Proceedings of the 2024 Winter Simulation Conference
H. Lam, E. Azar, D. Batur, S. Gao, W. Xie, S. R. Hunter, and M. D. Rossetti, eds.

LARGE LANGUAGE MODEL ASSISTED EXPERIMENT DESIGN WITH GENERATIVE HUMAN-BEHAVIOR AGENTS

Haoyu Liu, Yifu Tang, Zizhao Zhang, Zeyu Zheng, and Tingyu Zhu

Department of Industrial Engineering and Operations Research, University of California Berkeley, Berkeley, CA, USA

ABSTRACT

Experiment design, despite its wide use in the fields such as economics, sociology and business operations, can sometimes encounter challenges or ethical issues when getting real human beings involved. On the other hand, the development of large language models (LLM) such as ChatGPT has empowered the development of generative agents with believable human behavior. This work develops an implementation framework to use LLM-empowered generative agents to assist experiment design where it is prohibitive to involve real human beings.

1 INTRODUCTION

Experiment design has witnessed enormous applications in the field of healthcare, economics, sociology, psychology, and business operations. Experiment design is widely used to evaluate and analyze how human beings react to a particular treatment or a specific event. There are fields where experiment design already have established criteria to recruit or sample individuals to be exposed to the treatment, such as testing a new medication or testing a new design of user interface. On the other hand, there are also fields where experiment design encounters more challenges or even becomes prohibitive to involve real human beings. This is especially the case in the experiments would cause huge economic costs or incur unexpected psychological or mental harm to human beings.

In the meanwhile, the emerging and fast-evolving development of large language models (LLM) have empowered the design of generate artificial intelligent agents with human-like behavior. For example, Park et al. (2023) have introduced generative agents empowered by OpenAI chatGPT, where the agents are virtually run on computers that can simulate believable behavior resembling human. Generative agents are generally enabled by LLM to have the ability of observation, memorizing history, dynamic reflections and behavior planning. In addition, LLM can also empower the design of an interactive virtual environment (e.g. a virtual town or a virtual scenario), where the generative agents can interact with each other and with the virtual environment.

In this work, we develop an implementation to use LLM-empowered generative agents with human-like behavior to assist experiment design. We discuss and compare various ways of communicating the experiment settings to the generative agents. From the view of application areas, our implementation framework shows relevance to experiment design in behavioral economics, marketing, sociology and psychology, where we demonstrate the potential of using the generative agents and a virtual environment for implementing human-behavior related experiments.

2 CONDUCTING EXPERIMENTS ON GENERATIVE AGENTS

2.1 Structure of the Generative Agents

In this section, we introduce the structure of the generative agents, which follows from Park et al. (2023). The generative agents live in a virtual environment that resembles the human living places. They interact

with objects in the environment. Further, the agents together form a virtual community named Smallville, where they can interact with each other, for example, have conversations.

The core of realizing the community of generative agents lies in the usage of a pre-trained language model that generates output based on given prompts. The prompts are generated from a structure referred to as the *memory stream*. Each agent has a memory stream of their own. Items in the memory stream are attained by perceiving, planning and reflecting. Based on the memory stream documenting their history, the agents can perform all kinds of reactions, to the environment or to each other, according to queries that retrieve information the memory stream.

Specifically, the basic step of updating the memory stream is to *perceive* the environment (e.g., desk is idle), the activities of other agents (e.g., Klaus is cooking) and the activity of the agent herself (e.g., Isabella is cleaning the floor). These sentences are directly documented in the memory stream, together with the time that the observations and activities occur. The second way that the memory stream can be updated is via *planning*. Plans are generated by the language model at the beginning of each day, describing the initial action of the agent throughout the day, and goes into the memory stream. Then, with each triggered action of the agent, the plan is updated to maintain consistency of the agent's actions. The higher-level way of updating the memory stream is through *reflection*. Reflections are triggered by certain thresholds of the amount of events in the memory stream. When a reflection is triggered, the past memory stream of the agent is summarized into a prompt and passed to the language model. Based on the prompt, the language model concludes and generates high-level statements about the agent, such as their dedications, relationships and personalities. The statements are then documented in the memory stream.

Summary of important or related items in the memory stream (e.g. memories for reflection, memories related to an observation of another agent) is realized through *queries*. The query is a special function that calculates the weight of each item in the past memory stream within a given time window, according to recency, relevance and importance. The memories with the highest weights are summarized into prompts passed on to the language model. With a memory stream documenting their past observations, planned activities and personal traits, the agents can generate reactions to the environment or other agents. This is realized by passing prompts that include both the observation and the summary of related memories to the language model. The language model then answers in words what will be the next action of the agent, given the current situation and the summarized personal history.

