Automated Social Science:
A Structural Causal Model-Based Approach[*]

Benjamin S. Manning[†]   Kehang Zhu[†]   John J. Horton
MIT   Harvard   MIT & NBER

February 26, 2024

Abstract

We present an approach for automatically generating and testing, *in silico*, social scientific hypotheses. This automation is made possible with 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 automated 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 proposed and tested, finding evidence for some and not others. In the auction experiment, we show that the *in silico* simulation results closely match the predictions of auction theory, but elicited predictions of the clearing prices from an LLM are highly inaccurate. However, the LLM's clearing price predictions are dramatically improved if the model can condition on the fitted structural causal model. When given a proposed structural causal model for one of the scenarios, the LLM is good at predicting the signs of estimated effects, but it cannot reliably predict the magnitudes of those effects. This suggests that explicit social simulation gives the model insight not available purely through direct elicitation. In short, the LLM knows more than it can (immediately) tell.

[†]Both authors contributed equally to this work.

1 Introduction

There is much work on efficiently estimating models but comparatively little work on efficiently generating 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, and there is growing evidence that they indeed do (Binz and Schulz, 2023a; Li et al., 2024), simulation provides a reasonable substitute.[3] In this paper, we combine these ideas—automated hypothesis generation and automated *in silico* hypotheses testing—by using LLMs for both purposes.

A 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. 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 vary on that dimension—a straightforward transition from stated theory to experimental design and data generation. Furthermore, the structural causal model

---

[1]A few examples include generative adversarial networks being used 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 and recursive algorithms to identify heterogenous treatment effects (Athey and Imbens, 2016; Wager and Athey, 2018), and several others (Buyalskaya et al., 2023; Cai et al., 2023; Enke and Shubatt, 2023; Girotra et al., 2023; Parkes and Wellman, 2015; 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; Horton, 2023; Mei et al., 2024; Park et al., 2023a,b; Scherrer et al., 2024)

[3]See Horton (2023) for a discussion on the use of LLMs as economic agents for social science.

offers a pre-specified plan for estimation (Haavelmo, 1943, 1944; Jöreskog, 1970).[4]

We built a computational system implementing the structural causal model-based approach. Starting from a social science "scenario," an LLM is queried to propose outcomes of interest and the relevant agents. For each outcome proposed, potential exogenous causes are again elicited from the LLM. For example, a scenario of interest might be "two people bargaining over a mug." The LLM may propose "whether a sale occurs" as an outcome of interest, with a buyer and a seller as the relevant agents. The LLM might hypothesize that the buyer's willingness to pay may affect the outcome. The completed structural causal model contains endogenous variables (the outcomes), exogenous variables (the potential causes), and paths (the possible effects of the causes). Researchers can raise the LLM's temperature to encourage divergent idea generation and are free to modify whatever hypotheses the system proposes—the system's output is editable at every step.[5]

Agents that vary on the exogenous dimensions of the structural causal model are then generated. For example, in the bargaining scenario, each simulation will vary the buyer's willingness to pay. A separate LLM powers each agent—their conversations are simulated by the LLMs exchanging text. To measure outcomes, survey questions are formulated (e.g., ask the buyer, "Did a sale occur?"), and those questions are administered to the agents after each simulation run. This is possible because LLM-powered agents can each be given a "memory"—a transcript of what happened in the simulation. The answers to the survey questions are used as data to estimate the linear structural causal model.

The structural causal model framework is also useful because it describes exactly what needs to be measured as a downstream outcome subject to the exogenous manipulations of the causes. If we contrast this to, say, a more open-ended social

---

[4]That is in contrast to unstructured data analysis. When a group of social scientists has the same data set on some human behavior or outcome, they can reach very different conclusions when analyzing it independently (Engzell, 2023; Salganik et al., 2020).

[5]Temperature is a parameter that controls how an LLM samples from its response distribution. At temperature zero, the LLM responds deterministically, always outputting the same highest probability response. At higher temperatures, the LLM's responses become more stochastic. One way to think about prompting an LLM at higher temperatures is that it is more likely to generate unexpected responses to a given prompt.

simulation, collecting data on what happened can be challenging. For example, inspecting conversation logs for specific events can be cumbersome and error-prone.[6] This tight connection between theory and data is another key advantage of this approach.

We use this system to explore several social scenarios: (1) two people bargaining over a mug, (2) a lawyer interviewing for a job, (3) a bail hearing for tax fraud, and (4) an ascending price auction for a piece of art. We allow the LLM to propose the outcomes and causes for the first two scenarios and then run the simulations without intervention. For (3) and (4), we hand-selected hypotheses that we decided would be interesting to explore and to illustrate the complementarities between human and machine-derived research.

In terms of qualitative outcomes, the system uncovers several findings from the simulations. 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. The candidate passing the bar exam was the only important factor in her getting the job. Neither the candidate's height nor the interviewer's friendliness affected the outcome. A remorseful defendant was granted lower bail but was not so fortunate if his criminal history was extensive. However, the judge's case count before the hearing—which was hypothesized to matter—did not affect the final bail amount. An increase in the bidders' reservation prices causes an increase in the final price of the piece of art. These are not "counterintuitive" findings, but it is important to emphasize they were the result of empiricism, not just model introspection. But this does raise the question of whether a simulation is even necessary.

Instead of simulation, could an LLM simply do a "thought experiment" about the proposed *in silico* experiment and achieve the same insight—or is the actual simulation necessary? To test this idea, we describe the simulations that will be

---

[6]Research by Park et al. (2023a) has demonstrated the remarkable potential of simulating humans with groups of LLM-powered agents. They endow agents with personas and memory systems, allow them to interact in an open environment, and discover the emergence of human-like behaviors. These behaviors include throwing parties, going on dates, and making friends. While impressive, a problem with such open-ended social simulations is selecting and analyzing outcomes. To unveil insights, researchers may need to comb through thousands of lines of unstructured text.

run and ask the LLM to predict the results—both the path estimates and point predictions. To make this concrete, suppose we had the simple linear model $y = \beta X$ 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. Interestingly and reassuringly, the predictions made by auction theory are close to the empirical results—often matching the *in silico* simulations exactly.

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. Two-sided t-tests revealed that 11 out of 12 predictions significantly differed from the empirical estimates, often by many multiples. On average, the LLM predicts the path estimates will be 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 using leave-one-out cross-validation. For each X_i, we fit the structural causal model using all but the ith 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 more accurate than in the predict-y_i task without the fitted model. The mean squared error (MSE) is lower by almost an order of magnitude, and the predictions are much closer to those made by the theory.

Performing these experiments required a substantial software infrastructure, which will be entirely open-source. Although this paper focuses on the conceptual issues, the software's features are briefly worth discussing. The current capabilities of our system can broadly be broken down into three categories: (1) a researcher can use the system to explore hypotheses and the parameter space of social phenomena it-

eratively, (2) it can be used to quickly explore *any* LLM's behavior in a practically limitless set of social scenarios, and (3) the system enables nearly frictionless communication, replicability, and ease of sharing and building on results.

The remainder of this paper is structured as follows: In Section 2, we describe structural causal models and their importance in the approach. Section 3 provides an overview of the system. Section 4 provides some results generated using our system. Section 5 explores an LLM's capacity to predict the results of these experiments before they are conducted. Section 6 discusses the advantages of identifying causal structure ex-ante when analyzing data and the problems that arise when trying to determine such structure ex-post. Section 7 explains how the system generates structural causal models and agents, runs the simulated experiments, and estimates the model. The paper concludes in Section 8 with a discussion of future research and why automated *in silico* social science might be useful.

2 Hypotheses as structural causal models

Hypotheses stated in natural language can express rich ideas. However, they can be ambiguous, making it challenging to discern their 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 1 offers three ways we can interpret this causal statement: (1a) the budget and the sentimental attachment could independently affect whether a deal occurs, (1b) the budget could mediate the relationship between the attachment and the outcome, or (1c), the mediation could be reversed.[7]

For (1a), 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

---

[7]This list of interpretations is not exhaustive.

Figure 1: Valid graphical interpretations of the same natural language hypothesis.

(a) Independent causes   (b) Mediation   (c) Alternative mediation
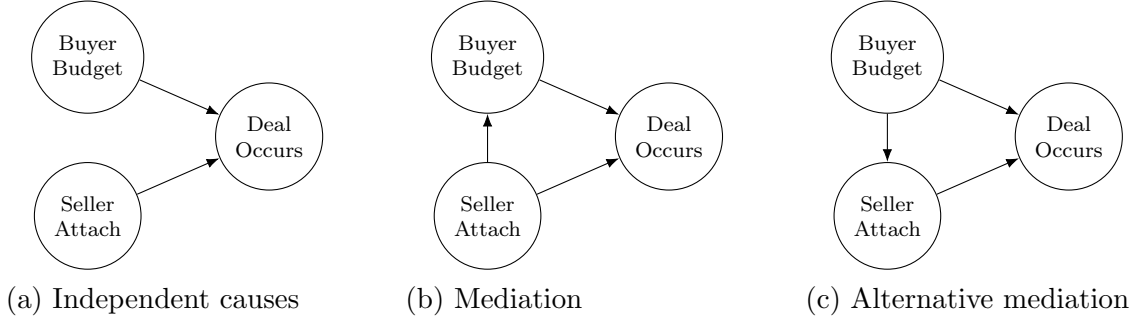
Notes: Each directed acyclic graph (DAG) is a valid causal interpretation of the following natural language hypothesis: "The buyer's budget and the seller's sentimental attachment to the mug causally affect whether a deal occurs." In contrast, each of the DAGs is unique in its declaration of the causal 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. Unless stated otherwise, we assume all DAGs specify linear relationships between variables.

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 (1b), 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 (1c), 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 insufficient for our purposes. Fortunately, there are other options.

