NBER WORKING PAPER SERIES

LARGE LANGUAGE MODELS:
AN APPLIED ECONOMETRIC FRAMEWORK

Jens Ludwig
Sendhil Mullainathan
Ashesh Rambachan

Working Paper 33344
http://www.nber.org/papers/w33344

NATIONAL BUREAU OF ECONOMIC RESEARCH
1050 Massachusetts Avenue
Cambridge, MA 02138
January 2025

This paper was supported by the Center for Applied Artificial Intelligence at the University of Chicago and the Altman Family Fund at MIT. We especially thank Haya Alsharif, Suproteem Sarkar, and Janani Sekar for excellent research assistance. We thank Peter Bergman, Peter Chang, Jimmy Lin, Larry Katz, Steven Ma, Marina Mancoridis, Lindsey Raymond, Andrei Shleifer, Cassidy Shubatt and David Yanagizawa-Drott as well as audiences at UIUC, UT Austin, and the University of Chicago "AI in Social Science" Conference for comments. Wharton Research Data Services (WRDS) was used in preparing this paper. This service and the data available thereon constitute valuable intellectual property and trade secrets of WRDS and/or its third-party suppliers. All interpretations and any errors are our own. The views expressed herein are those of the authors and do not necessarily reflect the views of the National Bureau of Economic Research.

Large Language Models: An Applied Econometric Framework
Jens Ludwig, Sendhil Mullainathan, and Ashesh Rambachan
NBER Working Paper No. 33344
January 2025
JEL No. C01, C45

ABSTRACT

How can we use the novel capacities of large language models (LLMs) in empirical research? And how can we do so while accounting for their limitations, which are themselves only poorly understood?We develop an econometric framework to answer this question that distinguishes between two types of empirical tasks. Using LLMs for prediction problems (including hypothesis generation) is valid under one condition: no "leakage" between the LLM's training dataset and the researcher's sample. No leakage can be ensured by using open-source LLMs with documented training data and published weights. Using LLM outputs for estimation problems to automate the measurement of some economic concept (expressed either by some text or from human subjects) requires the researcher to collect at least some validation data: without such data, the errors of the LLM's automation cannot be assessed and accounted for. As long as these steps are taken, LLM outputs can be used in empirical research with the familiar econometric guarantees we desire. Using two illustrative applications to finance and political economy, we find that these requirements are stringent; when they are violated, the limitations of LLMs now result in unreliable empirical estimates. Our results suggest the excitement around the empirical uses of LLMs is warranted – they allow researchers to effectively use even small amounts of language data for both prediction and estimation – but only with these safeguards in place.

Jens Ludwig
Harris School of Public Policy
University of Chicago
1307 East 60th Street
Chicago, IL 60637
and NBER
jludwig@uchicago.edu

Sendhil Mullainathan
MIT Department of Electrical Engineering
and Computer Science
and Department of Economics
77 Massachusetts Ave.
Room 38-401
Cambridge, MA 02139
and NBER
sendhil.mullainathan@gmail.com

Ashesh Rambachan
Department of Economics
MIT
50 Memorial Drive
Cambridge, MA 02142
asheshr@mit.edu

1 Introduction

Large language models (LLMs) are puzzling. They accomplish astonishing feats, but also exhibit surprising failures that are poorly understood. How can we use the novel capabilities of LLMs in empirical research while not taking their outputs at face value? In this paper, we analyze how to exploit these powerful yet unpredictable tools without compromising the rigor of our empirical research.[1]

Consider how we incorporate a more familiar algorithm into our research pipeline: ordinary least squares (OLS). At its core, OLS is nothing more than an algorithm that, when applied to a dataset, returns the coefficients $\widehat{\beta}$ that minimize the sum of squared residuals. Econometrics clarifies what assumptions the researcher must believe about the underlying data generating process (DGP) to justify different interpretations of the algorithm's output. If we want to interpret $\widehat{\beta}$ as the best linear unbiased estimator of the conditional expectation function, the Gauss-Markov theorem tells us what we need to be willing to assume (and defend) about the DGP. If we want to interpret $\widehat{\beta}$ causally, we need to be willing to defend the conditional independence assumption. In this sense, econometrics provides empirical researchers with the terms of a *contract*: to endow an estimate with a particular interpretation, these are the assumptions you need to defend.

Appropriately using LLMs requires exactly the same sort of contract: what specific assumptions about LLMs must the researcher defend in order to draw what specific inference? Yet at present no such contracts exist because producing one is challenging. Part of the problem is that precisely modeling the inner workings of LLMs is difficult. LLMs are a diverse, dynamic set of extraordinarily complex machine learning models involving many layers of interactivity and billions of parameters. Their training datasets and architectures (among many other details) are often intentionally obscured because LLMs are proprietary commercial products.[2] To further complicate matters, modeling the outputs of LLMs has proven to be equally difficult. Computer scientists struggle to characterize the brittleness of LLMs that lead their outputs to accomplish remarkable feats in some tasks yet produce bizarre failures in others.

We develop an econometric framework that provides contracts for LLMs despite the complexity of these models and without ex-ante assumptions about the quality of their outputs. Our framework considers settings in which a researcher uses an LLM to process text and produce outputs that can be related to traditional economic variables in downstream analyses. Precisely

[1] For a discussion of how LLMs can be used for an adjacent set of applications—as computational tools like others that occupy our workflow, such as search engines, and coding copilots, see Korinek (2023, 2024). See also Dell (2024). Our focus here is instead on the uses of LLMs not in day-to-day workflow but directly in empirical analysis. For discussion of the application of machine learning to economics more generally see Varian (2014); Mullainathan and Spiess (2017); Athey (2018); Gentzkow, Kelly and Taddy (2019).

[2] As an example, the GPT-4o technical report writes "this report contains no further details about the architecture (including model size), hardware, training compute, dataset construction, training method, or similar" (OpenAI, 2023). Different families of LLMs make different choices along each of these dimension, and the state-of-the-art is ever-evolving as new LLMs are released constantly.

because these algorithms are opaque and varied, we treat everything about how the LLM goes from its training data to a particular text response as a black box. The framework distinguishes between two types of empirical applications — prediction and estimation — and provides four main results.

Our first result relates to *prediction problems*, in which the researcher predicts a linked economic variable using the associated text. For example, in asset pricing, we might want to know how well stock prices can be predicted using news headlines. We show that for LLM outputs to be validly used in prediction problems, the LLM must satisfy a single condition we refer to as "*no training leakage*." Suppose the researcher prompts the LLM on each collected piece of text to form predictions, and evaluates the quality of the LLM's resulting predictions by calculating its sample average loss on the researcher's dataset. We show that the LLM's sample average loss only reflects its true out-of-sample predictive performance if and only if there is no overlap between the text in the LLM's training dataset and the researcher's own dataset. Intuitively, in this workflow the researcher is using the LLM as-if it were some sort of prediction function from text to the economic variable, and is then using their own collected dataset as-if it were a test sample. This is only valid if the LLM has not been trained on the test sample.

No training leakage is often violated in naive uses of LLMs precisely because their training datasets are both immense and intentionally obscured. This is a well-known challenge in computer science, where researchers worry that LLMs have been trained on the benchmark datasets commonly used to evaluate LLM performance within that field. We present new empirical results showing that the "no training leakage" condition is also violated in settings relevant to economists, through two applications to finance and political economy. We show that GPT-4o has likely been trained on economic datasets since it appears to have exactly memorized the text of many financial news headlines and Congressional legislation.[3]

Nonetheless prediction is a very promising use case. No training leakage can credibly hold in many empirical applications as a result of the choice of LLM. Specifically, researchers must use open-source LLMs that clearly document their underlying training data and/or provide a clear time-stamp beyond which the LLM has not been updated, such as the Llama family of models (Touvron et al., 2023; Dubey et al., 2024) or the StoriesLM family (Sarkar, 2024). Having made such a choice, the LLM can provide a tremendous advantage for prediction tasks using text. Predicting economic variables from text requires modeling the complex structure and semantics of language, which requires vast amounts of training data. Since open-source LLMs have already been pre-trained on large corpuses to learn this structure, they can serve as foundations upon which researchers can form accurate predictions from even fairly small datasets.

[3] In recent work, Sarkar and Vafa (2024) show LLMs used to predict a future outcome from some text can inadvertently draw on information from the time of the outcome rather than the time of the text. The authors refer to this as "look-ahead bias," which violates our "no training leakage" condition. See also Glasserman and Lin (2023).

Our second result relates to *estimation problems*, in which the researcher wants to relate some economic concept (e.g., positive or negative news, hawkish attitudes, or policy topic) expressed in text in order to estimate some downstream economic parameter. For example, in studying partisanship, how does the policy topic of a Congressional bill relate to the ideology of its sponsor? We assume the researcher has specified a resource-intensive procedure for measuring the economic concept that we would be satisfied with if it could be scaled. Because it may be prohibitively costly or time-consuming, the researcher hopes to automate their existing measurement procedure and instead use an LLM to cheaply produce these labels. At the same time, ample research in computer science documents the brittleness of LLM outputs across a range of tasks. How can we use LLM outputs in estimation problems knowing they are potentially imperfect substitutes for the measurements they seek to replace?

We highlight a constructive solution to this problem by borrowing an idea long known in economics: collect a small sample of benchmark data and use that data to empirically model the LLM's errors. While this is well-known in labor economics (e.g., Bound and Krueger, 1991; Bound et al., 1994; Bound, Brown and Mathiowetz, 2001) and well-studied in econometrics (e.g., Lee and Sepanski, 1995; Chen, Hong and Tamer, 2005; Schennach, 2016), it has only recently been revived in machine learning, such as Wang, McCormick and Leek (2020), Angelopoulos et al. (2023), Wei and Malik (2023), Egami et al. (2024), and Battaglia et al. (2024) among others. We illustrate the value of benchmark data in the context of linear regression, using the LLM's output as either a covariate or dependent variable. Through asymptotic arguments, we show that de-biasing LLM outputs using a benchmark sample preserves our usual econometric guarantees of consistency and asymptotic normality, while still drastically reducing the costs of measuring the economic concept of interest.

We further show that incorporating the imperfect LLM outputs in addition to the benchmark data can improve the precision of the downstream estimates in many cases. This gain may initially seem surprising. The LLM's role here is something like a missing-value imputer, and the standard approach for that in economics is for the researcher to use their own data to do some version of linear imputation. So where would the extra information be coming from that gets us some boost in precision? The answer is that the new information the LLM brings to bear comes from the large outside corpus of text the model has been pre-trained on; if the LLM is sufficiently accurate in a given application, it can amplify the validation data the researcher has incurred the cost of collecting. We empirically demonstrate this performance in finite samples with Monte Carlo simulations using data on Congressional legislation.

In this light, estimation is also a very promising use case for LLMs, provided researchers define the measurements they seek to automate with the LLM and collect benchmark data to empirically model the LLM's potential errors. Indeed, this is a small modification to existing practice as researchers often already collect such benchmark data in many empirical applications involving LLMs (e.g., Durvasula, Eyuboglu and Ritzwoller, 2024; Hansen and Kazinnik, 2024). By appropriately leveraging this safeguard, LLMs can offer researchers a free lunch: lowering the cost of data collection and in some cases improving statistical precision, while preserving the familiar econometric guarantees we desire in estimation problems.

Notice, however, that this productive use of LLMs in estimation problems requires the researcher to collect benchmark data using some existing measurement procedure. What if such benchmark data *cannot* be collected? Our third result establishes that, absent benchmark data, plug-in estimation using the LLM's outputs recovers the researcher's target estimate if and only if there is no error in the LLM's outputs; in other words, the LLM must reproduce the economic concept of interest.

Consequently, in the absence of benchmark data, a researcher using LLM outputs could directly argue that there are no errors in the LLM outputs; that is, in the earlier example, argue that the chosen LLM and prompt engineering strategy correctly labels the policy topic of all Congressional bills. However, even the most generous measures of LLM performance on benchmark tasks in computer science do not find perfect performance. Furthermore, computer science research documents the brittleness of LLMs — what seem like small modifications to a task (choice of LLM or specific prompt, for example) lead to substantial changes in LLM performance. We present new empirical results showing the implications of this brittleness for downstream estimation of economic parameters. In our applications to finance and political economy, we show that variability in the LLM's outputs across models and prompts leads to large changes in the magnitude, statistical significance and even sign of downstream regression coefficients.

Since it seems difficult to argue LLM outputs have no errors at this point, a researcher using LLM outputs without benchmark data faces two choices. First, the researcher might write down a statistical model of the LLM's errors, just as we would in the measurement error literature in econometrics. While tempting in its familiarity, this exercise seems heroic when applied to LLMs. We would be asserting that we can correctly model how the outputs of an algorithm we do not understand relate to a quantity that we do not observe. Second, the researcher could argue there is no existing measurement that we are trying to automate; instead the LLM's outputs are themselves the quantity of interest, not a proxy for some other quantity. While correct by assumption, it is difficult to think of meaningful applications where this would be satisfying. For example, would we be comfortable revisiting all published work using GPT-4o whenever GPT-5 happens to be released?