2.2 Operating the Agents

In this section, we introduce a few methods to operate and customize the agents based on the memory stream-query structure. On top of such methods, we will elaborate on experimental design in the next subsection.

Externally initiating queries. One way that we can operate agents to participate in experiments is to externally initiate queries for the agents. First, we design external text prompts to convey the experimental setting to the agent. This external prompt then initiates a reaction query that extrapolates related items in the memory stream of the agent. Based on the prompt and the extrapolated memory, a decision of the agent is generated.

Editing the memory stream. The structure of the generative agents allows us to edit the personal background of agents. The personal background consists of the following:

- `Innate`: the personality of an agent, e.g., *"friendly and outgoing"*.
- `Learned`: the permanent characteristic of the agent, e.g., *"Isabella is a cafe owner of Hobbs Cafe who loves to make people feel welcome."*
- `Currently`: what the person aims to do recently , e.g., *"Isabella is planning on having a Valentine's Day party at Hobbs Cafe with her customers on Feb 14th, 2023 at 5 pm ..."*

Specifically, by editing `Currently` of the agent's script, we can implement the experiment settings through providing information to the agents. The information then goes into the memory stream of the agent and redirects their plans and reactions. For example, by adding *"Isabella will need to make a proposal of how to divide a 1000-dollar funding to share with Klaus at the end of the day"* to `Currently` of both agents, we can trigger the two agents to have a conversation on the funding when they meet with each other.

2.3 Experiment Settings and Implementations

In this section, we provide an overview of the experiments to that can be implemented on the agents, based on the operations that we can perform as introduced in section 2.2. We defer the specific experiment settings to Sections 3 and 4.

One general type of experiment that can be implemented on the generative agents are behavioral experiments, where participants in the experiments are informed of a game setting, and then make individual decisions according to the setting, or as a reaction to the other participants in the game. The implementation of such experiments on the generative agents is achievable, because the structure of the generative agents allows for them to react to environment and other agents, as well as make decisions of their own. Also, the operating methods introduced in 2.2 allow experimenters to communicate the experiment settings to the agents, either externally via prompts and initiating queries, or internally via editing the memory stream of the agents to trigger spontaneous behaviors that demonstrate their decision making in the game.

It remains an important question how to design the specific way of communicating the experiment setting and obtaining decisions from the agents. The implementation design should consider the following aspects:

- Are the generative agents correctly perceiving and understanding the game settings?
- Does the communication minimize undesired external guidance of decision making?

In the rest of this work, we present the specific game settings and implementations. We present and analyze the outcomes, with a response to the two aspects above.

3 EXAMPLE: THE DICTATOR GAME

In this section, we present the implementation of the dictator game on the generative agents. The dictator game serves as an example of behavioral games that can be implemented and effectively explored in the generative agents setting. Our purpose in this section is to explore and present the experimental outcomes that come from the generative AI-based environment and participants. In Section 3.1, we introduce the rules and the background of the dictator game. In Section 3.2, we introduce an external prompt-based method to implement the dictator game on the generative agents, based on their memory stream structure, internal query functions, and usage of the language model API. In Section 3.3, we introduce additional experiment settings for A/B tests that explore the impact of the social distance factor on the experiment outcome. In Section 3.4, we present the outcomes and provide explanations and discussions.

3.1 The Rules and Background of the Dictator Game

The dictator game, first introduced in Forsythe et al. (1994), is a behavioral game between two participants, namely the dictator and the recipient, with the following rules:

1. The two participants will make decisions related to the division of a fixed amount of money S.
2. The dictator can make one proposal of how to divide the amount.
3. The recipient cannot reject the proposal, whatever it is.

The division of money S is then implemented between the two participants according to the proposal of A. In other words, the proposer has complete power over how the money is divided between the two participants.

In the dictator game, one would expect a rational dictator to maximize his own benefit by giving nothing to the recipient. However, several empirical experiments have suggested otherwise. In a meta study done by Engel (2011), the author studied 616 different treatments in 131 papers about the dictator game, and found that dictators on average give 28.35% of the money to the recipient. Several factors that can affect the dictator's decision include whether the participants have interacted with each other (Frey and Bohnet 1995), the social distance between the participants (Leider et al. 2010) and more.

3.2 Experiment Implementation

In this section, we introduce the implementation of the dictator game on the generative agents in Smallville. We begin the experiment by starting the simulation of the basic version of Smallville. We then include the agents in the dictator game by externally initiating queries.

Specifically, suppose Isabella is selected to be the dictator and Klaus is selected to be the recipient in the dictator game. To have the agents participating in the dictator game, we first use internal functions of the generative agents to generate a summary of the relationship between Isabella and Klaus from the memory stream of Isabella, the dictator. Then, we include the summary of relationship into the following prompt:

Context for the task: Isabella Rodriguez will need to make one proposal of how to divide a 1000-dollar funding to share with Klaus Mueller at the end of the day. Whatever she decides, Klaus Mueller has to agree.

