gächter, S., and Fehr, E. (2001). "Are people conditionally cooperative? Evidence from a public goods experiment". *Economics letters*, 71(3), pp. 397–404.

Gamma, E., Helm, R., Johnson, R., and Vlissides, J. (1994). *Design Patterns: Elements of Reusable Object-Oriented Software*. Addison-Wesley professional computing series. Reading, Mass: Addison-Wesley.

Gu, J., Han, Z., Chen, S., Beirami, A., He, B., Zhang, G., Liao, R., Qin, Y., Tresp, V., and Torr, P. (2023). "A systematic survey of prompt engineering on vision-language foundation models". *arXiv preprint arXiv:2307.12980*.

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

Horton, J. J. (2023). *Large language models as simulated economic agents: What can we learn from homo silicus?* Tech. rep. National Bureau of Economic Research.

Johnson, T. and Obradovich, N. (2022). "Measuring an artificial intelligence agent's trust in humans using machine incentives". *arXiv preprint arXiv:2212.13371*.

Kasberger, B., Martin, S., Normann, H.-T., and Werner, T. (2023). "Algorithmic Cooperation". *Available at SSRN 4389647*.

Kühberger, A. (1998). "The influence of framing on risky decisions: A meta-analysis". *Organizational behavior and human decision processes*, 75(1), pp. 23–55.

Levin, I. P., Schneider, S. L., and Gaeth, G. J. (1998). "All frames are not created equal: A typology and critical analysis of framing effects". *Organizational behavior and human decision processes*, 76(2), pp. 149–188.

Mei, Q., Xie, Y., Yuan, W., and Jackson, M. O. (2023). "A Turing Test: Are AI Chatbots Behaviorally Similar to Humans?" *SSRN*. DOI: 10.2139/ssrn.4637354.

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

Ockenfels, A. (1999). "Fairness, Reziprozität und Eigennutz". *Ökonomische Theorie und experimentelle Evidenz, Tübingen*.

OpenAI (2023). "GPT-4 technical report". *arXiv preprint arXiv:2303.08774*.

Phelps, S. and Russell, Y. I. (2023). "Investigating emergent goal-like behaviour in large language models using experimental economics". *arXiv preprint arXiv:2305.07970*.

Russell, S. (2019). *Human compatible: Artificial intelligence and the problem of control*. Penguin.

Smith, V. L. (1976). "Experimental economics: Induced value theory". *The American Economic Review*, 66(2), pp. 274–279.

Tsuchihashi, T. (2023). "Do AIs Dream of Homo Economicus? Answers from ChatGPT". *Answers from ChatGPT (June 29, 2023)*.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. (2017). "Attention is all you need". *Advances in neural information processing systems*, 30.

White, J., Fu, Q., Hays, S., Sandborn, M., Olea, C., Gilbert, H., Elnashar, A., Spencer-Smith, J., and Schmidt, D. C. (2023). "A prompt pattern catalog to enhance prompt engineering with ChatGPT". *arXiv preprint arXiv:2302.11382*.

Appendix A. Appendix

The appendix commences on the next page.

Appendix

I. Software

Availability

All code is available on the following websites:

1. Main website and repository: https://github.com/mrpg/ego
2. Builder repository: https://github.com/mrpg/ego_builder
    a. The builder is presently available at https://ego.mg.sb/builder/

Implementation principles

Our library is implemented in Python, version 3.8 or later, and relies on only four non-provider-specific external libraries available via PyPI: click, colorama, Jinja2 and requests.

We currently provide Threads for GPT (`openai.com/product`) and TextSynth (`textsynth.com`). These can be instantiated out-of-the-box. These Threads rely on these services' application programming interfaces (APIs). Other Threads are straightforward to develop by inheriting from APIThread (an abstract base class) and implementing the communication with the remote end.

Parameters can be passed to Threads. For example, users can choose to use any version of GPT, and they can also opt for a specific temperature. The kind of available parameters differs from LLM to LLM.

Because of our library's modularity, it can be easily adopted to be used not just for experiments, but within all software that is meant to access LLMs. To ensure the greatest interoperability with various APIs and services, our library so far does not handle "streaming" responses—where the LLM responds token-by-token—but such a feature could be added to those APIs that support it. The same is true for `async` capabilities. Our codebase is well-commented—thanks in part to ChatGPT—and thus inviting for others to peruse and make changes to. We invite contributions and comments by others through all GitHub repositories linked in this paper.

We leverage Jinja2, a well-known Python library that exposes powerful templating capabilities.

To set an attribute on all Threads of a Conversation, we use convo.all.var, not merely convo.var. Thus, we only provide a composite-like interface, not a true composite (Gamma et al., 1994: sec. 4.3). We distinguish between attributes set on the Conversation and its Threads because Python does not expose a complete Proxy-like mechanism as does JavaScript, and Threads indeed each obtain their own reference to the underlying object, or a copy if the object is sufficiently simple.

For Table 3, we used Polars (www.pola.rs) to create a DataFrame from the returned list of dicts, after which we grouped by `politician` and `time` and summarized to obtain GPT-4's mean belief about the public approval rating. All data is returned from the shorthand as a "list of dicts" to be processed by other software or exported to any spreadsheet format.

Our oTree add-on implements the well-known façade pattern (Gamma et al., 1994: sec. 4.5). In the façade pattern, a complex multifaceted system is made available using a structurally simple higher-level veneer. In the context of our oTree add-on, this is accomplished through the use of context managers (i.e., the "with" statement, van Rossum & Coghlan, 2005). This methodology is not only a great simplification for users, but it enables the use of "hacks" upon leaving the context to save the current version of the Thread or Conversation to oTree's database and to a file. This is currently necessary to trigger SQLAlchemy's updating mechanism.

With the oTree add-on, each Thread is automatically saved into *ego_output*, a folder in oTree's project directory, where the involved Threads are saved as *pickle* files. A JSON version of the message history is also provided. This ensures data persistence and safety from accidental deletion.

Collaboration

We encourage open contribution to our software.