At this point, in the absence of benchmark data, each of these available choices seems difficult to defend in empirical applications. By contrast, if benchmark data can be collected, our results imply that LLMs are already powerful tools for automating existing measurements that are costly to scale across large collections of text.

Our final result is to show that this framework is general enough to cover even some of the more novel research use cases of LLMs. For example, we argue that the use of LLMs for hypothesis generation has a similar structure to prediction problems. That means our framework can clarify what assumptions are required for the valid use of LLMs for this purpose—no training leakage. As another example, LLMs are increasingly being used to simulate human subject responses (to surveys, lab experiments, etc.). We argue that task has the same structure as estimation problems, and hence highlights the vital importance of collecting validation data (e.g., running experiments, collecting surveys, etc. from at least *some* real subjects).

None of these results required articulating how exactly the LLM goes from its training data to its output. Black-boxing new AI tools in this manner might initially seem unsatisfying, if not unnerving. Traditionally econometrics provided contracts for empirical tools that we designed ourselves. By contrast, economists increasingly sit on the *consumer* side of new AI models, and we are blind to most of what happens in their production. The consumer side can be a frustrating and confusing place to be, given that it is occupied by both impassioned enthusiasts and ardent skeptics. Both are to a degree right. AI tools can be productively used in empirical research but only if they are housed inside an econometric framework that does not require unrealistic assumptions in either direction. This paper provides one such framework, resulting in meaningful contracts for using LLMs in empirical research. Provided those contracts are adhered to, our results suggest that LLMs can expand the scope of empirical research without compromising its rigor.

2 Evaluating General-Purpose Technologies: the LLM Conundrum

Why might we be willing to directly plug LLM outputs into our empirical studies? For two reasons. The first relates to the impressive performance of LLMs on benchmark evaluations, combined with memorable examples of their capabilities. The second is our tendency to draw inferences and generalize from how LLMs performed on some tasks to how they are likely to perform on other tasks. After all, if we saw a human ace the math GRE, we would naturally assume this person would perform well on all manner of other math problems as well. We tend to make the same assumption about LLMs.

Yet both rationales turn out to be problematic. The domains on which LLMs perform well turn out to be selective, especially when compared to the grand ambition to have LLMs serve as general-purpose technologies. And the way that the capabilities of LLMs generalize across tasks turns out to be quite different than what we intuitively assume.

2.1 Benchmark Evaluations and Anthropomorphic Generalization

The central difference between LLMs and supervised learning algorithms is that LLMs aspire to serve as useful tools not just for a single task, but *all* tasks — to be a general-purpose technology (GPT). The scope of that ambition is both the most remarkable feature of LLMs and also their Achilles' heel.

Because supervised learning algorithms (e.g., convolutional networks) aim to tackle a single task (e.g., image classification), it is straightforward to evaluate any algorithm's prediction — and therefore to know how well an algorithm works and which alternative design works better. This was operationalized by the machine learning community through the creation of open public prediction tasks that competing algorithms could be deployed on to assess performance, the so-called *common task framework* (Donoho, 2024). The resulting competitions (e.g., the Netflix prize, ImageNet, etc.) helped spur innovation and progress.

But there is an obvious challenge in extending the common task framework to any new tool (like LLMs) that aspires to serve as a GPT: how does one rank-order the performance of competing LLMs when they aim to be useful on *any* task? How do we develop open common-task competitions to evaluate LLMs when there is no common task?

The current solution is to try building bigger and more diverse benchmark evaluations. For example, the "Beyond-the-Imitation-Game benchmark" (BIG-bench) collects problems on 204 tasks ranging from math and analogical reasoning questions to reading comprehension and social reasoning problems (Srivastava et al., 2022). The "Massive Multitask Language Understanding" (MMLU) benchmarks collect questions across 57 different scientific and humanistic disciplines (Hendrycks et al., 2020). It is also common to evaluate LLMs on standardized exams that aim to test general-purpose knowledge in people, such as the SAT, the GRE, and AP exams. If the collection of questions across these benchmark evaluations were actually what users ultimately wanted to deploy LLMs on, then comparing performance across language models could be reduced to constructing task performance metrics on these benchmarks.

However, we do not *intrinsically* care about these benchmark tasks — after all, relatively few researchers will ever deploy an LLM to take the SAT or answer abstract reasoning puzzles. Rather, we use these benchmark evaluations to generalize about the capabilities of LLMs on *new* tasks. This reflects a deeper bias we have when we engage with LLMs: we tend to engage in *anthropomorphic generalization*. Since it is difficult to imagine a person who could accomplish these feats yet fail on related (or even simpler) tasks, we tend to generalize the performance of an LLM onto new tasks the same way we would generalize human performance across tasks. Surely a large language model that can write a proof that there are infinitely many primes in the form of a poem (Bubeck et al., 2023) must be able to label text or answer surveys in economics research.

Evidence of anthropomorphic generalization comes from Vafa, Rambachan and Mullainathan (2024), who present online subjects with pairs of questions and ask subjects to predict whether a human (or algorithm) would get the second question correct depending on how they did with the first question. The authors refer to a person's ability to accomplish this task as the "human generalization function." While the human generalization function predicts the performance of other humans well, it is misaligned with the performance of LLMs — these models often get questions wrong that we expect them to get right based on our observations of their performance on previous tasks. Dreyfuss and Raux (2024) document a similar phenomenon, showing that users extrapolate the performance of LLMs based on an intuitive notion of task difficulty and explore implications for LLM usage.[4]

This heuristic of anthropomorphic generalization is surely one important reason we are willing to directly plug LLM outputs into our empirical research.

2.2 Impressive Feats, But Also Puzzling Examples

This anthropomorphic generalization of the LLM's capacity is understandable (given their seemingly human interface) but problematic. Recent work in computer science shows the capabilities of LLMs are brittle: For every example of impressive performance, there seems to be a counterexample that leaves users scratching their heads.

The type of brittleness that has attracted the most public attention is the periodic tendency of LLMs to *hallucinate*—to report back plausible-sounding "facts" that are not actually facts. For example, an LLM asked to provide proof that dinosaurs created a civilization will reply that there is fossil evidence of dinosaur tools, and even credit dinosaurs with the invention of primitive art forms like stone engravings (Szempruch, 2023). This tendency to hallucinate is not just limited to whimsical examples, as was discovered by the lawyer who used an LLM to prepare a legal brief, only to discover it cited a number of "cases" that are not real cases (Bohannon, 2023). These hallucinations are not rare events; one study found that LLMs hallucinate in 58% of legal applications (Dahl et al., 2024). While some may hope that there exists some future technological solution to hallucination, recent work provides statistical and computational arguments suggesting this may be inevitable — an intrinsic and unavoidable feature of the language generation problem (e.g., Kalai and Vempala, 2024; Xu, Jain and Kankanhalli, 2024).

While hallucinations are well-known, economists and other users who are impressed by some LLM's performance on the math SAT or math Olympiad might be surprised to see examples of how poorly these same LLMs perform on other math problems. For example, the same LLM that can reliably solve $(9/5)x+32$ cannot solve $(7/5)x+31$; an LLM that can reliably implement common ciphers such as shift by 13 cannot implement other variations (McCoy et al., 2024).

The performance of LLMs turns out to be remarkably sensitive to seemingly minor details. An LLM that can correctly answer a given multiple choice question will often answer the same question incorrectly after the order of the answers has been permuted (Zong et al., 2024). LLMs struggle on "counterfactual" versions of tasks — for example, being able to program a list sort in 0-based

[4] These findings are also backed up by experimental evidence from giving large language models to managers and consultants to solve different tasks, alternating between tasks on which the models do well and other tasks of *seemingly*-similar difficulty levels that the models handle poorly—what the authors call AI's "jagged technological frontier" (Dell'Acqua et al., 2023).

indexing but unable to perform the same task in 1-based indexing (Wu et al., 2024). Similar sensitivity has been documented on logical reasoning tasks (Lewis and Mitchell, 2024). Their behavior can be substantially influenced by appending an adversarial string to an existing question (Zou et al., 2023). The same GPT-4 model that demonstrates impressive spatial awareness in describing how to carefully stack a book, a laptop, nine eggs, a bottle and a nail does very poorly for stacking a pudding, a marshmallow, a toothpick and a glass of water (Mitchell, 2023).

Despite their many amazing feats and human-like interface, LLMs do not reason the way humans reason. An LLM trained on "A is B" will not know "B is A." For example, an LLM trained on the fact that "Tom Cruise's mother is Mary Lee Pfeiffer" cannot answer "Who is Mary Lee Pfeiffer's son?" (Berglund et al., 2023). LLMs struggle on variations of the question: "Alice has N brothers and she also has M sisters. How many sisters does Alice's brother have?" (Nezhurina et al., 2024). When asked how to ferry a single farmer and a single sheep from one side of a river to the other using a boat with enough room for one person and one animal, the LLM claims this requires at least three trips (e.g., Fraser, 2024*a*,*b*).

This accumulating body of evidence sets up the problem statement that we seek to address. Since LLMs are impressive, easy-to-use yet potentially brittle tools, under what conditions can researchers safely plug-in their outputs into empirical research?

3 An Empirical Framework for LLMs

In this section, we develop a framework that helps clarify what properties an LLM must satisfy in order for us to safely incorporate it into empirical research. While social scientists in practice tend to treat LLMs as an all-purpose black box, in reality the model's performance will hinge critically—as our framework demonstrates—on what text the algorithm is trained on, to what text we seek to apply it, and for what purpose.

3.1 Setting and the Researcher's Dataset

Let Σ^* denote the collection of strings (up to some finite length) in an alphabet with elements $\sigma \in \Sigma^*$. A *training dataset* is any collection of strings. We summarize a training dataset by the vector t, whose elements t_σ are sampling indicators for whether a particular string is collected in the training dataset.

Of course, for any given empirical question, not all strings are relevant; we denote those that are as $\mathcal{R} \subseteq \Sigma^*$ with elements $r \in \mathcal{R}$ that we refer to as text pieces. The *researcher's dataset* is also summarized by the vector d, whose elements d_σ are similarly sampling indicators for whether the researcher collected a particular string. The researcher only collects economically relevant text pieces, and so $d_\sigma = 0$ for all strings $\sigma \in \Sigma^* \setminus \mathcal{R}$.

Each text piece r is (or can be) linked to observable economic variables (Y_r, W_r), which can be thought of as economic outcomes that might be influenced by the text Y_r or candidate economic determinants that might influence the text W_r. Altogether the researcher observes (r, Y_r, W_r) for each collected text piece with $d_r = 1$. To make this more concrete, consider two illustrative empirical applications that we return to throughout the paper.

Example: Congressional legislation Consider descriptions of bills introduced in the United States Congress. Each text piece $r \in \mathcal{R}$ refers to a bill's brief description such as "`A bill to revise the boundary of Crater Lake National Park in the State of Oregon`." The economically relevant outcome Y_r might be whether the associated bill passed its originating chamber of Congress. The candidate economic determinant W_r of the bill's text might be the party affiliation or roll-call voting score of the bill's sponsor. ▲

Example: Financial news headlines Consider financial news headlines about publicly traded companies. Each text piece $r \in \mathcal{R}$ refers to a particular financial news headline such as "`Bank of New York Mellon Q1 EPS $0.94 Misses $0.97 Estimate, Sales $3.9B Misses $4.01B Estimate`." The economic outcome Y_r might be the company's realized return in some event window after the headline's publication date, while the candidate determinant W_r of the news headline itself could be the company's past fundamentals. ▲

Importantly, each text piece r expresses some economic concept for which it is costly in terms of either time or money to obtain measurements. A key assumption in much empirical research (often left implicit) is that there is *some* existing procedure that could in principle be applied to each text piece to measure the economic concept; that is, the economic concept is defined as the output of some mapping $f^*(r)$ for all text pieces $r \in \mathcal{R}$. If this measurement procedure $f^*(\cdot)$ could be scaled, the researcher would be satisfied to use the resulting measurements in downstream analyses.

For example, in some applications a researcher is willing to argue that a sufficiently reliable measure of the economic concept could be derived by having a single expert spend the time and effort needed to read each text piece carefully.[5] In other applications, the researcher may recruit teams of trained human labelers to read each text piece (e.g., Baker, Bloom and Davis, 2016). Further motivated by findings in psychology (e.g., Biemer et al., 2013; Kahneman, Sibony and Sunstein, 2021), the researcher may worry that human annotations of some text are themselves noisy measures for the economic concept, but is willing to accept an average (or majority vote) from some minimum number of labelers as a reliable measure.

Of course, the measurement procedure will vary in nature and difficulty across empirical applications; our assumption is only that there exists *some* procedure the researcher would be

[5] For example, Ash and Hansen (2023) write, "The most accurate approach to concept detection is perhaps direct human reading with appropriate domain expertise" (pg. 672). Hansen et al. (2023) write, "The most precise way of classifying [text pieces] is arguably via direct human reading" (pg. 6).

satisfied to use. As shorthand, we will write the economic concept as $V_r := f^*(r)$.