The graphs in Figure 1 are known as directed acyclic graphs (DAGs) and are a common way to 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 Sewall Wright almost 100 years ago (Wright, 1934), represent hypotheses as sets of equations. Suppose we assume the relationships between the variables in Figure 1 are linear. We can write an SCM for each of the DAGs. Figure 1a can be stated as:

$$DealOccurs = \beta_1 BuyerBudget + \beta_2 SellerAttachment + \epsilon; \tag{1}$$

Figure 1b as:

$$BuyerBudget = \beta_0 SellerAttachment + \eta \tag{2}$$

$$DealOccurs = \beta_1 BuyerBudget + \beta_2 SellerAttachment + \epsilon; \tag{3}$$

and Figure 1c as:

$$SellerAttachment = \beta_0 BuyerBudget + \eta \tag{4}$$

$$DealOccurs = \beta_1 BuyerBudget + \beta_2 SellerAttachment + \epsilon. \tag{5}$$

To be clear, an SCM is the set of equations that represent the causal relationships between variables, not the individual equations.[8] For example, Equations 2 and 3 define one SCM together. We could also write each SCM with quadratic or 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. No matter the form, any variable on the left-hand side of an equation in an SCM is an endogenous variable; it is caused by the variables on the right-hand side of the equation and any unobserved factors—the error terms (ϵ and η). The effects of the causes on the endogenous variables are captured by the coefficients in the equations. If a variable never appears on the left-hand side of an equation but does appear on

---

[8] Although in the case of equation 1, all of the causal relationships are possible to represent in a single equation. In this limited case, the equation is the SCM.

the right, it is exogenous. I.e., it is determined by external factors and is independent of all other variables in the SCM. If a variable never appears in any equation, then no stated causal relationship involving this variable exists.

Although DAGs only make claims about which variables cause others, they present a clean and intuitive way to visualize causal relationships. To take advantage of this visual clarity, we will use DAGs and linear SCMs interchangeably unless stated otherwise. That is, we presume Figure 1a implies Equation 1 and vice versa, Figure 1b implies the combined Equations 2 and 3, and so on. This assumption is not necessarily correct for any set of causal relationships but offers an unequivocal starting point to generate hypotheses. One can always test functional assumptions with experimental data and a known causal structure by estimating SCMs with different forms and comparing the results.

SCMs are useful for our approach because unambiguous causal statements allow for unambiguous estimation (Haavelmo, 1943, 1944; Jöreskog, 1970). Identification is algorithmic—one can determine whether it is possible to identify the effect of a given cause by examining the graph, and if possible, know precisely which variables must be controlled for (Pearl, 2009b). This also allows for unambiguous experimental design in the sense that algorithms can determine precisely which variables must be exogenously manipulated to identify the effect of a given cause.

Despite their effectiveness in our application, many social scientists, particularly economists, have been somewhat lukewarm towards the use of SCMs. The advantages here mirror that of the traditional use of formal models to state hypotheses: whether or not formal economic models are correct, their statements are unambiguous and testable.

3   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 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 not possible, 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.

The system is implemented in Python and uses GPT-4 for all LLM queries. 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 Section 7. 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 4, and the additional analyses in Sections 5 and 6.

The system takes as input some 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, and (8) plans a follow-on experiment. Figure 2 illustrates these steps, and we will briefly explore each in greater depth.

Figure 2: An overview of the automated system.
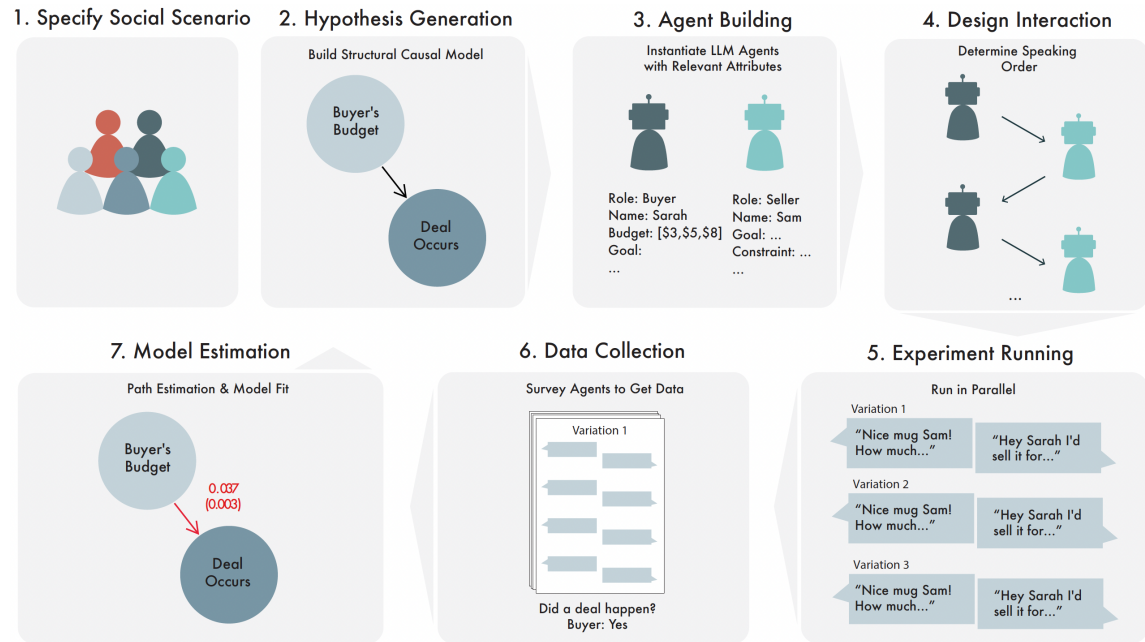
Notes: Each step in the process corresponds to an analogous step in the social scientific process as done by humans. The development of the hypothesis guides the experimental design, execution, and model estimation. Researchers can edit the system's decisions at any step in the process.

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.[9] We use `Typewriter text` to

---

[9]When we say "query an LLM," we mean this literally. We have written a prompt that the 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*

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 operationalizes the outcome 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 2), and stores the relevant information about the operationalizations associated with each variable.[10] 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 2, 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 gather data about the outcomes from the agents automatically 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 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

---

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 ouctomes and proposed causes. 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.

[10]The system generates several other pieces of information about each variable, which help guide the experimental design and data analysis. See Section 7 for further details.

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 Section 7, and they are presented in Figure 11, 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 2). 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 2), 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. This is even a place for possible experimentation as the structure of social interactions can impact various outcomes of interest (Jahani et al., 2023; Rajkumar et al., 2022; Sacerdote, 2001).

Finally, the system gathers the data for analysis. Outcomes are measured by asking the agents the survey questions (Figure 2, step 6) as determined before the experiment. The data is then used to estimate the linear SCM.[11] For our negotiation,

---

[11]Since all causes are exogenously manipulated in the experiment, there are no mediating relationships. More formally, the system estimates the fitted SCM in the form of Equation 1 and its presumed equivalent, Figure 1a.

that would be a simple linear model with a single path estimate for the effect of the buyer's budget on the probability of a deal—the final step in Figure 2. 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, the system can repeat the process. 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.

4   Results of experiments

We present results for four social scenarios explored using the system. In the first two scenarios, our involvement in the system's process was restricted to entering the description of the scenario. In the third scenario, we picked some causes and outcomes that we thought might be interesting. In the fourth, the auction, we edited some of the agents to make them symmetric. For these scenarios, the system took between 30 minutes and 3 hours to complete the entire social scientific process, including generating the hypothesis, running the experiment, and estimating the fitted SCM.

4.1   Bargaining over a mug

We first use the system to simulate "two people bargaining over a mug."[12]   The system selected a buyer and seller as the relevant agents, the outcome as whether a deal occurs, and the buyer's budget, the seller's minimum acceptable price, and the seller's emotional attachment to the mug as potential causes.

Table 1 provides the information generated by the system about the SCM and the experimental design. The leftmost column of the table lists the variables in the

---

[12]We put "two people bargaining over a mug" in quotes because it is the exact input we entered into the system for it to run the entire experiment.

SCM. The second column gives the name of the most important information for each variable (e.g., the type, the treatment variations etc.). The third column provides the realized value of each piece of information named in the second column. The system automatically generated all this information by iteratively querying the LLM.

Table 1: Information automatically generated to build the SCM and design the experiment for "two people bargaining over a mug."

| Variable Name	Information Name	Information Value
Whether or not a deal occurs for the mug (deal-for-mug)	Measurement Question	coordinator: did the buyer and seller explicitly agree on the price of the mug during their interaction?
	Variable Type	binary
Buyer's budget (buyers-budget)	Attribute Treatment Values (9)	['3', '6', '7', '8', '10', '13', '18', '20', '25']
	Proxy Attribute Name	your budget for the mug
	Relevant Agent	buyer
	Variable Type	continuous
Seller's minimum acceptable price for the mug (sell-min-mug)	Attribute Treatment Values (9)	['3', '5', '7', '8', '10', '13', '18', '20', '25']
	Proxy Attribute Name	your minimum acceptable price for the mug
	Relevant Agent	seller
	Variable Type	continuous
Seller's feelings of love towards the mug (sell-love-mug)	Attribute Treatment Values (5)	['no emotional attachment', 'slight emotional attachment', 'moderate emotional attachment', 'high emotional attachment', 'extreme emotional attachment']
	Proxy Attribute Name	your emotional attachment to the mug
	Relevant Agent	seller
	Variable Type	ordinal

Notes: The left-most column of the table lists the variables in the SCM. The second column gives the name of the most important pieces of information about each variable. The third column shows the realized value of each piece of information named in the second column. The proxy attribute and one of its values are directly provided to the relevant agent in each simulation.

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 Treatment Values" in

Table 1) 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 3 provides the fitted SCM. Each variable is given with its mean and variance. For ordinal variables (e.g., the seller's feelings of love), we treat the levels as numerical values. The raw path estimates and their standard errors are shown on the arrows. The buyer and seller reached a deal for the mug in roughly half of the simulations, and all three causes had a statistically significant effect on the probability of a deal.