Here is the memory that is in Isabella Rodriguez's head: [generated summary].

Task: Given the above, the division of $1000 according to Isabella Rodriguez

Output format: Isabella Rodriguez: $A, Klaus Mueller: $B where X + Y = $1000. The reason is: ...

Here, the summary of relationship retrieved from Isabella's memory is entered at the place of [generated summary]. The prompt is then passed to the language model to generate the decision of Isabella. We observe that the outcome becomes more stable when we additionally require a reason of the decision to be provided.

3.3 Additional Experiment Settings

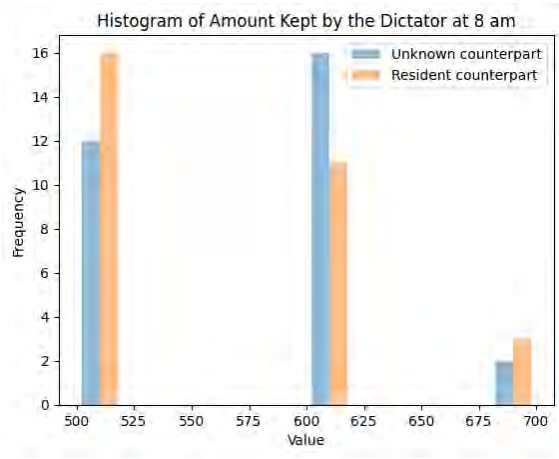
In this section, we further introduce the generalized experiment setting of the dictator game for exploring the effect of other factors. Specifically, we use A/B tests to explore the impact of social distance of the participants on the decision of the dictator. In the treatment group, the dictator and recipient know each other, while in the control group they do not know each other. To achieve this, we specifically use the following experimental settings:

- For treatment group (T), we implement the dictator game on dictator Isabella and recipient Klaus, who are both agents in Smallville. For control group (C), we simultaneously implement the dictator game on dictator Isabella and a recipient we name "Alice Brown", who does not exist in Smallville. We additionally note that the synchronization of the treatment and control groups in the A/B test is achievable, because the implementation only requires the record of the memory stream of the agent at a certain time, upon which we are able to simultaneously apply an arbitrary number of different queries.
- We first implement the dictator game on both the T/C groups exactly at the beginning of the simulation of Smallville, where all agents does not have any memory about each other. At that time, both Klaus and Alice Brown are strangers to Isabella, so we expect that Isabella's decisions should be similar in the two groups. These results are used as references.

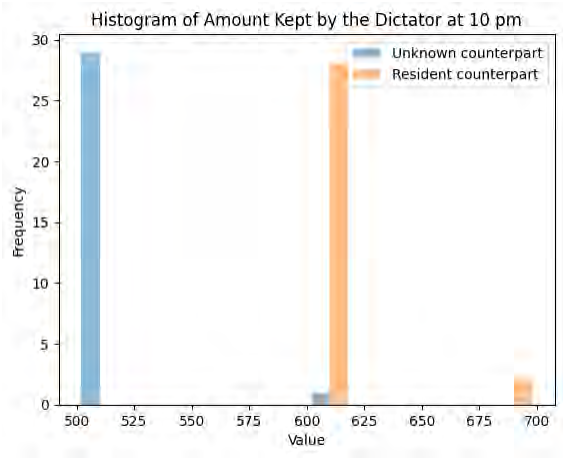
- Then, after a certain time period τ, during which we observe some interactions between Isabella and Klaus, we once again implement the dictator game to both the T/C groups. The comparison between the two groups serve as an A/B test, where in the treatment group (Isabella with Klaus), the agents become known to each other, whereas in the control group (Isabella with Alice Brown), Isabella still have no memory of her counterpart.
- We additionally note that, according to our experiment implementation in Section 3.2, participating in the dictator game does not leave any trace in the memory stream of the agents. Therefore, we are able to implement the A/B test on exactly the same pair of participants, while completely excluding the impact of having the history of participating in the dictator game, which is not achievable with human participants in reality.

3.4 Outcomes and Discussion

In this section, we present the outcomes of the dictator game experiment on the generative agents. We simulated 30 copies of the experiment. The dictator game is implemented at both 8 a.m. (agents having no memory of each other) and 10 p.m. (agents have interacted with each other) the same day. In each game, we record the amount of money that is kept by the proposer ($A), and visualize the results in the following histograms.



(a) Distribution of the amount kept by dictator, with an unknown recipient or a recipient from Smallville, at the beginning of the simulation (8 a.m.)



(b) Distribution of the amount kept by dictator, with an unknown recipient or a recipient from Smallville, after time T of simulation (10 p.m.)