No matter how the existing measurement procedure is defined, we face a *text processing problem*: measuring the economic concept requires processing each text piece r, and it may be prohibitively costly or time-consuming to do so in many applications. Absent a solution to this text processing problem, the researcher cannot collect V_r on all text pieces.

In settings like this, researchers would like to process the collected text pieces to tackle one of two types of economic analyses. The first is a *prediction problem* — predict the linked variable Y_r using the associated text piece r. For example, we might use the short text description of some Congressional legislation r to predict whether the bill passed its originating chamber of Congress Y_r. Or a researcher might try to use each financial news headline r to predict the company's realized return Y_r in some event window after the headline's publication date.

The second is an *estimation problem* — estimate some parameter that relates the economic concept V_r to the linked variables (Y_r, W_r). For example, if each Congressional bill's brief text description r expresses the policy topic V_r of the bill, such as whether it is related to defense, foreign affairs, or health, that topic could be related in a regression to the party affiliation or ideological voting score of the bill's sponsor W_r. Or if each financial news headline r expresses positive or negative news about some company's future V_r, that could be the outcome in a regression against past fundamentals W_r or the explanatory variable in a regression against future returns.

This is where LLMs come in. LLMs are general-purpose and easy-to-use models for processing text, so researchers would like to use them to tackle their text processing challenge in service of some prediction or estimation problem. When is that justified? To clarify those conditions, we next introduce LLMs into our framework.

3.2 Large Language Models

To capture how we typically interact with LLMs as black boxes — to generate responses given prompts without knowing or worrying much about their design or the contents of the training datasets — we define a *large language model* as any mapping from possible training datasets t to mappings between strings, where $\widehat{m}(\cdot;t): \Sigma^* \to \Sigma^*$ is its text generator when it is trained on dataset t and $\widehat{m}(\sigma;t)$ is the LLM's response when prompted by string σ.

While this definition is quite general, notice it has a specific implication: The LLM "algorithm" is actually *two* algorithms: a *training algorithm* that takes in any training dataset t and learns the mapping between strings; and what we call the *text generator*, $\widehat{m}(\cdot;t)$, which is the output of the training algorithm and is what the user interacts with to obtain responses to any given prompt. We return below to the importance of this implication.

The other important feature of our definition is that our analysis will not depend on how exactly an LLM's training algorithm and text generator accomplish their functionalities. This is intentional. We would like to understand the conditions under which we can use any given LLM to solve prediction and estimation problems. By analyzing them at this level of abstraction, we provide interpretable conditions on LLMs needed for empirical researchers to accomplish their objectives.

Furthermore, since the state of the art in natural language processing is constantly evolving, we should expect the exact implementation of training algorithms and text generators to change; studying LLMs at this level of abstraction ensures the durability of our analysis. In our framework, alternative LLMs, such as GPT-3.5-Turbo and GPT-4o versus Llama-3-8B and Llama-3-70B (or alternative snapshots of the same model) may all differ in their training algorithms and text generators, but researchers will now have a clear set of consistent conditions under which the output of any given model can (or cannot) be incorporated into empirical research.

3.2.1 Interpreting the Text Generator and the Training Algorithm

Our framework, even as general as it is, nonetheless captures all of the key design choices underlying LLMs. Some of those choices are made by the researcher themselves, while others are made by the algorithm builder — many of which may even be invisible to the researcher.

The researcher interacts entirely with the text generator $\widehat{m}(\cdot;t)$, which winds up capturing all choices that influence how an LLM generates responses from any prompt once it has been trained. For example, research in computer science has found that alternative prompt engineering strategies, such as personas, chain-of-thought reasoning, or few-shot prompting, materially affect the quality of responses given by an LLM to a fixed task or question (Liu et al., 2023; Wei et al., 2024; White et al., 2023; Chen et al., 2024). In our framework, any particular prompt engineering strategy can be cast as an alternative choice of the text generator $\widehat{m}(\cdot;t)$.

The text generator additionally captures alternative choices of other hyper-parameters that govern how the LLM randomly generates text in response to a given prompt, such as its "temperature," top-p sampling or top-k sampling parameters. An LLM models the probability distribution over tokens that are likely to come next given an existing string; these hyperparameters all govern the extent of randomness in how the text generator samples from its probability distribution over next tokens. Alternative LLMs may have different default choices of these parameters; various APIs give researchers access to choose these parameters themselves. By defining the text generator as a deterministic mapping from prompts to responses, our framework can be interpreted as focusing on the case in which these parameters are such that the LLM greedily only generates its most likely token. Our results extend naturally to stochastic text generators as well, but at the expense of more cumbersome notation.

We briefly note that the text generator $\widehat{m}(\cdot;t)$ also captures other design choices that are typically hidden to the user. Since it is computationally costly to generate text from the transformer architecture, there exist alternative "inference algorithms" that may be used, such as greedy decoding or beam search. Alternative inference algorithms may affect an LLM's outputs, and so alternative choices correspond to alternative text generators in our framework.

Finally, the training algorithm captures all aspects of the design and training process of an LLM that produces the text generator. In particular, it captures the architecture, such as the total number of parameters, the number of attention layers, its choice of proxy objective in pre-training (such as next token prediction or variants like masked token prediction), its context window, and the details of its optimization procedure (batch size, initialization strategies, etc.). We emphasize that the training algorithm captures both pre-training and any additional fine-tuning, such as instruction tuning and reinforcement learning from human feedback. In this sense, the training dataset t in our framework must be interpreted as containing *all* strings upon which the LLM was pre-trained and fine-tuned on.

4 Prediction with LLMs

With this framework in hand, we turn to identifying the conditions under which the LLM can be incorporated into an economics research pipeline. We start with what we defined above as a "prediction problem." A researcher would like to assess the predictability of the linked economic variable Y_r based on the text pieces r. The economic variable could be, say, a stock price or an indicator for whether some piece of legislation passed Congress, but (as we discuss further below) could also be something more abstract like a new hypothesis.

While the researcher could in principle build their own predictor based on text from scratch, doing so would require vast amounts of training data to first learn the sort of lower-dimensional representation of language needed to successfully predict from text. Since LLMs have already learned some lower-dimensional representation, having been trained on enormous datasets, we may hope to use them as a foundation upon which to tackle our prediction problem.

We establish the key condition under which the use of an LLM for this type of research question is valid: there must be no *leakage* between the LLM's training dataset and the researcher's context. Training leakage is a thorny problem for benchmark evaluations of LLMs in computer science. Nonetheless, we discuss how training leakage can be managed through careful choice of the researcher's context and LLM. As a result, LLMs *can* be valuable tools for solving prediction problems, provided we take the threat of training leakage seriously.

4.1 The Researcher's Prediction Problem

Suppose the researcher prompts an LLM, trained on some unknown training dataset t, to form predictions of the linked variable based on the text pieces, $\widehat{Y}_r = \widehat{m}(r;t)$.[6] For some non-negative loss

[6] Researchers may instead use a given LLM to construct embeddings for each text piece r, and the resulting embeddings may then be used as features by a supervised machine learning algorithm to predict Y_r. Our analysis

function $\ell(y,\tilde{y})$, the researcher then calculates the sample average loss of the LLM's predictions on their collected dataset:

$$\frac{1}{N}\sum_{r\in\mathcal{R}} D_r \ell(Y_r,\widehat{m}(r;t)), \tag{1}$$

where $N=\sum_r D_r$ is the number of text pieces collected by the researcher. We would like to draw conclusions about the predictability of the linked economic variable Y_r from the text piece r based on the LLM's sample average loss (Equation 1). What conditions must the LLM satisfy in order for this to be valid?

To answer this question, we define the researcher's inferential goal in the prediction problem. We associate the researcher with a *context* $Q(\cdot)\in\mathcal{Q}$ that summarizes their own sampling distribution over economically relevant text pieces and beliefs about the LLM's sampling distribution over training datasets; more precisely, $Q(\cdot)$ is a joint distribution over the sampling indicators (D,T). The researcher's context $Q(\cdot)$ therefore summarizes two distinct features: first, it defines the collection of text pieces over which the researcher would like to assess the predictability of Y_r — this is chosen and known to the researcher; second, since its exact contents are unknown, it also captures the researcher's uncertainty over what strings entered into the LLM's training dataset.

We assume that the collection of research contexts $\mathcal{Q}$ satisfies the following assumptions.

Assumption 1. Letting $t=\left(t_{\sigma_1},\dots,t_{\sigma_{|\Sigma^*|}}\right)'$ denote the vector of indicators summarizing the large language model's realized training dataset, all research contexts $Q(\cdot)\in\mathcal{Q}$ satisfy the following assumptions:

(i) For all values d, $Q(D=d,T=t)=\Pi_{\sigma\in\Sigma^*}Q(D_\sigma=d_\sigma,T_\sigma=t_\sigma)$.

(ii) $\mathbb{E}_Q[\sum_{r\in\mathcal{R}}D_r]=\mathbb{E}_Q[\sum_{r\in\mathcal{R}}D_r\,|\,T=t]$.

Assumption 1(i) states that the sampling process across strings is independent but need not be identically distributed, capturing the idea that not all strings may be sampled with equal probability. Assumption 1(ii) states that irrespective of the LLM's corpus, the researcher always samples the same number of text pieces on average. As notation, $q_\sigma^{T|D}(t_\sigma)=Q(T_\sigma=t_\sigma\,|\,D_\sigma=1)$ is the conditional probability the string is sampled by the LLM's training dataset given that it is sampled by the researcher. We denote marginal probabilities as $q_\sigma^T(t_\sigma)=Q(T_\sigma=t_\sigma)$ and $q_\sigma^D=Q(D_\sigma=1)$.

Conditional on the LLM's realized training dataset, under mild regularity conditions as the number of economically relevant text pieces grows large (see Appendix C.1), the researcher's

in this section equally applies to evaluating the performance of a prediction function that uses LLM embeddings as features.

sample average loss converges to

$$\frac{1}{N}\sum_{r\in\mathcal{R}} D_r \ell(Y_r,\widehat{m}(r;t)) - \frac{1}{\mathbb{E}_Q[\sum_{r\in\mathcal{R}} D_r \,|\, T=t]}\mathbb{E}_Q\left[\sum_{r\in\mathcal{R}} D_r \ell(Y_r,\widehat{m}(r;t)) \,|\, T=t\right] \xrightarrow{p} 0, \tag{2}$$

and it recovers a particular weighted average of the LLM's loss over all economically relevant text pieces.

With this in hand, we introduce the researcher's inferential goal in the prediction problem. What can the researcher conclude about the LLM's predictive performance in a research context given that the LLM is a black box?

The most relevant information the researcher has about the LLM is usually *not* about the different design choices made inside the black box, but rather about high-level properties regarding the model's behavior or output. For example, builders of LLMs report their performance on natural language processing benchmarks, such as BIG-Bench or MMLU, and standardized exams, like the SAT and GRE.

We model this by assuming the researcher only knows that a given LLM satisfies some property in their research context $Q(\cdot)$; more formally, $\widehat{m}(\cdot;t)\in\mathcal{M}$, where $\mathcal{M}$ is some collection of possible text generators. We refer to the collection $\mathcal{M}$ as a *guarantee*, and the researcher only knows the text generator satisfies this guarantee (e.g., it could be any text generator that receives a score of at least 330 on the GRE). Given an LLM with guarantee $\mathcal{M}$, the researcher would like to draw their conclusions in the prediction problem based on the sample average loss.

Definition 1. The large language model $\widehat{m}(\cdot;t)$ with guarantee $\mathcal{M}$ *generalizes* in the research context $Q(\cdot)$ if, for all text generators satisfying the guarantee $\widehat{m}(\cdot)\in\mathcal{M}$,

$$\mathbb{E}_Q\left[\sum_{r\in\mathcal{R}} D_r \ell(\widehat{m}(r),Y_r) \,|\, T=t\right] = \mathbb{E}_Q\left[\sum_{r\in\mathcal{R}} D_r \ell(\widehat{m}(r),Y_r)\right].$$

The large language model $\widehat{m}(\cdot;t)$ with guarantee $\mathcal{M}$ is a *general-purpose technology for prediction* if it generalizes in all research contexts $Q(\cdot)\in\mathcal{Q}$.

The above definition is an "out-of-sample" inferential goal in the prediction problem. The LLM $\widehat{m}(\cdot;t)$ with guarantee $\mathcal{M}$ generalizes in the research context $Q(\cdot)$ if evaluating its sample average loss recovers its average loss over the researcher's target collection of text pieces in their research context. Importantly, under Definition 1, the researcher's workflow in the prediction problem is justified for *any* LLM that satisfies the guarantee $\mathcal{M}$ — the researcher can ignore all of the details in the design of the LLM and safely proceed to draw conclusions based on the sample average loss knowing only the guarantee $\mathcal{M}$ is satisfied. We further say that the LLM $\widehat{m}(\cdot;t)$ with guarantee $\mathcal{M}$ is a general-purpose technology for prediction if this empirical workflow is justified in *all* possible contexts $Q(\cdot) \in \mathcal{Q}$. Knowing the guarantee is satisfied implies this workflow in the prediction problem is justified in any research context.