Figure 3: Fitted SCM for "two people bargaining over a mug."
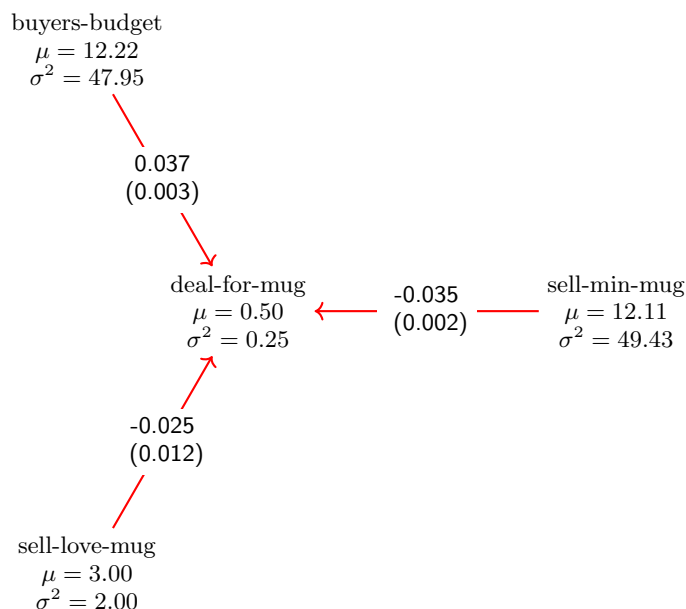
Notes: *Each variable is given with its mean and variance. The edges are labeled with their unstandardized path estimate and standard error. There were 405 simulations with these agents: ['buyer', 'seller'].*

We report standardized effect sizes estimate with $\hat{\beta}*$. Standardized effect sizes being "a one standard deviation increase in X causes a $\hat{\beta}*$ standard deviation increase in Y." A one-dollar increase in the buyer's budget caused an average increase of 3.7 percentage points in the probability of a deal ($\hat{\beta}* = 0.51$, $p < 0.001$). 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 ($\hat{\beta}^* = -0.07$, $p = 0.044$).

We also estimated another potential SCM incorporating interactions among all causal variables (Figure A.1). Of the three interactions, only the seller's attachment interacted with the seller's minimum acceptable price seemed to matter ($\hat{\beta}^* = 0.21$, $p = 0.036$). All three of the proposed causes were still independently significant in this specification.

4.2 A bail hearing

Next, we explore "a judge is setting bail for a criminal defendant who committed 50,000 dollars in tax fraud." 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. Here, the system chose the judge as a center agent and the prosecutor, defense attorney, and defendant as the non-center agents (in that order). This means the judge spoke first in every simulation, alternating with the other agents: judge, then prosecutor, then judge, then defense attorney, then judge, then defendant, and so on. As described in Section 7.3, we call this the "center-ordered" interaction protocol.

Table 2 shows that the outcome is the final bail amount, and the three selected 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.

Table 2: Information automatically generated to build the SCM and design the experiment for "a judge is setting bail for a criminal defendant who committed 50,000 dollars in tax fraud."

Variable Name	Information Name	Information Value
The bail amount set by the judge (bail-amt)	Measurement Question	judge: what was the bail amount you set for the defendant?
	Variable Type	continuous
Defendant's criminal history (def-crim-hist)	Attribute Treatment Values (7)	['0', '1', '2', '3', '6', '9', '12']
	Proxy Attribute Name	number of your prior convictions
	Relevant Agent	defendant
	Variable Type	count
Number of cases the judge has already heard that day (num-judge-cases)	Attribute Treatment Values (7)	['0', '2', '5', '9', '12', '18', '23']
	Proxy Attribute Name	number of cases you have already heard today
	Relevant Agent	judge
	Variable Type	count
Defendant's level of remorse (def-remorse)	Attribute Treatment Values (5)	['no expressed remorse', 'low expressed remorse', 'moderate expressed remorse', 'high expressed remorse', 'extreme expressed remorse']
	Proxy Attribute Name	your level of expressed remorse
	Relevant Agent	defendant
	Variable Type	ordinal

Notes: The left-most column of the table lists the variables in the SCM. The second column gives the name of the most important pieces of information about each variable. The third column shows the realized value of each piece of information named in the second column. The proxy attribute and one of its values are directly provided to the relevant agent in each simulation.

In the fitted SCM in Figure 4, 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 with borderline significance ($\hat{\beta}^* = -0.12$, and $p = 0.056$).

Figure 4: Fitted SCM 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'].*

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.2), none of the other interactions or the standalone causes have a significant effect, including the defendant's criminal history.

4.3 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 a job applicant and an employer as the agents. Unlike the previous simulations, we manually selected the variables in the SCM. Table 3 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.[13]

Table 3: Information automatically generated to build the SCM and design the experiment for "a person is interviewing for a job as a lawyer."

Variable Name	Information Name	Information Value
The decision of the employer to hire or not hire (hire-decision)	Measurement Question	employer: have you decided to hire the job applicant?
	Variable Type	binary
Whether the applicant passed the bar exam (bar-exam-pass)	Attribute Treatment Values (2)	['not passed', 'passed']
	Proxy Attribute Name	your bar exam status
	Relevant Agent	job applicant
	Variable Type	binary
The interviewer's level of friendliness (inter-friendly)	Attribute Treatment Values (5)	['2', '7', '12', '17', '22']
	Proxy Attribute Name	number of positive phrases to use during interview
	Relevant Agent	employer
	Variable Type	count
The job applicant's height (job-app-height)	Attribute Treatment Values (8)	['160', '165', '170', '175', '180', '185', '190', '195']
	Proxy Attribute Name	your height in centimeters
	Relevant Agent	job applicant
	Variable Type	continuous

Notes: The left-most column of the table lists the variables in the SCM. The second column gives the name of the most important pieces of information about each variable. The third column shows the realized value of each piece of information named in the second column. The proxy attribute and one of its values are directly provided to the relevant agent in each simulation.

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 ($n = 80$).

---

[13]There is evidence that height affects labor market outcomes (Case and Paxson, 2008; Vogl, 2014).

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 if so desired.

We can see in Figure 5 that only the applicant passing the bar has a clear causal 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. When we test for interactions, none are significant (Figure A.3).

Figure 5: Fitted SCM for "a person is interviewing for a job as a lawyer."
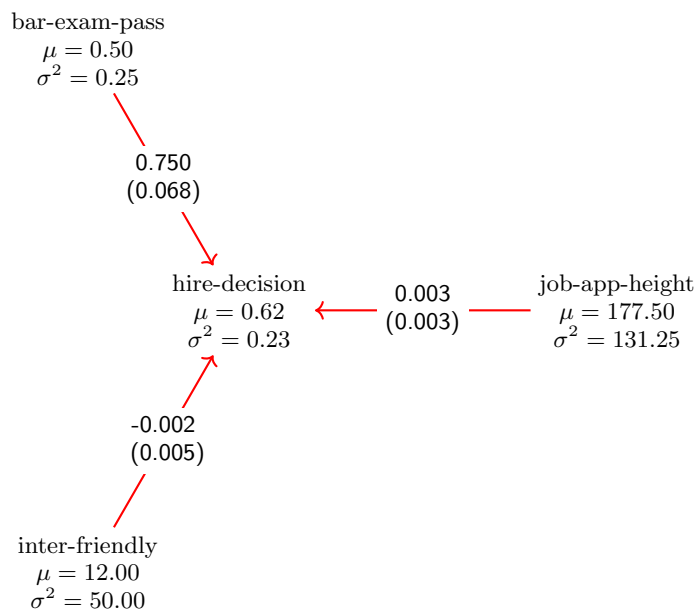
Notes: Each variable is given with its mean and variance. The edges are labeled with their un-standardized path estimate and standard error. There were 80 simulations with these agents: ['job applicant', 'employer'].

4.4 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 4 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.

Table 4: Information automatically generated to build the SCM and design the experiment for "3 bidders participating in an auction for a piece of art starting at fifty dollars."

Variable Name	Information Name	Information Value
Final price of the piece of art (final-art-price)	Measurement Question	auctioneer: what was the final bid for the piece of art at the end of the auction?
	Variable Type	continuous
Bidder 1's maximum budget for the piece of art (bid1-max-budget)	Attribute Treatment Values (7)	['$50', '$100', '$150', '$200', '$250', '$300', '$350']
	Proxy Attribute Name	your maximum budget for the piece of art
	Relevant Agent	bidder 1
	Variable Type	continuous
Bidder 2's maximum budget for the piece of art (bid2-max-budg)	Attribute Treatment Values (7)	['$50', '$100', '$150', '$200', '$250', '$300', '$350']
	Proxy Attribute Name	your maximum budget for the piece of art
	Relevant Agent	bidder 2
	Variable Type	continuous
Bidder 3's maximum budget for the piece of art (bid3-max-budg)	Attribute Treatment Values (7)	['$50', '$100', '$150', '$200', '$250', '$300', '$350']
	Proxy Attribute Name	your maximum budget for the piece of art
	Relevant Agent	bidder 3
	Variable Type	continuous

Notes: The left-most column of the table lists the variables in the SCM. The second column gives the name of the most important pieces of information about each variable. The third column shows the realized value of each piece of information named in the second column. The proxy attribute and one of its values are directly provided to the relevant agent in each simulation.

All four variables are operationalized in dollars. To maintain symmetry in the simulations, we also 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.

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 6 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 \$0.352, \$0.293, and \$0.313 increase in the final price for the piece of art for each respective bidder ($\hat{\beta}^* = 0.57$, $p < 0.001$; $\hat{\beta}^* = 0.47$, $p < 0.001$; $\hat{\beta}^* = 0.5$ $p < 0.001$). That is, a one-dollar increase in any bidder's budget increases the expected final price by roughly 33 cents. These quantities make sense as each bidder has a $\frac{1}{3}$ chance of being marginal.

Figure 6: Fitted SCM for "3 bidders participating in an auction for a piece of art starting at fifty dollars."
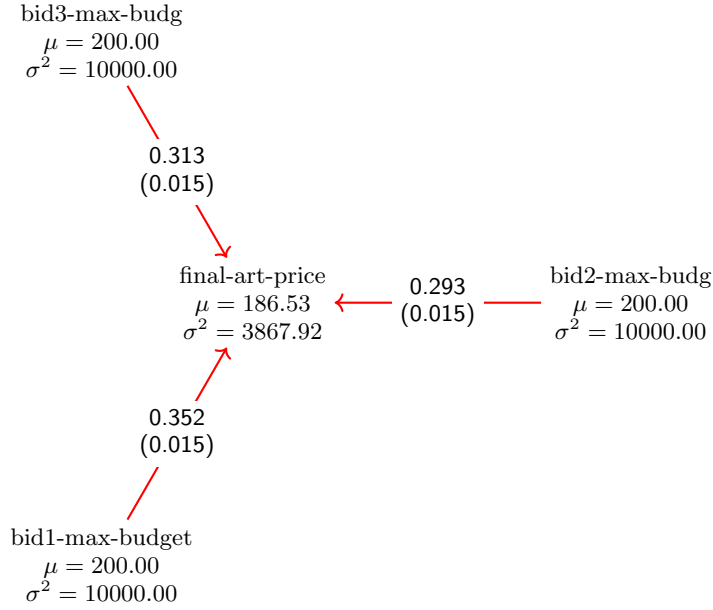
Notes: Each variable is given with its mean and variance. The edges are labeled with their unstandardized path estimate and standard error. There were 343 simulations with these agents: ['bidder 1', 'bidder 2', 'bidder 3', 'auctioneer'].