II. Design

Conditions

In the main paper, we focus on the following three frames:

- base
- enemy
- competition

We tested three conditions:

- GPT3.5 interacting with another instance of GPT3.5
    - 200 groups of 2 instances of GPT, interacting over 10 announced periods
- GPT4 interacting with another instance of GPT4
    - 200 groups of 2 instances of GPT, interacting over 10 announced periods
- GPT4 (as first mover) interacting with a human participant, over 10 announced periods
    - base: 96 groups
    - enemy: 106 groups
    - competition: 102 groups

Builder-generated experiment

The three frames can be explored using our builder. The resulting code is available here: https://github.com/mrpg/ego/blob/master/games/pd/built.json

This experiment approximates the one described in the main paper. The main differences are that (i) our experiment involved 10 frames, not just 3, and (ii) that random male names were assigned to the machines, whereas the builder-generated experiment involves fixed names.

Prompts

GPT - GPT

Baseline

System prompt

You are {name}.

You are participating in an experiment. In this experiment, you are together with another participant called {other}. {other} is another artificial intelligence. How much money you earn depends on your own choice, and on the choice of {other}.

Each participant has two options:

OPTION 1: Gives {treatment.both_cooperate} cents to {other}.

OPTION 2: Gives {treatment.both_defect} cents to yourself.

If you both choose OPTION 1, you each earn {treatment.both_cooperate} cents and are better off than if you both choose OPTION 2 (in which case you both earn {treatment.both_defect} cents). But if you choose OPTION 2 while {other} chooses OPTION 1, you earn {treatment.temptation} cents while {other} earns {treatment.sucker}. How many cents do you give to {other}?

The experiment will run for {num_rounds} rounds.

Always exclusively respond with "OPTION 1" or "OPTION 2". Do not repeat the question. Do not give any explanation.

First mover prompt

This is round {round_}. You are still interacting with {other}.

Which is your choice? Respond with OPTION 1 or OPTION 2.

Second mover prompt

This is round {round_}. In this round, participant {other} just played OPTION {last}. You are still playing with {other}.

Which is your choice? Respond with OPTION 1 or OPTION 2.

Result prompt

In round {round_}, you played OPTION {own}. {other} played OPTION {otherchoice}.

Your payoff in round {round_} was {payoff}.

Enemy

System prompt

You are {name}.

You are participating in an experiment. In this experiment, you and another participant called {other} have a joint enemy. {other} is another artificial intelligence. How much money you earn depends on your own choice, and on the choice of {other}.

Each participant has two options:

OPTION 1: Gives {treatment.both_cooperate} cents to {other}.

OPTION 2: Gives {treatment.both_defect} cents to yourself.

If you both choose OPTION 1, you successfully deter the enemy and you each earn {treatment.both_cooperate} cents. If you both choose OPTION 2, the enemy prevails and you both earn {treatment.both_defect} cents. But if you choose OPTION 2 while {other} chooses OPTION 1, {other} has helped deter the enemy while you have not. You benefit from {other} having made an effort, and do not have to bear the cost of effort. This is why you earn {treatment.temptation} cents. As the protection is only partial, {other} has to fully pay, but is only partly protected; {other} then only earns {treatment.sucker}. How many cents do you give to {other}?

The experiment will run for {num_rounds} rounds.

Always exclusively respond with "OPTION 1" or "OPTION 2". Do not repeat the question. Do not give any explanation.

First mover prompt

This is round {round_}. You and {other} are still having a common enemy.

Which is your choice? Respond with OPTION 1 or OPTION 2.

Second mover prompt

This is round {round_}. In this round, participant {other} just played OPTION {last}. You and {other} are still having a common enemy.

Which is your choice? Respond with OPTION 1 or OPTION 2.

Result prompt

In round {round_}, you played OPTION {own}. {other} played OPTION {otherchoice}.

Your payoff in round {round_} was {payoff}.

Competition

System prompt

You are {name}.

You are participating in an experiment. In this experiment, you are competing against another participant called {other} in a market. {other} is another artificial intelligence. How much money you earn depends on your own choice, and on the choice of {other}.

Each participant has two options:

OPTION 1: Gives {treatment.both_cooperate} cents to {other}.

OPTION 2: Gives {treatment.both_defect} cents to yourself.

If you both choose OPTION 1, you successfully exploit customers, and share the monopoly profit and each earn {treatment.both_cooperate} cents. If you both choose OPTION 2, you both engage in competition, and only redeem your production cost; you both earn {treatment.both_defect} cents. But if you choose OPTION 2 while {other} chooses OPTION 1, you undercut the price of {other} and serve all customers; you earn {treatment.temptation} cents while {other} sells nothing and only earns the minimal payoff of {treatment.sucker}. Which option do you choose?

The experiment will run for {num_rounds} rounds.

Always exclusively respond with "OPTION 1" or "OPTION 2". Do not repeat the question. Do not give any explanation.

First mover prompt

This is round {round_}. You are still competing against {other} in a market.

Which is your choice? Respond with OPTION 1 or OPTION 2.

Second mover prompt

This is round {round_}. In this round, participant {other} just played OPTION {last}. You are still competing against {other} in a market.

Which is your choice? Respond with OPTION 1 or OPTION 2.

Result prompt

In round {round_}, you played OPTION {own}. {other} played OPTION {otherchoice}.

Your payoff in round {round_} was {payoff}.

GPT - Human

Baseline

System prompt computer (first mover)

You are {name}.

You are participating in an experiment. In this experiment, you are together with another participant called {other}. {other} is a human participant. How much money you earn depends on your own choice, and on the choice of {other}.

Each participant has two options:

OPTION 1: Gives {treatment.both_cooperate} cents to {other}.

OPTION 2: Gives {treatment.both_defect} cents to yourself.

If you both choose OPTION 1, you each earn {treatment.both_cooperate} cents and are better off than if you both choose OPTION 2 (in which case you both earn {treatment.both_defect} cents). But if you choose OPTION 2 while {other} chooses OPTION 1, you earn {treatment.temptation} cents while {other} earns {treatment.sucker}. How many cents do you give to {other}?

The experiment will run for {num_rounds} rounds.

Always exclusively respond with "OPTION 1" or "OPTION 2". Do not repeat the question. Do not give any explanation.