To make this more concrete, let us return to our earlier empirical applications.

Example: Congressional legislation Consider researchers that would like to predict whether a bill was passed by either house of the United States Congress Y_r using only its short text description r. To do so, researchers select a given language model trained on the unknown training dataset $T = t$ and a particular prompt engineering strategy to generate responses $\widehat{m}(r;t)$. For example, researchers may prompt the language model as

```
Answer this question as if you were a helpful research assistant for a
political scientist.  Here is the description of a piece of legislation:
‘‘A bill to revise the boundary of Crater Lake National Park in the State
of Oregon.’’ Will this bill pass either the United States House of Rep-
resentatives or Senate?
```

Each researcher calculates the sample average loss of the LLM's predictions on their own collected sample of Congressional bills. Different researchers may study alternative sampling distributions over Congressional bills, studying particular Congresses, for example, or particular policy areas. When can any researcher that is using a given LLM in this manner reliably conclude that whether a bill passes Congress is predictable by its description? ▲

Example: Financial news headlines Consider researchers that would like to predict a company's realized returns Y_r based on the text of a financial news headline r. To do so, researchers select an LLM trained on the unknown training dataset $T = t$ and a particular prompting strategy in order to generate predictions $\widehat{m}(r;t)$. For example, researchers may prompt the LLM as

```
You are a knowledgeable financial analyst.  Here is a headline about Bank
of New York Mellon: ‘‘Bank of New York Mellon Q1 EPS $0.94 Misses $0.97
Estimate, sales $3.9B Miss $4.01B Estimate.’’ Will this headline increase
or decrease the Bank of New York Mellon’s stock price?
```

Each researcher calculates the sample average loss of the LLM's predictions on their own collected sample of financial news headlines. Different researchers may select alternative sampling distributions over financial news headlines, studying particular time periods or particular industries. When can any researcher using a given LLM in this manner reliably conclude that realized returns are predictable by financial news headlines? ▲

4.2 Training Leakage as a Threat to Prediction

We next clarify what guarantee $\mathcal{M}$ is necessary and sufficient for an LLM to generalize in a research context, and therefore whether it is a general-purpose technology for prediction.

Lemma 1. *Under Assumption 1, for any research context $Q(\cdot)\in\mathcal{Q}$ and text generator $\widehat{m}(\cdot)$,*

$$\mathbb{E}_Q\left[\sum_{r\in\mathcal{R}} D_r\ell(Y_r,\widehat{m}(r))\right]=\mathbb{E}_Q\left[\sum_{r\in\mathcal{R}} D_r\ell(Y_r,\widehat{m}(r))\,|\,T=t\right]-\mathbb{E}_Q\left[\sum_{r\in\mathcal{R}} D_r\left(\frac{q_r^{T|D}(t_r)}{q_r^T(t_r)}-1\right)\ell(Y_r,\widehat{m}(r))\right].$$

Proposition 1. *The large language model $\widehat{m}(\cdot;t)$ generalizes for research context $Q(\cdot)\in\mathcal{Q}$ if and only if it satisfies the guarantee $\mathcal{M}(Q)$ for*

$$\mathcal{M}(Q)=\left\{\widehat{m}(\cdot):\ -\mathbb{E}_Q\left[\sum_{r\in\mathcal{R}} D_r\left(\frac{q_r^{T|D}(t_r)}{q_r^T(t_r)}-1\right)\ell(Y_r,\widehat{m}(r))\right]=0\right\}. \qquad (3)$$

Consequently, the LLM $\widehat{m}(\cdot;t)$ is a general-purpose technology for prediction if and only if it satisfies the guarantee $\bigcap_{Q\in\mathcal{Q}}\mathcal{M}(Q)$.

Lemma 1 and Proposition 1 together establish that the given LLM generalizes if and only if there is no *training leakage* (Equation 3). The term $\frac{q_r^{T|D}(t_r)}{q_r^T(t_r)}-1$ captures any dependence between the researcher's sampling distribution over text pieces and their beliefs over the LLM's sampling distribution over strings. When this term is non-zero for a text piece, knowing that the researcher has sampled a text piece affects our beliefs about the likelihood the text piece has entered the LLM's training dataset. In this sense, Equation (3) captures the extent to which there is "overlap" between the researcher's dataset and the LLM's training dataset in the research context $Q(\cdot)$. If further the LLM predicts well on text pieces that are likely included in the training dataset, then Equation (3) will tend to be positive and so the sample average loss of the LLM will tend to overstate its performance. Our uncertainty over what exact strings entered into the LLM's training dataset acts like an omitted variables bias in the prediction problem.

4.3 Evidence on Training Leakage

Most existing evidence on training leakage sits in computer science and natural language processing and — unfortunately — indicates that training leakage is a pernicious problem in evaluating the capabilities of LLMs. There is ample evidence, for instance, that the training datasets of LLMs contain examples from common standardized-test benchmarks, such as the SAT, GRE and AP exams, as well as popular benchmark datasets in natural language processing (Sainz et al., 2023; Golchin and Surdeanu, 2024).[7] As a result there is growing skepticism about the reliability of evaluating the capabilities of LLMs on any dataset that is publicly available online (e.g., Wei et al., 2024; Ravaut et al., 2024). In hindsight, perhaps none of this evidence should be surprising; there are, after all, sizable financial incentives at stake in meaningfully improving performance

[7] There even exists a public running tabulation of all natural language processing benchmark datasets that have been found in the training datasets of various large language models (LM Contamination Index, 2024).

on these benchmarks with every new LLM that is released.

Since most of this research focuses on canonical computer science applications, it is only indirectly relevant for economics. The incentives for leakage of economically relevant text pieces into the training datasets of LLMs is potentially different. How much should we worry that naive uses of LLMs in economic prediction problems suffer from training leakage? To answer this question, we next test for training leakage in two empirical settings relevant to economists.

4.3.1 Assessing Training Leakage in Congressional Legislation

We first assess training leakage in an empirical setting relevant for economists studying politics and political economy: congressional legislation. We use data from the Congressional Bills Project (Wilkerson et al., 2023; Adler and Wilkerson, 2020), which contains the text description r for more than 400,000 bills proposed in the U.S. Congress. For each piece of legislation, we also observe whether it passed either the House or the Senate Y_r. For our empirical exercise, on a random sample of 10,000 Congressional bills introduced from 1973 to 2016, we explore whether the bill passed the House or the Senate can be predicted from the text of its description alone. A substantial literature in economics and political science studies the complicated strategic dynamics involved in congressional voting patterns; these dynamics make predicting legislative outcomes a challenging activity – especially using only a bill's short text description. Indeed, among these 10,000 randomly sampled bills, only 7.4% pass the House and 6.0% pass the Senate.

For our empirical analysis we generate predictions $\hat{Y}_r = \widehat{m}(r;t)$ based on each bill's text description r by prompting GPT-4o (see Appendix Figure A12 for the specific prompt). Impressively, we find that GPT-4o correctly predicts the bill's outcome 91.2% of the time in the House and 92.5% of the time in the Senate (left panel of Table 1). Taking these results at face value, it would be tempting to conclude that GPT-4o is able to detect subtle undercurrents in the bill's writing that point the way the political winds are blowing.

So what drives GPT-4o's ability to accurately predict whether a Congressional bill will pass the House or the Senate based only on its text description? The answer is that the the text of congressional legislation is likely included in GPT-4o's training dataset, producing training leakage in this prediction problem.

To evaluate training leakage, we prompt GPT-4o to complete each bill's text description based on only the first half of its text, following research in computer science such as Golchin and Surdeanu (2024) (see Appendix Figure A13 for the specific prompt). On 344 bills, GPT-4o completes the bill's text description *exactly* as it is written in the Congressional Bills Project database, indicating that not only was GPT-4o likely trained on these text pieces but it appears to have memorized them. Figure 1 provides two examples of original bill descriptions and GPT-4o's successful completions. On the other bills, GPT-4o's completed bill descriptions are close to the original bill descriptions. We construct word embeddings of GPT-4o's completed bill descriptions and the original bill descriptions, finding that the word embeddings of GPT-4o's descriptions are far closer to those of the originals than the average distance between the word embeddings of two randomly selected bills (left panel of Appendix Table A1).

One might wonder whether this training leakage could be addressed through clever prompt engineering — for example, by commanding GPT-4o to not pay attention to any information past a certain date. Despite applying this prompt engineering strategy to our sample of Congressional bills (see Appendix Figure A12 and A13 for associated prompts), we still find substantial evidence of training leakage. Even when explicitly told to not consider any information past the bill's introduction date in Congress, GPT-4o can still accurately predict its outcome in the House and the Senate based on these small snippets of text (right panel of Table 1), GPT-4o still exactly completes nearly the same number of bill descriptions as without the prompt engineering (330 versus 344) (Appendix Figure A1 for examples), and more generally the word embeddings of GPT-4o's completed descriptions remain quite close on average to those of the originals (right panel of Appendix Table A1).

With a little reflection, it should perhaps not be surprising that prompt engineering alone cannot fully address the problem of training leakage. This can be seen by remembering that any large language model actually consists of *two* algorithms, not just one. Prompting an LLM's text generator to not use data after a certain cutoff date does not lead to some algorithmic retraining using only data before the cutoff date — after all, prompt engineering only influences the text generator, not the underlying training algorithm.

4.3.2 Assessing Training Leakage in Financial News Headlines

We next consider another domain relevant for economists: financial markets. Existing work such as Glasserman and Lin (2023) and Lopez-Lira and Tang (2024) found that LLMs appear to predict company-level stock returns accurately using the text of pertinent financial news headlines. We complement this work by assessing the scope for training leakage.

We use publicly-available data on financial news headlines collected by Aenlle (2020), which captures financial news headlines r for nearly 4 million financial news headlines covering 6,000 publicly traded companies from 2009-2020. For our empirical exercise, we randomly sampled 10,000 financial news headlines from 2019 and prompt GPT-4o to complete each financial news headline based on only 50% of its text (see Appendix Figure A14 for the specific prompt). GPT-4o reproduces 60 financial news headlines exactly as they were written in the publicly available dataset, indicating that GPT-4o was likely trained on these headlines and memorized them. Figure 2 provides two examples of financial news headlines that GPT-4o successfully completed. Moreover, on all other headlines, word embeddings of GPT-4o's completions are quite close to those of the original headlines in the dataset (left panel of Appendix Table A2).

We again explore whether explicitly incorporating date restrictions into the LLM prompt moderate this evidence of training leakage (see Appendix Figure A14 for the associated prompts). Surprisingly, prompt engineering if anything appears to make the problem *worse*; GPT-4o now reproduces 73 headlines exactly — that is, including explicit date restrictions in our prompt leads to *stronger* evidence of memorization. And as with the congressional legislation, we continue to find that on average word embeddings of GPT-4o's completions with the date restriction are close to those of the original headlines.

In related work, Sarkar and Vafa (2024) provide a different type of evidence in support of training leakage in finance. Specifically, the authors prompt Llama 2 to predict the potential future risks for companies based on transcripts of their earnings calls from September-November 2019. In over 25% of earnings calls, the LLM discusses Covid-19, pandemic and supply chains as a threat to the company – a distinct yet related form of "look ahead bias" that comes from the LLM being trained on future information from the later time period than which the LLM's performance is being evaluated. See also Glasserman and Lin (2023). Altogether, our findings on memorization of financial news headlines and this existing work indicate that there is substantial risk of training leakage when using LLMs in finance.

4.4 Recommendations for Empirical Practice

We have shown that naive uses of large language models in economic prediction problems can suffer from training leakage, which cannot be fully addressed through simple prompt engineering strategies. However, prediction is still a promising use case since the "no training leakage" condition can be mechanically enforced through the researcher's choice of LLM.

In particular, researchers interested in using LLMs for prediction tasks in economics applications *must* use open-source LLMs that either clearly document their underlying training data and/or provide a clear time-stamp beyond which the LLM has not been re-trained.[8] As an example, the Llama family of models (Touvron et al., 2023; Dubey et al., 2024) or StoriesLM family (Sarkar, 2024) are posted publicly online with a fixed release date, and so researchers know that these models have not been trained on any strings past this release date.

The corollary is that economists absolutely *must not* use "closed" LLMs, such as the GPT family from OpenAI or the Claude family from Anthropic, which can only be accessed through

[8] For the two empirical examples we consider in this section, predicting future stock prices from corporate earnings reports, and predicting whether a Congressional bill passes using a snippet of the bill's text, avoiding leakage requires both use of a time-stamped LLM and ensuring that the text documents for which we wish to make predictions have not been used to train the LLM. In other applications just the latter condition will be enough—use of a time-stamped LLM is not necessary. Consider, for example, predicting whether a defendant is detained pretrial as a function of the courtroom transcript, predicting a patient's health status as a function of a doctor's notes in the medical records, and predicting whether a student will drop out of high school as a function of the written assignments they submit as part of their English classes.

their provider's chat interfaces or APIs. For such closed models, there is simply no way to know what the training data consists of, since that is typically proprietary information.[9] Nor is there any way to know what time period is covered by the closed model's training dataset, partly because these models may be continually fine-tuned or subtly tweaked over time.