5 LLM predictions for paths and points

It is worth reiterating that the results in the previous section were not generated through direct elicitation from an LLM but rather through experimentation. Although the experiments were fast and inexpensive, they were not free. This raises the question of whether the simulations were even necessary.[14] 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 come up with an SCM and elicit its predictions about the relationships between the variables.

To test this idea, we describe the simulations to the LLM and ask it to predict the results—path estimates and point predictions. Specifically, we modeled each scenario as $y = \beta X$, 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.[15] The experiments from Section 4 provided us with estimates for $\hat{\beta}$ (a 1 x k 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 (Krishna, 2009).[16] 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.

---

[14]The simulations in Section 4 cost between \$100 and \$700.

[15]The equation $y = \beta X$ is equivalent to a simple linear SCM.

[16]In formal terms, suppose b_i is the reservation price of bidder i. If there are n bidders, then $B = \{b_1, b_2, ..., b_n\}$ represents the set of all the bidders' reservation prices. The theory predicts that the auction clears at a price p where $p = \max(\{B \setminus \max(B)\})$.

5.1 Predicting y_i

For each combination of exogenous variables, we provide the LLM with a prompt that includes the experiment description, the scenario description, and the agents in the scenario.[17] 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 6.

Figure 7 presents a comparison of the LLMs predictions, the simulated experiments, and the predictions made by auction theory.[18] 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 auction 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 remains constant or even decreases as the second-highest reservation price across the agents increases. In contrast, auction theory is highly accurate in its predictions of the final bid price in the experiment—the black line often perfectly tracks the black triangles.[19] The 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 ($MSE_{Theory} = 128$).[20]

---

[17]In 80 out of the 343 simulations, the agents made the maximum number of statements (20) allowed by the system before the auction ended. We remove these observations because auction theory does not make predictions about partially completed auctions.

[18]We 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.5 shows the full set of predictions. The results are generally the same.

[19]There are a few observations where the empirical clearing price is slightly above or below the theory prediction. In most cases where it was off, this was due to the auctioneer incrementing the bid price above the second-highest reservation price in the last round.

[20]MSE is reported for all predictions, not just the subset shown in Figure 7.

Figure 7: Comparison of the LLM's predictions to the theoretical predictions and a subset of experimental results for the auction scenario.
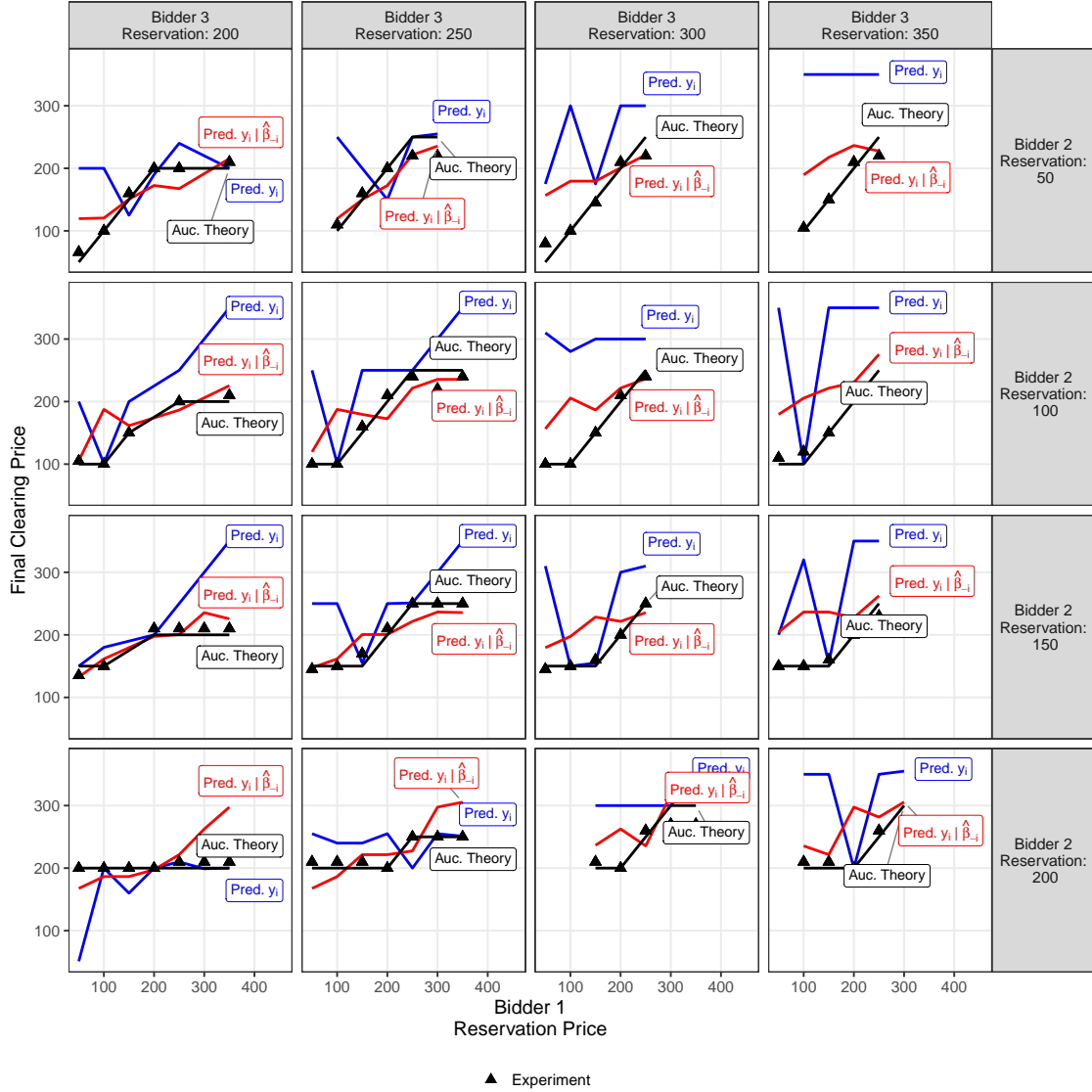
Notes: The columns correspond to the different reservation values for bidder 3 in a given simulation, and the rows correspond to the different reservation values for bidder 2. The y-axis is the clearing price, and the x-axis lists bidder 1's reservation price. The black triangles track the observed clearing 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 line is the LLM's predictions with the fitted SCM—the predict-$y_i|\hat{\beta}_{-i}$ task ($MSE_{y_i|\hat{\beta}_{-i}} = 1505$).

5.2 Predicting $\hat{\beta}$

We prompted the LLM to predict the path estimates and whether they would be statistically significant for each simulated experiment in Section 4. This is the predict-$\hat{\beta}$ task. We then compare the LLM's predictions to the fitted SCMs.

We provide the LLM with extensive information to make its prediction. This information includes the proposed SCM, the operationalizations of the variables, the number of simulations, and the possible treatment values. We provide the full prompt used to elicit the predictions in Figure A.6.

Table 5 shows the LLM's predictions for the path estimates. 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 whether the predicted sign of the estimate was correct, and column 7 the magnitude of the difference between the predicted and actual estimate ($|\frac{\text{Predicted}}{\text{Experiment}}|$).

Two-tailed t-tests rejected 11 out of 12 of the LLM's predictions as being the same as their corresponding empirical estimates. The only exception was the predicted effect of the judge's case count on the final bail amount in the tax fraud scenario, but the prediction was still more than two times larger than the actual estimate. The predictions 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 difference 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.

If LLMs are anything like humans in predicting behavior, it is not surprising that predicting path estimates is difficult. Humans often overestimate the effect sizes of interventions in behavioral experiments (Gandhi et al., 2023). Even experienced social scientists can struggle to accurately predict the results of experiments within their fields of expertise (Broockman et al., 2023; Dimant et al., 2022; Milkman et al.,

Table 5: GPT-4's predictions for the path estimates for the experiments in Section 4.

Scenario (Outcome)	Exogenous Variable	Path Estimate (SE)	GPT-4 Guess	Two-tailed T-Test	GPT-4 Sign Correct	$\left\lvert \frac{\text{Predicted}}{\text{Experiment}} \right\rvert$ 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	0.352* (0.015)	0.5*	$p < 0.001$	Yes	1.42
	Bidder 2 Valuation	0.293* (0.015)	0.5*	$p < 0.001$	Yes	1.71
	Bidder 3 Valuation	0.313* (0.015)	0.5*	$p < 0.001$	Yes	1.60
Bail Hearing (Bail Amount)	Defendant's Previous Convictions	521.53* (206.567)	5000*	$p < 0.001$	Yes	9.59
	Judge Case Number 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 4. 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.

2021, 2022).

What is clear from these predictions is that there is some information that the LLM did not have access to before the simulations but was revealed through experimentation on independent versions of itself.

5.3 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, we use the data from the experiments to estimate $\hat{\beta}_{-i}$, the path estimates from the SCM excluding the ith experiment. We then prompt the LLM to predict the outcome for each X_i given $\hat{\beta}_{-i}$, performing leave-one-out cross-validation.

The red line in Figure 7 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.[21]

However, the LLM's predictions on the predict-$y_i | \hat{\beta}_{-i}$ task are still not as accurate as the predictions made by auction theory ($MSE_{Theory} = 128$).[22] 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 second-highest reservation price of the bidders. 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.

---

[21]Since we use LOOCV, a given X_i is not used to estimate $\hat{\beta}_{-i}$ when predicting y_i. The prediction is, therefore, out of sample.

[22]It is also less accurate than the mechanical predictions made by the fitted SCM using LOOCV $MSE_{Mechanistic} = 725$. Either the LLM cannot do the math or is still conditioning on other information beyond the path estimates when making its predictions.