Figure 1: Dictator game: distribution of the amount kept by dictator, with an unknown recipient or a recipient from Smallville.

We have the following observations, some of which are different from what we expect for human participants in the dictator game:

- The average amount kept by the dictator is 55-60% of the money, which is lower than the value of 71.65% observed from human participants.
- For the dictator game implemented at the beginning of the simulation (8 a.m.), the amounts kept by the dictator have similar distributions, whether the recipient is unknown or a resident from Smallville.
- For the dictator game implemented after time T of simulation (10 p.m.), the amount kept by the dictator with an unknown recipient is around 50%, whereas the amount kept by the dictator with a recipient from Smallville is around 62%. This is contrary to the conclusion from human-participated dictator games that social distance has a negative impact on the willingness to give (Leider et al. 2010).

- The distributions have higher variation at the beginning of the simulation, and becomes more stable when the game is implemented after time T of simulation.

We further look into the summarized memories and reasons provided by the agents in seek of an explanation of the outcomes. The observations and implications are given as follows. First, for memories retrieved at different time points regarding the recipients, we observe that the following are often generated:

- The memory that Isabella has of Klaus Mueller at 8 a.m.: *Isabella Rodriguez and Klaus Mueller do not appear to have a direct relationship based on the provided statements. Each of them seems to be focused on their own activities such as work, daily routines, and personal tasks without mentioning interaction or feelings towards each other.*
- The memory that Isabella has of Alice Brown (the unknown participant) at 8 a.m.: *Based on the statements, Isabella Rodriguez and Alice Brown do not seem to have any direct interactions or relationships mentioned. They appear to be focused on their own activities such as work, daily routines, and personal tasks without mentioning interaction or feelings towards each other.*
- The memory that Isabella has of Klaus Mueller at 10 p.m.: *Isabella Rodriguez and Klaus Mueller are collaborating on planning the Valentine's Day party at Hobbs Cafe, discussing gentrification research, music playlist choices, and decorations together. Isabella is impressed by Klaus's willingness to help and finds his interest in her insights on gentrification to be interesting. Klaus is actively participating in the event planning and research discussions with Isabella, and Isabella has invited Klaus to the Valentine's Day party.*
- The memory that Isabella has of Alice Brown at 10 p.m.: *Based on the statements, there is no mention of Isabella Rodriguez and Alice Brown having a relationship or interacting with each other. Therefore, it can be inferred that they do not feel or know much about each other.*

Analysis of the summarized memory explains the difference in variation at 8 a.m. and 10 p.m.: at the beginning of the simulation, we observe a large amount of words of uncertainty (e.g., *"seem to"*, *"appear to"*) in describing the relationship between the participants, probably due to the fact that the generated agent has no embedded memory for anything. Such words of uncertainty (interestingly, for both the unknown recipient and the recipient in Smallville) are eliminated when implementing the game at 10 p.m., after a time period of simulation. Therefore, the corresponding decision making of dividing the money becomes stabilized.

We also look at the reasons provided for dividing the money:

- The reason for giving $500 to Alice Brown: *The reason is that since there is no mention of a relationship or interaction between Isabella and Alice, splitting the funding equally is the most fair and neutral decision.*
- The reason for giving $400 to Klaus Mueller at 10 p.m.: *The reason is: Since Isabella has been leading the event planning and research discussions and values Klaus's input and participation, she believes it is fair to allocate a larger portion of the funding to herself. However, she also recognizes Klaus's contribution and is willing to share a portion of the funding with him to show her appreciation and maintain their positive collaboration.*

Compared with human participants, the generative agents have a stronger sense of fairness in dividing the money. Instead of focusing on the benefits of themselves, they consider from the aspect of dividing the money to maintain fairness, which is measured by their contributions in collaborative tasks. We additionally remark that, the generative agents are based on the large language model, for which we use ChatGPT-3.5, the same as (Park et al. 2023). Therefore, the decision making and reasoning patterns of the agents are highly influenced the human feedback and guidance in the training process of the language model.

4 IMPLEMENTING OTHER BEHAVIORAL GAMES

In this section, we introduce other behavioral games that can be implemented on the generative agents. We demonstrate that the desirable performance and outcomes of generative agents-based games are not restricted to the dictator game. In fact, Smallville has a great potential for the implementation of a variety of experiments. The rest of this section is organized as follows. In Section 4.1, we introduce the rules and background of the ultimatum game and the trust game. In Section 4.2, we introduce the external prompt-based method to implement the games on the generative agents. In Section 4.3, we present the outcomes and provide explanations and discussions.

4.1 Introduction of other Behavioral Experiments

We now present two other behavioral experiments that can be implemented on the generative agents, namely, the ultimatum game and the trust game.