The difficulty of defining the possible limits to the training datasets of closed language models is not a hypothetical concern. For example, researchers in natural language processing now regularly caution against sending test data to the APIs or chat interfaces of closed LLMs since these data may be used in further fine-tuning or the development of new models (Jacovi et al., 2023) — indeed, Balloccu et al. (2024) estimates that 263 benchmarks in natural language processing may have been inadvertently leaked to OpenAI through use of the chat interface and API, and Cheng et al. (2024) questions the validity of publicly stated "knowledge cutoffs" in closed LLMs. Most worryingly for researchers, Barrie, Palmer and Spirling (2024) illustrates that the results of submitting the same prompt to GPT-4o yields results that change month-to-month, despite there being no publicly announced changes to the underlying model. That is a clear demonstration that closed LLMs are continually being tweaked in unknown ways. These findings are once again perhaps unsurprising. There are sizable financial incentives at stake in continually improving the performance of closed LLMs; after all, these LLMs are designed to be *commercial products*, not scientific research tools.

In principle there is a special case where, given the expression in Equation (3), there is mechanically no training leakage.

Corollary 1. *Under Assumption 1, there is no training leakage (3) in the research context $Q(\cdot)$ if, for all $r \in \mathcal{R}$, $q_r^D \in \{0,1\}$.*

In words, there is no training leakage if the researcher always observes all text pieces in their research context; in this case, there is no "out-of-sample" generalization required in the researcher's inferential goal. But in many ways that condition is incompatible with the very goal of using LLMs for prediction tasks, which by construction involves building tools to make predictive inferences about never-seen-before instances. In contrast, this no-generalization condition may be more relevant for estimation problems where the LLM can wind up serving in practice as a measurement tool. The use of LLMs for economic estimation tasks is the problem to which we turn next.

5 Plug-In Estimation with LLMs

In this section, we turn to the conditions under which such models could potentially serve as a general-purpose technology for estimation problems. We define those as ones in which the researcher would like to measure the economic concept V_r expressed by each text piece r in order

[9] Another candidate reason it may be kept confidential in practice is, for better or worse, to minimize concerns about legal liability for copyright infringement.

to estimate some downstream parameter of interest. Since the measurement process $f^*(\cdot)$ defining the economic concept is costly to scale, the researcher would like to instead use an LLM and plug in the model's outputted labels in place of the measurements she would have otherwise collected.

In addition to no training leakage, we argue that this practice requires that there is no *measurement error* in the large language model's labels of the economic concept. In other words, the LLM must reproduce the existing measurement process everywhere—for every dataset and for every task. This assumption, stated so directly, is as a conceptual matter unlikely to be true. As an empirical matter, the assumption of no measurement error contradicts even the most optimistic results of benchmark evaluations of LLM performance as reported by either tech companies or computer science researchers. We extend that evidence here by empirically illustrating the severity of the implications of LLM measurement error for downstream economic parameter estimation. In both of our empirical applications (financial news headlines and Congressional legislation), we show that seemingly minor implementation choices such as which LLM to use or how to prompt the model affect not only the labels the LLM assigns to a given piece of text, but also substantially affect the size, significance and even sign of economic parameters estimated using those LLM labels.

The good news is that there is a fix: collect a sample of benchmark data using the existing measurement procedure and empirically model the LLM's error. The corollary of this fix is that economists should *not* use the LLM's output for plug-in estimation without this type of validation data and measurement error correction.

5.1 The Researcher's Estimation Problem

To illustrate what guarantee is needed to use an LLM to solve an estimation problem, let us further formalize what we mean by an estimation problem.

The researcher specifies some parameter of interest $\theta \in \Theta$ and a moment condition $g(\cdot)$ that, in principle, identifies this parameter. Consequently, if the researcher collected the economic concept by applying the existing measurement procedure $f^*(\cdot)$ to each text piece r, she would be satisfied to calculate the parameter estimate

$$\widehat{\theta}^* = \arg\min_{\theta \in \Theta} \frac{1}{N} \sum_{r \in \mathcal{R}} D_r g(V_r, W_r; \theta). \tag{4}$$

For example, letting $g(V_r, W_r; \theta) = (V_r - W_r'\theta)^2$, then the researcher would like to understand how the economic concept V_r relates to the linked variables W_r, and $\widehat{\theta}^*$ corresponds to the sample regression coefficient. Due to the text processing problem, however, the researcher cannot calculate $\widehat{\theta}^*$ directly.

To solve the text processing problem, suppose the researcher prompted an LLM to measure the economic concept on each text piece, $\widehat{V}_r := \widehat{m}(r; t)$, and plugged the LLM's labels into the moment condition

$$\widehat{\theta} = \arg\min_{\theta \in \Theta} \frac{1}{N} \sum_{r \in \mathcal{R}} D_r g(\widehat{m}(r;t), W_r; \theta). \tag{5}$$

We would like to draw conclusions about the target estimate $\widehat{\theta}^*$, which relies on measurements of the economic concept $V_r := f^*(r)$, based on the feasible estimate $\widehat{\theta}$, which relies on the LLM's labels $\widehat{V}_r := \widehat{m}(r;t)$. What conditions must the LLM satisfy in order for this workflow to be valid?

To answer this question, we must define the researcher's inferential goal in the estimation problem. We associate the researcher with a research context $Q(\cdot) \in \mathcal{Q}$, where the collection $\mathcal{Q}$ satisfies Assumption 1. We further assume that the researcher could study any moment condition $g(\cdot) \in \mathcal{G}$ satisfying the following assumption.

Assumption 2. For all $g(\cdot) \in \mathcal{G}$, $g(\cdot)$ is differentiable and there exists some $\overline{G} > 0$ such that $\left| \frac{\partial g(v, W_r; \theta)}{\partial \theta} \right| \leq \overline{G}$ for all $r \in \mathcal{R}$ and $\theta \in \Theta$.

Conditional on the LLM's realized training dataset, under mild regularity conditions as the number of economically relevant text pieces grow large (see Appendix C.1), the researcher's sample moment condition converges to

$$\frac{1}{N} \sum_{r \in \mathcal{R}} D_r g(\widehat{m}(r;t), W_r; \theta) - \frac{1}{\mathbb{E}[\sum_{r \in \mathcal{R}} D_r]} \mathbb{E}\left[\sum_{r \in \mathcal{R}} D_r g(\widehat{m}(r;t), W_r; \theta) \,|\, T = t \right] \xrightarrow{p} 0 \tag{6}$$

at any parameter value $\theta \in \Theta$.

With this in hand, we introduce the following definition. Given an LLM with guarantee $\mathcal{M}$, the researcher would like to recover the moment condition defined using the economic concept.

Definition 2. The LLM $\widehat{m}(\cdot;t)$ with guarantee $\mathcal{M}$ *automates* the existing measurement procedure $f^*(\cdot)$ for the moment condition $g(\cdot)$ in research context $Q(\cdot)$ if, for all models satisfying the guarantee $\widehat{m}(\cdot) \in \mathcal{M}$ and all $\theta \in \Theta$,

$$\mathbb{E}_Q\left[\sum_{r \in \mathcal{R}} D_r g(\widehat{m}(r), W_r; \theta) \,|\, T = t \right] = \mathbb{E}_Q\left[\sum_{r \in \mathcal{R}} D_r g(V_r, W_r; \theta) \right].$$

The LLM $\widehat{m}(\cdot;t)$ with guarantee $\mathcal{M}$ is a *general-purpose technology for estimation* if it automates the researcher's measurement process for all moment conditions $g(\cdot) \in \mathcal{G}$ and research contexts $Q(\cdot) \in \mathcal{Q}$.

In words, the LLM $\widehat{m}(\cdot;t)$ with guarantee $\mathcal{M}$ automates the existing measurement procedure $f^*(\cdot)$ for the economic concept if and only if the population moment condition that plugs in the LLM-constructed labels recovers the population moment condition that plugs in the economic concept. In this case, the researcher's workflow in the estimation problem is justified—she can safely ignore all of the details in the design of the LLM and proceed knowing only the guarantee $\mathcal{M}$ is satisfied. We further say that the LLM $\widehat{m}(\cdot;t)$ with guarantee $\mathcal{M}$ is a general-purpose technology for estimation if this empirical workflow is justified for all possible research contexts $Q(\cdot)\in\mathcal{Q}$ and moment conditions $g(\cdot)\in\mathcal{G}$.

5.2 Measurement Error as a Threat to Estimation

We next clarify what type of guarantee $\mathcal{M}$ is necessary and sufficient for an LLM to automate the existing measurement procedure in a particular research context and moment condition, and therefore whether it is a general-purpose technology for estimation. We first decompose the difference between the plug-in moment condition and the target moment condition into two terms.

Lemma 2. *Under Assumption 1, for any research context $Q(\cdot)\in\mathcal{Q}$, moment condition $g(\cdot)\in\mathcal{G}$ and text generator $\widehat{m}(\cdot)$, $\mathbb{E}_Q[\sum_{r\in\mathcal{R}}D_r g(\widehat{m}(r),W_r;\theta)\,|\,T=t]-\mathbb{E}_Q[\sum_{r\in\mathcal{R}}D_r g(V_r,W_r;\theta)]$ equals*

$$\left(\mathbb{E}_Q\left[\sum_{r\in\mathcal{R}}D_r g(\widehat{m}(r),W_r;\theta)\right]-\mathbb{E}_Q\left[\sum_{r\in\mathcal{R}}D_r g(V_r,W_r;\theta)\right]\right)+\mathbb{E}_Q\left[\sum_{r\in\mathcal{R}}D_r\left(\frac{q_r^{T|D}(t_r)}{q_r^{T}(t_r)}-1\right)g(\widehat{m}(r),W_r;\theta)\right]. \quad (7)$$

The final term in Equation (7) is familiar from our analysis of prediction problems—it represents training leakage between the LLM's training dataset and the researcher's context. As discussed in the previous section, training leakage can be controlled through an appropriate choice of open LLM and the researcher's context (e.g., Corollary 1). Consequently, we will maintain the assumption that the LLM satisfies the guarantee of no training leakage.

We will instead focus on what additional guarantee is needed to control the first term in Equation (7). Towards this, we introduce some additional notation. We define $\mathcal{M}(Q,\delta)$ to be the collection of text generators satisfying $\|\widehat{m}(\cdot)-f^*(\cdot)\|_{\infty,Q}=\max_{r\in\mathcal{R}:\,q_r^D>0}|\widehat{m}(r;t)-f^*(r)|\leq\delta$. We further say that a moment condition $g(\cdot)$ is *sensitive* to the economic concept V_r in research context $Q(\cdot)$ if $q_r^D>0$ and there exists some $\underline{G}>0$ such that $|\frac{\partial g(v,W_r;\theta)}{\partial v}|\geq\underline{G}$ for all v,θ. Let $\mathcal{R}(g,Q)$ denote the collection of sensitive text pieces.

Lemma 3. *Consider a researcher studying any moment condition $g(\cdot)\in\mathcal{G}$ in research context $Q(\cdot)\in\mathcal{Q}$. Then, for all $\theta\in\Theta$ and $\widehat{m}(\cdot)\in\mathcal{M}(Q,\delta)$ satisfying no training leakage,*

$$\left|\mathbb{E}_Q\left[\sum_{r\in\mathcal{R}}D_r g(\widehat{m}(r),W_r;\theta)\,|\,T=t\right]-\mathbb{E}_Q\left[\sum_{r\in\mathcal{R}}D_r g(V_r,W_r;\theta)\right]\right|\leq\overline{G}\delta. \quad (8)$$

But, for all $\theta\in\Theta$, there exists $\widehat{m}(\cdot)\in\mathcal{M}(Q,\delta)$ that satisfies no training leakage such that, for $\delta(r)$ defined in the proof,

$$\left|\mathbb{E}_Q\left[\sum_{r\in\mathcal{R}}D_r g(\widehat{m}(r),W_r;\theta)\,|\,T=t\right]-\mathbb{E}_Q\left[\sum_{r\in\mathcal{R}}D_r g(V_r,W_r;\theta)\right]\right|\geq\underline{G}\left(\sum_{r\in\mathcal{R}(g,Q)}|\delta(r)|q_r^D\right). \quad (9)$$

The guarantee $\mathcal{M}(Q,\delta)$ bounds the error of the LLM for the existing measurement procedure $f^*(\cdot)$ in the research context $Q(\cdot)$. While knowing this guarantee is satisfied is sufficient to bound the error introduced by plugging in the labels into the researcher's estimation problem (Equation 8), that is not enough to ensure that the LLM automates the existing measurement process. There exist text generators satisfying the guarantee $\mathcal{M}(Q,\delta)$ whose labels lead to meaningful errors in the researcher's estimation problem (Equation 9). In other words, if the researcher only knows that the guarantee $\mathcal{M}(Q,\delta)$ is satisfied, it could still be the case that the LLM's labels exhibit small errors that lead to non-trivial bias for the downstream economic parameters we seek to recover — for example, if the LLM's errors are correlated with other economic variables W_r. The brittleness of an LLM is pernicious in this sense. Knowing the guarantee $\mathcal{M}(Q,\delta)$ is satisfied does not ensure that the LLM successfully automates the existing measurement process.