6 Advantages of ex-ante identified causal structure

The benefits of the SCM-based approach are clear—it allows us to build systems that can automatically generate and experimentally test hypotheses. These experiments reveal information not immediately available to an LLM through direct elicitation. However, automation is not the only advantage of using SCMs to guide data generation and analysis.

A randomized experiment on human subjects is often not possible. Observational data may be the only option available. While inferring causal structure from the data in these cases may be appealing, it is often difficult to do so. For example, large-scale simulations can generate massive amounts of unstructured observational data that can be analyzed for various purposes (Park et al., 2023a). While this can be informative, attempting to determine causal structure ex-post can be problematic. In this section, we discuss how assuming or searching for causal structure in data can lead to misidentification and how using SCMs avoids this problem.

6.1 Assuming causal structure from data

All estimates in the fitted SCMs in Section 4 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.[23] I.e., the coefficients on the fitted SCM are identified. For example, in the bargaining experiment, perhaps we are interested in the length of the conversation as an outcome, even though it was not a part of the original SCM. The conversation length could be operationalized as the sum of the number of statements made by all agents, and we can use the transcript from the finished experiment to measure it. We can then fit an SCM with the data and get unbiased estimates of

---

[23]When we say "downstream," we mean any variable whose value is realized after the agents begin interacting in the simulated conversations.

the exogenous variables' effect on the conversation's length.

Figure 8a shows this fitted SCM using the data from the experiment in Section 4. Both the buyer's budget and the seller's minimum price have a significant effect on the length of the conversation ($p < 0.001$; $p = 0.026$), but the seller's emotional attachment does not ($p = 0.147$).

Figure 8: Comparison of the true and misspecified SCMs.

(a) Correctly specified SCM   (b) Misspecified SCM
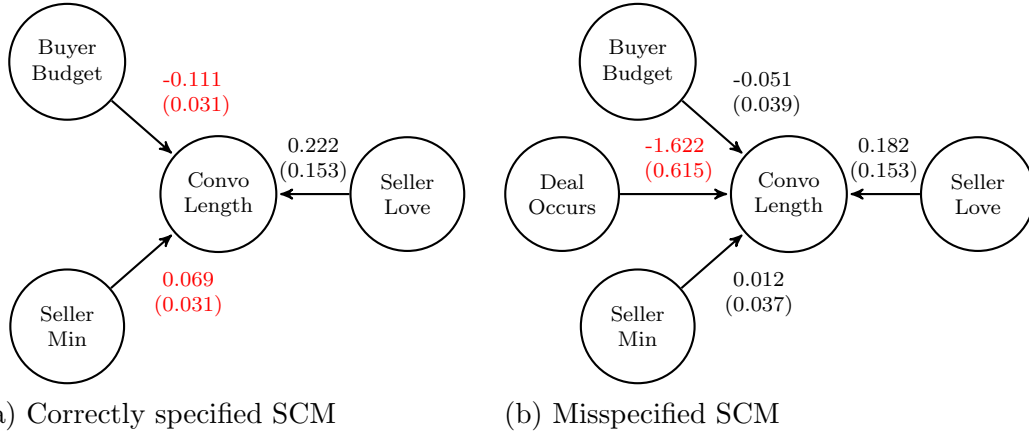
Notes: Statistically significant paths are marked in red ($\alpha = 0.05$). Each path is given with its estimated coefficient and standard error in parentheses. Both SCMs are estimated using the data from the bargaining scenario in Section 4. Subfigure (a) provides a correctly specified SCM from a randomized 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.

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). If we want to estimate the causal relationships between these variables, we would have to make untestable assumptions. 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 all causally affect the length of the conversation.

Figure 8b provides the fitted SCM for this proposed causal structure. Only whether a deal was made was estimated to have a significant effect on the length

of the conversation ($p = 0.008$). But we know this is wrong. We have the true causal structure in Figure 8a 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. Here, they are insignificant and far closer to zero ($p = 0.189$; $p = 0.755$). Whether or not the deal occurred is a bad control that biases the estimates—it is probably codetermined with the length of the conversation.[24]

The informed econometrician may presume that she would never make such a mistake, but many researchers are not so savvy.[25] We were unsure of it until we had unbiased estimates from the correctly specified SCM as a reference. There are also many kinds of bad controls, and many of them 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 has many variables, even when their relationships may seem obvious.

The SCM-based approach avoids the 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.

6.2   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

---

[24]We cannot be sure about the causal relationship between the length of the conversation and whether a deal was made because neither is exogenously varied in the experiment. All we know is that controlling for whether or not a deal occurs induces bias, as we have the experiment as a reference.

[25]LLMs are definitely not yet savvy enough to avoid this mistake.

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.[26]

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.).[27] 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 4.[28]

In some experiments, the algorithm incorrectly identified the causal structure. Figure 9 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.
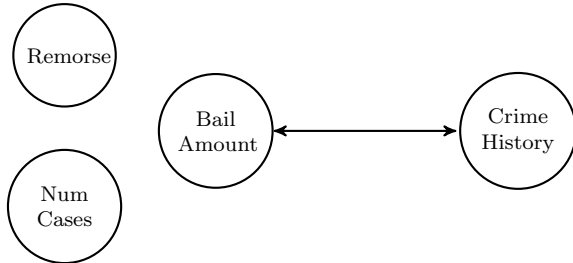
The GES algorithm identified the defendant's criminal history and the bail amount as the only variables in the scenario with any causal relationship. This is partially correct—we know from the experiment that an increase in the defendant's previous convictions caused an increase in the average bail amount. However, the algorithm identified the causal relationship as equally likely in either direction. There was 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

---

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

[27]The number of possible DAGs grows exponentially with the number of nodes. For example, for $n = 1, 2, 3$, and 4 nodes, there are 1, 3, 25, and 543 possible DAGs. This is a combinatorial explosion, and it is not feasible to evaluate all potential models for a large number of nodes, which presents further problems for this approach.

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

Figure 9: 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 4 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.

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.

7   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 the prompt. 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.

7.1 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 is in contrast to 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. Exogenous variables serve as inputs to the experiment, whose values can be deterministically manipulated to identify causal effects. Our 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 in the format of Equation (1) and its presumed equivalent Figure 1a—as a simple linear model. Our 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 (like the SCM in Figure 1b). 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.

7.1.1 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 in Table A.1. Each piece of information about a variable is stored by the system and is then used to determine subsequent information in consecutive scenario-neutral prompts. This is a kind of "chain-of-thoughts prompting", or the process of 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 to directly map the possible realizations of said variable 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 1a.[29] 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

---

[29]We continue the practice from Section 3 of using `typewriter text` to denote example information from the system.

a variable's possible values.[30]

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`, 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 refrain from asking the LLM to perform mathematical functions whenever possible, as they often make mistakes.

---

[30]For binary variables, the levels are the two possible outcomes. For nominal variables, the levels comprise the categories representing different groups or types the variable can realize. A category labeled "other" (or an equivalent term) is always included to account for any values that do not fit into the specified categories. For example, if a nominal variable was "the color of the agent's hair," the levels might be: {Brown, Blond, Black, Grey, White, Other}. 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.

7.1.2 Exogenous causes

Besides 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 7.2.

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`.

7.2 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.[31] We take a minimalist approach, endowing our agents with only goals, constraints, 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.

7.2.1 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.[32] Examples of buyer and seller agents with their attributes are provided in Figure 10.

For each agent, the system queries the LLM for a random name. The agents behave better in the simulations when they have identifiers to address one another, although this feature can be disabled. The system queries an LLM again, this time for a goal and then a constraint.

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 vari-

---

[31]The methods have varied, ranging from endowing agents with interesting attributes (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.

[32]The system already has the agent's roles from the construction of the SCM.

Figure 10: Example agents generated by the system for "two people bargaining over a mug"

Basic Information	**Your role is**: Seller	**Your role is**: Buyer
	Your name: Samuel	**Your name**: Beatrice
Goals & Constraints	**Goal**: Your goal is to sell the mug at the highest price possible	**Goal**: Try to purchase the mug at the lowest price possible
	Constraint: Must not accept a price below your minimum selling price	**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 are endowed with 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.

ables 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.

7.2.2 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.

7.3 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 four 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' per-

sonalities, 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 11 provides the menu, and we discuss each in turn.

7.3.1 Turn-taking protocols

The first interaction protocol is the `ordered` protocol (Figure 11, 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 11, 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 11, 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

Figure 11: Menu of interaction protocols for the system to choose from for a given scenario.
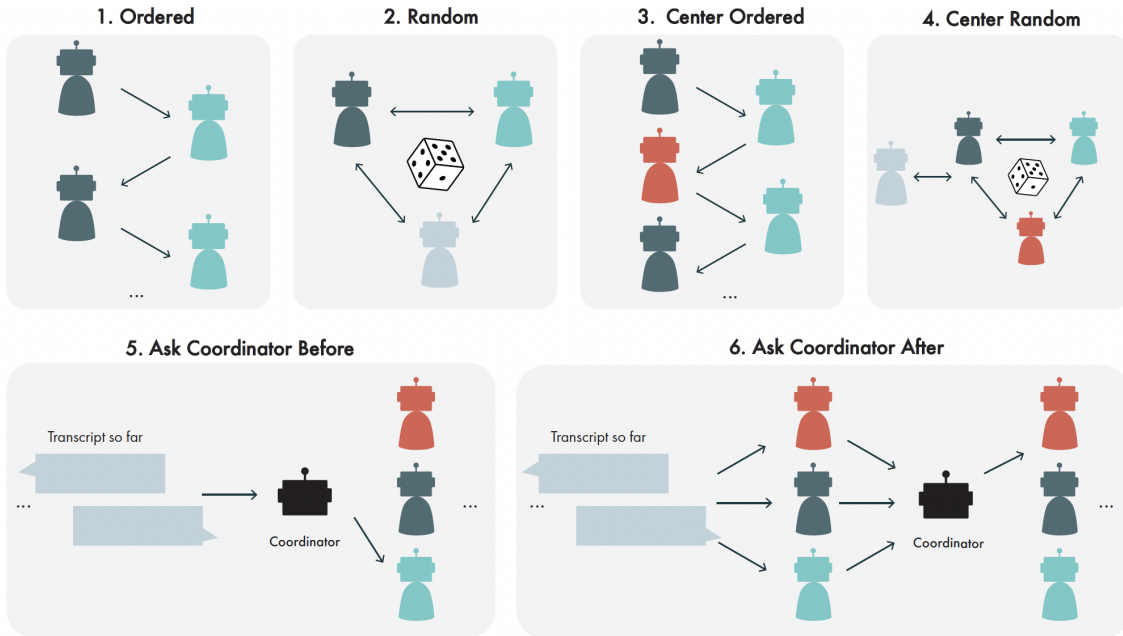
Notes: (1) The agents speak in a predetermined order. (2) The agents speak in a random order. (3) A central agent alternates speaking with non-central agents in a predetermined order. (4) A central agent alternates speaking with non-central agents in random order. (5) A separate LLM (whom we call the coordinator) determines who speaks next based on the conversation. (6) Each agent responds in private to the conversation so far, and the coordinator realizes one of the responses.