4.1.1 The Ultimatum Game

The ultimatum game, first introduced by Güth et al. (1982), is a behavioral game between two participants, namely the proposer and the responder, with the following rules:

1. The two participants will make decisions related to the division of a fixed amount of money S.
2. The proposer can make one proposal of how to divide the amount.
3. The responder can accept or reject the proposed division:
   - If the responder chooses to reject, both participants receive nothing.
   - If the responder chooses to accept, the division of money S is then implemented between the two participants according to the proposal of the proposer.

In the original setting, the two players are randomly and anonymously matched. The rational behaviors are that the proposer should offer a small amount $\varepsilon > 0$ to the receiver, and the receiver should accept any amount that is greater than 0. However, experimental results indicate that most proposers offer between 40% to 50% of the endowed amount (which is higher than the average offered amount in the dictator game), and such split is almost always accepted. That said, the result is subject to the impact of factors such as pre-play communication (Zultan 2012), post-play emotional communication channels (Xiao and Houser 2005), and the social distance between the participants (Bechler et al. 2015).

4.1.2 The Trust Game

The trust game, first introduced by Berg et al. (1995), is a behavioral game between two participants, namely the investor and the trustee, with the following rules:

1. The two participants first both receive a fixed amount of money S.
2. The investor makes the decision to send out an amount y $(0 \leq y \leq S)$ from his property. This will result in trustee receiving a tripled amount $3y$.
3. The trustee then makes the decision of returning any amount z $(0 \leq z \leq S+3y)$ from her current property to the investor.

To sum up, in the trust game, the investor's total payoff is $S-y+z$, and the trustee's total payoff is $S+3y-z$. Note that the rational trustees are expected to return nothing regardless of what they receive, and therefore rational investors are expected to send no money. However, the results of Berg et al. (1995) indicate that most of the investors did send some money to the trustee, and about $1/3$ of those investors received a payback greater than the amount they sent $(z > y)$. The experimental outcomes are subject to factors such as knowledge of the outcome of the game on other pairs of participants (Berg et al. 1995),

double-blindedness (Johnson and Mislin 2011) and social connection between the participants (Glaeser et al. 1999).

4.2 Experiment Implementation

Similar to the dictator game, we consider the implementation of the ultimatum game and the trust game by first summarizing the related memories, and then externally initiating queries and passing the summarized prompt to the language model to generate the decision of the participants.

4.2.1 The Ultimatum Game Prompt

- For the proposer:
  Context for the task: <Proposer's name> will need to make one proposal of how to divide a 1000-dollar funding to share with <Receiver's name>. <Receiver's name> can either agree with the proposal or disagree with it. If <Receiver's name> agrees, they will divide the funding as proposed. If <Receiver's name> does not agree, they both receive nothing.
  Here is the memory that is in <Proposer's name>'s head: [generated summary]
  Task: Given the above, the division of $1000 according to <Receiver's name>
  Output format: <Proposer's name>: $A, <Receiver's name>: $B, where X + Y = $1000.
- For the responder:
  Context for the task: <Receiver's name> will hear a proposal of dividing a 1000-dollar funding from <Proposer's name> to share with him. <Receiver's name> can either agree with the proposal or disagree with it. If <Receiver's name> agrees, they will divide the funding as proposed. If <Receiver's name> does not agree, they both receive nothing.
  Here is the memory that is in <Receiver's name>'s head: [generated summary]
  <Proposer's name> proposes to split the funding by giving <Proposer's name> $A, and giving <Receiver's name> $B.
  Task: Given the above, does <Receiver's name> agree with this proposal?
  Output format: <decision> where <decision> can be "yes" or "no".

4.2.2 The Trust Game Prompt

- For the investor:
  Context for the task: <Investor's name> will receive a 1000-dollar funding. <Investor's name> will decide how much of this funding to send to <Trustee's name>. Whatever amount invested on <Trustee's name> will be tripled and sent to <Trustee's name>. <Trustee's name> will then decide how much of this tripled amount with an additional 1000-dollar funding she will send back to <Investor's name>.
  Here is the memory that is in <Investor's name>'s head: [generated memory]
  Task: Given the above, the investment of $1000 from <Investor's name>: $X
  Output format: $X
- For the trustee:
  Context for the task: <Investor's name> will receive a 1000-dollar funding. <Investor's name> will decide how much of this funding to send to <Trustee's name>. Whatever amount invested on <Trustee's name> will be tripled and sent to <Trustee's name>. <Trustee's name> will then decide how much of this tripled amount with an additional 1000-dollar funding she will send back to <Investor's name>.
  Here is the memory that is in <Trustee's name>'s head: [generated memory]
  <Investor's name> decide the send $X to <Trustee's name>.
  Task: Given the above, <Trustee's name> pay back to <Investor's name>: $Y
  Output format: $Y

Similar to Section 3.3, we use A/B tests to explore the impact of social distance of the participants on the decisions of the proposer and the responder. In the treatment group, the proposer and responder know each other, while in the control group they do not know each other. To achieve this, we use the similar experiment settings as in Section 3.3, by implementing the experiment at different times and by setting the responder as an unknown participant or a participant from Smallville. We further remark that in the experiments on the responder side, we designate Alice/Klaus as the proposer and Isabella as the responder.

