

Automated Social Science: Language Models as Scientist and Subjects

By

Benjamin S. Manning

B.A., Washington University in St. Louis, 2017

M.P.P., Harvard University, 2021

SUBMITTED TO THE DEPARTMENT OF MANAGEMENT IN PARTIAL
FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF

MASTER OF SCIENCE IN MANAGEMENT RESEARCH

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

September 2024

©2024 Benjamin S. Manning. All rights reserved.

The author hereby grants to MIT a nonexclusive, worldwide, irrevocable, royalty-free license to exercise any and all rights under copyright, including to reproduce, preserve, distribute and publicly display copies of the thesis, or release the thesis under an open-access license.

Authored by: Benjamin S. Manning
Department of Management
July 31th, 2024

Certified by: John J. Horton
Department of Management
Thesis supervisor

Accepted by: Eric So
Professor, Global Economics and Finance
Faculty Chair, MIT Sloan PhD Program

Automated Social Science: Language Models as Scientist and Subjects

By

Benjamin S. Manning

Submitted to the Department of Management on July 30th, 2024 in partial fulfillment of the requirements for the Degree of Master of Science in Management Research

ABSTRACT

We present an approach for automatically generating and testing, in silico social scientific hypotheses. This automation is made possible by recent advances in large language models (LLM), but the key feature of the approach is the use of structural causal models. Structural causal models provide a language to state hypotheses, a blueprint for constructing LLM-based agents, an experimental design, and a plan for data analysis. The fitted structural causal model becomes an object available for prediction or the planning of follow-on experiments. We demonstrate the approach with several scenarios: a negotiation, a bail hearing, a job interview, and an auction. In each case, causal relationships are both proposed and tested by the system, finding evidence for some and not others. We provide evidence that the insights from these simulations of social interactions are not available to the LLM purely through direct elicitation. When given its proposed structural causal model for each scenario, the LLM is good at predicting the signs of estimated effects, but it cannot reliably predict the magnitudes of those estimates. In the auction experiment, the in silico simulation results closely match the predictions of auction theory, but elicited predictions of the clearing prices from the LLM are inaccurate. However, the LLM's predictions are dramatically improved if the model can condition on the fitted structural causal model. In short, the LLM knows more than it can (immediately) tell.

Thesis supervisor: John J. Horton

Title: Associate Professor, Information Technology

List of Figures

1	An overview of the automated system.	9
2	Experimental design and fitted SCM for “two people bargaining over a mug.”	14
3	Experimental design and fitted SCM for “a judge is setting bail for a criminal defendant who committed 50,000 dollars in tax fraud.” . . .	15
4	Experimental design and fitted SCM for “a person is interviewing for a job as a lawyer.”	17
5	Experimental design and fitted SCM for “3 bidders participating in an auction for a piece of art starting at fifty dollars.”	19
6	Comparison of the LLM’s predictions to the theoretical predictions and a subset of experimental results for the auction scenario.	22
7	Comparison of the true and misspecified SCMs.	26
8	Incorrect causal structure identified by the GES algorithm for the tax fraud experiment.	28
A.1	Example agents generated by the system for “two people bargaining over a mug”	44
A.2	Menu of interaction protocols for the system to choose from for a given scenario.	46
A.3	Valid graphical interpretations of the same natural language hypothesis.	51
A.4	Fitted SCM with interaction terms for “two people bargaining over a mug.”	53
A.5	Fitted SCM with interaction terms for “a judge is setting bail for a criminal defendant who committed 50,000 dollars in tax fraud.” . . .	54
A.6	Fitted SCM with interaction terms for “a person is interviewing for a job as a lawyer.”	55
A.7	Fitted SCM with interaction terms for “3 bidders participating in an auction for a piece of art starting at fifty dollars.”	56
A.8	Fitted SCM for auction with bidder’s reservation prices and second highest bid as exogenous variables.	59
A.9	Fitted SCM for auction and second highest bid as exogenous variables.	59
A.10	Comparison of the LLM’s predictions to the theoretical predictions and all experimental results for the auction scenario.	60
A.11	Prompt used to elicit LLM predictions for the Predict- $\hat{\beta}$ task.	61

List of Tables

- A.1 GPT-4's predictions for the path estimates for the experiments in Section 3 at temperature 0. 57
- A.2 GPT-4's predictions for the path estimates for the experiments in Section 3 at temperature 1. 58
- A.3 Example of the information generated for each variable in an SCM. . . 62

1 Introduction

There is much work on efficiently estimating econometric models of human behavior but comparatively little work on efficiently generating and testing those models to estimate. Previously, developing such models and hypotheses to test was exclusively a human task. This is changing as researchers have begun to explore automated hypothesis generation through the use of machine learning.¹ But even with novel machine-generated hypotheses, there is still the problem of testing. A potential solution is simulation. Researchers have shown that Large Language Models (LLM) can simulate humans as experimental subjects with surprising degrees of realism.² To the extent that these simulation results carry over to human subjects in out-of-sample tasks, they provide another testing option (Horton, 2023). In this paper, we combine these ideas—automated hypothesis generation and automated *in silico* hypothesis testing—by using LLMs for both purposes. We demonstrate that such automation is possible. We evaluate the approach by comparing results to a setting where the real-world predictions are well known and test to see if an LLM can be used to generate information that it cannot access through direct elicitation.

The key innovation in our approach is the use of structural causal models to organize the research process. Structural causal models are mathematical representations of cause and effect (Pearl, 2009b; Wright, 1934) and have long offered a language for expressing hypotheses.³ What is novel in our paper is the use of these models as a blueprint for the design of agents and experiments. In short, each explanatory variable describes something about a person or scenario that has to vary for the effect to be identified, so the system “knows” it needs to generate agents or scenarios that

¹A few examples include generative adversarial networks to formulate new hypotheses (Ludwig and Mullainathan, 2023), algorithms to find anomalies in formal theories (Mullainathan and Rambachan, 2023), reinforcement learning to propose tax policies (Zheng et al., 2022), random forests to identify heterogenous treatment effects (Wager and Athey, 2018), and several others (Buyalskaya et al., 2023; Cai et al., 2023; Enke and Shubatt, 2023; Girotra et al., 2023; Peterson et al., 2021).

²(Aher et al., 2023; Argyle et al., 2023; Bakker et al., 2022; Binz and Schulz, 2023b; Brand et al., 2023; Bubeck et al., 2023; Fish et al., 2023; Mei et al., 2024; Park et al., 2023)

³In an unfortunate clash of naming conventions, some disciplines have alternative definitions for the term “structural” when discussing formal models. Here, structural does not refer to the definition traditionally used in economics. See Appendix B for a more detailed explanation.

vary on that dimension—a straightforward transition from stated theory to experimental design and data generation. Furthermore, the structural causal model offers a pre-specified plan for estimation (Haavelmo, 1943, 1944; Jöreskog, 1970).

We built an open-source computational system implementing this structural causal model-based approach. The system can automatically generate hypotheses, design experiments, run those experiments on independent LLM-powered agents, and analyze the results. We use this system to explore several social scenarios: (1) two people bargaining over a mug, (2) a bail hearing for tax fraud, (3) a lawyer interviewing for a job, and (4) an open ascending price auction with private values for a piece of art. We allow the system to propose the hypotheses for the first two scenarios and then run the experimental simulations without intervention. For (3) and (4), we demonstrate the system’s ability to accommodate human input at any point by selecting the hypotheses ourselves and editing some of the agents, but otherwise, we allow the system to proceed autonomously.

Though yet to be optimized for novelty, the system formulates and tests multiple falsifiable hypotheses—from which it generates several findings. The probability of a deal increases as the seller’s sentimental attachment to the mug decreases, and both the buyer’s and the seller’s reservation prices matter. A remorseful defendant is granted lower bail but is not so fortunate if his criminal history is extensive. However, the judge’s case count before the hearing—which was hypothesized to matter—does not affect the final bail amount. The candidate passing the bar exam is the only important factor in her getting the job. Neither the candidate’s height nor the interviewer’s friendliness affect the outcome.

The auction scenario is particularly illuminating. An increase in the bidders’ reservation prices causes an increase in the clearing price, a clearing price that is always close to the second-highest reservation amongst the bidders. These simulation results closely match the theory (Maskin and Riley, 1985) and what has been observed empirically (Athey et al., 2011).

None of the findings from the system’s experiments are “counterintuitive,” but it is important to emphasize they were the result of empiricism, not just model introspection. However, this does raise the question of whether the simulations

are even necessary. Instead of simulation, could an LLM simply do a “thought experiment” about the proposed *in silico* experiment and achieve the same insight? To test this idea, we describe the experiments that will be simulated and ask the LLM to predict the results—both the path estimates and point predictions. The path estimates being the coefficients in the linear structural causal model. To make this concrete, suppose we had the simple linear model $y = X\beta$ to describe some scenario, and we ran an experiment to estimate $\hat{\beta}$. We describe the scenario and the experiment to the LLM and ask it to predict y_i given a particular X_i (a “predict- y_i ” task). Separately, we ask it to predict $\hat{\beta}$ (a “predict- $\hat{\beta}$ ” task). Later, we examine how the LLM does on the predict- y_i task when it has access to the fitted structural causal model (i.e., $\hat{\beta}$).

In the predict- y_i task, we prompt the LLM to predict the outcome y_i given each possible combination of the X_i ’s from the auction experiment. Direct elicitation of the predictions for y_i in the auction experiment is wildly inaccurate. The predictions are even further from the theory than the empirical results.

In the predict- $\hat{\beta}$ task, the LLM is asked to predict the fitted structural causal model’s path estimates for all four experiments, provided with contextual information about each scenario. On average, the LLM predicts the path estimates are 13.2 times larger than the experimental results. Its predictions are overestimates for 10 out of 12 of the paths, although they are generally in the correct direction.

We repeat the predict- y_i task, but this time, we provide the LLM with the experimental path estimates. For each X_i , we fit the structural causal model using all but the i th observation and then ask the LLM to predict y_i given X_i and this fitted model. In this “predict- $y_i|\hat{\beta}_{-i}$ ” task, the predictions are far better than in the predict- y_i task without the fitted model. The mean squared error is six times lower, and the predictions are much closer to those made by the theory. The process of experimentation can improve the LLM’s capacity to predict its behavior.

We design and implement an approach to automated social science because LLMs possess latent information about human behavior that can be systematically explored and extracted (Burns et al., 2023; Scherrer et al., 2024). These models are trained to predict the next token in a sequence of text from a massive human-generated corpus.

From this straightforward objective, the models develop a remarkably sophisticated model of the world, at least as captured in text (Bubeck et al., 2023; Gurnee and Tegmark, 2023; Patel and Pavlick, 2021). And while there are many situations where LLMs are imperfect proxies for humans (Cheng et al., 2023; Santurkar et al., 2023), there is also a growing body of work demonstrating that experiments with LLMs as subjects can predict human behavior in never-before-seen tasks (Binz and Schulz, 2023a; Li et al., 2024). Rapid and automated exploration of these models’ behavior could be a powerful tool to efficiently generate new insights about humans. Our contribution is to demonstrate that it is possible to create such a tool: a system that can simulate the entire social scientific process without human input at any step.

The remainder of this paper is structured as follows: Section 2 provides an overview of the system. Section 3 provides some results generated using our system. Section 4 explores an LLM’s capacity to predict the results in Section 3. Section 5 discusses the advantages of using SCMs over other methods for studying causal relationships in simulations of social interactions. The paper concludes in Section 6.

2 Overview of the system

To perform this automated social science, we needed to build a system. The system intentionally mirrors the experimental social scientific process. These steps are, in broad strokes:

1. Social scientists start by selecting a topic or domain to study (e.g., misinformation, auctions, bargaining, etc).
2. Within the domain, they identify interesting outcomes and some causes that might affect the outcomes. These variables and their proposed relationships are the hypotheses.
3. They design an experiment to test these hypotheses by inducing variation in the causes and measuring the outcomes.
4. After designing the experiment, social scientists determine how they will analyze the data in a pre-analysis plan.

5. Next, they recruit participants, run the experiment, and collect the data.
6. Finally, they analyze the data per the pre-analysis plan to estimate the relationships between the proposed causes and outcomes.

While any given social scientist might not follow this sequence exactly, whatever their approach may be, the first two steps should always guide the later steps—the development of the hypothesis guides the experimental design and model estimation. Of course, many social scientists must often omit steps 3-5 when a controlled experiment is impossible, but they typically have some notion of the experiment they would like to run.

To build our system, we formalized a sequence of these steps analogous to those listed above. The system executes them autonomously. Since the system uses AI agents instead of human subjects, it can *always* design and execute an experiment.

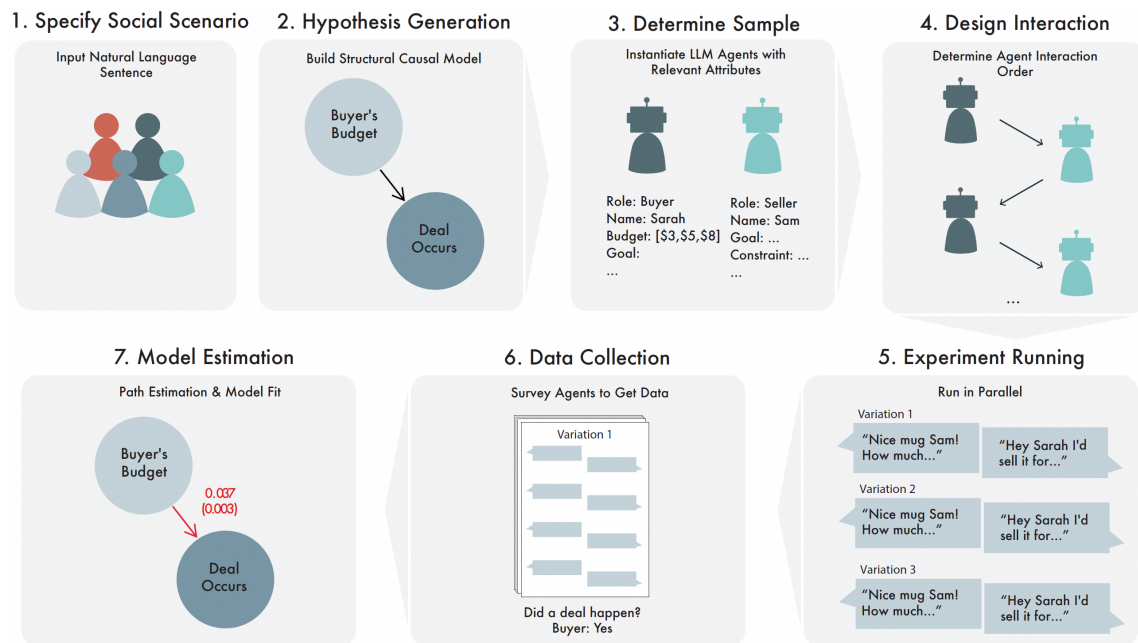
Structural causal models (SCM) are essential to the system’s design because they make unambiguous causal statements, which allow for unambiguous estimation and experimental design.⁴ Algorithms can determine precisely which variables must be exogenously manipulated to identify the effect of a given cause (Pearl, 2009b). If the first two steps in the social scientific process are building the SCM, the last four can be directly determined subject to the SCM. Such precision makes automation possible as the system only relies on a few key early decisions. Otherwise, the space of possible choices for the latter steps would explode, rendering automation infeasible.

The system is implemented in Python and uses GPT-4 for all LLM queries. Its decisions are editable at every step. The overview in this section is a high-level description of the system, but there are many more specific design choices and programming details in Appendix A. For the purposes of most readers, the high-level overview should be sufficient to understand the system’s process, the results we present in Section 3, and the additional analyses in Sections 4 and 5.