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 11, 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 11, 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 twice in a row.

7.3.2 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.[33] Agents are provided a live count of the remaining statements during the conversation.

7.3.3 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 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 the variable is categorical, the system repeats this process, except it generates a mapping for each raw value to a list of dummy variables. 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 unless there is a categorical variable, which always uses dummy variables. In this case, the categorical variable will add $k - 1$ columns for that variable, where k is the number of categories.

---

[33]Limiting 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.

7.4 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).[34] 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.

7.5 Follow-on experiments

Although we have not yet automated 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 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.

8 Conclusion

This paper presents an approach to automated social science made possible through the use of SCMs and LLMs. In this final section, we will discuss some features of the system, some additional theoretical considerations, and areas for future research.

---

[34]For 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.

8.1 Use-cases and benefits of the system

Why might our systems and simulations with LLM-power agents be useful for social science research? One view is that these kinds of simulations are simple dress rehearsals for "real" social science. A more expansive and exciting view is that the LLM agents are close enough stand-ins for human subjects that these simulations would yield insights that 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, such as those used to explore how individual preferences can lead to surprising social patterns (Schelling, 1969, 1971).[35] This view would mirror recent advances in the use of machine learning for protein folding (Jumper et al., 2021) and material discovery (Merchant et al., 2023).

The system presented in this paper can generate these controlled experimental simulations en masse. That contrasts most academic social science research as currently practiced (Almaatouq et al., 2022). 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. Automated experimental social simulations could help alleviate many of the problems that pervade the social sciences, improving the reliability and transparency of the research process (Camerer et al., 2018; Camerer, 2022; Simmons et al., 2011; Simonsohn et al., 2014; Yarkoni, 2022). There is still, of course, the fundamental jump from simulations to human subjects.

8.2 Interactivity

The system allows a scientist to monitor its entire process. Should a researcher disagree with or be uncertain about a decision made by the system, they can probe

---

[35]See Horton (2023) for a full discussion on the differences between traditional agent-based modeling and the use of LLM-powered agents. This position reflects our views as it was written recently by authors on this paper.

47

the system regarding its choice. This allows the researcher to either (1) understand why the decision was made, (2) ask the system to come up with a different option for that decision, or (3) input their own custom choice for that decision.

A researcher can even ignore much of the automation process and fill in the details themselves. They can choose the variables of interest, their operationalizations, the attributes of the agents, how the agents interact, or customize the statistical analysis, among other decision points. Different parts of the system can also accommodate different types of LLMs simultaneously. For example, a researcher could use GPT-4 to generate hypotheses and Llama-2-70B to power the agents' simulated social interactions.

8.3 Replicability

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

The system's entire procedure is exportable as a JSON file with the fitted SCM.[36] 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, perfectly replicate the experiment, or extend the work.

8.4 Future research

While designing our system, we encountered several areas for new research. 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

---

[36]A JSON (JavaScript Object Notation) is a data format that is easy for humans to read and write and easy 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.

information, personalities, and other traits are not included in the agent's attributes unless they are a part of the SCM. To improve the fidelity of the simulations, it might make sense to add some or all of these attributes to the agents. It is not only unclear how to optimize this process but also which set of initial attributes needs optimization.

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. 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 whom we dub the 'coordinator." The coordinator functions as a quasi-omniscient assistant who can read through transcripts and make choices about the speaking order of other agents in the simulations. 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.

Lastly, 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 social 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 and Guido Imbens, "Recursive partitioning for heterogeneous causal effects," *Proceedings of the National Academy of Sciences*, 2016, *113* (27), 7353–7360.

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.

Broockman, David E., Joshua L. Kalla, Christian Caballero, and Matthew Easton, "Political Practitioners Poorly Predict Which Messages Persuade the Public," Technical Report, University of California, Berkeley 10 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.

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.

Case, Anne and Christina Paxson, "Stature and status: Height, ability, and labor market outcomes," *Journal of political Economy*, 2008, *116* (3), 499–532.

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.

Dimant, Eugen, Elena Giulia Clemente, Dylan Pieper, Anna Dreber, Michele Gelfand, Michael Hallsworth, Aline Holzwarth, Piyush Tantia, and Behavioral Science Units Consortium, "Politicizing mask-wearing: predicting the success of behavioral interventions among republicans and democrats in the U.S.," *Scientific Reports*, May 2022, *12* (1), 7575.

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 Gölz, David C Parkes, Ariel D Procaccia, Gili Rusak, Itai Shapira, and Manuel Wüthrich, "Generative Social Choice," *arXiv preprint arXiv:2309.01291*, 2023.

Fudenberg, Drew, Jon Kleinberg, Annie Liang, and Sendhil Mullainathan, "Measuring the Completeness of Economic Models," *Journal of Political Economy*, 2022, *130* (4), 956–990.

Gandhi, Linnea, Anoushka Kiyawat, Colin Camerer, and Duncan J. Watts, "Hypothetical Nudges Provide Misleading Estimates of Real Behavior Change," Technical Report, University of Pennsylvania 2023. Available at OSF Preprints: `https://osf.io/preprints/psyarxiv/c7mkf`.

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.

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.

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.

Jahani, Eaman, Samuel P. Fraiberger, Michael Bailey, and Dean Eckles, "Long ties, disruptive life events, and economic prosperity," *Proceedings of the National Academy of Sciences*, 2023, *120* (28), e2211062120.

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.

Krishna, Vijay, *Auction theory*, Academic press, 2009.

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.

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.

Milkman, Katherine L., Dena Gromet, Hung Ho, Joseph S. Kay, Timothy W. Lee, Pepi Pandiloski, Yeji Park, Aneesh Rai, Max Bazerman, John Beshears, Lauri Bonacorsi, Colin Camerer, Edward Chang, Gretchen Chapman, Robert Cialdini, Hengchen Dai, Lauren Eskreis-Winkler, Ayelet Fishbach, James J. Gross, Samantha Horn, Alexa Hubbard, Steven J. Jones, Dean Karlan, Tim Kautz, Erika Kirgios, Joowon Klusowski, Ariella Kristal, Rahul Ladhania, George Loewenstein, Jens Ludwig, Barbara Mellers, Sendhil Mullainathan, Silvia Saccardo, Jann

Spiess, Gaurav Suri, Joachim H. Talloen, Jamie Taxer, Yaacov Trope, Lyle Ungar, Kevin G. Volpp, Ashley Whillans, Jonathan Zinman, and Angela L. Duckworth, "Megastudies improve the impact of applied behavioural science," *Nature*, December 2021, *600* (7889), 478–483.

Milkman, Katherine, Linnea Gandhi, Mitesh S. Patel, Heather N. Graci, Dena M. Gromet, Hung Ho, Joseph S. Kay, Timothy W. Lee, Jake Rothschild, Jonathan E. Bogard, Ilana Brody, Christopher F. Chabris, Edward Chang, Gretchen B. Chapman, Jennifer E. Dannals, Noah J. Goldstein, Amir Goren, Hal Hershfield, Alex Hirsch, Jillian Hmurovic, Samantha Horn, Dean S. Karlan, Ariella S. Kristal, Cait Lamberton, Michelle N. Meyer, Allison H. Oakes, Maurice E. Schweitzer, Maheen Shermohammed, Joachim Talloen, Caleb Warren, Ashley Whillans, Kuldeep N. Yadav, Julian J. Zlatev, Ron Berman, Chalanda N. Evans, Rahul Ladhania, Jens Ludwig, Nina Mazar, Sendhil Mullainathan, Christopher K. Snider, Jann Spiess, Eli Tsukayama, Lyle Ungar, Christophe Van den Bulte, Kevin G. Volpp, and Angela L. Duckworth, "A 680,000-person megastudy of nudges to encourage vaccination in pharmacies," *Proceedings of the National Academy of Sciences*, 2022, *119* (6), e2115126119.

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`.

OpenAI, "GPT-4 System Card," Technical Report, OpenAI 2023.

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.

Park, Yang Jeong, Daniel Kaplan, Zhichu Ren, Chia-Wei Hsu, Changhao Li, Haowei Xu, Sipei Li, and Ju Li, "Can ChatGPT be used to generate scientific hypotheses?," *arXiv preprint arXiv:2304.12208*, 2023.

Parkes, David C and Michael P Wellman, "Economic reasoning and artificial intelligence," *Science*, 2015, *349* (6245), 267–272.

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.

Rajkumar, Karthik, Guillaume Saint-Jacques, Iavor Bojinov, Erik Brynjolfsson, and Sinan Aral, "A causal test of the strength of weak ties," *Science*, 2022, *377* (6612), 1304–1310.

Rosseel, Yves, "lavaan: An R Package for Structural Equation Modeling," *Journal of Statistical Software*, 2012, *48* (2), 1–36.

Sacerdote, Bruce, "Peer Effects with Random Assignment: Results for Dartmouth Roommates*," *The Quarterly Journal of Economics*, 05 2001, *116* (2), 681–704.

Salganik, Matthew J., Ian Lundberg, Alexander T. Kindel, Caitlin E. Ahearn, Khaled Al-Ghoneim, Abdullah Almaatouq, Drew M. Altschul, Jennie E. Brand, Nicole Bohme Carnegie, Ryan James Compton, Debanjan Datta, Thomas Davidson, Anna Filippova, Connor Gilroy, Brian J. Goode, Eaman Jahani, Ridhi Kashyap, Antje Kirchner, Stephen McKay, Allison C. Morgan, Alex Pentland, Kivan Polimis, Louis Raes, Daniel E. Rigobon, Claudia V. Roberts, Diana M. Stanescu, Yoshihiko Suhara, Adaner Usmani, Erik H. Wang, Muna Adem, Abdulla Alhajri, Bedoor AlShebli, Redwane Amin, Ryan B. Amos, Lisa P.