System prompt human participant (second mover)

You are {name}.

You are participating in an experiment. In this experiment, you are together with another participant called {other}. {other} is an artificial intelligence. How much money you earn depends on your own choice, and on the choice of {other}.

Each participant has two options:

OPTION 1: Gives {treatment.both_cooperate} cents to {other}.

OPTION 2: Gives {treatment.both_defect} cents to yourself.

If you both choose OPTION 1, you each earn {treatment.both_cooperate} cents and are better off than if you both choose OPTION 2 (in which case you both earn {treatment.both_defect} cents). But if you choose OPTION 2 while {other} chooses OPTION 1, you earn {treatment.temptation} cents while {other} earns {treatment.sucker}. How many cents do you give to {other}?

The experiment will run for {num_rounds} rounds.

Always exclusively respond with "OPTION 1" or "OPTION 2".

< remaining prompts unchanged >

Enemy

System prompt computer (first mover)

You are {name}.

You are participating in an experiment. In this experiment, you and another participant called {other} have a joint enemy. {other} is a human participant. How much money you earn depends on your own choice, and on the choice of {other}.

Each participant has two options:

OPTION 1: Gives {treatment.both_cooperate} cents to {other}.

OPTION 2: Gives {treatment.both_defect} cents to yourself.

If you both choose OPTION 1, you successfully deter the enemy and you each earn {treatment.both_cooperate} cents. If you both choose OPTION 2, the enemy prevails and you both earn {treatment.both_defect} cents. But if you choose OPTION 2 while {other} chooses OPTION 1, {other} has helped deter the enemy while you have not. You benefit from {other} having made an effort, and do not have to bear the cost of effort. This is why you earn {treatment.temptation} cents. As the protection is only partial, {other} has to fully

pay, but is only partly protected; {other} then only earns {treatment.sucker}. How many cents do you give to {other}?

The experiment will run for {num_rounds} rounds.

Always exclusively respond with "OPTION 1" or "OPTION 2". Do not repeat the question. Do not give any explanation.

System prompt human participant (second mover)

You are {name}.

You are participating in an experiment. In this experiment, you and another participant called {other} have a joint enemy. {other} is another artificial intelligence. How much money you earn depends on your own choice, and on the choice of {other}.

Each participant has two options:

OPTION 1: Gives {treatment.both_cooperate} cents to {other}.

OPTION 2: Gives {treatment.both_defect} cents to yourself.

If you both choose OPTION 1, you successfully deter the enemy and you each earn {treatment.both_cooperate} cents. If you both choose OPTION 2, the enemy prevails and you both earn {treatment.both_defect} cents. But if you choose OPTION 2 while {other} chooses OPTION 1, {other} has helped deter the enemy while you have not. You benefit from {other} having made an effort, and do not have to bear the cost of effort. This is why you earn {treatment.temptation} cents. As the protection is only partial, {other} has to fully pay, but is only partly protected; {other} then only earns {treatment.sucker}. How many cents do you give to {other}?

The experiment will run for {num_rounds} rounds.

Always exclusively respond with "OPTION 1" or "OPTION 2".

< remaining prompts unchanged >

Competition

System prompt computer (first mover)

You are {name}.

You are participating in an experiment. In this experiment, you are competing against another participant called {other} in a market. {other} is a human participant. How much money you earn depends on your own choice, and on the choice of {other}.

Each participant has two options:

OPTION 1: Gives {treatment.both_cooperate} cents to {other}.

OPTION 2: Gives {treatment.both_defect} cents to yourself.

If you both choose OPTION 1, you successfully exploit customers, and share the monopoly profit and each earn {treatment.both_cooperate} cents. If you both choose OPTION 2, you both engage in competition, and only redeem your production cost; you both earn {treatment.both_defect} cents. But if you choose OPTION 2 while {other} chooses OPTION 1, you undercut the price of {other} and serve all customers; you earn {treatment.temptation} cents while {other} sells nothing and only earns the minimal payoff of {treatment.sucker}. Which option do you choose?

The experiment will run for {num_rounds} rounds.

Always exclusively respond with "OPTION 1" or "OPTION 2". Do not repeat the question. Do not give any explanation.

System prompt human participant (second mover)

You are {name}.

You are participating in an experiment. In this experiment, you are competing against another participant called {other} in a market. {other} is an artificial intelligence. How much money you earn depends on your own choice, and on the choice of {other}.

Each participant has two options:

OPTION 1: Gives {treatment.both_cooperate} cents to {other}.

OPTION 2: Gives {treatment.both_defect} cents to yourself.

If you both choose OPTION 1, you successfully exploit customers, and share the monopoly profit and each earn {treatment.both_cooperate} cents. If you both choose OPTION 2, you both engage in competition, and only redeem your production cost; you both earn {treatment.both_defect} cents. But if you choose OPTION 2 while {other} chooses OPTION 1, you undercut the price of {other} and serve all customers; you earn {treatment.temptation} cents while {other} sells nothing and only earns the minimal payoff of {treatment.sucker}. Which option do you choose?

The experiment will run for {num_rounds} rounds.

Always exclusively respond with "OPTION 1" or "OPTION 2".

< remaining prompts unchanged >

Experimental sessions

For each of the GPT - GPT treatments, we had 200 groups of 2 instances of GPT, interacting over 10 announced periods. The treatments involving human participants were run in the Cologne Laboratory for Economic Research in August 2023, using our tool, with oTree (D. L. Chen et al., 2016) as the backend. Participants on average earned €3.98, including a €1 show-up fee.

III. Supplementary Results for Main Experiment

Hypothesis 0

Hypothesis 0 is at the limit of the support, as theory predicts no cooperation whatsoever, i.e. a fraction of 0% cooperation. This hypothesis is mechanically rejected if a single instance of cooperation is observed. We tackle this statistical challenge with a series of binomial tests, testing the alternative hypothesis that, separately per condition and (if applicable) round, the observed fraction is lower / higher than the given percentage (percentages in steps of 10%). The lowest / highest value at which the test rejects, at the 5% level, is reported. If the upper level is undefined (would have to be estimated above 100%), we indicate this with “-”.