⁴We use simple linear SCMs unless stated otherwise. This assumption is not necessarily correct but offers an unequivocal starting point to generate hypotheses. Functional assumptions can be tested by comparing fitted SCMs with various forms using data generated from a known causal structure. Section B in the appendix provides a more detailed discussion of SCMs.

The system takes as input a scenario of social scientific interest: a negotiation, a bail decision, a job interview, an auction, and so on. Starting with (1) this input, the system (2) generates outcomes of interest and their potential causes, (3) creates agents that vary on the exogenous dimensions of said causes, (4) designs an experiment, (5) executes the experiment with LLM-powered agents simulating humans, (6) surveys the agents to measure the outcomes, (7) analyzes the results of the experiment to assess the hypotheses, which can be used to plan a follow-on experiment. Figure 1 illustrates these steps, and we will briefly explore each in greater depth.

Figure 1: An overview of the automated system.



Notes: Each step in the process corresponds to an analogous step in the social scientific process as done by humans. The development of the hypothesis guides the experimental design, execution, and model estimation. Researchers can edit the system’s decisions at any step in the process.

The first step is to generate hypotheses as SCMs based on the social scenario, the scenario being the only necessary input to the system. This is done by querying an LLM for the relevant agents and then interesting outcomes, their potential causes, and methods to operationalize and measure both.⁵ We use `Typewriter` text to

⁵When we say “query an LLM,” we mean this literally. We have written a prompt that the

indicate example output from the system. Suppose the social scenario is “two people bargaining over a mug.” The LLM may generate whether a deal occurs for the mug as an outcome and operationalize it as a binary variable with a ‘‘1’’ when a deal occurs and a ‘‘0’’ when it does not. It then generates potential exogenous causes and their operationalizations: the buyer’s budget, which is operationalized as the buyer’s willingness to pay in dollars. The system takes each of these variables, constructs an SCM (see the second step in Figure 1), and stores the relevant information about the operationalizations associated with each variable.⁶⁷ From this point on, the SCM serves as a blueprint for the rest of the process, namely the automatic instantiation of agents, their interaction, and the estimation of the linear paths.

The second step is to construct the relevant agents—the Buyer and the Seller in Figure 1, step 3. By “construct,” we mean that the system prompts independent LLMs to be people with sets of attributes. These attributes are the exogenous dimensions of the SCM, dimensions that are varied in each simulation. I.e., the different experimental conditions. For the current scenario, a Budget is provided to the buyer that can take on values of {\$5, \$10, \$20, \$40}. By simulating interactions of agents that vary on the exogenous dimensions of the SCM, the data generated can be used to fit the SCM.

Next, the system generates survey questions to automatically gather data about the outcomes from the agents once each simulation is complete. An LLM can easily generate these questions when provided with information about the variables in the SCM (e.g., asking the buyer, “Did a deal happen?”). All LLM-powered agents in

system provides to an LLM with the scenario. For example, the prompt used to generate the relevant agents is: *In the following scenario: “{scenario}”, who are the individual human agents in a simple simulation of this scenario?* Where “{scenario}” is replaced with the scenario of interest. The LLM then returns a list of agents, which are stored in the system and can be used in follow-on prompts, prompts that generate things like the outcomes and proposed causes. The system contains over 50 pre-written prompts used to execute the entire scientific process.

⁶The system generates several other pieces of information about each variable, which help guide the experimental design and data analysis. See Appendix A for further details.

⁷The graph in the second step of Figure 1 is a directed acyclic graph (DAG). For convenience, we will use DAGs to represent SCMs throughout the paper and assume they imply a simple linear model unless stated otherwise.

our system have “memory.” They store what happened during the simulation in text, making it easy to ask them questions about what happened.

Fourth, the system determines how the agents should interact. LLMs are designed to generate text in sequence. Since independent LLMs power each agent, one agent must finish speaking before the next begins. This necessitates a turn-taking protocol to simulate the conversation. We programmed a menu of six ordering protocols, from which an LLM is queried to select the most appropriate for a given scenario. We describe each protocol in Appendix A, and they are presented in Figure A.2, but in our bargaining scenario with two agents, there are only two possible ways for the agents to alternate speaking. In this case, the system selects: **speaking order**: (1) **Buyer**, (2) **Seller**, (step 4, Figure 1). The speaking order can be flexible in more complex simulations with more agents, such as an auction or a bail hearing.

Now, the system runs the experiment. The conditions are simulated in parallel (step 5 in Figure 1), each with a different value for the exogenous dimensions of the SCM—the possible budgets for the buyer.

The system must also determine when to stop the simulations. There is no obvious rule for when a conversation should end. Like the halting problem in computer science—it is impossible to write a universal algorithm that can determine whether a given program will complete (Turing, 1937)—such a rule for conversations does not exist. We set two stopping conditions for the simulations. After each agent speaks in a simulation, an external LLM is prompted with the transcript of the conversation and asked if the conversation should continue. If yes, the next agent speaks; otherwise, the simulation ends. Additionally, we limit the total number of agent statements to twenty. One could imagine doing something more sophisticated both with the social interactions and the stopping conditions in the future.

Finally, the system gathers the data for analysis. Outcomes are measured by asking the agents the survey questions (Figure 1, step 6) as determined before the experiment. The data is then used to estimate the linear SCM. For our negotiation, that would be a simple linear model with a single path estimate (i.e., linear coefficient) for the effect of the buyer’s budget on the probability of a deal—the final step in Figure 1. Note that an SCM specifies, ex-ante, the exact statistical analyses

to be conducted after the experiment—akin to a pre-analysis plan. This step of the system’s process is, therefore, mechanical.

The system, as outlined, is automated from start to finish—the SCM and its accompanying metadata serve as a blueprint for the rest of the process. Once there is a fitted SCM, this process can be repeated. Although we have not automated the transition from one experiment to the next, the system can generate new causal variables, induce variations, and run another experiment based on the results of the first.

3 Results of experiments

We present results for four social scenarios explored by the system. In the first two scenarios, our involvement was restricted to entering the scenario description. In the third and fourth scenarios, we selected the hypotheses and edited some of the agents, but the system designed and executed the experiments. We intervened in the latter scenarios not because the system is incapable of simulating these scenarios autonomously but to demonstrate the system’s capacity to accommodate human input at any point while still generating exciting results.

3.1 Bargaining over a mug

We first use the system to simulate “two people bargaining over a mug”—this phrase being in quotes because it was the only input needed for the system to simulate the following process. The system identified the buyer and seller as the relevant agents and selected the buyer’s budget, the seller’s minimum acceptable price, and the seller’s emotional attachment to the mug as potential causes affecting the outcome of whether a deal occurs.

Table 2a provides the information generated by the system about the SCM and the experimental design. The topmost row, the simulation details, provides high-level information about the structure of the simulation. The remaining rows provide information about the variables in the SCM and how they were operationalized. The

system automatically generated all this information by iteratively querying an LLM.

The three exogenous variables were operationalized as the buyer’s budget in dollars, the seller’s minimum acceptable price in dollars, and the seller’s emotional attachment as an ordinal scale from “no emotional attachment” to “extreme emotional attachment.” The system chose nine values (the “Attribute Treatments” in Table 2a) to vary for each of the first two causes and five for the seller’s feelings of love towards the mug (one for each level of the scale). This led to $9 \times 9 \times 5 = 405$ experimental runs of the simulated conversation between the buyer and seller.

Figure 2b provides the fitted SCM. The outcome variable is given with its mean and variance. The raw path estimates and their standard errors are shown on the arrows. For ordinal variables (e.g., the seller’s feelings of love), we treat the levels as numerical values. The buyer and seller reached a deal for the mug in half of the simulations, and all three causes had a significant effect on the probability of a deal.

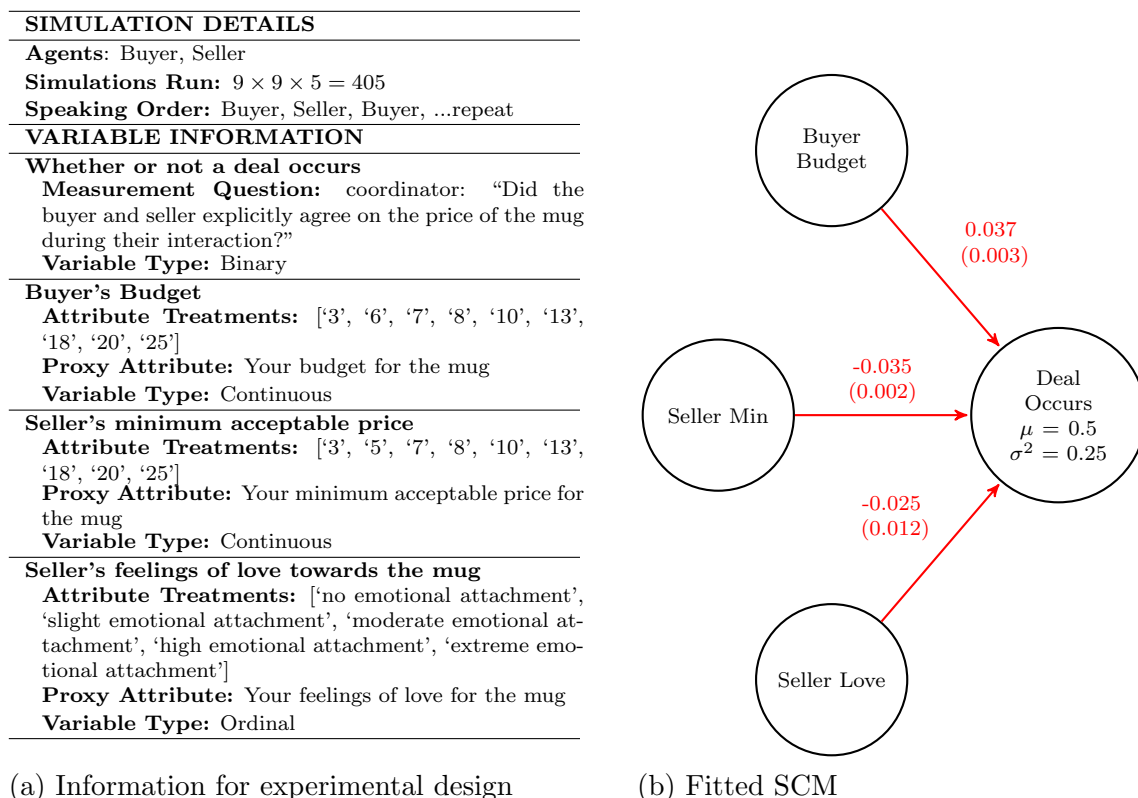
A one-dollar increase in the buyer’s budget caused an average increase of 3.7 percentage points in the probability of a deal ($\hat{\beta}^* = 0.51$, $p < 0.001$).⁸ A one-dollar increase in the seller’s minimum acceptable price caused an average decrease of 3.5 percentage points in the probability of a deal occurring ($\hat{\beta}^* = -0.49$, $p < 0.001$). Finally, a one-unit increase in the ordinal scale of the seller’s love for the mug, such as going from moderate emotional attachment to high emotional attachment, caused an average decrease of 2.5 percentage points in the probability of a deal ($\hat{\beta}^* = -0.07$, $p = 0.044$).

3.2 A bail hearing

Next, we explore “a judge is setting bail for a criminal defendant who committed 50,000 dollars in tax fraud.” Table 3a shows that the system selected a judge, defendant, defense attorney, and prosecutor as the relevant agents. In this scenario, the system selected a more flexible interaction protocol than the one used in the previous experiment. The judge was chosen as a center agent and, in order, the

⁸We report standardized effect size estimates with $\hat{\beta}^*$. Standardized effect sizes being “a one standard deviation increase in X causes a $\hat{\beta}^*$ standard deviation increase in Y.”

Figure 2: Experimental design and fitted SCM for “two people bargaining over a mug.”



(a) Information for experimental design

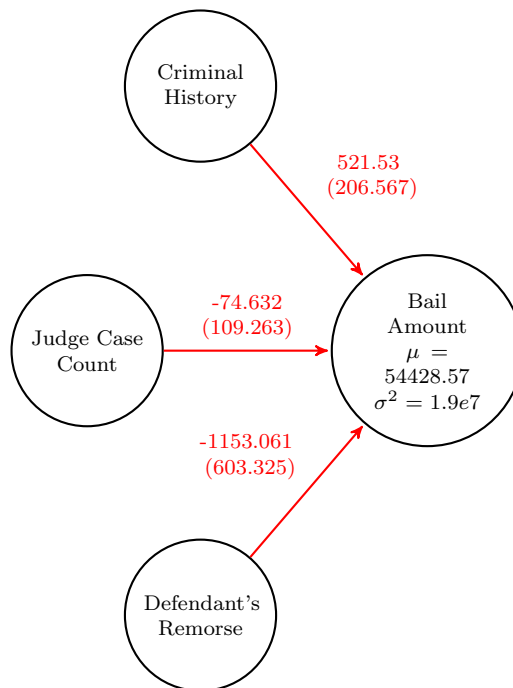
(b) Fitted SCM

Notes: Figure 2a provides the information automatically generated by the system to execute the experiment for its proposed hypothesis. This includes the high-level structure of the simulations, how the outcome is measured, and the treatment variations for each cause. The fitted SCM in Figure 2b shows the experiment results. The outcome is given with its mean and variance. The edges are labeled with their unstandardized path estimate and standard error. We assume a simple linear model for the SCM, such that the above graph can also be written as $DealOccurs = 0.037BuyerBudget - 0.035MinPrice - 0.025SellerLove$.

prosecutor, defense attorney, and defendant as the non-center agents. This means the judge spoke first in every simulation, alternating with the other agents: judge, prosecutor, judge, defense attorney, judge, defendant, and so on. As described in Section A.3, we call this the “center-ordered” interaction protocol.

Figure 3: Experimental design and fitted SCM for “a judge is setting bail for a criminal defendant who committed 50,000 dollars in tax fraud.”

SIMULATION DETAILS
Agents: Judge, Defendant, Defense attorney, Prosecutor
Simulations Run: $7 \times 7 \times 5 = 243$
Speaking Order: Judge, Prosecutor, Judge, Defense Attorney, Judge, Defendant, ... repeat
VARIABLE INFORMATION
Bail amount set by the judge
Measurement Question: Judge: “What was the bail amount you set for the defendant?”
Variable Type: Continuous
Defendant’s criminal history
Attribute Treatments: [‘0’, ‘1’, ‘2’, ‘3’, ‘6’, ‘9’, ‘12’]
Proxy Attribute: Number of your prior convictions
Variable Type: Count
Prior case count for judge that day
Attribute Treatments: [‘0’, ‘2’, ‘5’, ‘9’, ‘12’, ‘18’, ‘23’]
Proxy Attribute: Number of cases you have already heard today
Variable Type: Count
Defendant’s level of remorse
Attribute Treatments: [‘no expressed remorse’, ‘low expressed remorse’, ‘moderate expressed remorse’, ‘high expressed remorse’, ‘extreme expressed remorse’]
Proxy Attribute: Your level of expressed remorse
Variable Type: Ordinal



(a) Information for experimental design

(b) Fitted SCM

Notes: Figure 3a provides the information automatically generated by the system to execute the experiment for its proposed hypothesis. Figure 3b shows the fitted SCM from the experiment.

The system chose the outcome to be the final bail amount, and the three proposed causes are the defendant’s criminal history, the number of cases the judge has already heard that day, and the defendant’s level of remorse. The number of cases the judge already heard that day and the defendant’s level of remorse are operationalized literally, as the count of cases the judge has heard and five ordinal levels of possible outward expressions of remorsefulness. The defendant’s criminal history

is operationalized as the number of previous convictions.