Argyle, Livia Baer-Bositis, Moritz Büchi, Bo-Ryehn Chung, William Eggert, Gregory Faletto, Zhilin Fan, Jeremy Freese, Tejomay Gadgil, Josh Gagné, Yue Gao, Andrew Halpern-Manners, Sonia P. Hashim, Sonia Hausen, Guanhua He, Kimberly Higuera, Bernie Hogan, Ilana M. Horwitz, Lisa M. Hummel, Naman Jain, Kun Jin, David Jurgens, Patrick Kaminski, Areg Karapetyan, E. H. Kim, Ben Leizman, Naijia Liu, Malte Möser, Andrew E. Mack, Mayank Mahajan, Noah Mandell, Helge Marahrens, Diana Mercado-Garcia, Viola Mocz, Katariina Mueller-Gastell, Ahmed Musse, Qiankun Niu, William Nowak, Hamidreza Omidvar, Andrew Or, Karen Ouyang, Katy M. Pinto, Ethan Porter, Kristin E. Porter, Crystal Qian, Tamkinat Rauf, Anahit Sargsyan, Thomas Schaffner, Landon Schnabel, Bryan Schonfeld, Ben Sender, Jonathan D. Tang, Emma Tsurkov, Austin van Loon, Onur Varol, Xiafei Wang, Zhi Wang, Julia Wang, Flora Wang, Samantha Weissman, Kirstie Whitaker, Maria K. Wolters, Wei Lee Woon, James Wu, Catherine Wu, Kengran Yang, Jingwen Yin, Bingyu Zhao, Chenyun Zhu, Jeanne Brooks-Gunn, Barbara E. Engelhardt, Moritz Hardt, Dean Knox, Karen Levy, Arvind Narayanan, Brandon M. Stewart, Duncan J. Watts, and Sara McLanahan**, "Measuring the predictability of life outcomes with a scientific mass collaboration," *Proceedings of the National Academy of Sciences*, 2020, *117* (15), 8398–8403.

Santurkar, Shibani, Esin Durmus, Faisal Ladhak, Cinoo 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*.

Simmons, Joseph P., Leif D. Nelson, and Uri Simonsohn, "False-Positive Psychology: Undisclosed Flexibility in Data Collection and Analysis Allows Presenting Anything as Significant," *Psychological Science*, 2011, *22* (11), 1359–1366. PMID: 22006061.

Simon, Herbert A., *The Sciences of the Artificial, 3rd Edition* number 0262691914. In 'MIT Press Books.', The MIT Press, September 1996.

Simonsohn, Uri, Leif D. Nelson, and Joseph P. Simmons, "P-curve: A key to the file-drawer," *Journal of Experimental Psychology: General*, 2014, *143* (2), 534–547.

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.

Vogl, Tom S, "Height, skills, and labor market outcomes in Mexico," *Journal of Development Economics*, 2014, *107*, 84–96.

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.

Yarkoni, Tal, "The generalizability crisis," *Behavioral and Brain Sciences*, 2022, *45*, e1.

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.
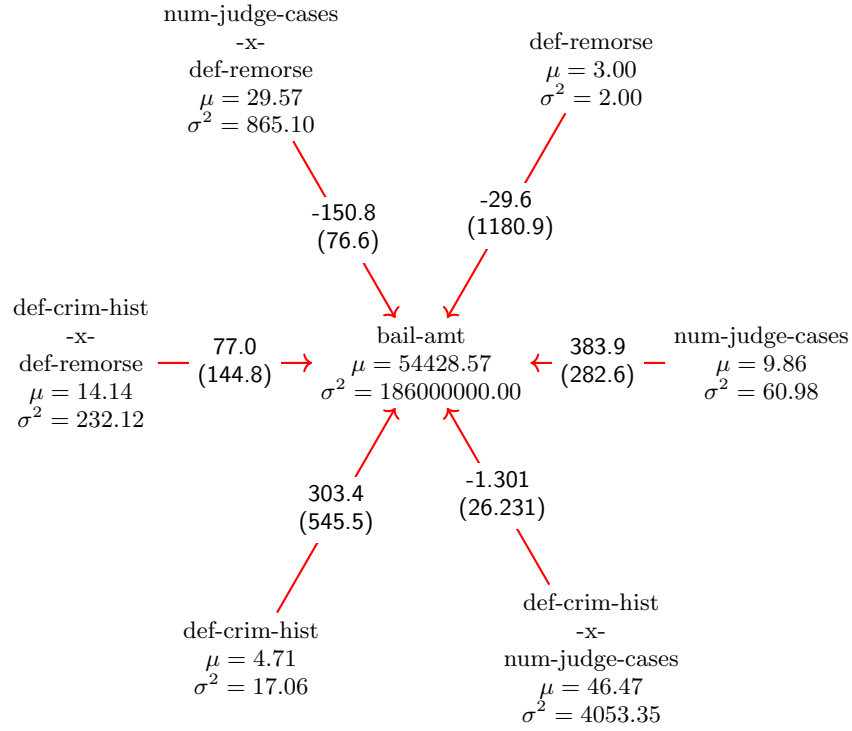
A Additional figures and tables

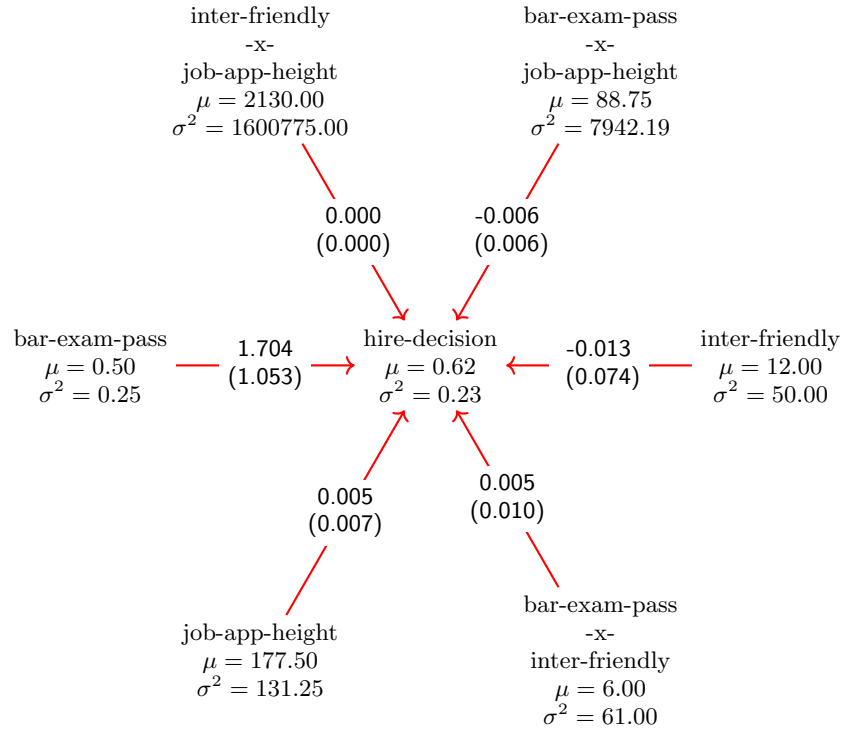Figure A.1: Fitted SCM for "two people bargaining over a mug."
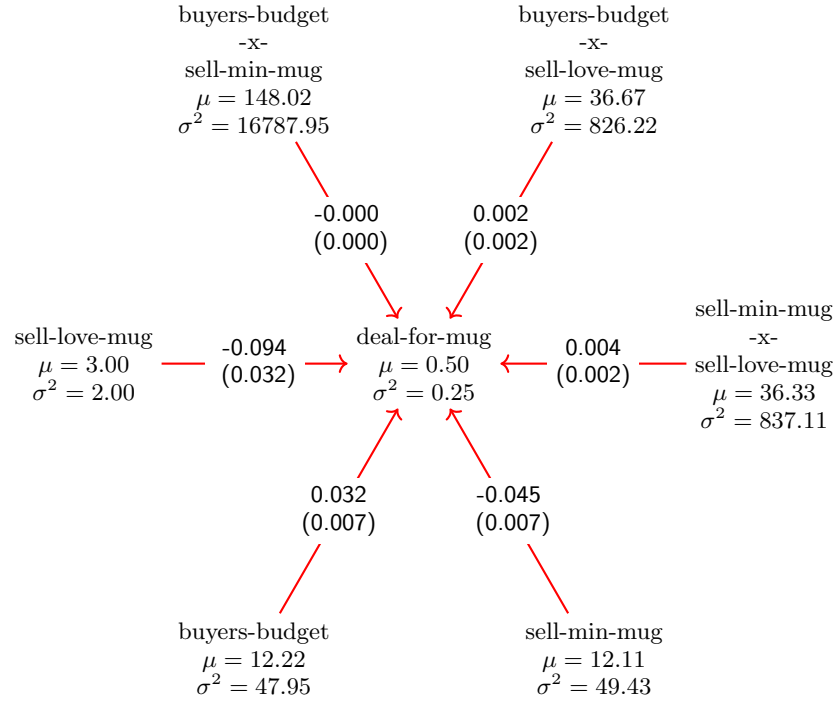
Notes: Each variable is given with its mean and variance. The edges are labeled with their unstandardized path estimate and standard error. There were 405 simulations with these agents: ['buyer', 'seller'].