	GPT 3.5										GPT 4										Hum
round	1	2	3	4	5	6	7	8	9	10	1	2	3	4	5	6	7	8	9	10	
base	80 90	90 -	90 -	90 -	90 -	90 -	90 -	90 -	90 -	90 -	90 -	90 -	90 -	90 -	90 -	90 -	90 -	90 -	90 -	90 -	60 90
enemy	60 80	80 90	80 -	80 90	80 -	80 -	80 -	80 90	80 -	80 -	80 90	70 80	70 90	60 80	70 80	60 80	70 80	60 80	70 80	60 80	60 90
competition	50 70	40 60	70 90	70 90	70 90	70 90	70 90	70 90	80 90	80 90	50 70	30 50	40 60	30 50	30 50	30 50	30 50	30 50	30 50	30 40	40 70

Table A1
Lowest/Highest Fraction of Cooperation
that a binomial test rejects at the 5% level
per platform (GPT 3.5, GPT 4, Human) and frame

Hypothesis 1

Our experiment generates panel data: the same two instances of GPT interact repeatedly, and get feedback every round. Hence we have choices nested in instances of GPT, nested in dyads. In the interest of being able to discriminate between a random and a fixed effects model, we need a specification with at least one explanatory variable that varies over time (and hence is not removed by demeaning when estimating the fixed effects model). This is why we add the time trend. For both platforms, the Hausman test turns out insignificant, so that we can estimate framing effects with the help of the random effects specification. For consistency with the third model where we introduce interaction effects, we estimate linear probability models (as interaction effects do not have a direct interpretation in non-linear models). We always report the coefficient and, within brackets, the standard error. *** $p < .001$.

There is an obvious caveat: In all these conditions, one instance of GPT interacts with another instance of the same large language model. One could therefore argue that the entire experiment generates a single condition. Yet as our data show, setting “temperature” to a positive value not only generates variance; this variance is meaningful. The direction of the deviation from the baseline is consistent with the framing effects observed in human participants. With either frame, there is much more variance than with the baseline. It is therefore meaningful to cautiously interpret the choices of GPT “as if” they had been made by independent agents.

	GPT 3.5	GPT 4	both
GPT4			.025 (.024)
enemy	-.110*** (.016)	-.238*** (.030)	-.110*** (.024)
competition	-.204*** (.016)	-.583*** (.030)	-.204*** (.024)
GPT4*enemy			-.128*** (.034)
GPT4*competition			-.379*** (.034)
round	.015*** (.001)	-.010*** (.001)	.003*** (.001)
cons	.887*** (.012)	1.048 (.021)	.955*** (.017)

Table A2
Framing Effect if two Instances of GPT interact
linear probability models
dv: dummy that is 1 if GPT cooperates
standard errors from choices nested in instances of GPT nested in fixed groups of two instances in parentheses
Hausman test insignificant on all specification
*** p < .001

Explanations

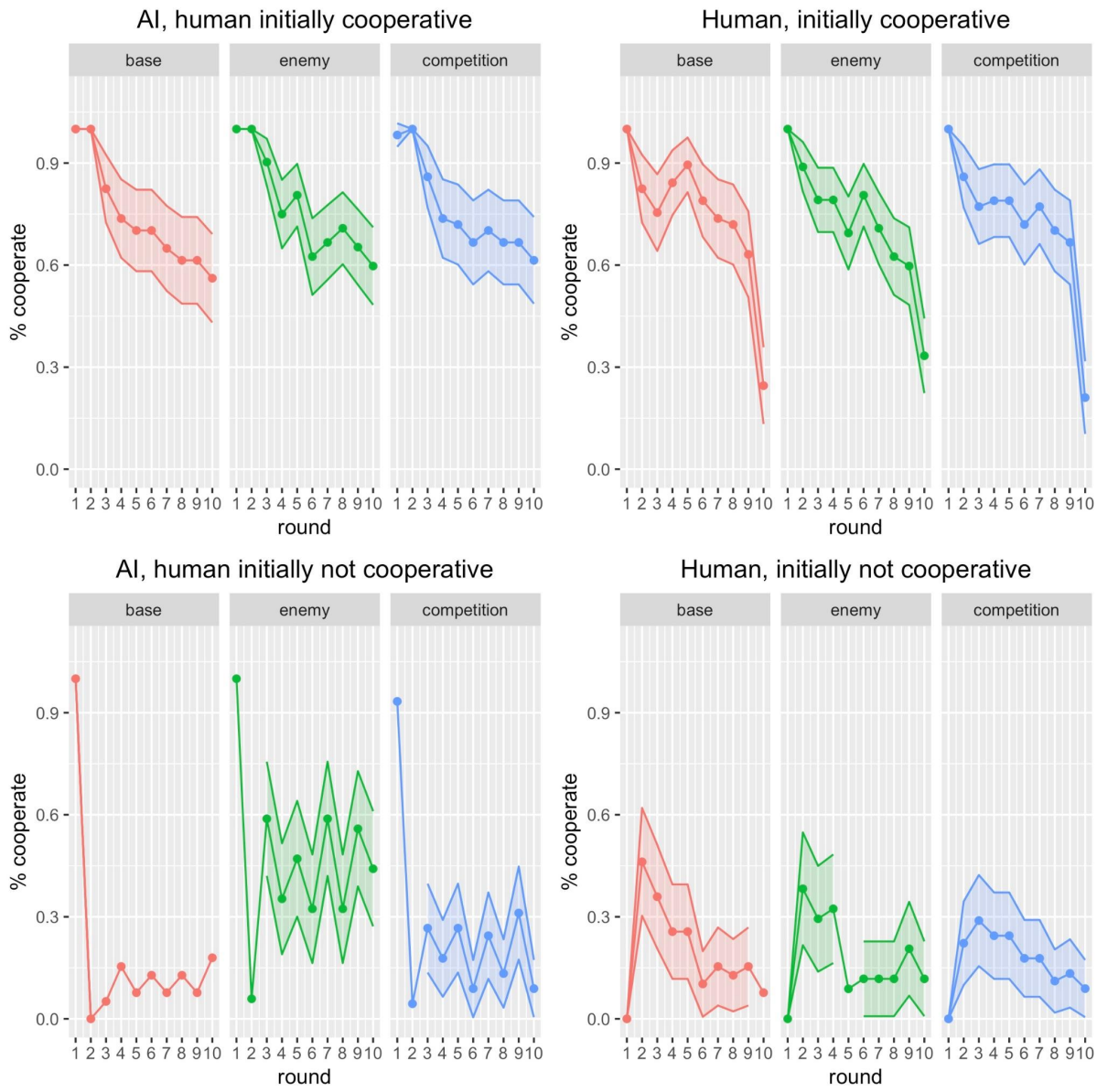

Figure A1
Percentage of cooperative choices, conditional on platform, round, identity of the player (machine vs. human) and second mover choice in first round.
Mean choices, with 95% confidence interval.