The fitted SCM in Figure 3b shows that only the defendant’s criminal history had a significant effect on the final bail amount with each additional conviction causing an average increase of \$521.53 in bail ($\hat{\beta}^* = 0.16$, $p = 0.012$). It is unclear whether the defendant’s remorse affected the final bail amount. The effect size was small but non-trivial ($\hat{\beta}^* = -0.12$, and $p = 0.056$).

When we estimated the SCM with interactions, the interaction between the judge’s case count and the defendant’s remorse was nontrivial ($\hat{\beta}^* = -0.32$, $p = 0.047$). In this specification (Figure A.5), none of the other interactions or the stand-alone causes have a significant effect, including the defendant’s criminal history.

3.3 Interviewing for a job as a lawyer

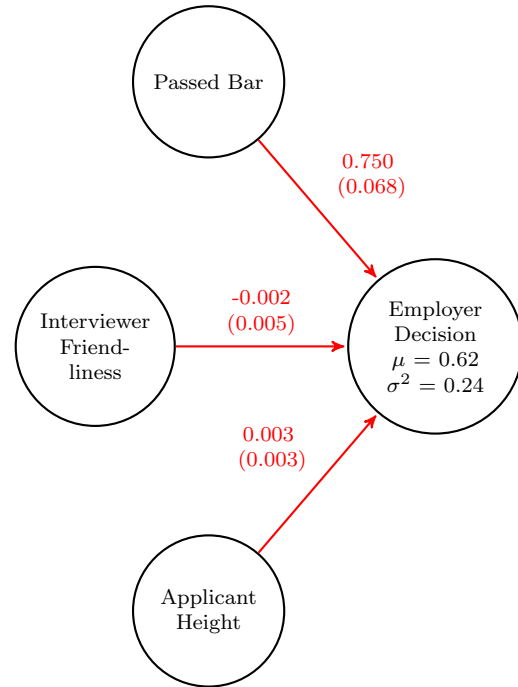
In our third simulated experiment, we chose the scenario “a person interviewing for a job as a lawyer.” The system determined that a job applicant and an employer were the agents. Unlike the previous simulations, we manually selected the variables in the SCM. Table 4a shows that these were the employer’s hiring decision as the outcome and whether the applicant passed the bar, the interviewer’s friendliness, and the job applicant’s height as the potential causes.

The system operationalized the causes as a binary variable for passing the bar, the job applicant’s height in centimeters, and the interviewer’s friendliness as the proposed number of friendly phrases to use during the simulation. Since one of the causes is a binary variable, the only potential cause in all our scenarios of this type, the sample size for the experimental simulations of this scenario is smaller ($n = 80$). By default, the system runs a factorial experimental design for all proposed values of each cause. With only two possible values for the job applicant passing the bar (as opposed to 5 varied treatment values for the interviewer’s friendliness and 8 for the applicant’s height), this limits the possible combinations of the causal variables to $2 \times 5 \times 8 = 80$. A researcher could run more simulations to increase the sample size.

We can see in Figure 4b that only the applicant passing the bar has a clear causal

Figure 4: Experimental design and fitted SCM for “a person is interviewing for a job as a lawyer.”

SIMULATION DETAILS
Agents: Interviewer, Job Applicant
Simulations Run: $2 \times 5 \times 8 = 405$
Speaking Order: Interviewer, Job Applicant, Interviewer, ...repeat
VARIABLE INFORMATION
Employer’s Decision
Measurement Question: Employer: “Have you decided to hire the job applicant?”
Variable Type: Binary
Whether Applicant Passed Exam
Attribute Treatments: [‘Passed’, ‘Not’]
Proxy Attribute: Your bar exam status
Variable Type: Binary
Interviewer’s level of friendliness
Attribute Treatments: [‘2’, ‘7’, ‘12’, ‘17’, ‘22’]
Proxy Attribute: Number of positive phrases to use during interview
Variable Type: Count
Job applicant’s height
Attribute Treatments: [‘160’, ‘165’, ‘170’, ‘175’, ‘180’, ‘185’, ‘190’, ‘195’]
Proxy Attribute: Your height in centimeters
Variable Type: Continuous



(a) Information for experimental design

(b) Fitted SCM

Notes: Figure 4a provides the information automatically generated by the system to execute the experiment for the proposed hypothesis. Figure 4b shows the fitted SCM from the experiment.

effect on whether the applicant gets the job. This is the largest standardized effect we see across the simulations in the four scenarios ($\hat{\beta}^* = 0.78$, $p < 0.001$). On average, whether or not the applicant passes the bar increases the probability she gets the job by 75 percentage points. None are significant when we test for interactions (Figure A.6).

3.4 An auction for a piece of art

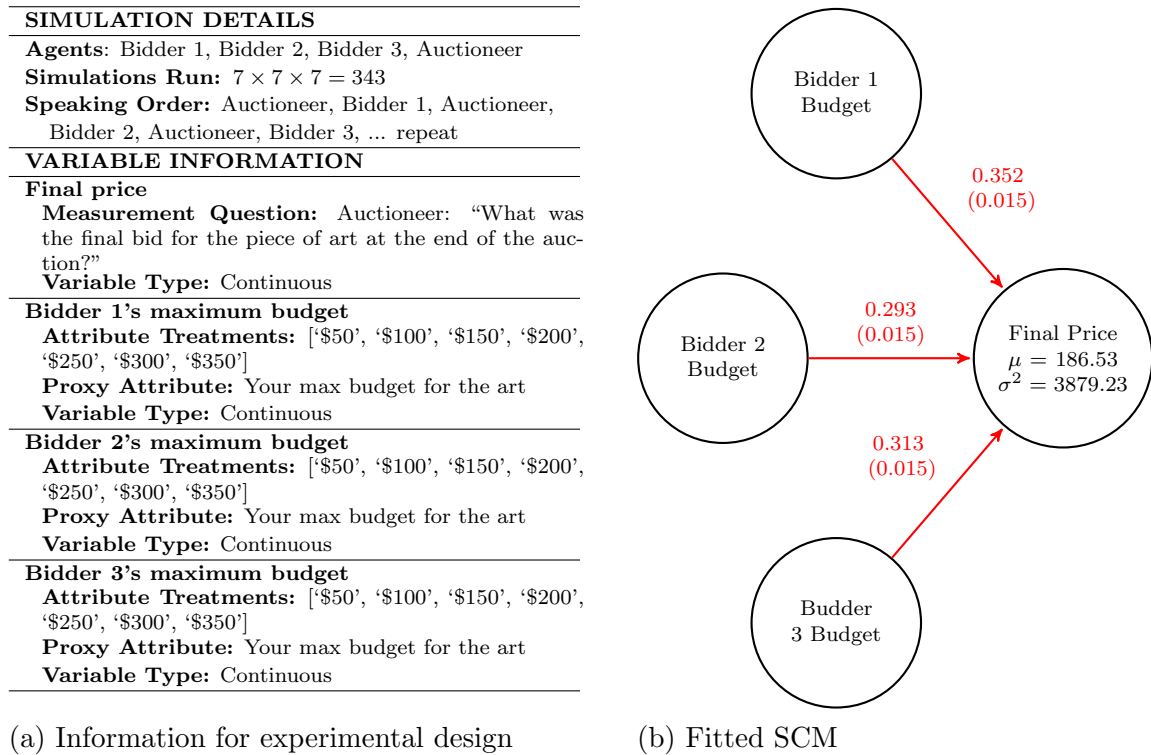
Finally, we explored the scenario of “3 bidders participating in an auction for a piece of art starting at fifty dollars.” Table 5a shows that the causes are each bidder’s maximum budget for the piece of art, and the outcome is the final price of the piece of art—all of which we selected.

All four variables are operationalized in dollars. To maintain symmetry in the simulations, we manually selected the same proxy attribute for the three bidders: “your maximum budget for the piece of art.” Each bidder had the same seven possible values for their attribute, leading to $7^3 = 343$ simulations of the auction. These budgets are private values, meaning that unless a bidder publicly reveals their budget, the other bidders are unaware of its value.

Like the tax fraud scenario, the system chose the center-ordered interaction protocol for these simulations. The auctioneer was selected as the central agent, and the other agents were bidder 1, bidder 2, and bidder 3, who alternated with the auctioneer in that order.

Figure 5b provides the results. All three causal variables had a positive and statistically significant effect on the final price. A one-dollar increase in any of the bidder’s budgets caused a \$0.35, \$0.29, and \$0.31 increase in the final price for the piece of art for each respective bidder ($\hat{\beta}^* = 0.57$, $p < 0.001$; $\hat{\beta}^* = 0.47$, $p < 0.001$; $\hat{\beta}^* = 0.5$, $p < 0.001$). These quantities make sense as each bidder has a $\frac{1}{3}$ chance of being marginal.

Figure 5: Experimental design and fitted SCM for “3 bidders participating in an auction for a piece of art starting at fifty dollars.”



Notes: Figure 5a provides the information automatically generated by the system to execute the experiment for the proposed hypothesis. Figure 5b shows the fitted SCM from the experiment.

4 LLM predictions for paths and points

It is worth reiterating that the results in the previous section were not generated by directly prompting an LLM but rather through experimentation. Although the experiments were fast and inexpensive, they were not free—in total, they took about 5 hours to run and cost over \$1,000. This raises the question of whether the simulations were even necessary. Could an LLM do a “thought experiment” (i.e., make a prediction based on a prompt) about a proposed *in silico* experiment and achieve the same insight? If so, we should just prompt the LLM to generate an SCM and elicit its predictions about the relationships between the variables.

To test this idea, we describe some of the simulations to the LLM and ask it to predict the results—path estimates and point predictions.⁹ Specifically, we model each scenario as $y = X\beta$, where y is an $n \times 1$ vector and X is a $n \times k$ matrix. Here, n is the number of simulations, and k is the number of proposed causes. The experiments from Section 3 provided us with estimates for $\hat{\beta}$ (a $k \times 1$ vector). We describe the scenario and the experiment to the LLM and ask it to independently predict y_i given each X_i (a predict- y_i task) as well as to predict $\hat{\beta}$ (a predict- $\hat{\beta}$ task).

The LLM’s y_i predictions are highly inaccurate compared to those from auction theory, which predicts that the clearing price will be the second highest valuation in an open-ascending price auction with private values (Maskin and Riley, 1985). The LLM is also unable to accurately predict the path estimates ($\hat{\beta}$) of the fitted SCM. Finally, we examine how the LLM does on the predict- y_i task when provided with an SCM fit on all of the data except for the corresponding X_i (the predict- $y_i|\hat{\beta}_{-i}$ task). While the additional information dramatically improves the LLM’s predictions, they are still less accurate than those made by auction theory.

⁹All predictions are made by the LLM once at temperature 0. When we elicit these predictions many times at higher temperatures, the results are similar.

4.1 Predicting y_i

We supply the LLM with a prompt detailing the simulation and experimental design for various bidder reservation price combinations in the auction experiment.¹⁰ We then ask the LLM to predict the clearing price for the auction. This gives us a point prediction for each simulated auction (i.e., each unique row X_i in X) used to generate the fitted SCM in Figure 5b.

Figure 6 presents a comparison of the LLMs predictions, the simulated experiments, and the predictions made by auction theory.¹¹ The columns correspond to the different reservation values for bidder 3 in a given simulation, and the rows correspond to the different reservation values for bidder 2. The y-axis is the final bid price, and the x-axis lists bidder 1’s reservation price. The black triangles track the observed clearing price in each simulated experiment, the black line shows the predictions made by the theory, and the blue line indicates the LLM’s predictions without the fitted SCM—the predict- y_i task.

The LLM performs poorly at the predict- y_i task. The blue line is often far from the black triangles and sometimes decreases as the second-highest reservation price across the agents increases. In contrast, auction theory is accurate in its predictions of the final bid price in the experiment—the black line often perfectly tracks the black triangles.¹² The mean squared error (MSE) of the LLM’s predictions in the predict- y_i task ($MSE_{y_i} = 8628$) is an order of magnitude higher than that of the theoretical predictions ($MSE_{Theory} = 128$), and the predictions are even further from the theory than they are from the empirical results ($MSE_{y_i-Theory} = 8915$).¹³

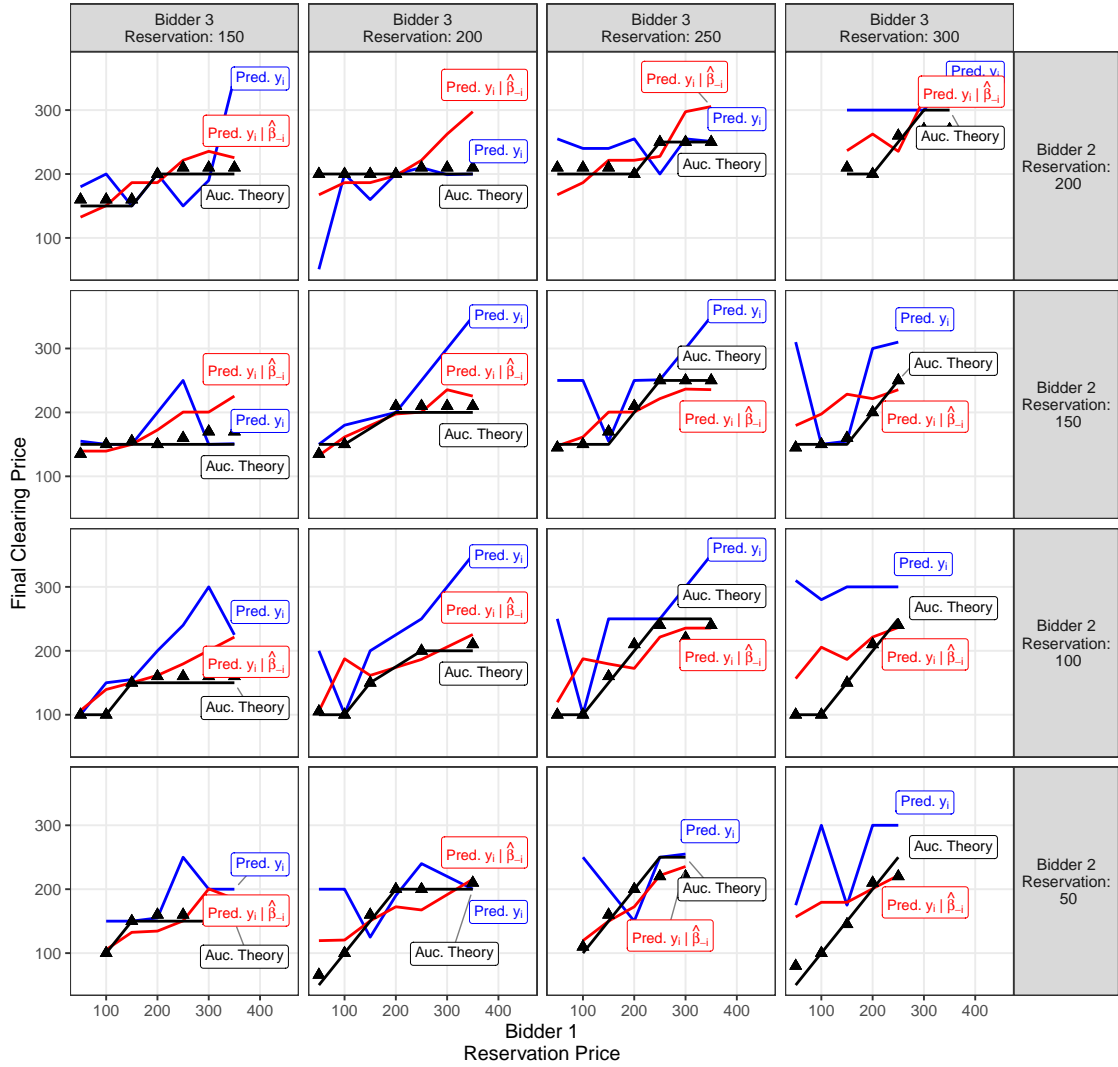
¹⁰In 80/343 simulations, the agents made the maximum number of statements (20) allowed by the system before the auction ended. We remove these observations because, without additional information, auction theory does not make predictions about partially completed auctions.