So Lemma 3 immediately implies that an LLM is a general-purpose technology for estimation if and only if it satisfies a strong guarantee that its outputs have no errors in reproducing the existing measurement process. Put simply, researchers can safely ignore the details of the LLM's design no matter the research context studied and economic question being asked if and only if the LLM has the following strong guarantee: its labels reproduce the existing measurement process everywhere.

Proposition 2. *Suppose the LLM $\widehat{m}(\cdot;t)$ satisfies the guarantee of no training leakage in all research contexts $Q(\cdot)\in\mathcal{Q}$ and moment conditions $g(\cdot)\in\mathcal{G}$. Provided there exists some $g(\cdot)\in\mathcal{G}$ that is sensitive to the economic concept for any $r\in\mathcal{R}$, then the language model is a general-purpose technology for estimation if and only if $\widehat{m}(\cdot;t)$ satisfies the guarantee $\mathcal{M}(Q,0)$ for all research contexts $Q(\cdot)$.*

5.3 Evidence on Measurement Error

Using LLM outputs in an estimation problem requires the strong guarantee that the LLM reproduces the existing measurement process $f^*(\cdot)$. This guarantee is often implicitly invoked in many research uses of LLMs. It is also quite intuitively appealing from observing the LLM's impressive performance on some tasks, although, as we discussed in Section 2, that intuition turns out to be misleading. Indeed, no LLM performs perfectly on existing benchmark evaluations, and a growing body of research in computer science provides evidence documenting a variety of surprising errors with LLMs. What remains unknown is the degree to which this brittleness matters in practice for economics research.

That is the question to which we turn next. We empirically demonstrate that LLM measurement error has substantively important implications for what economists ultimately care about: downstream parameter estimation. We illustrate this point for the canonical application of linear regression and the two empirical examples introduced earlier, of financial news headlines and Congressional bills.

5.3.1 Linear Regression with LLMs

Consider a researcher who would like to relate the economic concept V_r and linked variables W_r by estimating a linear regression. Due to the text processing problem, this is infeasible; instead the researcher prompts an LLM and reports the associated plug-in regression. Of course, there are two types of regressions the researcher may run. First, she may regress the economic concept on the linked variables

$$V_r = W_r'\beta^* + \epsilon_r, \text{ and } \widehat{m}(r;t) = W_r'\beta + \widetilde{\epsilon}_r. \tag{10}$$

Second, she may regress the linked variable on the economic concept

$$W_r = V_r'\alpha^* + \nu_r, \text{ and } W_r = \widehat{m}(r;t)'\alpha + \widetilde{\nu}_r. \tag{11}$$

The bias of the plug-in regression coefficients β and α depends on how the LLM's error $\Delta_r = \widehat{m}(r;t) - V_r$ varies across text pieces. These are well-known results, dating back to Bound et al. (1994), that we restate in our framework.

Proposition 3. *Consider the research context $Q(\cdot)$ and assume that $q_r^T(t_r) = q_r^{T|D}(t_r)$ for all $r \in \mathcal{R}$.*

(i) Defining $\lambda_{\Delta|W}$ to be the coefficients in the regression of $\Delta_r = \widehat{m}(r;t) - V_r$ on W_r in the research context $Q(\cdot)$, then $\beta = \beta^* + \lambda_{\Delta|W}$.

(ii) Defining $\lambda_{V|\widehat{V}}$ to be the regression coefficients of V_r on $\widehat{m}(r;t)$ and $\lambda_{\eta|\widehat{V}}$ to be the regression coefficients of η_r on $\widehat{m}(r;t)$ in the research context $Q(\cdot)$, then $\alpha = \lambda_{V|\widehat{V}}\alpha^* + \lambda_{\eta|\widehat{V}}$.

In other words, when the economic concept is the dependent variable, the bias of the plug-in regression coefficient β is summarized by the extent to which the LLM's error $\Delta_r = \widehat{m}(r;t) - V_r$ covaries with the linked variable W_r in the research context. Imperfect automation of the existing measurement process induces an omitted variable bias in the plug-in regression coefficient. While its expression is more complicated when the economic concept is the covariate, the bias of the plug-in regression coefficient α can again be summarized by how the economic concept covaries with the LLM's error.

To concretely illustrate why researchers must be appropriately skeptical of claims that the labels of LLMs perfectly automate existing measurements in economic settings, we return to our earlier examples of financial news headlines and congressional legislation. If they are equivalent to existing measurements, it must be that the parameter estimates based on LLM labels should not be sensitive to specific implementation choices, such as the particular choice of large language model or the specific prompt engineering strategy. By contrast, Proposition 3 implies that variation in the estimates across models and prompting strategies must be driven by variation in the associated measurement error. If not—if the parameter estimates *are* sensitive to these LLM implementation choices—then clearly the labels of some of these LLMs cannot be treated as if they successfully automate existing measurements.

5.3.2 Assessing Measurement Error in Financial News Headlines

We return to the dataset from Section 4.3 of financial news headlines for 6,000 publicly traded stocks from 2009-2020. We focus on all financial news headlines published in 2019. We observe the text of each headline r, its publication date, and the ticker of the stock it refers to. Onto each financial news headline r, we merge the stock's realized returns within various windows after the publication date W_r (e.g., 1 day, 5 days, and 10 days after publication) as well as its lagged returns before the publication date (e.g., 1 day, 2 days and 3 days before publication) (Beta Suite by WRDS, 2024).

Financial news headlines express various economic concepts that we could measure and relate to realized stock returns. In this exercise, we focus on one simple economic concept: is the headline positive, negative or neutral news for the company? As an example, consider the headline

> `Bank of New York Mellon Q1 EPS $0.94 Misses $0.97 Estimate, sales $3.9B Miss $4.01B Estimate.`

Surely this is negative news for Bank of New York Mellon since it indicates the bank underperformed relative to market expectations. Now consider the headline

> `Disney And Charter Communications Announce Comprehensive Distribution Agreement; New Multi-Year Deal Delivers Full Suite Of Sports, News And Entertainment Networks From Walt Disney Television And ESPN To Spectrum Customers.`

This is reasonably labeled as positive news for Charter Communications since the distribution agreement with Disney enables Charter Communications to expand its offerings to existing and potential customers. While we could in principle read every single financial news headline published in 2019, this process $f^*(\cdot)$ would be painstaking.

So it is natural to instead turn to LLMs to solve this text processing problem. For our own analysis, we take each financial news headline r and prompt LLMs to label each news headline as positive, negative or neutral news for the company it refers to, as well as score the magnitude of this assessment and the LLM's confidence in the label, $\widehat{V}_r := \widehat{m}(r;t)$. This requires making several practical choices (as does any effort to solve any text processing problem with an LLM): what specific LLM to use? What prompt engineering strategy? If alternative answers to these practical questions lead to meaningfully different labels and downstream estimates, this would indicate non-ignorable errors in the LLM outputs.

To examine this possible sensitivity, we separately prompt GPT-3.5-Turbo, GPT-4o, and GPT-4o-mini to label each financial news headline r, using nine alternative prompts to each model. Our two base prompts provide the LLM with the text of the headline r and ask it to label whether this news is positive, negative, or neutral for the company; we also ask the LLM to provide its confidence and a magnitude in the label. Our two base prompts differ in how they ask the model to format its reply: filling-in-the-blanks text or a structured JavaScript Object Notation (JSON) object. The exact wordings of our base prompts are provided in Appendix Figure A15.

To round out our list of nine different prompts, we also vary them in two other ways as well. These variations are motivated by existing computer science research on prompting LLMs, which provides a large menu of prompt engineering strategies that can be appended to a base prompt and have been found to meaningfully affect the performance of LLMs on natural language processing tasks (Liu et al., 2023; Wei et al., 2024; White et al., 2023; Chen et al., 2024). First, we append to the prompt a request to the LLM to adopt one of four different "personas," like "knowledgeable economic agent" or "expert in finance" (see Appendix Figure A16 for the exact wording). We also append three prompt modifiers that ask the LLM to "think carefully" or "think step-by-step" and provide an explanation for its answer (see Appendix Figure A17 for the exact wording). Altogether, for each financial news headline r, we obtain labels $\widehat{V}_r^{m,p}$ associated with three different LLMs m and nine different prompting strategies p.

We first calculate the pairwise agreement in the labels produced by alternative prompting strategies p. The results, summarized in Figure 3 separately for each model m, show substantial variation in the pairwise agreement in the labels produced. For GPT-3.5-Turbo, our base prompt with fill-in-the-blank formatting only produces the same label as our base prompt with JSON formatting on 79% of financial news headlines. Commanding GPT-3.5-Turbo to behave as a "knowledgeable economic agent" produces the same label as commanding it to behave as a "successful trader" on 91.6% of financial news headlines. We see similar variation across pairs of prompting strategies for GPT-4o and GPT-4o-mini. Prompting GPT-4o to "think step-by-step" produces the same label as our base prompt with fill-in-the-blank formatting on 84.3% of headlines; whereas prompting GPT-4o to answer like an "expert in finance" versus an "expert in the economy" leads to agreement on 94.6% of headlines. Across models, there appear to be no consistent patterns in which pairs of prompting strategies tend to have the most agreement.

Of course, economists don't care about prompt responses per se; we care about what happens when we use these prompt responses to estimate economically relevant parameters. To what extent does this variation in LLM output matter for downstream parameter estimation?

To investigate this, for each model m and prompt p, we next regress the realized returns of each stock within 1-day, 5-days and 10-days after the headline's publication date Y_r on binary indicators for whether the LLM's labels $\widehat{V}_r^{m,p}$ are positive versus negative (with neutral the omitted reference category), controlling for the model's reported magnitudes and lagged realized returns. We report separately the coefficients $\widehat{\beta}_{m,p}$ on whether the LLM labels the headline as positive and whether the LLM labels it as negative, as well as the coefficient's associated t-statistics with standard errors clustered at the date and company level.

Figure 4 summarizes the variation in the resulting t-statistics across models and prompts in the regression of the LLM's labels $\widehat{V}_r^{m,p}$ against 1-day, 5-day and 10-day realized returns. Table 2 summarizes the variation in the coefficient estimates across models and prompts for all horizons of realized returns. In Appendix Figures A3 and Appendix Table A3, we show results that controls for the LLM's assessed confidence; results are qualitatively similar.

For the exact same set of financial news headlines, simply adjusting the prompting strategy or using a different LLM can yield remarkably different results. For example, looking at the coefficient on the positive headline indicator and 1-day realized returns, the results range from a t-statistic of nearly -6 up to $+2$ (Figure 4). Furthermore, the variation in coefficients is meaningful in terms of economic magnitudes even when the estimates share the same sign. This problem is even worse for returns on longer horizons, as the bottom panels of Figure 4 show. There are many combinations of prompt and model that produce entirely different directions and magnitudes of the relationship between the positive/negative label and realized returns.

5.3.3 Assessing Measurement Error in Congressional Legislation

We next return to the Congressional legislation data described earlier. For each bill, we observe the text of its description r as well as a collection of additional linked economic variables W_r, such as the party affiliation of the bill's sponsor, whether the bill originated in the Senate, and an ideological score – the DW1 roll call voting record – of the bill's sponsor. A natural political-economy question might be how these variables shape the topic of each bill, V_r. Could we draw valid inferences using LLMs to collect those labels rather than relying on human annotation?

For our analysis, we randomly select 10,000 Congressional bills and separately prompt GPT-3.5-Turbo and GPT-4o to label each Congressional bill for its policy area using alternative prompting strategies, including base prompts that modify the requested format, persona modifications, chain-of-thought modifications, and even few-shot examples (see Appendix E.4 for the specific prompts we used). Consequently, for each Congressional bill description r, we obtain labels $\widehat{V}_r^{m,p}$ associated with two different large language models m and twelve prompting strategies p. Appendix Figure A4 calculates the pairwise agreement in the labels produced by alternative prompting strategies p. We again find substantial variation in the pairwise agreement in the labels produced, and there appear to be no consistent patterns in which pairs of prompting strategies tend to have the most agreement.