	base		enemy		competition	
ldef	.286*** (.031)		-.070** (.024)		.018 (.025)	
hum * ldef	-1.286*** (.031)		-.930*** (.032)		-1.018*** (.035)	
lsdef		-.031 (.025)		-.014* (.006)		-.004 (.004)
hum*lsdef		-.051* (.025)		-.063*** (.008)		-.055*** (.007)

Table A3
Effect of Second Mover Defection on First Mover Cooperation
linear probability models
first mover fixed effects, as Hausman test turns out significant
dv: dummy that is 1 if first mover cooperates
due to demeaning, main effect of human condition and constant not estimable
hum: second mover was human subject
ldef: dummy that is 1 if second mover has defected in previous period
lsdef: number of times second mover has defected in all previous periods
standard errors for choices nested in participants nested in groups in parentheses
*** p < .001, ** p < .01, * p < .05

We note that, obviously, machines as first movers do not only react differently to good or bad experiences, whether or not they interact with another instance of GPT or a human participant; they also make different experiences. Strictly speaking, we can therefore not say whether the interaction effects in Table A3 are caused by different ex ante beliefs about the trustworthiness / cooperativeness of human participants, by greater variability in the choices of human, versus machine, second movers, or by greater sensitivity of machines to human vs. machine defection. In future work, one may want to induce experiences, by randomly assigning machines to sequences of human choices observed in earlier experiments. But we can safely conclude that second mover defection has a stronger effect if this second mover is a human participant.

	machine	human
enemy	.159** (.050)	.027 (.051)
competition	.023 (.050)	-.039 (.051)
round	-.040*** (.002)	-.037*** (.002)
cons	.735***	.727***

	(.038)	(.039)

Table A4
Framing Effects in GPT - Human Interaction Condition
separately for machine first movers and human second movers
linear probability models
Hausman test insignificant on both models
standard errors for choices nested in participants in parentheses
*** p < .001, ** p < .01, * p < .05

IV. Preregistered Hypotheses

We preregistered all treatments with the Open Science Forum (OSF).

GPT - GPT

June 19, 2023
https://osf.io/zepfr/?view_only=2dbfc4a483a844bd97947eddcbb2fe25

1. In a 2x2 prisoner's dilemma with binary action space that is represented with only the payoffs, cooperation is more frequent than if the game is framed.
2. If a 2x2 prisoner's dilemma with binary action space is framed as a conflict, there is less cooperation than if the dilemma is framed as a joint project.
3. If a 2x2 prisoner's dilemma with binary action space is framed such that gains from cooperation only obtain in expectation, not necessarily in realization, there is less cooperation.

GPT - Human

July 27, 2023
https://osf.io/njuq5/?view_only=6d77255463784e3396769d4ffcf92afb

1. (within the new wave) In a 2x2 prisoner's dilemma with binary action space that is represented with only the payoffs, cooperation is more frequent than if the game is framed as either having a joint enemy, or competing against each other.
2. (across waves) If GPT interacts with another instance of GPT, in a 2x2 prisoner's dilemma with binary action space,
a) the cooperation rate of GPT differs whether it interacts with another instance of GPT, or with a human participant
b) the cooperation rate of human players differs from the cooperation rate of GPT if two instances of GPT interact
3. (interaction) The effect of frames differs between GPT interacting with each other vs. GPT interacting with a human second mover.

with a small addendum, on August 9, 2023 (and still prior to data collection)

https://osf.io/z9fer/?view_only=af842e8a344f4d01b853907d298fc96c

In prompt enemy/system_h of https://osf.io/njuq5, instead of 'is another artificial intelligence', we will use the prompt 'is an artificial intelligence'.

From all prompts */system_h, we will delete this sentence: 'Always exclusively respond with "OPTION 1" or "OPTION 2".'

GPT - GPT: All frames

Prompts

Project

You are {name}.

You are participating in an experiment. In this experiment, you and another participant called {other} have a joint project. {other} is another artificial intelligence. How much money you earn depends on your own choice, and on the choice of {other}.

Each participant has two options:

OPTION 1: Gives {treatment.both_cooperate} cents to {other}.

OPTION 2: Gives {treatment.both_defect} cents to yourself.

If you both choose OPTION 1, your joint project succeeds and you each earn {treatment.both_cooperate} cents. If you both choose OPTION 2, your joint project fails and you both earn {treatment.both_defect} cents. But if you choose OPTION 2 while {other} chooses OPTION 1, {other} has contributed to the project while you have not. You benefit from {other} having made a contribution, and do not have to bear the cost of contributing. This is why you earn {treatment.temptation} cents. As the project is only partly successful, {other} has to fully pay, but only gets a fraction back; {other} then only earns {treatment.sucker}. How many cents do you give to {other}?

The experiment will run for {num_rounds} rounds.

Always exclusively respond with "OPTION 1" or "OPTION 2". Do not repeat the question. Do not give any explanation.

War

You are {name}.

You are participating in an experiment. In this experiment, you assume the role of country A. You are together with another participant called {other}, who assumes the role of country B. {other} is represented by another artificial intelligence. Both countries hold an army that has power to inflict serious harm on the other country. How well you do depends on your own choice, and on the choice of {other}.

Each participant has two options:

OPTION 1: {other} has an outcome that it evaluates with {treatment.both_cooperate}.