¹¹We provide only a subset of the results in the main text as it is difficult to visualize all of them in a single figure. Figure A.10 shows the full set of predictions. The results are similar.

¹²There are a few observations where the empirical clearing price is slightly above or below the theory prediction. In most cases, it was off because the auctioneer incremented the bid price above the second-highest reservation price in the last round.

¹³MSE is reported for all predictions, not just the subset shown in Figure 6.

Figure 6: Comparison of the LLM’s predictions to the theoretical predictions and a subset of experimental results for the auction scenario.



▲ Experiment

Notes: The columns correspond to the different reservation values for bidder 3 in a given simulation, and the rows correspond to the different reservation values for bidder 2. The y-axis is the clearing price, and the x-axis lists bidder 1’s reservation price. The black triangles track the observed clearing price in each simulated experiment; the black line shows the predictions made by auction theory ($MSE_{Theory} = 128$), the blue line indicates the LLM’s predictions without the fitted SCM—the $predict-y_i$ task ($MSE_{y_i} = 8628$), and the red line is the LLM’s predictions with the fitted SCM—the $predict-y_i|\hat{\beta}_{-i}$ task ($MSE_{y_i|\hat{\beta}_{-i}} = 1505$).

4.2 Predicting $\hat{\beta}$

We prompted the LLM to predict the path estimates and whether they would be statistically significant for the simulated experiments in Section 3. This is the predict- $\hat{\beta}$ task. We then compare the LLM’s predictions to the fitted SCMs. With four experiments and three causes in each, we generate 12 predictions.

We provide the LLM with extensive information to make its predictions for each experiment. This information includes the proposed SCM, the operationalizations of the variables, the number of simulations, and the possible treatment values. Each prediction is elicited once at temperature 0.

The predictions are shown in Table A.1. They were, on average, 13.2 times larger than the actual estimates, and 10/12 of the predictions were overestimates. Even when we remove the largest overestimate, the average magnitude of the ratio between the predicted and actual estimates is still 5.3. The sign of the estimate was correct in 10/12 predictions, and 10/12 correctly guessed whether or not the estimate would be statistically significant. The results are similar when we repeat the predictions at a higher temperature and take their average (see Table A.2).

4.3 Predicting $y_i|\hat{\beta}_{-i}$

The LLM was, on average, off by an order of magnitude for both the predict- y_i task and the predict- $\hat{\beta}$ task, but maybe it can do better with more information. For each X_i in the auction simulations, we use the data from the experiment to estimate $\hat{\beta}_{-i}$, the path estimates from the SCM excluding the i th observation. We then prompt the LLM to predict the outcome for each X_i given $\hat{\beta}_{-i}$.

The red line in Figure 6 provides these new predictions. The LLM’s predictions are much closer to the actual outcomes when it has access to a fitted SCM ($MSE_{y_i|\hat{\beta}_{-i}} = 1505$) as opposed to when it does not ($MSE_{y_i} = 8628$), even though all the predictions are out of sample and every X_i is unique. However, the LLM’s predictions on the predict- $y_i|\hat{\beta}_{-i}$ task are still not as accurate as the predictions made by the theory ($MSE_{Theory} = 128$) and are still further from the theory than they are

from the empirical results ($MSE_{y_i|\hat{\beta}_{-i}-Theory} = 1761$).¹⁴ There is clearly room for improvement. That improvement is feasible with the system: there exists an SCM perfectly consistent with auction theory. Only one exogenous variable was missing: the bidders’ second-highest reservation price. If allowed to generate and test enough potential causes, our system could have selected this variable as a possible cause by itself. In this case, the fitted SCM would have matched the theoretical predictions.¹⁵

5 Identifying causal structure ex-ante

The SCM-based approach offers a promising new method for studying simulated behavior at scale. However, there are other options for such rapid exploration. Others have designed large, quasi-unstructured simulations that generate exciting results. For example, Park et al. (2023) simulates a community of LLM agents with personas and memory systems, allowing them to interact freely for an extended period. Without explicit instructions, the agents exhibit human-like behaviors such as throwing parties, dating, and forming friendships.

While impressive and informative, a problem with such open-ended social simulations is that selecting and analyzing outcomes can take time and effort. To unveil insights, researchers may need to comb through thousands of lines of unstructured text. If they are interested in casual relationships, they may need to infer the causal structure ex-post, which can be problematic. In contrast, the SCM framework describes exactly what needs to be measured as a downstream outcome subject to the exogenous manipulations of the cause. Identification is guaranteed. In this section, we discuss how assuming or searching for causal structure in observational data—the type generated from massive open-ended simulations—can lead to misidentification and how using SCMs avoids this problem.

¹⁴It is also less accurate than the mechanical predictions made by the fitted SCM using the same procedure $MSE_{Mechanistic:y_i|\hat{\beta}_{-i}} = 725$. Maybe the LLM cannot do the math, is still conditioning on other information beyond the path estimates when making its predictions, or, like humans, is ignoring relevant information when making choices (Handel and Schwartzstein, 2018).

¹⁵When we do fit this SCM manually in Figure A.9, the coefficient is close to one ($\beta = 0.912$), and almost all the variance in the outcome is explained ($R^2 = 0.977$).

5.1 Assuming causal structure from data

All estimates in the fitted SCMs in Section 3 are unbiased. We know this because the data comes from an experiment, and we randomized on the causal variables. A nice feature of a perfectly randomized experiment is that we can get unbiased measurements of any downstream endogenous outcome relative to the exogenous manipulations.¹⁶ I.e., the coefficients on the fitted SCM are identified. For example, in the bargaining experiment, we may be interested in the length of the conversation as an outcome, even though it was not a part of the original SCM. The conversation length can be operationalized as the sum of the number of statements made by all agents, and we can measure it using the transcript from the finished experiment. We can then fit an SCM with the data and get unbiased estimates of the effect of the exogenous variables on the conversation’s length.

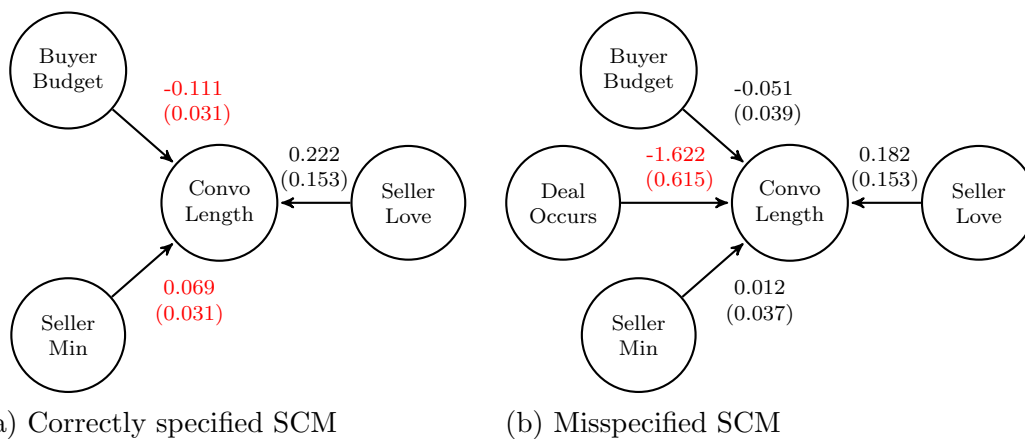
Figure 7a shows this fitted SCM using the data from the experiment in Section 3. Both the buyer’s budget and the seller’s minimum price affect the length of the conversation ($p < 0.001$; $p = 0.026$), but the seller’s emotional attachment does not ($p = 0.147$).

Suppose we did not know the actual causal structure of these scenarios or that the data came from an experiment. All we have are the data for the original three causes, the conversation length, and whether a deal was made (the original outcome). We must make untestable assumptions if we want to estimate the causal relationships between these variables. For example, one could reasonably presume that the buyer’s budget, the seller’s minimum price, the seller’s emotional attachment, and whether a deal was made causally affect the conversation’s length.

Figure 7b provides the fitted SCM for this proposed causal structure. Only the estimate for whether a deal was made is significant ($p = 0.008$). But we know this is wrong. We have the true causal structure in Figure 7a from a perfectly randomized experiment, and both the buyer’s and the seller’s reservation prices had a significant effect on the length of the conversation. In this misspecified SCM, both paths are insignificant and far closer to zero ($p = 0.189$; $p = 0.755$). The presence of a deal is

¹⁶When we say “downstream,” we mean any variable whose value is realized after the agents begin interacting in the simulated conversations.

Figure 7: Comparison of the true and misspecified SCMs.



Notes: Statistically significant paths are marked in red ($\alpha = 0.05$). Each path is given with its estimated coefficient and standard error in parentheses. Both SCMs are estimated using the data from the bargaining scenario in Section 3. Subfigure (a) provides a correctly specified SCM from a randomized experiment. Subfigure (b) shows a misspecified SCM based on an assumed structure. The path estimates of the buyer’s budget and the seller’s minimum price go from significant in the correctly specified SCM to insignificant and far closer to zero in the misspecified SCM.

a bad control—it is likely codetermined with the length of the conversation—biasing the other path estimates.

The informed econometrician may presume that she would never make such a mistake, but many researchers are not so savvy.¹⁷ Indeed, we were unsure until we compared the fitted models. Many bad controls are less obvious than those in this example (Cinelli et al., 2022). It is easy to misspecify a model when the data is observational and multidimensional.

The SCM-based approach avoids bad controls. The generation of the data is based on the causal structure. There is no need to instrument endogenous variables and presume their causal relationships. Exogenous variation is explicitly induced in the SCM to identify the causal relationships ex-ante. Even if we do not know how a new outcome is incorporated into the causal structure, we can always reference how it is affected by the exogenous variables by fitting a simple linear SCM.

¹⁷LLMs are definitely not yet savvy enough to avoid this mistake.

5.2 Searching for causal structure in data

Another strategy for identifying causal relationships when the underlying structure is unknown is to let the data speak for itself. For example, we could use an algorithm to find the model that makes the data most likely. There are many ways to do this, none of which can always, or even consistently, identify the correct causal relationships from observational data (Pearl, 2009a). These algorithms take as input potential variables of interest (a graph with no edges, only nodes) and data for these variables. They output a proposed DAG that best fits the data.¹⁸

The simplest algorithm is to generate all possible DAGs for existing variables and then evaluate each model based on some criteria (e.g., maximum likelihood, Bayesian information criterion, etc.). Another method is to add edges that maximize the criteria greedily. This approach can be further improved by penalizing the model for complexity (based on additional criteria) and removing edges until the model is greedily optimized. The second approach is the Greedy Equivalence Search (GES) algorithm (Chickering, 2002), which we used on the data and from all the experiments in Section 3.¹⁹

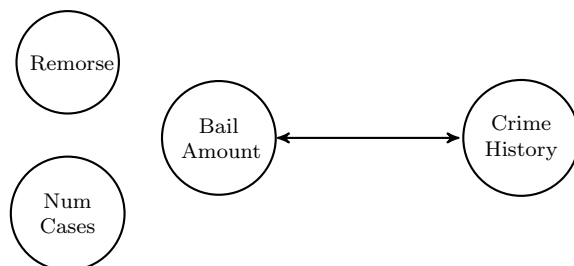
In some experiments, the algorithm incorrectly identified the causal structure. Figure 8 provides the DAG identified by the GES algorithm for the tax fraud scenario. As a reminder, the original causal variables are the defendant’s previous convictions, the judge’s number of cases heard that day, and the defendant’s level of remorse, and the outcome is the bail amount. The algorithm has no information about which variables are exogenously varied, just the raw data.

The GES algorithm identified the defendant’s criminal history and the bail amount as the only variables in the scenario with any causal relationship. This is partially correct—we know from the experiment that an increase in the defendant’s previous convictions caused an increase in the average bail amount. However, the algorithm identified the causal relationship as equally likely in either direction. There was

¹⁸These algorithms often do not presume a functional form, so we refer to hypotheses as DAGs, not SCMs, in this section.

¹⁹The GES algorithm is not perfectly stable; different runs on the same data can produce different results, which is its own problem.

Figure 8: Incorrect causal structure identified by the GES algorithm for the tax fraud experiment.



Notes: The Greedy Equivalence Search (GES) algorithm can incorrectly identify the causal structure of observational data. In the tax fraud scenario, we know from Figure 3b and the accompanying experiment that an increase in the defendant’s previous convictions caused an increase in the average bail amount. However, the algorithm identified the causal relationship as equally likely in either direction. Without the correctly specified DAG, a researcher would have to assume the causal structure of the data, which can be problematic.

no more evidence in the data that the defendant’s criminal history caused the bail amount than the bail amount caused the defendant’s criminal history. And while we know that the former is correct from our experiment, a researcher using the algorithm without the correctly specified DAG would not. They would have to make an assumption, which, as we have shown, can be problematic.

The SCM-based approach avoids search problems, as we never need to search for the causal structure given the data. Instead, we generate the data based on a proposed causal structure. Even if we want to measure a new outcome on the existing experimental data, we have already identified the sources of exogenous variation.

We should note that problems with searching for or assuming causal structures from data are not new. Pearl (2009a) makes a similar point many times. However, social scientists have never had the tools to induce exogenous variation and explore causal relationships at scale in many different scenarios.

6 Conclusion

This paper demonstrates an approach to automated *in silico* hypothesis generation and testing made possible through the use of SCMs. We implemented the approach by building a computational system with LLMs and provided evidence that simulations can elicit information from an LLM that was not ex-ante available to the model. We also showed that such simulations produce results that are highly consistent with theoretical predictions made by the relevant economic theory. In this final section, we will discuss implications and identify areas for future research.

6.1 Controlled experimentation at scale

How might automated social scientific systems be useful for social science? One view is that these simulations are simple dress rehearsals for “real” social science. They may be helpful for tasks such as experimental design, power calculations, and improved pre-analysis plans. A more expansive and exciting view is that these simulations would yield insights that sometimes generalize to the real world. This is a view that sees these agents as a step forward in representing humans far beyond classical methods in agent-based modeling, like those used to explore how individual preferences can lead to surprising social patterns (Schelling, 1969, 1971). This view would mirror recent advances in machine learning for protein folding (Jumper et al., 2021) and material discovery (Merchant et al., 2023). For now, these simulations probably lie somewhere in between these views, but there is value in the middle ground.

In contrast to most academic research as currently practiced (Almaatouq et al., 2022; Camerer, 2022), the automated exploration of LLMs could generate controlled and replicable experimental simulations en masse. This contrast is important. In the social sciences, context can heavily influence results. Outcomes that hold true for one population may not for another. Even within the same population, a change in environment can nullify or flip results (Lerner et al., 2004). Studying humans is also expensive and time-consuming, which makes rapid, inexpensive, and replicable exploration valuable. There is still, of course, the fundamental jump from simulations

to human subjects.

6.2 Replicability

Replicating social science experiments with human subjects can be difficult (Camerer et al., 2018). The exact procedures used in experiments are often unclear (Engzell, 2023). In contrast, the system allows for nearly frictionless communication and replication of its experimental designs.

The system’s entire procedure is exportable as a JSON file with the fitted SCM.²⁰ This JSON includes every decision the system makes, including natural language explanations for the choices and the transcripts from each simulation. These JSONs can be saved or uploaded at any point in the system’s process. A researcher could run experiments and post the JSON and results online. Other scientists could inspect, replicate the experiment, or extend the work.