We then separately regress the labeled economic concept — in this case, the policy topic of the bill — against linked covariates W_r, separately reporting the coefficients $\widehat{\beta}_{m,p}$ and their associated t-statistics. Figure 5 summarizes the variation in the resulting t-statistics across models and prompts: each column shows the result of using LLM labels $\widehat{V}_r^{m,p}$ for alternative bill topics as the dependent variable (i.e., Health, Banking, Finance and Domestic Commerce, Defense, Government Operations, and Public Lands and Water Management), and each row shows the result using a different linked covariate W_r (i.e., whether the bill's sponsor was a Democrat, whether the bill originated in the Senate, and the DW1 score of the bill's sponsor). Table 3 summarizes the coefficient estimates across models and prompts for each choice of labeled policy topic and the covariate. As was the case with financial news headlines, for each combination of labeled policy topic and linked covariate in the Congressional bills dataset we see substantial variability in the resulting t-statistics across different LLMs and prompting strategies.

5.4 Recommendations for Empirical Practice

When we see an LLM ace the SAT or GRE we *must* resist the natural urge to anthropomorphize the algorithm and assume it is capable of performing all sorts of other tasks that a human who performs well on standardized tests could be expected to carry out. Viral examples of their capabilities and performance on benchmark evaluations do not justify the strong guarantee needed for LLM's to automate an existing measurement procedure in estimation problems.

The good news is that there is a solution. The key to safely using this new technology is to actually look back to old lessons in econometrics: invest in collecting measurements $V_r := f^*(r)$ on a small benchmark or validation sample and use it to de-bias the plug-in estimate based on the LLM labels $\widehat{V}_r$. The virtues of benchmark data in dealing with measurement error is well-known – see, for example, classic work such as Bound and Krueger (1991), Bound et al. (1994), Lee and Sepanski (1995), and Bound, Brown and Mathiowetz (2001). It has been recently revived in machine learning; see Wang, McCormick and Leek (2020), Angelopoulos et al. (2023), Wei and Malik (2023), Egami et al. (2024), and Battaglia et al. (2024) among many others.

In the rest of this section, we demonstrate the value of debiasing LLM labels in the context of linear regression. We review the mechanics of debiasing a linear regression in which the LLM is the dependent variable and provide intuition about when it will work well based on asymptotic arguments. We then show this empirically in finite samples using our earlier example on Congressional legislation before turning to briefly discuss how these results carry over to the case of using the LLM's output as an explanatory rather than dependent variable.

5.4.1 Debiasing LLM Labels: Conceptual Results for Linear Regression

Let us return to the case where the researcher wishes to regress the economic concept V_r on the linked variables W_r, but relies on the LLM labels instead and reports the results of estimating the plug-in regression $\widehat{m}(r;t) = W_r'\beta + \widetilde{\epsilon}_r$. As noted above, the bias of the plug-in regression coefficient

β is summarized by the extent to which the LLM measurement error $\Delta_r = \widehat{m}(r;t) - V_r$ covaries with the linked variable W_r in the research context. Our discussion here immediately extends to regressing the linked variable on the economic concept (Equation 11); see Appendix C.2 for details.

Proposition 3 described above has two implications that both hinge on collecting benchmark data containing measurements $V_r := f^*(r)$. More specifically, on a random subset of the researcher's dataset, we will now assume that the researcher collects the label V_r. We refer to the collection of text pieces on which the researcher now observes $(r, W_r, \widehat{m}(r;t), V_r)$ as the validation sample, and we refer to the remaining text pieces on which she observes $(r, W_r, \widehat{m}(r;t))$ as the primary sample.

Indeed, in many empirical applications that use LLMs to solve an estimation problem, it is already common for researchers to collect such a validation sample (e.g., Durvasula, Eyuboglu and Ritzwoller, 2024; Hansen and Kazinnik, 2024). Often the validation data is used to only report the overall accuracy of the LLM's labels for the economic concept; although, as we have shown, this is not sufficient to ensure valid inference for whatever downstream parameters the labels are used to estimate.

The real (often unrealized) value of the validation sample comes from enabling the researcher to empirically estimate the bias associated with the plug-in regression; that is, the researcher can estimate $\widehat{\lambda}_{\Delta|W}$ by forming $\Delta_r = \widehat{m}(r;t) - V_r$ on the validation sample and regressing it on the linked variable W_r. By making the bias of the LLM estimable, there are two implications for empirical researchers.

First, the validation sample provides the researcher with a direct target to optimize when fine-tuning or prompt engineering a given LLM. Since her goal is to ultimately run some downstream regression, her goal is to obtain quality downstream estimates rather than simply maximizing the accuracy of the resulting labels — which, as we saw in the Congressional legislation application, can provide a misleading picture. For any choice of LLM m and prompt p, the researcher can estimate the bias of the plug-in regression $\widehat{\lambda}^{m,p}_{\Delta|W}$ associated with the labels $\widehat{V}^{m,p}_r$; and thereby select the combination that results in the smallest bias.

Second, more importantly, for any choice of an LLM and prompting strategy, the validation sample can be used to bias-correct the plug-in regression constructed on the primary sample. In other words, rather than directly reporting the plug-in regression $\widehat{\beta}$, the researcher can instead report the bias-corrected estimator

$$\widehat{\beta}^{debiased} = \widehat{\beta} + \widehat{\lambda}_{\Delta|W}. \tag{12}$$

This is exceedingly simple to implement: it only involves running two linear regressions, regressing $\widehat{m}(r;t)$ on W_r in the primary sample and Δ_r on W_r in the validation sample.

As we show in more detail in Appendix C.2, the bias-corrected estimator has desirable theoretical properties. Consider the case in which the researcher's dataset is a random sample of economically relevant text pieces, a fraction ρ_p of all text pieces enter into the primary sample, and a fraction ρ_v of all text pieces enter into the validation sample. As the number of economically relevant text pieces grows large, the bias-corrected estimator $\widehat{\beta}^{debiased}$ is consistent for the target regression coefficient β^*. Moreover, it can be shown that $\widehat{\beta}^{debiased}$ is asymptotically normal with limiting variance given by

$$\sigma_W^{-4}\left(\frac{1-\rho_p}{\rho_p}\sigma^2_{\widehat{V}W}+2\sigma_{\widehat{V}W}\sigma_{\Delta W}+\frac{1-\rho_v}{\rho_v}\sigma^2_{\Delta W}\right), \tag{13}$$

where σ_W is the standard deviation of the linked variable W_r across all text pieces, $\sigma_{\widehat{V}W}$ is the standard deviation of the product $\widehat{V}_r \times W_r$, and $\sigma_{\Delta W}$ is defined analogously. Consequently, the precision of the bias-corrected estimator depends on the relative size of the validation sample versus the primary sample as well as the variability of the LLM's label $\widehat{V}_r$ and measurement error Δ_r across text pieces.

Of course, if the researcher invests effort to collect existing measurements on a validation sample, a natural question arises: why bother with the possibly mismeasured LLM labels on the primary sample? The researcher could just directly estimate the target regression on the validation sample only and report $\widehat{\beta}^*$. To answer that question we can analyze the limiting behavior of the validation-sample only regression estimator: it is also asymptotically normal with limiting variance given by $\sigma_W^{-4}\frac{1-\rho_v}{\rho_v}\sigma^2_{VW}$. Comparing this expression with that for the bias-corrected regression, the latter is more precise if

$$\frac{1-\rho_p}{\rho_p}\sigma^2_{\widehat{V}W}+2\sigma_{\widehat{V}W}\sigma_{\Delta W}\leq\frac{1-\rho_v}{\rho_v}\left(\sigma^2_{VW}-\sigma^2_{\Delta W}\right). \tag{14}$$

Unsurprisingly, this comparison depends on the relative size of the validation sample versus the primary sample. But notice that it further depends on the relative standard deviation of the existing measurement V_r and the LLM's error Δ_r. Importantly, Equation (14) therefore implies that the bias-corrected regression coefficient can be more precisely estimated than the validation-sample-only regression coefficient if the LLM's labels are sufficiently accurate.[10]

Intuitively, in this exercise, the LLM's role is something like a missing-value imputer for the existing measurement on the primary sample. Consequently, provided the LLM is sufficiently accurate in a given application, the researcher can obtain a possible free-lunch: use the LLM to solve the text processing problem and still deliver precise estimates of the target regression coefficient. Importantly, the LLM outputs are not substitutes for an existing measurement process;

[10] This phenomenon has been documented in recent machine learning research, such as Angelopoulos et al. (2023) and associated work, that combines validation data with the outputs of machine learning models to estimate downstream parameters.

instead, they can serve to cheaply amplify a small sample of validation data.

Of course, to further emphasize, whether this occurs in practice will depend on the specific empirical application. We turn next to illustrating the performance of the bias-corrected regression coefficient in finite samples for our Congressional legislation example.

5.4.2 Monte Carlo Simulations based on Congressional Legislation

The Congressional legislation data described earlier is particularly well-suited to illustrate the performance of the bias-corrected regression because we observe existing measurements of each bill's policy topic. The Congressional Bills Project trained teams of human annotators to label the description of each Congressional bill r for its major policy topic area $V_r := f^*(r)$, describing whether the bill falls into one of twenty possible policy areas such as health or defense. This human labeling process was quite labor intensive. The Congressional Bills Project states that all annotators were trained for a full academic quarter before beginning this task (Jones et al., 2023). Given the widespread use of this dataset, researchers are comfortable plugging in these measurements into their downstream analysis; it is therefore natural to ask: can an LLM automate this measurement procedure $f^*(\cdot)$?

We start off by reporting a surprising twist on the results reported in the previous section, where we found that plugging in LLM labels for each bill's policy topic as the dependent variable in a regression led to substantial biases and variation across possible choices of LLM and prompting strategy. The surprising twist is that we observe this substantial variability in downstream parameter estimates even though each LLM and prompt leads to a label that is nearly equally accurate for the existing label V_r on average.

Figure 6 plots the accuracy of the LLM label $\widehat{V}_r^{m,p}$ for the existing label V_r across all combinations of LLMs m and prompting strategies p. While GPT-4o tends to be a bit more accurate than GPT-3.5-Turbo, for a given LLM, accuracy is remarkably invariant to prompting strategy. Yet we have seen that for a given LLM the downstream parameter estimates in a regression using these labels varies enormously across different prompting strategies.

To what extent can collecting a small amount of validation data address this problem? To answer that question we carry out a Monte Carlo simulation exercise to examine how well the plug-in regression and bias-corrected regression recover the target coefficient β^*.

More specifically, for a given policy topic V_r (e.g., health, defense, etc.), linked covariate W_r (e.g., whether the bill's sponsor was a Democrat, etc.) and pair of large language model m and prompting strategy p, we randomly sample 5,000 bills from our dataset of 10,000 bills. On this random sample of 5,000 bills, we first calculate the plug-in coefficient $\widehat{\beta}$ from regressing $\widehat{V}_r^{m,p}$ on the linked variable W_r. We next randomly reveal the existing label V_r on 250 (i.e. 5%) of our random sample of 5,000 bills, which produces a validation sample. We then calculate the bias-corrected coefficient $\widehat{\beta}^{debiased}$. We repeat these steps for 1,000 randomly sampled datasets. Across each simulation, we calculate the average bias of the plug-in coefficient and the bias-corrected coefficient for the target regression β^* associated with regressing the existing label V_r on the linked variable W_r on all 10,000 bills. We repeat this exercise for each possible combination of bill topic V_r, linked covariate W_r, LLM m (either GPT-3.5-Turbo or GPT-4o) and prompting strategy p. This allows us to summarize how the plug-in regression performs against the bias-corrected regression across a wide variety of possible regression specifications and choices of LLM and prompting strategy. In Appendix D.1, we report results from Monte Carlo simulations that vary the size of the validation sample and we find the similar results even when the validation sample contains as few as 125 bills.

Figure 7 and the top panel of Table 4 compares the average bias of the plug-in coefficient $\widehat{\beta}$ and the bias corrected coefficient $\widehat{\beta}^{debiased}$ for the target regression β^* (normalized by their standard deviations) across possible combinations of bill topic V_r, linked covariate W_r, LLM m, and prompting strategy p. For most regression specifications and pairs of LLM and prompting strategy, the simple plug-in regression suffers from substantial biases for the target regression. Indeed, focusing on GPT-3.5-Turbo, the 5th percentile yields a bias that is fully -1.90 standard deviations away from the target regression coefficient and the 95th percentile is fully 2.21 standard deviations off. By contrast, using the validation sample to debias the LLM labels yields estimates that are reliably unbiased for the target regression — indeed, the bias-corrected regression coefficient performs remarkably well across all regression specifications and pairs of LLM and prompting strategy.

We have shown above that measurement error in the LLM labels leads not only to biased point estimates, but correcting using a validation sample fixes this problem. Does this extend to our inference statements? For each regression specification and pair of LLM and prompting strategy, we further calculate the fraction of simulations in which a 95% confidence interval centered at either the plug-in coefficient or the bias-corrected coefficient includes the target regression β^*. The bottom panel of Table 4 summarizes the distribution of coverage results across all combinations of regression specification and pair of LLM and prompting strategy. For GPT-3.5-Turbo (left panel, bottom), the median coverage obtained from plugging in the language model's label includes the true parameter 82% of the time; the 5^{th} percentile confidence interval includes the true parameter 38% of the time. In contrast, the nominal 95% confidence interval for the bias-corrected regression coefficient has approximately correct coverage across all regression specifications and pairs of LLM and prompting strategy. The right-hand panel shows qualitatively similar results with GPT-4o as well.