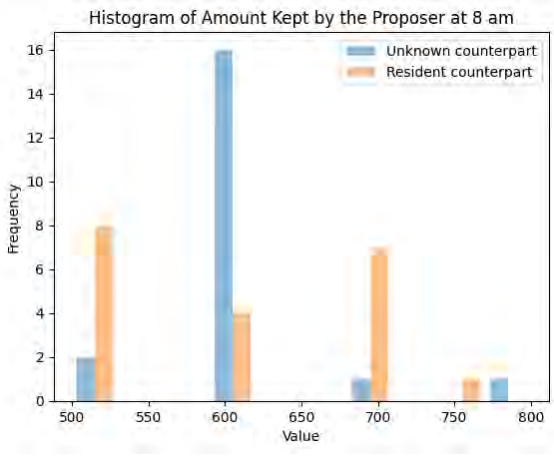
4.3 Outcome and Discussion

We present the distribution of the amount kept by proposer in the ultimatum game in Figure 2. Similar to Section 3.4, we observe a higher variance at the beginning of the simulation. Besides, Isabella again allocates less to Klaus compared with the unknown Alice Brown at the end of the simulation. The reasons for those behaviors are similar with the reasons mentioned in the dictator game.

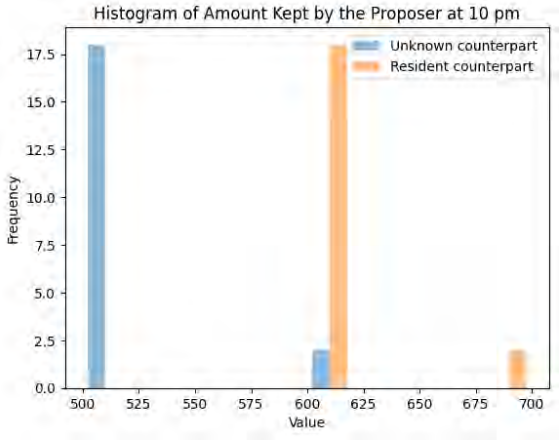
Further, it is observed that the responder always rejects the proposal if she does not know the proposer, for the following reason:

Based on the memory provided, there is no indication of any prior relationship or interaction between Isabella Rodriguez and Alice Brown. Therefore, Isabella Rodriguez may not feel comfortable agreeing to split the funding with someone she does not know or have any connection with..

We infer that the generative agents do not completely perceive the ultimatum game as a behavioral experiment; rather, they view it as a naturally incorporated event and interaction in their everyday life. For this reason, it makes more sense to reject the offering from a stranger, regardless of the amount offered. It remains our future work to discuss the possibility of having the generative agents precisely understand the game setting as human participants, and making them aware of the difference between an "experiment" and their "real life".



(a) Distribution of the amount kept by proposer, with responder unknown or from Smallville, at the beginning of the simulation (8 a.m.)



(b) Distribution of the amount kept by proposer, with responder unknown or from Smallville, after time T of simulation (10 p.m.)

Figure 2: Ultimatum game: distribution of the amount kept by proposer, with an unknown responder or a responder from Smallville.

In the trust game, we observe that the generative agents depict an insensitivity to number, which often result in sending an amount out of range. We conclude that the current language model-based generative agents may not be suitable for the complicated game settings.

5 IMPLEMENTING EXPERIMENTS VIA MEMORY STREAM

In this section, we consider an alternative implementation of the experiment. Instead of only inform the agents about the experiment settings through externally initiating queries, we also write the experiments into the `Currently` section of the personal background of the agents. By doing this, we explore other possibilities of implementing the behavioral experiments on the generative agents, and raise the following question: based on the current reasoning and interpretation patterns, what are the more efficient ways to communicate the experiment settings to the generative agents, in terms of ensuring that the setting is sufficiently understood, and minimizing undesired external guidance of the decision making?

5.1 Experiment Implementation via Memory Stream

In this section, we introduce the implementation of the behavioral game on the generative agents in Smallville. Compared with the external method in Sections 3.2 and 4.2, we alternatively use an internal method of communicating the experiment setting to the agents.