6.3 Future research

While designing our system, we encountered several areas for new research. Progress in these areas could greatly improve the fidelity of the simulations and the system’s utility. First is the problem of which attributes to endow an LLM-powered agent beyond those immediately relevant to the proposed exogenous variables. For example, demographic information, personalities, and other traits are not included in the agent’s attributes unless they are a part of the SCM. Adding some or all of these attributes to the agents might improve the fidelity of the simulations. However, it is unclear how to optimize this process.

Second, we encountered the problem of engineering social interactions between LLM agents. LLMs are designed to exchange text in sequence, necessitating a protocol for turn-taking that reflects the natural ebb and flow of human conversation.

²⁰A JSON (JavaScript Object Notation) is a data format that is easy for humans to read and write and for machines to parse and generate. It is commonly used for transmitting data in web applications, as a configuration and data storage format, and for serializing and transmitting structured data over a network.

In an initial attempt to address this problem, we created a menu of flexible agent-ordering mechanisms. We also introduced an additional LLM-powered agent into our version of the system as a quasi-omniscient assistant who can read through transcripts and make choices about the speaking order of other agents in the simulations (see Section A.3). There are probably better ways to determine the speaking order of agents.

A related problem is the question of when to stop the simulations. Like Turing’s halting problem, there is likely no universal rule for when conversations should end, but there are probably better rules than those we have implemented. A Markov model approximating the distribution of agents speaking, estimated from real conversation data, might provide more naturalistic results for simulating and ending interactions, but that is an idea for future work.

Finally, if we can build a system that can automate one iteration of the scientific process and determine a follow-on experiment, a clear next step is to set up an intelligently automated research program. This would involve using outcomes from the simulations to inform continuous cycles of experimentation. Then, a researcher could intelligently explore a given scenario’s parameter space. How to optimize this exploration amongst so many possible variables will be an important problem to solve.

As presented in this paper, the system provides only one possible implementation of the SCM-based approach. We made many subjective decisions. Other researchers might implement the approach with different design choices. There is room for improvement and exploration.

References

Aher, Gati V, Rosa I Arriaga, and Adam Tauman Kalai, “Using large language models to simulate multiple humans and replicate human subject studies,” in “International Conference on Machine Learning” PMLR 2023, pp. 337–371.

Almaatouq, Abdullah, Thomas L. Griffiths, Jordan W. Suchow, Mark E.

- Whiting, James Evans, and Duncan J. Watts**, “Beyond Playing 20 Questions with Nature: Integrative Experiment Design in the Social and Behavioral Sciences,” *Behavioral and Brain Sciences*, 2022, p. 1–55.
- Argyle, Lisa P, Ethan C Busby, Nancy Fulda, Joshua R Gubler, Christopher Rytting, and David Wingate**, “Out of one, many: Using language models to simulate human samples,” *Political Analysis*, 2023, *31* (3), 337–351.
- Atari, M., M. J. Xue, P. S. Park, D. E. Blasi, and J. Henrich**, “Which Humans?,” Technical Report 09 2023. <https://doi.org/10.31234/osf.io/5b26t>.
- Athey, Susan, Jonathan Levin, and Enrique Seira**, “Comparing open and Sealed Bid Auctions: Evidence from Timber Auctions*,” *The Quarterly Journal of Economics*, 02 2011, *126* (1), 207–257.
- Bakker, Michiel, Martin Chadwick, Hannah Sheahan, Michael Tessler, Lucy Campbell-Gillingham, Jan Balaguer, Nat McAleese, Amelia Glaese, John Aslanides, Matt Botvinick, and Christopher Summerfield**, “Fine-tuning language models to find agreement among humans with diverse preferences,” in S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, eds., *Advances in Neural Information Processing Systems*, Vol. 35 Curran Associates, Inc. 2022, pp. 38176–38189.
- Binz, Marcel and Eric Schulz**, “Turning large language models into cognitive models,” 2023.
- and –, “Using cognitive psychology to understand GPT-3,” *Proceedings of the National Academy of Sciences*, 2023, *120* (6), e2218523120.
- Brand, James, Ayelet Israeli, and Donald Ngwe**, “Using GPT for Market Research,” *Working paper*, 2023.
- Bubeck, Sébastien, Varun Chandrasekaran, Ronen Eldan, Johannes Gehrke, Eric Horvitz, Ece Kamar, Peter Lee, Yin Tat Lee, Yuanzhi Li, Scott Lundberg, Harsha Nori, Hamid Palangi, Marco Tulio Ribeiro,**

- and Yi Zhang**, “Sparks of Artificial General Intelligence: Early experiments with GPT-4,” 2023.
- Burns, C, H Ye, D Klein, and J Steinhardt**, “Discovering latent knowledge in language models without supervision,” in “International Conference on Learning Representations (ICLR)” 2023.
- Buyalskaya, Anastasia, Hung Ho, Katherine L. Milkman, Xiaomin Li, Angela L. Duckworth, and Colin Camerer**, “What can machine learning teach us about habit formation? Evidence from exercise and hygiene,” *Proceedings of the National Academy of Sciences*, 2023, *120* (17), e2216115120.
- Cai, Alice, Steven R Rick, Jennifer L Heyman, Yanxia Zhang, Alexandre Filipowicz, Matthew Hong, Matt Klenk, and Thomas Malone**, “DesignAID: Using Generative AI and Semantic Diversity for Design Inspiration,” in “Proceedings of The ACM Collective Intelligence Conference” CI ’23 Association for Computing Machinery New York, NY, USA 2023, p. 1–11.
- Camerer, Colin, Anna Dreber, Felix Holzmeister, Teck-Hua Ho, Jurgen Huber, Magnus Johannesson, Michael Kirchler, Gideon Nave, Brian A. Nosek, Thomas Pfeiffer, Adam Altmejd, Nick Buttrick, Taizan Chan, Yiling Chen, Eskil Forsell, Anup Gampa, Emma Heikensten, Lily Hummer, Taisuke Imai, Siri Isaksson, Dylan Manfredi, Julia Rose, Eric-Jan Wagenmakers, and Hang Wu**, “Evaluating the Replicability of Social Science Experiments in Nature and Science between 2010 and 2015,” *Nature Human Behaviour*, Aug 2018, *2* (9), 637–644.
- Camerer, Colin F.**, “The apparent prevalence of outcome variation from hidden “dark methods” is a challenge for social science,” *Proceedings of the National Academy of Sciences*, 2022, *119* (52), e2216020119.
- Cheng, Myra, Tiziano Piccardi, and Diyi Yang**, “CoMPosT: Characterizing and Evaluating Caricature in LLM Simulations,” *ArXiv*, 2023, *abs/2310.11501*.

- Chickering, David Maxwell**, “Optimal structure identification with greedy search,” *Journal of machine learning research*, 2002, 3 (Nov), 507–554.
- Cinelli, Carlos, Andrew Forney, and Judea Pearl**, “A crash course in good and bad controls,” *Sociological Methods & Research*, 2022, p. 00491241221099552.
- Engzell, Per**, “A universe of uncertainty hiding in plain sight,” *Proceedings of the National Academy of Sciences*, 2023, 120 (2), e2218530120.
- Enke, Benjamin and Cassidy Shubatt**, “Quantifying Lottery Choice Complexity,” Working Paper 31677, National Bureau of Economic Research September 2023.
- Fish, Sara, Paul Gözl, David C Parkes, Ariel D Procaccia, Gili Rusak, Itai Shapira, and Manuel Wüthrich**, “Generative Social Choice,” *arXiv preprint arXiv:2309.01291*, 2023.
- Girotra, Karan, Lennart Meincke, Christian Terwiesch, and Karl T Ulrich**, “Ideas are dimes a dozen: Large language models for idea generation in innovation,” *Available at SSRN 4526071*, 2023.
- Gurnee, Wes and Max Tegmark**, “Language Models Represent Space and Time,” 2023.
- Haavelmo, Trygve**, “The statistical implications of a system of simultaneous equations,” *Econometrica, Journal of the Econometric Society*, 1943, pp. 1–12.
- , “The probability approach in econometrics,” *Econometrica: Journal of the Econometric Society*, 1944, pp. iii–115.
- Handel, Benjamin and Joshua Schwartzstein**, “Frictions or Mental Gaps: What’s Behind the Information We (Don’t) Use and When Do We Care?,” *Journal of Economic Perspectives*, February 2018, 32 (1), 155–178.
- Hernán, Miguel A. and James M. Robins**, *Causal Inference: What If*, Boca Raton: Chapman & Hall/CRC, 2020.

- Horton, John J.**, “Large language models as simulated economic agents: What can we learn from homo silicus?,” Technical Report, National Bureau of Economic Research 2023.
- Imai, Kosuke, Dustin Tingley, and Teppei Yamamoto**, “Experimental Designs for Identifying Causal Mechanisms,” *Journal of the Royal Statistical Society Series A: Statistics in Society*, 11 2012, 176 (1), 5–51.
- Jumper, John, Richard Evans, Alexander Pritzel, Tim Green, Michael Figurnov, Olaf Ronneberger, Kathryn Tunyasuvunakool, Russ Bates, Augustin Žídek, Anna Potapenko et al.**, “Highly accurate protein structure prediction with AlphaFold,” *Nature*, 2021, 596 (7873), 583–589.
- Jöreskog, Karl G.**, “A GENERAL METHOD FOR ESTIMATING A LINEAR STRUCTURAL EQUATION SYSTEM*,” *ETS Research Bulletin Series*, 1970, 1970 (2), i–41.
- Lerner, Jennifer S., Deborah A. Small, and George Loewenstein**, “Heart Strings and Purse Strings: Carryover Effects of Emotions on Economic Decisions,” *Psychological Science*, 2004, 15 (5), 337–341. PMID: 15102144.
- Li, Peiyao, Noah Castelo, Zsolt Katona, and Miklos Sarvary**, “Frontiers: Determining the Validity of Large Language Models for Automated Perceptual Analysis,” *Marketing Science*, 2024, 0 (0), null.
- Ludwig, Jens and Sendhil Mullainathan**, “Machine Learning as a Tool for Hypothesis Generation,” Working Paper 31017, National Bureau of Economic Research March 2023.
- Maskin, Eric S. and John G. Riley**, “Auction Theory with Private Values,” *The American Economic Review*, 1985, 75 (2), 150–155.
- Mastroianni, Adam M., Daniel T. Gilbert, Gus Cooney, and Timothy D. Wilson**, “Do conversations end when people want them to?,” *Proceedings of the National Academy of Sciences*, 2021, 118 (10), e2011809118.

- Mei, Qiaozhu, Yutong Xie, Walter Yuan, and Matthew O. Jackson**, “A Turing test of whether AI chatbots are behaviorally similar to humans,” *Proceedings of the National Academy of Sciences*, 2024, *121* (9), e2313925121.
- Merchant, Amil, Simon Batzner, Samuel S Schoenholz, Muratahan Aykol, Gowoon Cheon, and Ekin Dogus Cubuk**, “Scaling deep learning for materials discovery,” *Nature*, 2023, pp. 1–6.
- Mullainathan, Sendhil and Ashesh Rambachan**, “From Predictive Algorithms to Automatic Generation of Anomalies,” Technical Report May 2023. Available at: <https://ssrn.com/abstract=4443738> or <http://dx.doi.org/10.2139/ssrn.4443738>.
- Park, Joon Sung, Joseph C O’Brien, Carrie J Cai, Meredith Ringel Morris, Percy Liang, and Michael S Bernstein**, “Generative agents: Interactive simulacra of human behavior,” *arXiv preprint arXiv:2304.03442*, 2023.
- Patel, R. and E. Pavlick**, “Mapping language models to grounded conceptual spaces,” in “Proceedings of the International Conference on Learning Representations” 2021, p. 79.
- Pearl, J., M. Glymour, and N.P. Jewell**, *Causal Inference in Statistics: A Primer*, Wiley, 2016.
- Pearl, Judea**, “Causal inference in statistics: An overview,” *Statistics Surveys*, 2009, *3* (none), 96 – 146.
- , *Causality*, Cambridge university press, 2009.
- Peterson, Joshua C., David D. Bourgin, Mayank Agrawal, Daniel Reichman, and Thomas L. Griffiths**, “Using large-scale experiments and machine learning to discover theories of human decision-making,” *Science*, 2021, *372* (6547), 1209–1214.

- Rosenbusch, Hannes, Claire E. Stevenson, and Han L. J. van der Maas,** “How Accurate are GPT-3’s Hypotheses About Social Science Phenomena?,” *Digital Society*, July 2023, *2* (2), 26.
- Rosseel, Yves,** “lavaan: An R Package for Structural Equation Modeling,” *Journal of Statistical Software*, 2012, *48* (2), 1–36.
- Santurkar, Shibani, Esin Durmus, Faisal Ladhak, Cino Lee, Percy Liang, and Tatsunori Hashimoto,** “Whose Opinions Do Language Models Reflect?,” 2023.
- Schelling, Thomas C,** “Models of segregation,” *The American economic review*, 1969, *59* (2), 488–493.
- , “Dynamic models of segregation,” *Journal of mathematical sociology*, 1971, *1* (2), 143–186.
- Scherrer, Nino, Claudia Shi, Amir Feder, and David Blei,** “Evaluating the moral beliefs encoded in llms,” *Advances in Neural Information Processing Systems*, 2024, *36*.
- Simon, Herbert A.,** *The Sciences of the Artificial, 3rd Edition* number 0262691914. In ‘MIT Press Books.’, The MIT Press, September 1996.
- Turing, A. M.,** “On Computable Numbers, with an Application to the Entscheidungsproblem,” *Proceedings of the London Mathematical Society*, 1937, *s2-42* (1), 230–265.
- Törnberg, Petter, Diliara Valeeva, Justus Uitermark, and Christopher Bail,** “Simulating Social Media Using Large Language Models to Evaluate Alternative News Feed Algorithms,” 2023.
- Wager, Stefan and Susan Athey,** “Estimation and Inference of Heterogeneous Treatment Effects using Random Forests,” *Journal of the American Statistical Association*, 2018, *113* (523), 1228–1242.

Wei, Jason, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Ed H. Chi, Quoc Le, and Denny Zhou, “Chain of Thought Prompting Elicits Reasoning in Large Language Models,” *CoRR*, 2022, *abs/2201.11903*.

Wright, Sewall, “The method of path coefficients,” *The annals of mathematical statistics*, 1934, 5 (3), 161–215.

Zheng, Stephan, Alexander Trott, Sunil Srinivasa, David C Parkes, and Richard Socher, “The AI Economist: Taxation policy design via two-level deep multiagent reinforcement learning,” *Science advances*, 2022, 8 (18), eabk2607.

A Implementation details

The first step in the system’s process is to query an LLM for the roles of the relevant agents in the scenario. When we say “query an LLM,” we mean this quite literally. We have written a scenario-neutral prompt that the system provides to an LLM, with the scenario added to it. The prompt is scenario-neutral because we can reuse it for any scenario. The prompt takes the following format:

In the following scenario: “{scenario_description}”, Who are the individual human agents in a simple simulation of this scenario?

where {scenario_description} is replaced with the scenario of interest. The LLM then returns a list of agents relevant to the scenario, and we have various checking mechanisms to ensure the LLM’s response is valid.

The system contains over 50 pre-written scenario-neutral prompts to gather all the information needed to generate the SCM, run the experiment, and analyze the results. These prompts have placeholders for the necessary information aggregated in the system’s memory as it progresses through the different parts of the process.

A.1 Constructing variables and drawing causal paths

The system builds SCMs variable-by-variable. It queries an LLM for an outcome involving the agents in the social scenario of interest. We refer to outcomes as endogenous variables because their values are realized during the experiment. This contrasts exogenous variables, the causes, whose values are determined before the experiment.