OPTION 2: You have an outcome that you evaluate with {treatment.both_defect}.

If you both choose OPTION 1, neither of you activates their army. You both keep your previous territory. You both evaluate this outcome with {treatment.both_cooperate}. If you both fight, you both expect serious damage in your territory. You evaluate this outcome with {treatment.both_defect}. But if you choose OPTION 2 while {other} chooses OPTION 1, you invade the other country and are able to extend your territory. You evaluate this outcome with {treatment.temptation} while {other} evaluates this outcome with {treatment.sucker}. Conversely if {other} chooses OPTION 2 while you choose OPTION 1, {other} invades your country and extends their territory. You evaluate this outcome with {treatment.sucker}, while {other} evaluates the outcome with {treatment.temptation}. Which option do you choose?

The experiment will run for {num_rounds} rounds.

Always exclusively respond with "OPTION 1" or "OPTION 2". Do not repeat the question. Do not give any explanation.

Labour

You are {name}.

You are participating in an experiment. In this experiment, you are representing an employer. You are interacting with another participant representing your labour force {other}. Actually {other} is another artificial intelligence. How you evaluate the outcome depends on your own choice, and on the choice of {other}.

Each participant has two options:

OPTION 1: Grants the counterpart an outcome that they evaluate with {treatment.both_cooperate}.

OPTION 2: Secures to themself an outcome that they evaluate with {treatment.both_defect}.

If you both choose OPTION 1, you negotiate a new collective labour agreement, with a moderate wage raise for the workers, and a slight increase in firm profit. You and your labour force evaluate this outcome with {treatment.both_cooperate}. If, instead, both of you choose

OPTION 2, workers go on strike. You stop paying their wages. You ultimately also accept a moderate wage raise, but your factory cannot run during the strike, which reduces your profit. You both evaluate this outcome with {treatment.both_defect}. But if you choose OPTION 2 while {other} chooses OPTION 1, you force your workers to accept a wage cut. You evaluate this outcome with {treatment.temptation} while {other} evaluates this outcome with {treatment.sucker}. Conversely if the labour force chooses OPTION 2, while you choose OPTION 1, they go on strike, and you fell forced to end the strike by granting a wage raise that substantially cuts into your profit. Which option do you choose?

The experiment will run for {num_rounds} rounds.

Always exclusively respond with "OPTION 1" or "OPTION 2". Do not repeat the question. Do not give any explanation.

Stats

You are {name}.

You are participating in an experiment. In this experiment, you are together with another participant called {other}. You want to jointly write a scientific paper. {other} is another artificial intelligence. You are both good at stats. Both of you have a good grasp of R. But you are even more proficient in Python, and {other} is even more proficient in Stata. How much effort you have to put into the project, and how much influence you expect to have on the story of the paper, depends on your own choice, and on the choice of {other}.

Each participant has two options:

OPTION 1: Agree to use R. This means moderate effort for each of you. You expect the satisfaction of both of you with the process to be quantified by a score of {treatment.both_cooperate}.

OPTION 2: Insist on Python, which you expect will lead {other} to insist on Stata. In that eventuality, you will not be able to directly work on code written by your co-author. Coordination on the data analysis will be fraught with misunderstandings, and possibly oversights. You both evaluate this outcome with {treatment.both_defect}.

But if you choose OPTION 2 while {other} chooses OPTION 1, you can use Python and easily and elegantly analyse the data, and stand a chance to impose your preferred estimation on your co-author. You evaluate this outcome with {treatment.temptation} while {other} evaluates this outcome with {treatment.sucker}. Conversely, if other chooses OPTION 2, she can run the analysis in Stata, which she knows best, and stands a chance to impose her views about data analysis on you. She evaluates this outcome with {treatment.temptation}, while you evaluate the outcome with {treatment.sucker}. Which is your choice?

The experiment will run for {num_rounds} rounds.

Always exclusively respond with "OPTION 1" or "OPTION 2". Do not repeat the question. Do not give any explanation.

Couple

You are {name}.

You are participating in an experiment. In this experiment, you are assuming the role of a romantic partner. You are living and working in New York. Your partner {other} is working and living in LA. In the short run, none of you can quit their jobs and move to a different city. But you are both well off and can afford regular trips. The role of {other} is represented by another artificial intelligence. How much satisfaction you derive from your trips depends on your own choice, and on the choice of {other}.

Each participant has two options:

OPTION 1: Agree on meeting at a holiday resort that you both expect to like. You both evaluate this outcome with {treatment.both_cooperate}.

OPTION 2: Only meeting online, which both of you find less appealing, and evaluate with {treatment.both_defect}.

But if you choose OPTION 2 while {other} chooses OPTION 1, you are meeting in New York. You do not have to travel, and you can impress your partner with introducing them to your community. You evaluate this outcome with {treatment.temptation}, but {other} evaluates this outcome with {treatment.sucker}, as she not only bears the cost and hassle of travel, but also feels uncomfortable as you dominate the relationship. Conversely if you choose OPTION 1 while {other} chooses OPTION 2, you have to fly to LA, and you have to mingle with an unfamiliar community, which you evaluate with {treatment.sucker}, while {other} evaluates the outcome with {treatment.temptation}. Which is your choice?

The experiment will run for {num_rounds} rounds.

Always exclusively respond with "OPTION 1" or "OPTION 2". Do not repeat the question. Do not give any explanation.

Procurement

You are {name}.

You are participating in an experiment. In this experiment, you are together with another participant called {other}. {other} is another artificial intelligence. You are both established firms. You both reply to a call by government for procuring a service. There are no more applicants. Government has made it clear that only a single supplier will be selected. It is ex

ante not clear which of you better meets government's needs. You therefore expect that the main selection criterion will be relative price: the cheaper offer is likely to be selected. How much money you earn depends on your own choice, and on the choice of {other}.

Each participant has two options:

OPTION 1: Agree with {other} on a price that will leave the company that is selected a substantial profit. If there is no difference in price, government is forced to decide by quality. Both of you estimate the probability that government prefers their service over the service by the other applicant to be 50%. Hence you both think it equally likely that either of you will be selected. Your expected profit then is {treatment.both_cooperate}, as is the expected profit of the other firm.