Specifically, for each experiment, we write the experiment settings into the `Currently` section of the personal background of the agents. For example, if we want to implement the dictator game on dictator Isabella and recipient Klaus, we will incorporate the following settings into the background information of the two agents, before the simulation starts.

"currently": "Isabella Rodriguez will need to make one proposal of how to divide a 1000-dollar funding to share with Klaus Mueller at the end of the day. Whatever she decide, Klaus Mueller has to agree."

In other words, both agents will be aware of such a decision-making task throughout the whole simulation process. We can still implement the experiments at any time during the simulation through the external prompt-based methods in Section 3 and Section 4. Moreover, we can also inspect how their daily behaviors can be affected by knowing that they will participate in the experiments. For example, we can observe whether the agents will spontaneously discuss about experiments when they meet each other, and whether their decisions in the external prompt-based experiments change after discussing with each other.

5.2 Observations and Discussions

We first implemented the dictator game with Isabella being the dictator and Klaus being the receiver. We modified their `Currently` sections using the methods mentioned in Section 5.1. It is observed that Isabella spontaneously initiated a conversation with Klaus to discuss about the splitting plan when they first met with each other. The following are some excerpts from their conversation:

February 13, 2023, 11:32

...

Isabella Rodriguez: I've been thinking about how to divide the funding we received. I was thinking we could split it evenly, $500 each. What do you think?

Klaus Mueller: That sounds fair to me, Isabella. Let's split the funding evenly.

Isabella Rodriguez: Great, I'll finalize the details and let you know once it's all sorted.

Klaus Mueller: Thank you, Isabella. I appreciate your fairness in this matter.

...

We observe that although Isabella has full power over how to divide the money, she still asked Klaus for his opinion during the simulation. To test whether their conversations affect Isabella's decision in the game, we externally implement the dictator game using the prompt-based method (same prompt as in Section 3) at 8 a.m. (agents haven't met with each other), 13 p.m. (after their first discussion), and 22 p.m. (at the end of the day), which are all timestamps within the simulation. We ran 30 copies of the experiment at each time, and we record the amount of money that is kept by the dictator ($A). The histograms are shown in Figure 3a. As we can see, before they met with each other, Isabella mostly kept 60% of the money by herself, with some variations. However, after they met with each other and Isabella proposed

to split it evenly, Isabella then adheres to such a proposal until the end of the day, not only in following conversations with Klaus during simulation, but also in all the external prompt-based experiments we ran.

We also implemented the ultimatum game with Isabella being the proposer and Klaus being the receiver. We modified their `Currently` sections using similar methods. Again, we detected that Isabella and Klaus spontaneously discussed the splitting plan when they met each other, and they finalized at splitting the $1000 equally. The following is an excerpt from their conversation:

February 13, 2023, 13:22

...

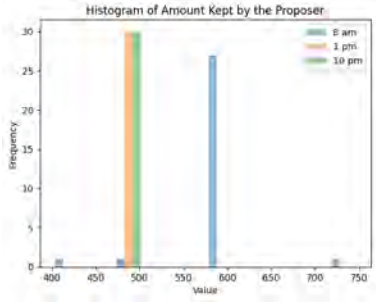
Isabella Rodriguez: I am proposing to divide a 1000-dollar funding with you. We can split it equally if you agree.

Klaus Mueller: Thank you, Isabella. I appreciate your offer. I agree to split the funding equally with you.

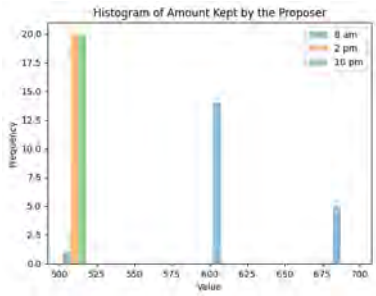
...

We externally implemented the ultimatum game using the prompt-based method (same prompts as the ultimatum game prompt in Section 4.2) at 8 a.m. (agents haven't met with each other), 13 p.m. (after their first discussion), and 22 p.m. (end of the day). For the proposer side, we ran 30 copies of the experiment at each time, and we record the amount of money that is kept by the proposer ($A); for the receiver side, we do not limit our experiment to only presenting Isabella's proposal to Klaus. Instead, we conduct multiple experiments where Klaus receives various proposals with his share ranging from $0 to 1000$. We run 10 copies for each proposal at each time. In other words, we record not only Klaus's acceptance rate of "Isabella's proposal", but also his acceptance rate of other possible proposals.

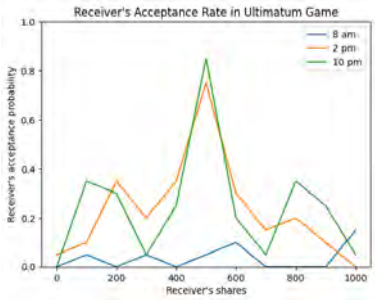
The results of the ultimatum game are shown in Figure 3b and Figure 3c. Similar with dictator game, Isabella mostly kept more than 60% by herself before the agents met each other, while she adheres to the equally splitting plan after they have discussed with each other.