The system queries the LLM for a list of possible exogenous causes of the endogenous variable, generating a hypothesis as an SCM.²¹ Exogenous variables serve as inputs to the experiment, whose values can be deterministically manipulated to identify causal effects. The system assumes that when an exogenous variable causes an endogenous variable, a single causal path is proposed from the exogenous variable to the endogenous variable. More formally, the system always generates SCMs as a simple linear model. The system currently generates all SCMs with one endogenous variable and as many exogenous causes as a researcher desires. We do little optimization here, although the system can test for interaction terms. In future iterations of the system, a researcher could choose outcomes and causes they are interested in, score hypotheses by interestingness, and generate more complex hypotheses with mediating endogenous variables.²²

A.1.1 Endogenous outcomes

For each endogenous variable, the system generates an operationalization, a type, the units, the possible levels, the explicit questions that need to be asked to measure the variable’s realized value, and how the answers to those questions will be aggregated to get the final data for analysis. Examples of all information collected about the variables in an SCM are provided in Table A.3. The system stores each piece of information about a variable and uses it to determine subsequent information in con-

²¹There is growing evidence that LLMs can be quite good at coming up with ideas and generating hypotheses (Girotra et al., 2023; Rosenbusch et al., 2023).

²²Parallel and crossover experimental designs can be used to identify mediating causal relationships (Imai et al., 2012). These experiments require few assumptions, which are often more plausible when researchers have more control over the experiment, as they usually do with LLMs.

secutive scenario-neutral prompts. This is a kind of “chain-of-thoughts prompting,” or breaking down a complex prompt into a series of simpler prompts. This method can dramatically improve the quality and robustness of an LLM’s performance (Wei et al., 2022).

The first piece of information determined for each endogenous variable is the operationalization. That is, how the possible realizations of said variable can be directly mapped to measurable outcomes that can be observed and quantified. Suppose the outcome variable is **whether or not a deal occurred** from the SCM in Figure 2b.²³ The system could operationalize this as a binary variable, where ‘‘1’’ means a deal occurred and ‘‘0’’ does not. It then stores this information and uses it in a scenario-neutral prompt to choose the variable type.

All variables are determined to be one of five mutually exclusive “types.” These are continuous, ordinal, nominal, binary, or count. By selecting a unique type for each variable, the system can accommodate different distributions when estimating the fitted SCM after the experiment.

Each variable also has units. The units are the specific measure or standard used to represent the variable’s quantified value. This information is used to improve the robustness and consistency of the system’s output when querying the LLM for other information about a variable.

The levels of the variable represent all of the values the variable can realize in a short list. They can take on different forms depending on the variable type, but they all follow a general pattern where they are defined by the range and nature of a variable’s possible values.²⁴

To measure the endogenous outcome, the system generates survey questions for one of the agents. For example, to measure **whether or not a deal occurred**,

²³We continue the practice from Section 2 of using `typewriter text` to denote example information from the system.

²⁴For binary variables, the levels are the two possible outcomes. For ordinal variables, the levels include all possible values that the ordinal variable could take on as determined by its operationalization. The levels are selected for count and continuous variables by segmenting the range of possible values into discrete intervals. In cases where the variable does not have a defined maximum or minimum, categories such as “above X” or “below Y” are included to ensure all possible values are covered.

the system could ask the buyer or the seller, “Did you agree to buy the mug?” Or, if the endogenous variable was the final price of the mug, the system could ask one of the agents, “How much did you sell the mug for?” Even though the simulations have yet to be conducted, the system generates survey questions. As with pre-registration, this reduces unneeded degrees of freedom in the data collection process after the experiment.

Most endogenous variables are measured with only one question. In this case, the answer to this question is the only information needed to quantify the variable. Sometimes, it takes more than one survey question to measure a variable. Maybe the variable is the **average satisfaction of the buyer and the seller**, a variable that requires two separate measurements to quantify. In this case, the system generates separate measurement questions to elicit the buyer’s and the seller’s satisfaction. Then, the system averages the answers to the questions to measure the variable.

We pre-programmed a menu of 6 mechanical aggregation methods: finding the minimum, maximum, average, mode, median, or sum of a list of values. If the system needs to combine the answers to multiple questions to measure a variable, it queries an LLM to select the appropriate aggregation method. Then, the system uses a pre-written Python function to perform said aggregation. We avoid asking the LLM to perform mathematical functions whenever possible, as they often make mistakes.

A.1.2 Exogenous causes

In addition to the explicit measurement questions and data aggregation method, the system collects the same information for the exogenous variables as it does for the endogenous variables. For exogenous variables, these two pieces of information are unnecessary for measurement. In each simulation of the social scenario, a different combination of the values of the exogenous variables is initialized. This is how the system induces variation in an experiment, so the treatments are always known to the system ex-ante.

Causal variables can have one of two possible “scopes.” The scope can be specific to an individual agent or the scenario as a whole. This scope determines how the

system induces variation in the exogenous variables—at the agent or scenario level. Individual-level variables are further designated as either public or private. If private, the variable’s values are only provided to one agent; if public, they are treated as common knowledge to all agents in the scenario.

The system induces variation in the exogenous variables by transforming them into manageable proxy attributes for the agents. The system queries an LLM to create a second-person phrasing of the operationalized variable provided to the agent (or agents, depending on the scope). For instance, with the **buyer’s budget** variable, the attribute could be “**your budget**” for the buyer. These attributes will be assigned to the agents, which we discuss in Section A.2.

With the proxy attribute for the variable, the system queries an LLM for possible values the attribute can take on. These are the induced variations—the treatment conditions for the simulated experiments. By default, the system uses the levels, or a value within each level, of the variable for the possible variation values. For example, these could be $\{\$5, \$10, \$20, \$40\}$ for the **buyer’s budget**.

A.2 Building hypothesis-driven agents

In conventional social science research, human subjects are catch as catch can. Here, we have to construct them from scratch. By “construct” we mean that we prompt an LLM to be a person with a set of attributes. This is quite literal; for example, we could construct an agent in a negotiating scenario with the following prompt:

“You are a buyer in a negotiation scenario with a seller. You are negotiating over a mug. You have a budget of \$20.”

We can construct an agent with any set of attributes we want, which raises the question of what attributes we should use.

We already have the attributes that will be varied to test the SCM, but there are many others we could include. Some work has explored the endowing of agents with many different attributes, but it is unclear what is optimal, sufficient, or even necessary.²⁵ We take a minimalist approach, endowing our agents with goals, constraints,

²⁵The methods have varied, ranging from endowing agents with potentially interesting attributes

roles, names, and any relevant proxy attributes for the exogenous variables. In the future, we could integrate large numbers of diverse agents, perhaps constructed to be representative of some specific population.

A.2.1 Assigning agents attributes


The system collects information for agents independently, similar to its one-at-a-time approach with the variables in the SCM. The system randomly selects an agent, determines its attributes, and then moves on to the next agent. Examples of buyer and seller agents with their attributes are provided in Figure A.1.

For each agent, the system queries the LLM for a random name. Agents perform better in simulations with identifiers to address one another, although this feature can be disabled. An agent’s name can also be varied as a proposed exogenous cause. The system then queries an LLM again, this time for a goal and then a constraint, which we discuss in the following subsection.

Finally, the system cross-checks the values of the proxy attributes between the agents to ensure they overlap appropriately. For example, if the two exogenous variables in the SCM were the `buyer’s budget` and the `seller’s minimum acceptable price`, the system would check to make sure that the `seller’s minimum acceptable price` is not invariably higher than the `buyer’s budget`. We let the LLM determine if these attribute values overlap appropriately. If any discrepancies are found, the system queries the LLM again to resolve them with new values for the proxy attributes. Otherwise, the simulated experiment would waste time and resources because the induced variations were not supported across reasonable values. For example, if the `buyer’s budget` was always below the `seller’s minimum acceptable price`, then they might never make a deal.

(Argyle et al., 2023; Horton, 2023) to using American National Election Study data to create “real” people (Törnberg et al., 2023) to demonstrating that endowing demographic information does not necessarily represent a population of interest (Atari et al., 2023; Santurkar et al., 2023). There is a balance to be struck. While attributes can provide a rich and nuanced simulation, they can also lead to redundancy, inefficiency, and unexpected interactions. In contrast, too few attributes might result in an oversimplified and unrealistic portrayal of social interactions.

Figure A.1: Example agents generated by the system for “two people bargaining over a mug”



Basic Information	Your role is: Seller Your name: Samuel	Your role is: Buyer Your name: Beatrice
Goals & Constraints	Goal: Your goal is to sell the mug at the highest price possible Constraint: Must not accept a price below your minimum selling price	Goal: Try to purchase the mug at the lowest price possible Constraint: Do not offer a price higher than your maximum budget
Exogenously Varied Attributes	Your sentimental attachment: [no attachment, ..., extreme attachment]	Your budget: [\$5, \$10, \$20, \$40]

Notes: In all simulations, agents have a randomly generated name, role, goal, constraint, and proxy attributes for the exogenous variables. To simulate the experiment for the agents in this figure, the system will generate four versions of the seller and four versions of the buyer, each with one of the values for the exogenously varied attributes (assuming there are four possible values for “Your sentimental attachment”). That is $4 \times 4 = 16$ treatments.

A.2.2 The importance of agent goals

Unlike, say, economic agents, whose goals are expressed via explicit utility functions, the LLM agent’s goals are expressed in natural language. In the context of our bargaining scenario, an example goal generated by our system for the seller is to `sell the mug at the highest price possible`. An example constraint is to `not accept a price below your minimum selling price`. These goals and constraints are oriented towards value, but they do not have to be; these are merely the ones generated by the system. A constraint could just have easily been `do not ruin your reputation with your negotiating partner`.

We do not take a prescriptive stance on what these goals *should* be. We let the system decide what is reasonable. These goals can, of course, also be the object of

study in their own right; researchers can vary them or choose their own, but they are seemingly fundamental to any social science for reasons laid out in Simon (1996). Therefore, explicit goals are a requirement for agents in our system.

A.3 Simulation design and execution

LLMs are designed to produce text. And since an independent LLM powers each agent, one agent must finish speaking before the next begins. So, in any multi-agent simulation, there must be a speaking order, which raises the question of how the system should determine this speaking order. Unfortunately, most human conversations do not have an obvious order; people collectively figure out how to interact. We centralize this process, but we could imagine a consensus protocol for who speaks next.

In more straightforward settings with only two agents (e.g., two people bargaining over a mug), the only possible conversational order is for the agents to alternate speaking. As the number of agents in interaction increases beyond two, the number of possible speaking orders grows factorially. For example, with three agents, there are $3! = 6$ ways to order them; with 4 agents, $4! = 24$ orderings, and so on. However, the number of possible orderings of the agents is only part of the complexity.

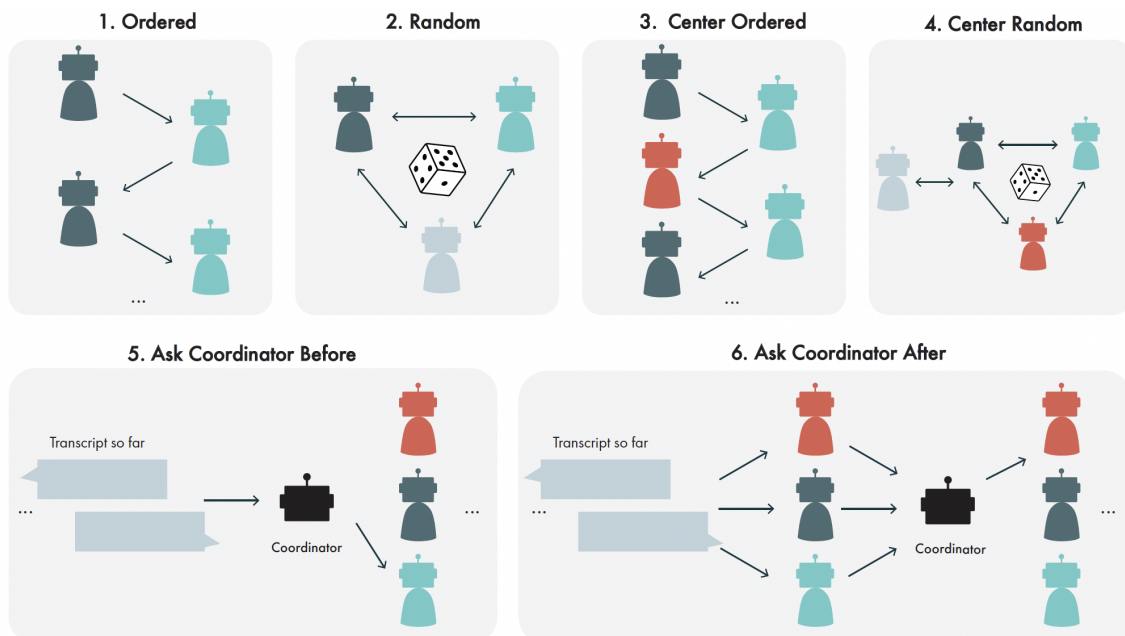
Who speaks next in a given conversation is a product of the participants' personalities, the setting of the conversation, the social dynamics between the speakers, the emotional state of the participants, and many other factors. They are also adaptive—often, the speaking order changes throughout a conversation. For example, in a court proceeding, the judge usually guides the interaction—signaling who speaks between the lawyers, witnesses, and the jury. Each contributes at various and irregular intervals depending on both the type and stage of the proceeding. In a family of two parents and two children, the order of who speaks next varies greatly. It might depend on the parents' moods or how annoying the children have been that day. In contrast, the teacher is typically the main speaker in a high school classroom, although this varies depending on the classroom activity, such as a lecture versus a group discussion. No simple universal formula exists for who speaks next in such

diverse settings.

Like the aggregation methods for outcomes determined by multiple measurement questions, we designed a menu of six interaction protocols. The system queries an LLM to select the appropriate protocol for a given scenario. Figure A.2 provides the menu, and we discuss each in turn.

A.3.1 Turn-taking protocols

Figure A.2: Menu of interaction protocols for the system to choose from for a given scenario.



Notes: (1) The agents speak in a predetermined order. (2) The agents speak in a random order. (3) A central agent alternates speaking with non-central agents in a predetermined order. (4) A central agent alternates speaking with non-central agents in random order. (5) A separate LLM (whom we call the coordinator) determines who speaks next based on the conversation. (6) Each agent responds privately to the conversation so far, and the coordinator realizes one of the responses.

The first interaction protocol is the **ordered** protocol (Figure A.2, option 1), where the agents speak in a predetermined order and continue repeatedly speaking in that order until the simulation is complete. Next is the **random** protocol. An agent

is randomly selected to speak first (Figure A.2, option 2). Then, each subsequent speaker is randomly selected, with the only restriction being that no agent can speak twice in a row.

In more complex scenarios with a central agent—an agent that speaks more than all others—like an auction with an auctioneer or a teacher in a classroom, the system can choose the **central-ordered** or **central-random** protocols (Figure A.2, options 3 and 4). The former features a central agent who interacts alternately with a series of non-central agents, following a predetermined order among the non-central agents. The latter also has a central agent alternating with the non-central agents but in random order. Whenever there is an order of agents or a central agent, we also query the system to determine this order.

Finally, we designed two interaction protocols that provide more flexibility. These interaction protocols involve a separate LLM-powered agent: “the coordinator.” The coordinator can read through transcripts of the conversations and make decisions about the simulations when necessary. It can also answer measurement questions after the experiment. The agents are not aware of the coordinator. The use of the coordinator is the only part of the system that needs quasi-omniscient supervision. Fortunately, LLMs perform so well that they can be used to automate this role.

In the **coordinator-before** protocol (Figure A.2, option 5), the coordinator is given the transcript of the conversation after each agent speaks. Then, it selects the next speaker.

In the **coordinator-after** protocol (Figure A.2, option 6), after each agent speaks, all the agents respond, but only the coordinator can see the responses along with the transcript of the conversation up to that point. Then, the coordinator chooses the response to “realize” as the real response. The realized response is added to the conversation’s transcript, and the rest are deleted as if they had never been made. The only limitation in either of the coordinator protocols is that no agent can speak consecutively.