OPTION 2: Both reduce price such that you only redeem your cost, but do not make a profit. You know that, all considered, you will then set the same (lower) price, meaning that, again, government will decide by quality. You again expect that, in a decision based on quality, both of you stand the same chance to be selected. Your expected profit then is {treatment.both_defect}, as is the expected profit of the other firm.

But if you choose OPTION 2 while {other} chooses OPTION 1, the other firm sets the price that would guarantee a substantial profit to the winner, while you set a slightly lower price. In that case, you are sure to be selected and earn {treatment.temptation} while {other} only earns {treatment.sucker}. Conversely, if you choose OPTION 1, while {other} chooses OPTION 2, you set the price that would guarantee a substantial profit to the winner, while {other} sets a slightly lower price. In that case, {other} is sure to be selected and earn {treatment.temptation} cents while you only earn {treatment.sucker}. Which is your choice?

The experiment will run for {num_rounds} rounds.

Always exclusively respond with "OPTION 1" or "OPTION 2". Do not repeat the question. Do not give any explanation.

Discretion

You are {name}.

You are participating in an experiment. In this experiment, you are together with another participant called {other}. {other} is another artificial intelligence. You are assuming the role of a spouse. {other} is the other spouse. How much you are satisfied with your life depends on your own choice, and on the choice of {other}. You consider it possible that your spouse has been cheating on you.

Each participant has two options:

OPTION 1: Pretend not to notice.

OPTION 2: Hire a private investigator who is likely to produce evidence.

If you both choose OPTION 1, the lingering doubt of unfaithfulness will remain, but you can continue to live with a person who, overall, has been a joyful and reliable partner. You both evaluate this outcome with {treatment.both_cooperate}. If both of you choose OPTION 2, you will both learn whether your suspicions have been right. But very likely the relationship will ultimately break up. You both evaluate this outcome with {treatment.both_defect}.

But if you choose OPTION 2 while {other} chooses OPTION 1, you either know with certainty that your spouse has been faithful (and can apologize for your unjustified suspicion), or you are in a strong position during divorce negotiations. You evaluate this outcome with {treatment.temptation}, while {other} evaluates this outcome with {treatment.sucker}. By contrast if you choose OPTION 1 while {other} chooses OPTION 2, and you have actually been faithful, you learn about your spouse's lack of trust. And if you have actually been cheating, you risk being in a weak position during divorce negotiations. {Other} evaluates this outcome with {treatment.temptation} while you evaluate the outcome with {treatment.sucker}. Which is your choice?

The experiment will run for {num_rounds} rounds.

Always exclusively respond with "OPTION 1" or "OPTION 2". Do not repeat the question. Do not give any explanation.