Figure A.2: Fitted SCM for "a judge is setting bail for a criminal defendant who committed 50,000 dollars in tax fraud."

num-judge-cases
-x-
def-remorse
$\mu = 29.57$
$\sigma^2 = 865.10$

def-remorse
$\mu = 3.00$
$\sigma^2 = 2.00$

-150.8
(76.6)

-29.6
(1180.9)

def-crim-hist
-x-
def-remorse
$\mu = 14.14$
$\sigma^2 = 232.12$

77.0
(144.8)

bail-amt
$\mu = 54428.57$
$\sigma^2 = 186000000.00$

383.9
(282.6)

num-judge-cases
$\mu = 9.86$
$\sigma^2 = 60.98$

303.4
(545.5)

-1.301
(26.231)

def-crim-hist
$\mu = 4.71$
$\sigma^2 = 17.06$

def-crim-hist
-x-
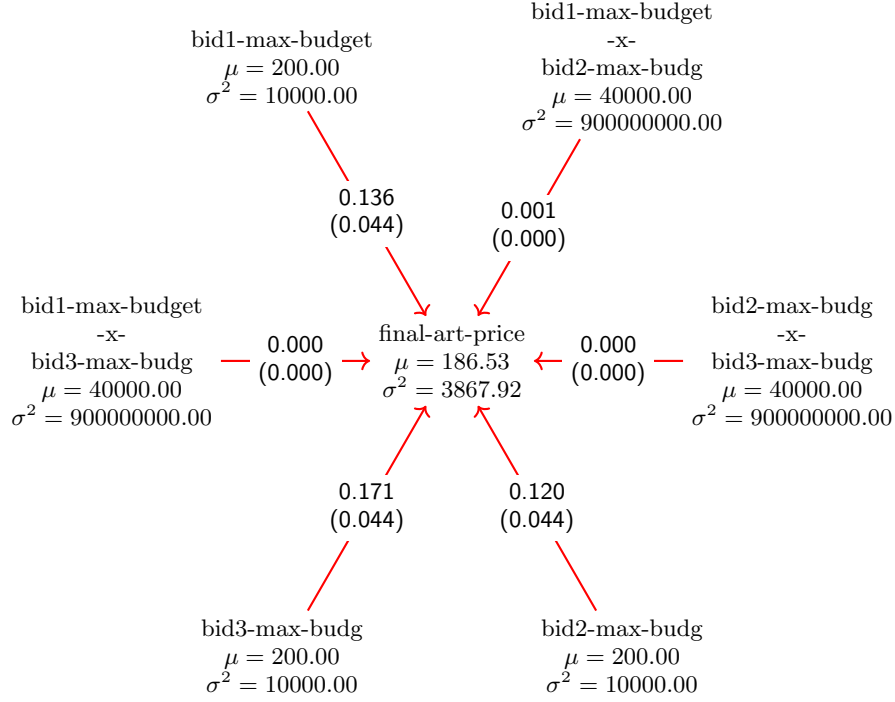num-judge-cases
$\mu = 46.47$
$\sigma^2 = 4053.35$

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.3: Fitted SCM 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 un-standardized path estimate and standard error. There were 80 simulations with these agents: ['job applicant', 'employer'].*

Figure A.4: Fitted SCM 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'].

Figure A.5: Comparison of the LLM's predictions to the theoretical predictions and all experimental results for the auction scenario.
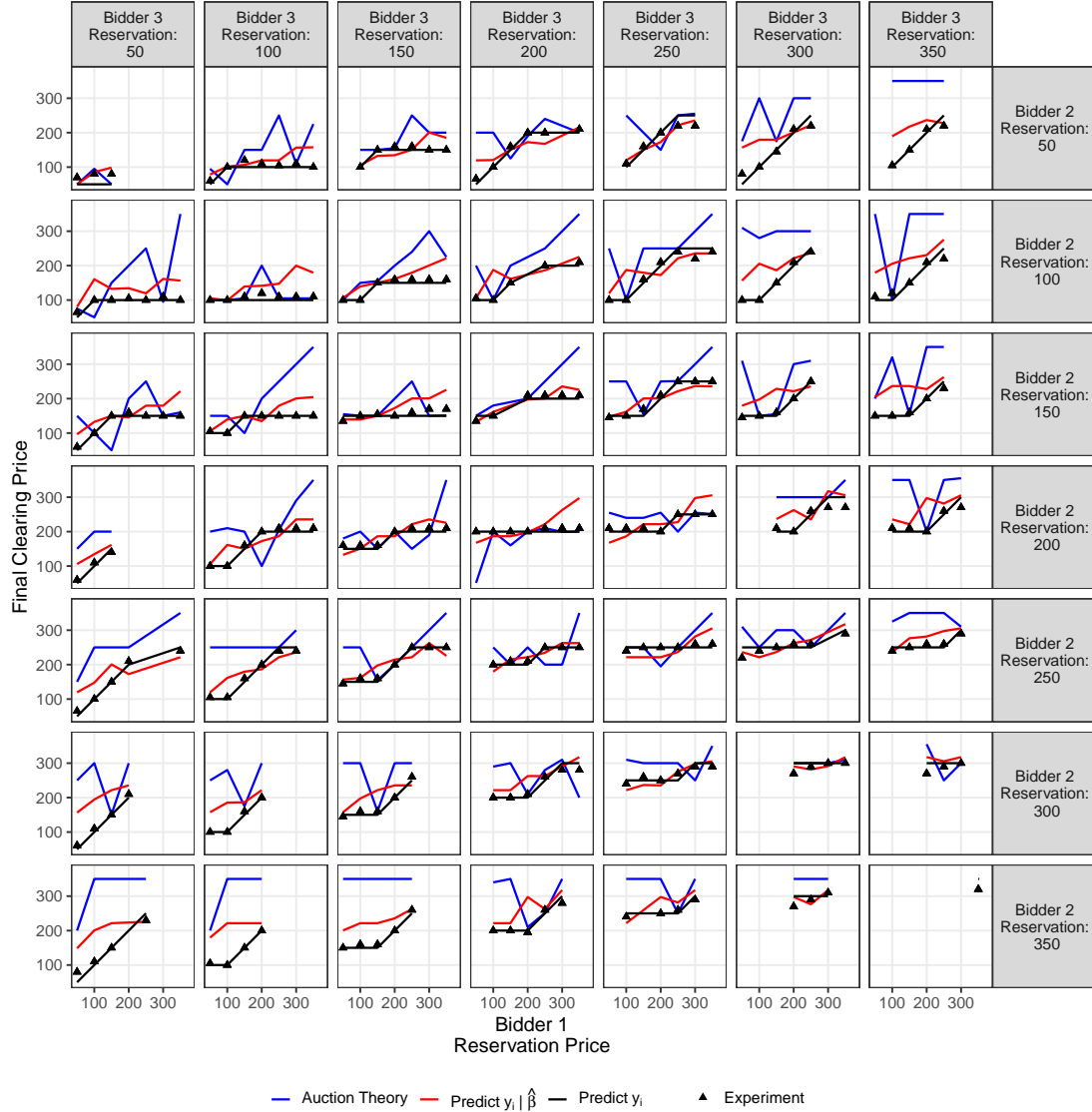
Notes: The columns correspond to the different reservation values for bidder 3 in a given simulation, and the rows correspond to the different reservation values for bidder 2. The y-axis is the clearing price, and the x-axis lists bidder 1's reservation price. The black triangles track the observed clearing 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.6: Prompt used to elicity 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 Treatment Values" 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 3 and Table 1.

Table A.1: 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 Treatment Values	-	{$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 1a.

B   Additional features of the SCM-based approach

B.1   LLM alignment and safety

One way to view our system is that it allows an LLM to "imagine" hypothetical situations before they happen. This is similar to how humans simulate different versions of an event in their mind, a mental dress rehearsal, to improve their understanding of a situation without experiencing it. For example, when an employee wants to ask their boss for a raise, they may imagine the conversation and possible counterfactual repetitions to prepare for the real thing. Our system does this hypothetical counterfactual simulation with more control on a much larger scale with complete independence between the simulations. It lets an LLM acquire social scientific knowledge autonomously.

This suggests a way to transfer the relationships from the black box LLM into human-interpretable hypotheses that can be explicitly tested. We can imagine using this sort of automated and iterative hypothesis testing as a "top-down" approach to exploring the behavior of any LLM (Binz and Schulz, 2023b). Top-down exploration could allow researchers to quickly identify when an LLM's behavior deviates from "what a human would do" (or any other measure of behavior) in a given situation. Then, this information can be used better to align the LLM with a given set of objectives. A large portion of the LLM evaluation process is still done by humans (OpenAI, 2023). While a human should always be in the loop, efficiency can be gained with an easily interpretable and automated approach.

B.2   Interpreting hypotheses from data

As noted in Section 1, a recent and exciting trend in the social sciences, specifically in economics (e.g., lotteries and bail decisions), is the use of machine learning to generate novel hypotheses (Enke and Shubatt, 2023; Ludwig and Mullainathan, 2023; Peterson et al., 2021). The approach to generate these hypotheses can be broadly summarized as follows.

First, a very large data set is acquired with a clear outcome and possible ex-

planatory variables. At least one of these variables is "unstructured," in the sense that it does not fit neatly into predefined data models or is not easily quantifiable. This could include text, images, audio, etc. Then, a black-box deep neural network is trained to predict the outcome from the explanatory variables with the highest possible accuracy.

Next, an economic model of interest (e.g., expected utility theory) is used to predict the outcome on the same data set. The model's predictions are compared to those made by the deep neural network. Invariably, the neural network is far better at predicting the outcome than the economic model, even on a holdout test data set.[37] This difference in predictive power is generally not surprising—the unstructured explanatory variables (the images, text, etc.) often contain a lot of latent information that the economic model does not capture.[38] However, due to the black-box nature of the neural network, it is unclear which relationships in the data it has identified to comparatively predict the outcome so well.

The identification of these hidden relationships and subsequent transformation into human-interpretable features is the generation of novel hypotheses. Unfortunately, this transformation is generally non-obvious, time-consuming, and expensive. Methods to transform the hidden relationships into human-interpretable features include building new complex machine-learning models, running multiple experiments or surveys on human subjects, hand-coding variables of interest, and a combination of all three. None of these are guaranteed to be successful. This is not to say that the process is not valuable, but it has its practical limitations.

In contrast, hypotheses generated as SCMs are always easy to interpret. They are directed graphs with variables labeled in natural language. All that is needed to generate a new hypothesis is a proposed causal path between two variables—one of the main purposes of the system presented in this paper.

One way to view the system is as a tool for transforming information from an

---

[37]The fraction of an economic model's maximum possible predictable variation can account for is the model's "completeness" (Fudenberg et al., 2022). In this case, the ratio of the predictive power of the economic model to the predictive power of the machine learning model. When a model is complete, this ratio is ≈ 1 because all possible predictable variation is accounted for.

[38]Formal economic models generally do not incorporate unstructured data in their predictions.

LLM (a large black-box neural network) into an interpretable SCM—similar to the methods discussed above. But with the SCM-based approach, this process is automated, inexpensive, fast, and interpretable.