A.3.2 Executing the experimental simulations

The system runs each experimental simulation in parallel, subject to the computational constraints of the researcher’s machine. When the exogenous variable’s values present too many combinations to sample from, a subset is randomly selected. In every simulation, agents are provided with a description of the scenario, their unique private attributes, the other agents’ roles, any public or scenario-level attributes, and access to the transcript of the conversation. Then, they interact according to the chosen interaction protocol. However, none of the protocols specify when the simulation should end.

It is not obvious how to construct an optimal, nor even good, stopping rule. Human conversations are unpredictable and do not always end when we expect them to or want them to (Mastroianni et al., 2021). An analogous issue is the halting problem in computer science, which is the problem of determining when, if ever, an arbitrary computer program will stop. Turing (1937) proved that no universal algorithm exists to solve the halting problem.

We implemented a two-tier mechanism to determine when to stop each simulation. These apply to all interaction protocols. After each agent speaks, the coordinator receives the transcript and decides if the conversation should continue—a yes or no decision. Additionally, simulations are limited to 20 statements across all agents in the scenario, not including the coordinator.²⁶ Agents are provided a live count of the remaining statements during the conversation.

A.3.3 Post-simulation survey and data collection

After the experiment, the system conducts a post-experiment survey. As determined during the SCM construction, the system asks the relevant agents or the coordinator the survey questions to measure the outcome variable in each simulation. The system then takes this question’s raw answer and saves it as an observation along with the

²⁶Limiting the number of turns in the simulation is partially a convenience. As of the time of running the simulations for this paper, GPT-4 has a maximum token limit of 8,192 tokens, and the system must provide each agent with the entire conversation up to that point each time they need to speak.

values of the exogenous variables. If there is no reasonable answer to the question, say, if the outcome is conditional, then the system will report an *NA* for the variable’s value.

Once the system has the answer to the survey question, it queries an LLM with the survey question, the agent’s response, and information about the variable’s type to determine its correct numerical value as a string. If the variable is a count or continuous variable, it is converted into an integer or a float. If the variable is ordinal or binary, the system queries an LLM to map it to a whole-number integer sequence. If multiple survey questions determine a variable, the system aggregates the answers to the questions using the method selected during the SCM construction phase. Then, it converts the aggregated value to the appropriate type. After parsing the data for each outcome, the system has a data frame with one column of numerical values for each variable in the SCM.

A.4 Path estimation & model fit

With a complete dataset and the proposed SCM, the system can estimate the linear SCM without further queries to an LLM. The system uses the R package lavaan to estimate all paths in the model (Rosseel, 2012).²⁷ The system can standardize all estimates, estimate interactions and non-linear terms, and view various summary statistics for each variable. It can also provide likelihood ratio, Wald, and Lagrange Multiplier tests to evaluate the model fit and compare path estimates. The system can do any statistical estimation or test that is built into lavaan.

A.5 Follow-on experiments

Although we have yet to automate this process, the system can perform follow-on experiments. Insignificant exogenous variables from the first experiment can be dropped. Then, the system could query an LLM for new exogenous variables based

²⁷For those familiar with lavaan and Python, the system automatically generates the correctly formatted string in lavaan syntax using a Python dictionary that stores the structure of the SCM in key-value pairs.

on what might be interesting, given the already tested causal paths. The system would use the same agents and interaction protocol, but the agents would vary on the new exogenous variables and the old ones that were significant in the first experiment. Theoretically, the system can run follow-on experiments ad infinitum, and we can imagine future models that could be very good at proposing potential causal relationships.

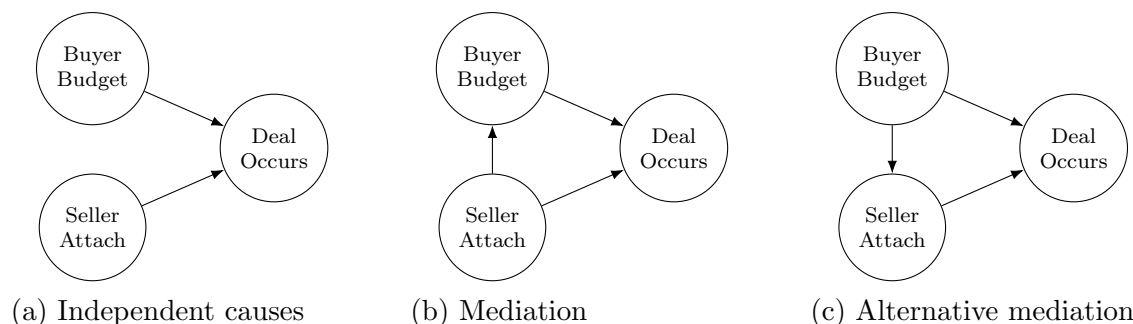
B Hypotheses as structural causal models

Hypotheses stated in natural language can be ambiguous, making it challenging to discern precise implied causal relationships. Suppose a researcher is interested in two-person bargaining scenarios with a buyer and a seller. And she has the following natural language hypothesis about two people bargaining over a mug: “the buyer’s budget and the seller’s sentimental attachment to the mug causally affect whether a deal occurs.” Figure A.3 offers three ways we can interpret this causal statement: (A.3a) the budget and the sentimental attachment could independently affect whether a deal occurs, (A.3b) the budget could mediate the relationship between the attachment and the outcome, or (A.3c), the mediation could be reversed.²⁸

For (A.3a), an example could be an online marketplace where the buyer and seller cannot communicate. When the buyer has a higher budget, she is more likely to buy the mug. If the seller is more sentimentally attached to the mug, he may raise the price and, therefore, lower the probability of a deal. However, without any form of communication, these causal variables would not affect each other. For (A.3b), if the buyer and the seller can communicate and the seller realizes that the buyer is willing to spend more, he might become more attached to the mug and value it higher because of the increased potential sale price. Finally, for (A.3c), the mediated relationship could be reversed. If the buyer sees that the seller is attached to the mug, this may cause her to increase her budget, which increases the probability of a deal. The ambiguity of stating even simple hypotheses makes natural language

²⁸This list of interpretations is not exhaustive.

Figure A.3: Valid graphical interpretations of the same natural language hypothesis.



Notes: Each directed acyclic graph (DAG) is a valid causal interpretation of the following natural language hypothesis: “The buyer’s budget and the seller’s sentimental attachment to the mug causally affect whether a deal occurs.” In contrast, each DAG is unique in its declaration of the causal relationships. In DAGs, each arrow represents a direct causal relationship, and the absence of an arrow between two variables indicates no causal relationship. If a variable is not included in the graph, then there is no stated causal relationship about this variable. While DAGs are unambiguous in their causal claims about which variables cause which other variables, they do not make any claims about the functional form of the relationships between variables.

insufficient for our purposes.

The graphs in Figure A.3 are directed acyclic graphs (DAGs) and represent causal relationships. DAGs unambiguously state whether a variable is a direct cause of another variable—the direction of the arrow indicates the direction of the causal relationship (Hernán and Robins, 2020). The absence of an arrow between two variables indicates no causal relationship. If a variable is not included in the graph, then there is no stated causal relationship involving this variable.

While DAGs are clear in their claims about which variables cause others, they do not make any statements about the functional form of the relationships between variables. In contrast, structural causal models unambiguously state the causal relationships between variables *and* the functional forms of these relationships (Pearl et al., 2016).

Structural causal models (SCM), as first explored by Wright (1934), represent hypotheses as sets of equations. Suppose we assume the relationships between the

variables in Figure A.3 are linear. We can write an SCM for each of the DAGs. Figure A.3a can be stated as:

$$DealOccurs = \beta_1 BuyerBudget + \beta_2 SellerAttachment + \epsilon; \quad (1)$$

Figure A.3b as:

$$BuyerBudget = \beta_0 SellerAttachment + \eta \quad (2)$$

$$DealOccurs = \beta_1 BuyerBudget + \beta_2 SellerAttachment + \epsilon; \quad (3)$$

and Figure A.3c as:

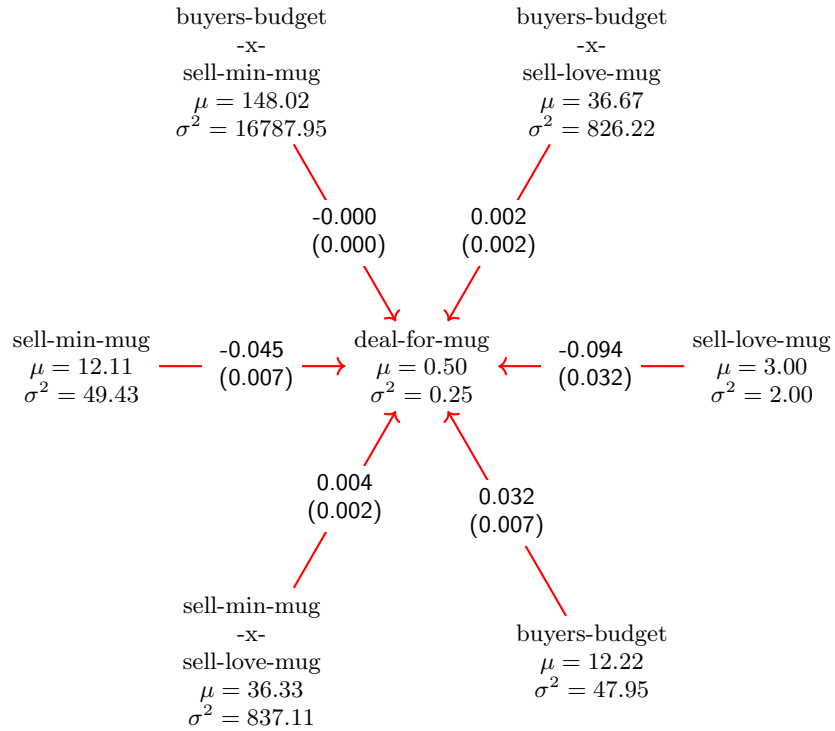
$$SellerAttachment = \beta_0 BuyerBudget + \eta \quad (4)$$

$$DealOccurs = \beta_1 BuyerBudget + \beta_2 SellerAttachment + \epsilon. \quad (5)$$

The set of equations representing the causal relationships between variables makes the SCM. We could also write each SCM with interaction terms for some or all of the causes or even use other types of link functions, and these would all be equally valid representations of the corresponding DAGs.

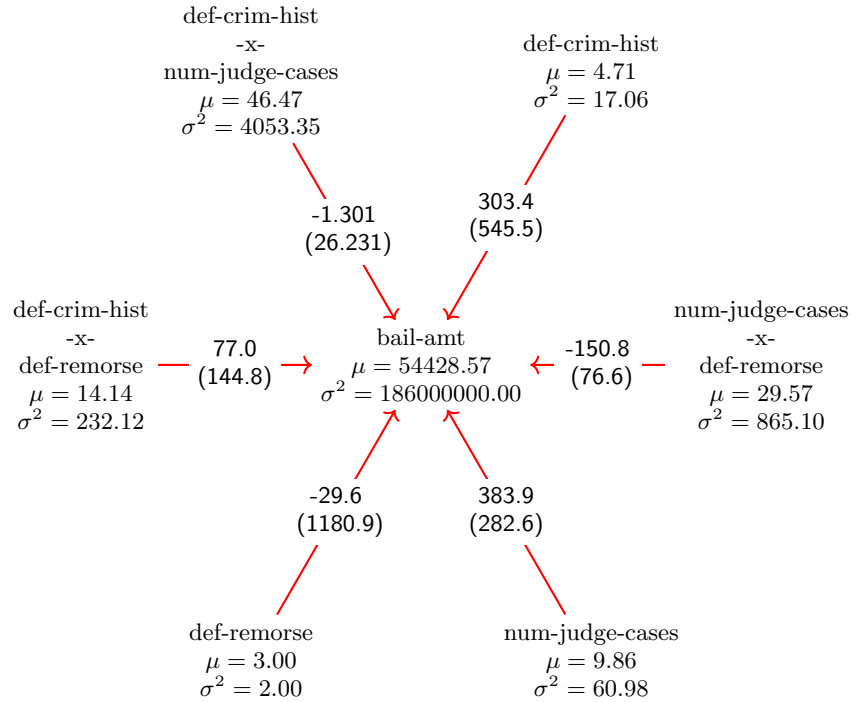
C Additional figures and tables

Figure A.4: Fitted SCM with interaction terms for “two people bargaining over a mug.”



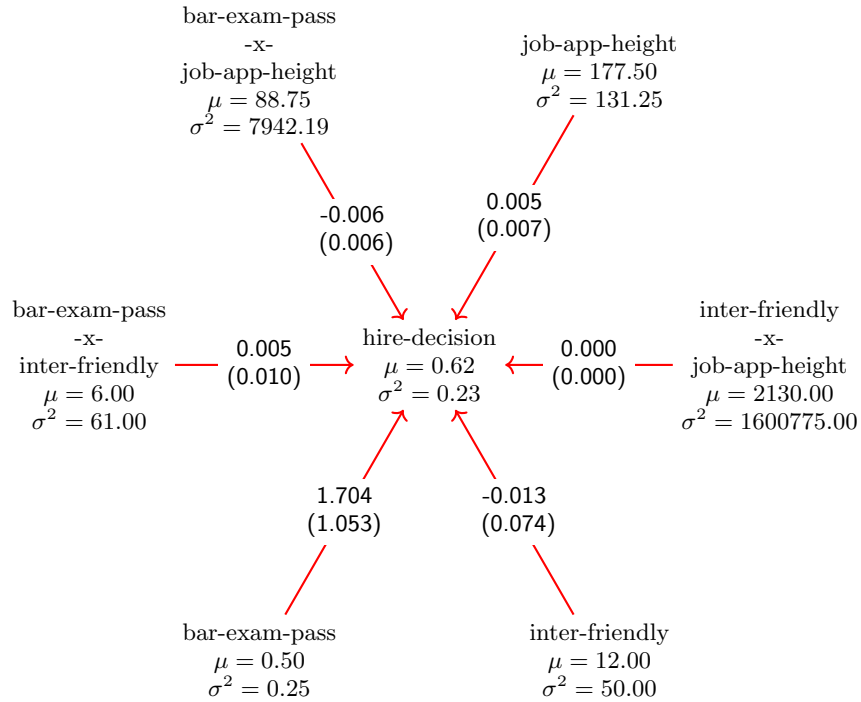
Notes: Each variable is given with its mean and variance. The edges are labeled with their unstandardized path estimate and standard error. There were 405 simulations with these agents: [‘buyer’, ‘seller’].

Figure A.5: Fitted SCM with interaction terms for “a judge is setting bail for a criminal defendant who committed 50,000 dollars in tax fraud.”