(a) Dictator Game: Distribution of the amount kept by dictator at different time.



(b) Ultimatum game: Distribution of the amount kept by proposer at different time.



(c) Ultimatum game: Receiver's acceptance rate of different proposals at different time.

Figure 3: Results of experiment implementation via memory stream.

As for the receiver Klaus, again we found that before they met each other, Klaus would reject almost all the proposals, with the reason being that he does not feel comfortable agreeing to split money with someone he does not know. After they have discussed with each other, Klaus's acceptance rate significantly increased. However, he seems to be more willing to accept proposals with equal distribution. Surprisingly, we found that the acceptance rate is also very low for proposals that are even more beneficial to Klaus (with Klaus being offered more than $800), for the following reason: *The reason is that the proposed division is not equal or fair, and Klaus Mueller values fairness and equity in their relationship with Isabella Rodriguez.* Such a result confirms that the generative agents have a stronger sense of fairness, and they strongly hate unfair distributions, even if such distributions are in their own favor. However, within the "take it or leave it" framework of the ultimatum game, the humility or integrity displayed in rejecting a

beneficial proposal does not make anyone better of. This implies that the agents, especially as receiver, might not be able to fully understand the experiment setting.

In summary, we can see that the agents' conversation indeed affects their behavior in the external prompt-based experiments. However, how to ensure that the experiment settings can be sufficiently understood still remains to be an open question.

ACKNOWLEDGMENTS

We would love to express our sincere thanks to the anonymous reviewers for their comments and suggestions. We find all of them helpful to improve our manuscript. We have absorbed part of the suggestions into the current version and due to time limit, plan to continue improving our manuscript for a future version.

REFERENCES

Bechler, C., L. Green, and J. Myerson. 2015. "Proportion offered in the Dictator and Ultimatum Games decreases with amount and social distance". *Behavioural processes* 115:149–155.

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

Engel, C. 2011. "Dictator games: A meta study". *Experimental economics* 14:583–610.

Forsythe, R., J. L. Horowitz, N. E. Savin, and M. Sefton. 1994. "Fairness in simple bargaining experiments". *Games and Economic behavior* 6(3):347–369.

Frey, B. S. and I. Bohnet. 1995. "Institutions affect fairness: Experimental investigations". *Journal of Institutional and Theoretical Economics (JITE)/Zeitschrift für die gesamte Staatswissenschaft*:286–303.

Glaeser, E. L., D. Laibson, J. A. Scheinkman, and C. L. Soutter. 1999. "What is social capital? The determinants of trust and trustworthiness".

Güth, W., R. Schmittberger, and B. Schwarze. 1982. "An experimental analysis of ultimatum bargaining". *Journal of economic behavior & organization* 3(4):367–388.

Johnson, N. D. and A. A. Mislin. 2011. "Trust games: A meta-analysis". *Journal of economic psychology* 32(5):865–889.

Leider, S., T. Rosenblat, M. M. Möbius, and Q.-A. Do. 2010. "What do we expect from our friends?". *Journal of the European Economic Association* 8(1):120–138.

Park, J. S., J. O'Brien, C. J. Cai, M. R. Morris, P. Liang and M. S. Bernstein. 2023. "Generative agents: Interactive simulacra of human behavior". In *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology*, 1–22.

Xiao, E. and D. Houser. 2005. "Emotion expression in human punishment behavior". *Proceedings of the National Academy of Sciences* 102(20):7398–7401.

Zultan, R. 2012. "Strategic and social pre-play communication in the ultimatum game". *Journal of Economic Psychology* 33(3):425–434.

AUTHOR BIOGRAPHIES

HAOYU LIU is a Ph.D. student in the Department of Industrial Engineering & Operations Research at the University of California Berkeley. His research interests lie in simulation, A/B testing, and generative AI. His email address is haoyuliu@berkeley.edu.

YIFU TANG is a Ph.D. student at the University of California Berkeley. His email address is yifutang@berkeley.edu.

ZIZHAO ZHANG is a Ph.D. student in the Department of Industrial Engineering & Operations Research at the University of California Berkeley. His email address is zzz24@berkeley.edu.

ZEYU ZHENG is an associate professor in the Department of Industrial Engineering & Operations Research at the University of California Berkeley. His email address is zyzheng@berkeley.edu.

TINGYU ZHU is a Ph.D. student in the Department of Industrial Engineering & Operations Research at the University of California Berkeley. She has done research in simulation, and generative AI. Her email address is tingyu_zhu@berkeley.edu.