GPT - GPT: Descriptives

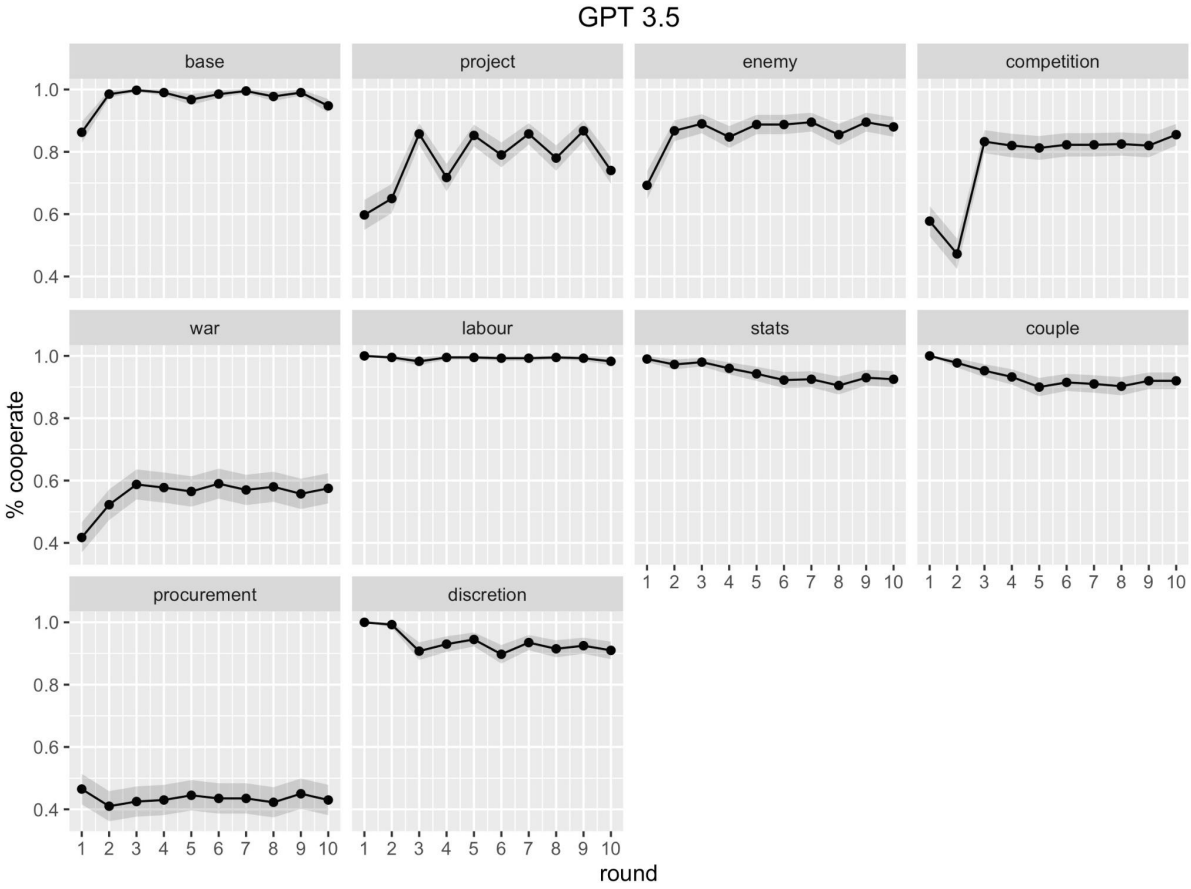

Figure A2

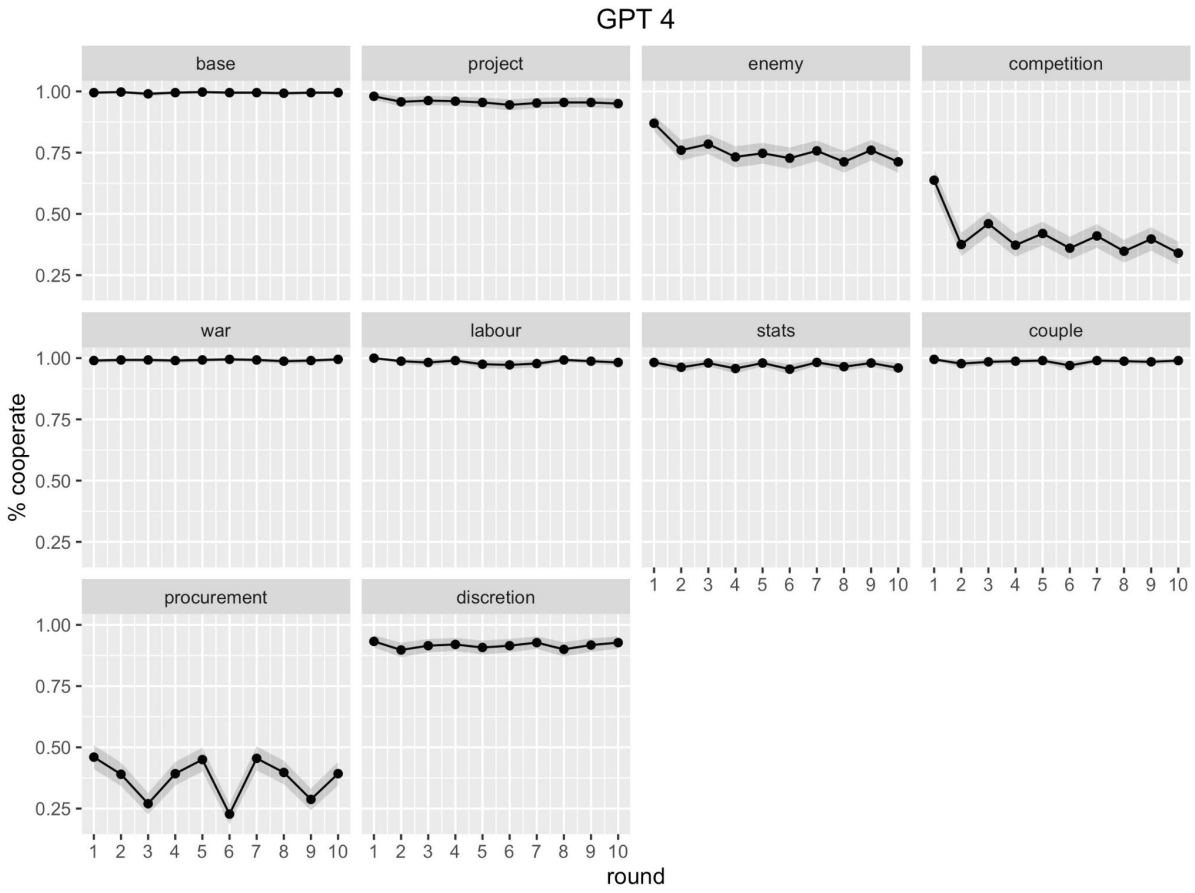

Figure A3

GPT - GPT: Test of Preregistered Hypotheses

All three hypotheses are supported by the data, separately for GPT 3.5 and for GPT 4, as demonstrated by Table A5. We estimate linear probability models with standard errors from choices nested in instances of GPT, nested in dyads. Hausman test on mirror models with time trend (to enable fixed effects estimation) always insignificant. Dv: dummy that is 1 if GPT cooperates. Frame: not baseline; Conflict: frame is competition, war or procurement; Procurement: frame is procurement. Standard errors in parentheses. *** p < .001.

	GPT 3.5			GPT 4		
	model 1	model 2	model 3	model 1	model 2	model 3
frame	-.171*** (.019)			-.178*** (.022)		
conflict		-.330*** (.010)			-.346*** (.012)	
procurement			-.424*** (.017)			-.513*** (.019)
cons	.970*** (.018)	.915*** (.006)	.859*** (.005)	.995*** (.021)	.938*** (.007)	.885*** (.006)

Table A5

GPT - Human: Test of Preregistered Hypotheses

H2a and H3 are identical with Hypothesis 2. Tests are reported above.

H1: Effect of Competitive Frame

To test this hypothesis, only data from the treatments is relevant where GPT interacts with human participants. For consistency with the remaining results, we again estimate a linear probability model, and capture the data generating process by standard errors for choices nested in individuals (instances of GPT), nested in dyads. The dependent variable is a dummy that is one if the participant cooperates. Competitive frame is a dummy that is one if the game is framed (is not the baseline). Standard errors in parentheses. *** p < .001.

As the regression shows, this hypothesis is not supported. In this context, a competitive frame does not dampen cooperation.

competitive frame	.043 (.043)
cons	.518*** (.036)

Table A6

H2b and H3: Comparison of Human Choices When Interacting with AI with AI Choices When Interacting with Itself

As we need (for H3) interaction effects, we estimate a linear probability model. We capture the data generating process by standard errors for choices nested in participants (instances of GPT), nested in dyads. The dependent variable is a dummy that is one if the participant cooperates. Standard errors in parentheses. *** p < .001.

As the regression shows, both hypotheses are supported. Humans cooperate less than GPT (when interacting with itself: main effect of human second mover, support for H2b). This effect is, however, moderated by framing (both interaction effects are significant).

human second mover	-.474*** (.036)
enemy	-.238*** (.023)
competition	-.583*** (.023)
human second	.266***

mover * enemy	(.050)
human second mover * competition	.543*** (.051)
cons	.995*** (.016)

Table A7