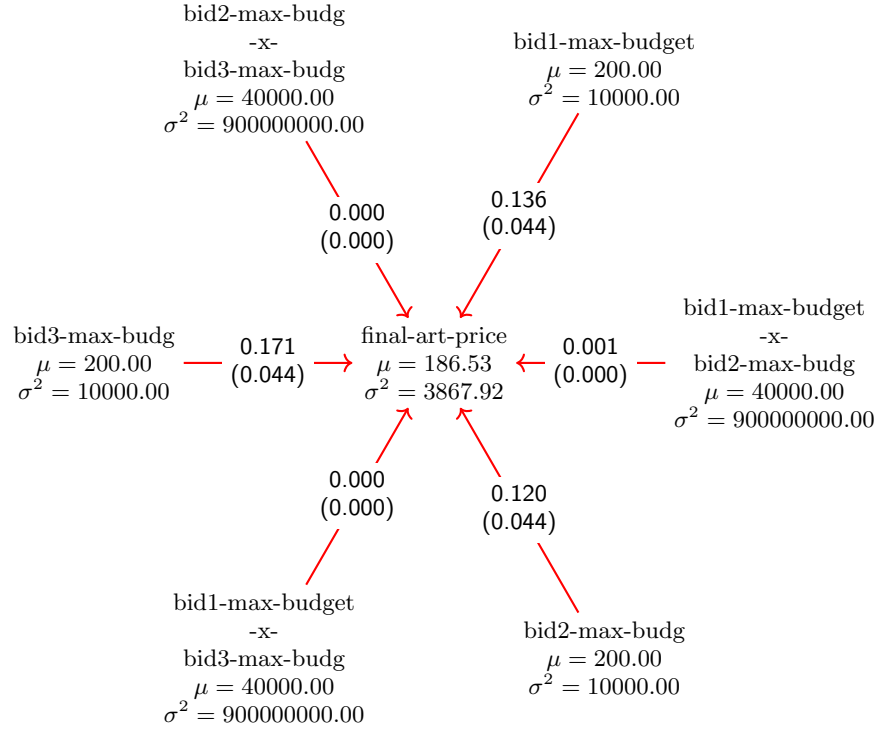
Notes: Each variable is given with its mean and variance. The edges are labeled with their unstandardized path estimate and standard error. There were 245 simulations with these agents: [‘judge’, ‘defendant’, ‘defense attorney’, ‘prosecutor’].

Figure A.6: Fitted SCM with interaction terms for “a person is interviewing for a job as a lawyer.”



Notes: Each variable is given with its mean and variance. The edges are labeled with their unstandardized path estimate and standard error. There were 80 simulations with these agents: [‘job applicant’, ‘employer’].

Figure A.7: Fitted SCM with interaction terms for “3 bidders participating in an auction for a piece of art starting at fifty dollars.”



Notes: Each variable is given with its mean and variance. The edges are labeled with their unstandardized path estimate and standard error. There were 343 simulations with these agents: [‘bidder 1’, ‘bidder 2’, ‘bidder 3’, ‘auctioneer’].

Table A.1: GPT-4’s predictions for the path estimates for the experiments in Section 3 at temperature 0.

Scenario (Outcome)	Exogenous Variable	Path Estimate (SE)	GPT-4 Guess	Two-tailed T-Test	GPT-4 Sign Correct	$ \frac{\text{Predicted}}{\text{Experiment}} $ Estimates
Mug Bargaining (Deal Made)	Buyer’s Budget	0.037* (0.003)	0.05*	$p < 0.001$	Yes	1.35
	Seller’s Min Price	-0.035* (0.002)	-0.07*	$p < 0.001$	Yes	2.00
	Seller’s Attachment	-0.025* (0.012)	0.02	$p < 0.001$	No	0.80
Art Auction (Final Price)	Bidder 1 Budget	0.35* (0.015)	0.5*	$p < 0.001$	Yes	1.43
	Bidder 2 Valuation	0.29* (0.015)	0.5*	$p < 0.001$	Yes	1.72
	Bidder 3 Valuation	0.31* (0.015)	0.5*	$p < 0.001$	Yes	1.610
Bail Hearing (Bail Amount)	Defendant’s Previous Convictions	521.53* (206.567)	5000*	$p < 0.001$	Yes	9.59
	Judge Cases That Day	-74.632 (109.263)	-200	$p = 0.252$	Yes	2.68
	Defendant’s Remorse	-1153.061 (603.325)	-3000*	$p = 0.002$	Yes	2.60
Lawyer Interview (Gets Job)	Passed Bar	0.750* (0.068)	0.6*	$p = 0.03$	Yes	0.80
	Interviewer Friendliness	-0.002 (0.005)	0.2	$p < 0.001$	No	100.00
	Applicant’s Height	0.003 (0.003)	0.1	$p < 0.001$	Yes	33.33

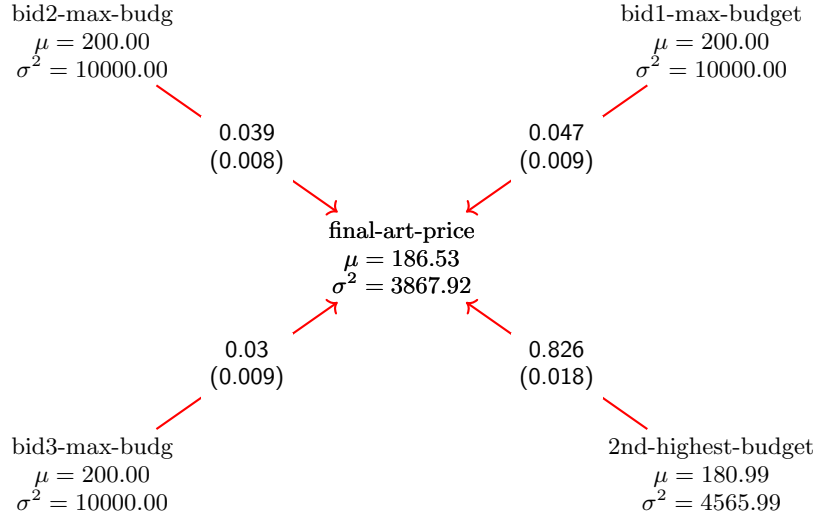
Notes: The table provides GPT-4’s prediction for the path estimate for each experiment in Section 3. From left to right, column 1 provides the scenario and outcome, column 2 provides the causal variable name, column 3 the path estimate and its standard error, and column 4 shows the LLM’s prediction for the path estimate and whether it was predicted to be statistically significant. Column 5 gives the p-value of a two-tailed t-test comparing the predictions to the results, column 6 is whether the predicted sign of the estimate was correct, and column 7 is the magnitude of the difference between the predicted and actual estimate.

Table A.2: GPT-4’s predictions for the path estimates for the experiments in Section 3 at temperature 1.

Scenario (Outcome)	Exogenous Variable	Path Estimate (SE)	GPT-4 Guess	Two-tailed T-Test	GPT-4 Sign Correct (SE)	$ \frac{\text{Predicted}}{\text{Experiment}} - \text{Estimates} $
Mug Bargaining (Deal Made)	Buyer’s Budget	0.037* (0.003)	0.117* (0.016)	$p < 0.001$	Yes	3.16
	Seller’s Min Price	-0.035* (0.002)	0.008* (0.018)	$p = 0.019$	No	0.23
	Seller’s Attachment	-0.025* (0.012)	0.062 (0.013)	$p < 0.001$	No	2.48
Art Auction (Final Price)	Bidder 1 Budget	0.35* (0.015)	1.279* (0.501)	$p = 0.064$	Yes	3.65
	Bidder 2 Valuation	0.29* (0.015)	1.263* (0.501)	$p = 0.053$	Yes	4.36
	Bidder 3 Valuation	0.31* (0.015)	1.269* (0.501)	$p = 0.056$	Yes	4.09
Bail Hearing (Bail Amount)	Defendant’s Previous Convictions	521.53* (206.567)	1785.192* (157.347)	$p < 0.001$	Yes	3.42
	Judge Cases That Day	-74.632 (109.263)	644.316* (79.919)	$p < 0.001$	No	8.63
	Defendant’s Remorse	-1153.061 (603.325)	-879.945* (92.700)	$p = 0.09$	Yes	0.76
Lawyer Interview (Gets Job)	Passed Bar	0.750* (0.068)	0.408* (0.018)	$p = 0.998$	Yes	0.54
	Interviewer Friendliness	-0.002 (0.005)	0.236* (0.015)	$p = 0.999$	No	118
	Applicant’s Height	0.003 (0.003)	0.108 (0.009)	$p = 0.999$	Yes	36

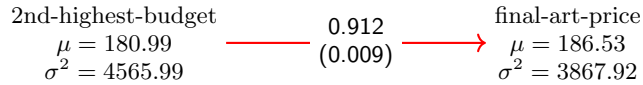
Notes: The table provides GPT-4’s prediction for the path estimate for each experiment in Section 3. Each prediction is the average of 100 prompts at temperature 1. From left to right, column 1 provides the scenario and outcome, column 2 provides the causal variable name, column 3 the path estimate and its standard error, and column 4 shows the LLM’s average prediction for the path estimate and whether it was predicted to be statistically significant more than 50% of the time. The given standard error is for the mean of the predictions, not the LLM’s prediction for the standard error. Column 5 gives the p -value of a two-tailed t -test comparing the average prediction to the results, column 6 is whether the predicted sign of the estimate was correct more than 50% of the time, and column 7 is the magnitude of the difference between the predicted and actual estimate.

Figure A.8: Fitted SCM for auction with bidder’s reservation prices and second highest bid as exogenous variables.



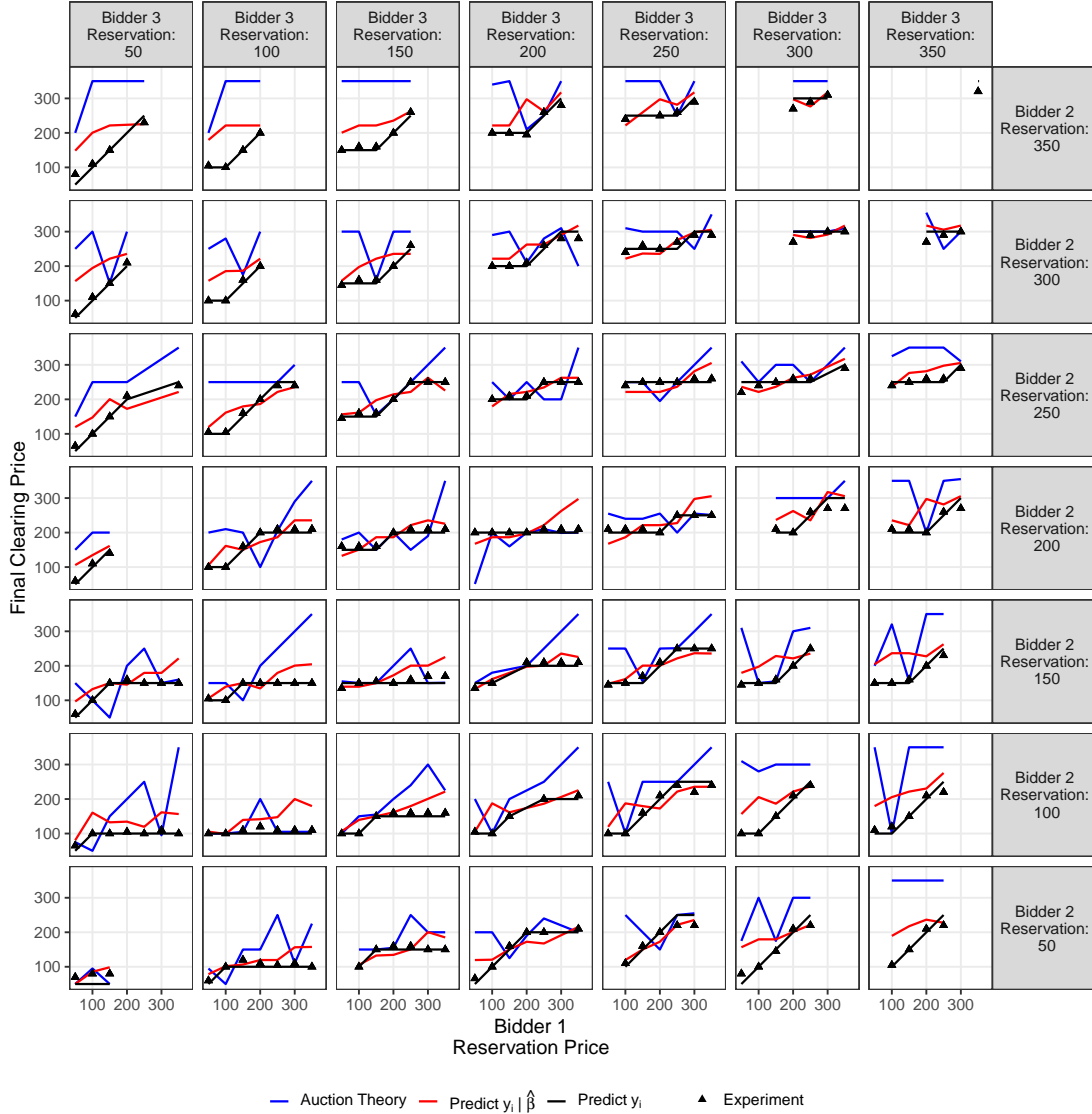
Notes: Each variable is given with its mean and variance. The edges are labeled with their unstandardized path estimate and standard error. There were 343 simulations with these agents: [‘bidder 1’, ‘bidder 2’, ‘bidder 3’, ‘auctioneer’].

Figure A.9: Fitted SCM for auction and second highest bid as exogenous variables.



Notes: Each variable is given with its mean and variance. The edges are labeled with their unstandardized path estimate and standard error. There were 343 simulations with these agents: [‘bidder 1’, ‘bidder 2’, ‘bidder 3’, ‘auctioneer’].

Figure A.10: Comparison of the LLM’s predictions to the theoretical predictions and all experimental results for the auction scenario.



Notes: The columns correspond to the different reservation values for bidder 3 in a given simulation, and the rows correspond to the different reservation values for bidder 2. The y-axis is the clearing price, and the x-axis lists bidder 1’s reservation price. The black triangles track the observed clearing price in each simulated experiment, the black line shows the predictions made by auction theory ($MSE_{Theory} = 128$), the blue line indicates the LLM’s predictions without the fitted SCM—the predict- y_i task ($MSE_{y_i} = 8628$), and the red curve is the LLM’s predictions with the fitted SCM—the predict- $y_i | \hat{\beta}_{-i}$ task ($MSE_{y_i | \hat{\beta}_{-i}} = 1505$).

Figure A.11: Prompt used to elicit LLM predictions for the Predict- $\hat{\beta}$ task.

I have just run an experiment to estimate the paths in the SCM from the TIKZ diagram below, which is delineated by triple backticks. We ran the experiment on multiple instances of GPT-4, once for each combination of the different “Attribute Treatments” in the accompanying table. This table also includes information about the variables and the individual agents involved in the scenario. Your task is to predict the point estimates for the paths in the SCMs as accurately as possible based on the experiments. You can see the summary statistics of the treatment variables below each variable name in the Tikz Diagram. We want to know how good you are at predicting the outcomes of experiments run on you. Make sure you consider the correct units for both the cause and the outcome for each path. Please output your answer in the following form and do not include any other text: {'predictions': dictionary of point estimate predictions for each path} {'sig': dictionary of whether or not each path is significant} ‘‘‘Figure X and Table X’’’

Notes: For each experiment, we input the accompanying table and the TIKZ diagram into the LLM between the triple backticks. For example, for the bargaining scenario, these are Figure 2b and Table 2a.

Table A.3: Example of the information generated for each variable in an SCM.

Information Type	Deal Occurred (Endogenous)	Buyer's Budget (Exogenous)	Seller's Attachment (Exogenous)
Operationalization	1 if a deal occurs, 0 otherwise	Max amount the buyer will spend	Seller's emotional attachment level on a scale
Variable Type	Binary	Continuous	Ordinal
Units	Binary	Dollars	Levels of attachment
Levels	{0, 1}	{\$0-\$5, ..., \$40+}	{Low, ..., High}
Explicit Measurement Questions	Buyer: "Did a deal occur?"	-	-
Data Aggregation Method	Single Value	-	-
Scenario or Individual	-	Individual	Individual
Varied Attribute Proxies	-	"Your budget"	"Your attachment level"
Attribute Treatments	-	{\$3, ..., \$45}	{no attachment, ..., extreme attachment}

Notes: Each row shows a different piece of information generated for the variables in the SCM. The first column represents the type of information, the second column represents the information for the endogenous variable, and the third and fourth columns represent the information for the exogenous variables. This is example information based on the SCM in Figure A.3a.