We can use these data to also answer the question: If we have already gone to the trouble of collecting measurements V_r in a validation sample, what is the additional value of the LLM labels in the primary sample? We could now in principle just directly run the target regression on the validation sample.

Figure 8 illustrates what could be gained by using the LLM on the primary sample. For each regression specification and pair of LLM and prompting strategy, we calculate the mean square error of the bias-corrected coefficient and the validation-sample only coefficient for the target regression β^*. We then calculate the average mean square error across all 1000 simulations for each regression specification and pair of LLM and prompting strategy, and Figure 8 plots the resulting distribution across all choices of bill topic V_r, covariate W_r, and pair of model-and-prompting strategy.

In this empirical setting, we find that the average mean square error of the validation-sample-only coefficient is always higher (less precisely estimated) compared to the bias-corrected coefficient. This suggests that there is information extracted from the LLM's labels on the primary sample, and further illustrates that precision improvements, as predicted by the asymptotic comparison in Equation (14), are possible in finite samples. In Appendix D.1, we show that these precision gains are present even when the validation sample contains on 125 bills and dissipate once the validation sample is around 25% of the researcher's sample. The share of applications in which it would actually be feasible (from a time and cost perspective) to collect gold-standard labels for more than a quarter of the sample is an open question. We suspect that for many applications, a combination of LLM labels and a small amount of validation data is likely to be the optimal approach.

These results focused on using the LLM's labels for the economic concept V_r as the dependent variable in a linear regression. As we show in Appendix D.2, we find similar results for the case in which the economic concept is an explanatory variable in a regression. In particular, we continue to find that plug-in regression with LLM labels leads to substantial biases and coverage distortions; by contrast, bias-correcting using a small validation sample effectively eliminates bias and restores the coverage of conventional confidence intervals. For many regression specifications and pair of LLM and prompting strategy, we again find that the bias-corrected coefficient improves on the mean-square error of the validation-sample-only coefficient.

Taken together, these results indicate that estimation is a promising use case for LLMs, provided the researcher collects benchmark data to debias the resulting estimates. When done correctly, LLMs can then lower the cost of data collection and in some cases improve statistical precisions, while preserving the familiar econometric guarantees we desire.

5.4.3 Estimation with LLMs in the Absence of Benchmark Data

This productive use of LLMs in estimation problems requires the researcher to collect benchmark data using some existing measurement procedure $f^*(\cdot)$. But what if benchmark data cannot be collected? Our earlier results highlight that the LLM is a general-purpose technology for estimation if and only if its outputs have no errors for reproducing the economic concept. Consequently, in the absence of benchmark data, a researcher using LLM outputs in an estimation problem faces three choices, each of which seem difficult to defend in empirical applications at present.

First, in a particular application, a researcher could attempt to justify using the LLM's outputs by directly arguing that there are no errors for the economic concept. However, even the most generous evaluations of LLM performance in computer science do not find perfect performance – for example, GPT-4o achieved 86.4% performance on MMLU (OpenAI, 2023). A researcher proceeding down this path must somehow argue why an LLM's output exactly reproduces the economic concept even though it is imperfect on popular benchmarks and why the growing body of evidence documenting the brittleness of LLMs does not apply to their specific application.

Second, the researcher might acknowledge that there are errors in the LLM's outputs and instead write down a statistical model of the LLM's errors, just as we would in the measurement error literature. While tempting in its familiarity, this exercise seems heroic when applied to LLMs. Consider the following simple, stylized model: for a given LLM m, its labels across prompting strategies p, denoted $\widehat{V}_r^{m,p}$, are on average equal to the economic concept V_r and their errors $\Delta_r^{m,p} = V_r - \widehat{V}_r^{m,p}$ are independent. Such an assumption about LLM errors is easy to state and it would suggest particular solutions: perhaps the researcher can average across prompting strategies or the researcher could use one prompting strategy as an instrument for another. But when we take this assumption to ground, the cracks are apparent. Again, no LLM achieves perfect performance on benchmark evaluations; so on what grounds are we confident that the average of prompting strategies is correct or unbiased for the economic concept? Furthermore, for a given LLM m, its labels across prompting strategies p are surely correlated with one another—indeed, the training algorithm was applied to the same underlying training dataset and prompt engineering only influences the text generator. Even worse, across language models m, there is surely substantial overlap in training datasets—we often heuristically describe LLMs as being trained on all of the internet and there is only one internet after all. On what basis is it reasonable to act as-if these labels are independent or satisfy a weak dependence condition needed to invoke a law of large numbers across prompts? Taking a step back, a researcher proceeding down this path is effectively asserting that we can correctly model how the outputs of an algorithm we do not understand relate to a quantity that we do not observe.

Finally, an even more radical choice is to argue that there is no existing measurement $f^*(\cdot)$ that we would like to automate, and instead assert that the LLM's outputs themselves $\widehat{V}_r$ are the quantity of interest. This choice restores the validity of plug-in estimation by assumption since the plug-in estimate is now the researcher's target. However, it is difficult to think of empirical applications in which we would be satisfied with this choice since it radically changes the underlying question. Rather than studying how partisanship relates to the policy topics of legislation, we would instead be studying how partisanship relates to, for example, the labels of GPT-4o prompted with a "helpful research assistant" persona. When the computer science community inevitably generates a new prompt engineering strategy, should we publish a new paper examining how partisanship relates to the resulting new labels? When OpenAI inevitably releases GPT-5, should we revisit all published work using labels produced by GPT-4o? These are challenging but unavoidable questions that require answers if a researcher proceeds down this path.

Taken together, in the absence of validation data, each of these available choices seems difficult to defend. By contrast, whenever validation data can be collected, LLMs can be powerful tools for automating existing measurements that are costly to scale across large collections of text.

5.4.4 Replicable Estimation with LLMs

Finally, this discussion also clarifies an important practical question: what does replicable estimation mean when using LLM outputs? What should a researcher using LLM outputs in an estimation problem include in their replication files and what quantities should we hope to reproduce? It is useful to consider what we typically include in more familiar situations — for example, using the CPS to estimate the causal effect of some state-level policy change. In this case, the researcher would post code that implements all steps to go from the original CPS data to their reported estimates. With these files in hand, we often have two goals in mind when evaluating the replicability of the researcher's work.

Our first goal is often "mechanical" replication: could a third party re-execute the researcher's provided code to go from the underlying data to the reported estimate? In the case of estimation using LLM outputs, what is needed for mechanical replication is familiar: the researcher should first include their generated dataset containing the LLM outputs $\widehat{m}(r;t)$ and the linked economic variables W_r (i.e., the equivalent of providing the underlying CPS data), and code to produce their estimates from that data.

Of course, the generated dataset containing the LLM outputs was produced through the researcher's specific choice of LLM and prompting strategies. Additionally the researcher should additionally document the choice of model and the specific prompts so that the resulting LLM outputs themselves can be mechanically replicated. This additional form of mechanical replication poses a challenge to using closed-source LLMs like the GPT family from OpenAI or the Claude family from Anthropic (in addition to the concerns about training leakage we already highlighted), since these models may be continually fine-tuned or tweaked over times. That means the same prompt run on the same model at different times can deliver different outputs, as shown in Barrie, Palmer and Spirling (2024).[11] Consequently, mechanical replication of the LLM outputs themselves requires the researcher to use open-source LLMs with fixed and published weights.

In addition to mechanical replication, a second goal we have in evaluating the CPS researcher's work is what we might call "conceptual replication:" we would like to let third parties assess how alternative choices in defining the estimation sample may affect the resulting estimates. In carrying forward this analogy to using LLM outputs, there appears to be novel dimensions along which to conduct conceptual replications: how sensitive are the researcher's estimates to alternative choices

[11] Barrie, Palmer and Spirling (2024) provide many other valuable practical recommendations towards the mechanical replicability of research involving LLM outputs.

of prompting strategies (e.g., persona modifications, chain-of-thought modifications, or few-shot examples) and alternative choices of LLMs (e.g., the GPT family versus the Claude family versus the LLama family)? Given the pace of development of new models and the breadth of possible prompting strategies, the garden of forking paths now appears to be endless.

However, provided the researcher collects benchmark data and debiases whatever LLM output they collect, our results above imply that alternative choices of prompting strategy and LLM target the *same* empirical quantity: the parameter of interest θ^* defined using the researcher's existing measurement process. Consequently, suppose a replicator tries an alternative prompting strategy or revisits an analysis after GPT-5 is released. By applying the researcher's code that debiases whatever new LLM output they produce, their resulting confidence interval should include the same target θ^* with high probability. Since the researcher's original confidence interval also covers θ^* with high probability, we should likely see agreement between the replicator's inferences drawn using a new prompt or new model and the researcher's original conclusions. In other words, for conceptual replication, it is essential for the researcher to provide benchmark data. In this case, the researcher must defend the existing measurement process used to produce the benchmark data, rather than defend the specific choice of LLM and prompting strategy used to cheaply amplify that data.

To summarize, our framework suggests that empirical researchers using LLM outputs for estimation should include: (i) the benchmark data they collected, (ii) the specific prompt and open-source model used to generate LLM outputs; and (iii) the familiar final data and code used to produce their reported estimates. With this in hand, replicators can verify both the mechanical replication of the reported estimates and the LLM outputs, and importantly engage in conceptual replications that explore variations in not just the prompts but also alternative models.

6 Extensions to Novel Uses of LLMs

So far we have applied our framework to familiar empirical problems related to prediction and estimation. But LLMs are increasingly being used in ways entirely new to economics research, such as for automated hypothesis generation, or to simulate the responses of human subjects. In this section, we show that our framework is broad enough to cover even novel LLM uses like these, partly because these new uses can often be viewed as variants of prediction and estimation problems.

6.1 LLMs as Hypothesis Generators

In Section 4, we considered whether LLM outputs can be used for prediction — the task at the heart of supervised learning. By reinterpreting its components, our framework also gives us a way to understand recent work using LLMs as tools for hypothesis generation. For example, Batista and Ross (2024) use LLMs to generate hypotheses about which headlines will lead to more user engagement (see also Zhou et al., 2024). Han (2024) prompts LLMs to generate hypotheses about candidate instrumental variables given a text description of various empirical settings. In computer science, Si, Yang and Hashimoto (2024) prompt LLMs to generate new research ideas in natural language processing.[12]

In this domain, the text pieces $r \in \mathcal{R}$ are the prompts for the hypothesis generation task, such as the candidate headlines or the description of the empirical setting, and $\widehat{m}(r;t) = \widehat{Y}_r$ is the LLM's returned hypothesis, such as its explanation for what drives user engagement or its suggested instrumental variable. The central difference is that there is no observed economic variable Y_r that $\widehat{Y}_r$ seeks to reproduce. Instead, in each hypothesis generation exercise, the researcher (sometimes implicitly) provides a scoring rule to assess the quality of the resulting hypothesis $\ell(\widehat{m}(r;t))$ and the researcher then reports the average quality of the generated hypotheses, $\frac{1}{N}\sum_{r\in\mathcal{R}} D_r \ell(\widehat{m}(r;t))$.

Cast in this light, using LLMs as tools for hypothesis generation bears striking resemblance to our analysis of using LLMs as tools for prediction. In assessing whether LLMs are good tools for hypothesis generation, we would like to generalize their quality in a particular research context $Q(\cdot)$ (some setting in which we would like to generate hypotheses); more precisely, make inferences about $\mathbb{E}_Q[\sum_{r\in\mathcal{R}} D_r \ell(\widehat{m}(r;t)]$.

Our previous results then immediately imply that the needed guarantee is that the LLM satisfies *no training leakage* (Proposition 1). For hypothesis generation, no training leakage now has an interesting interpretation: do we believe that the LLM has already seen this particular prompt or setting for hypothesis generation? If so, we would naturally be concerned that the LLM is merely mimicking hypotheses it has seen before – the quality of its hypotheses on the provided task r does not generalize to its performance on novel tasks. For example, in evaluating whether an LLM can generate instrumental variable strategies, we might worry that it has likely been trained on text pieces describing (say) distance to the nearest college as a useful instrument for measuring the returns to education. In this case, observing that the LLM reproduces these existing IV strategies may not provide us with good guidance about its performance in a novel, previously-unseen empirical setting. As before, using open source LLMs with documented training data and public weights can address the threat of no training leakage for hypothesis generation.

6.2 LLMs as Human Subject Simulators

A growing body of research argues that LLMs can be used to simulate the responses of human subjects in experiments and surveys — producing what are sometimes referred to as "in-silico" subjects. For example, Horton (2023) argues that LLMs are "implicit computational models of humans" and therefore "provide responses similar to what we might expect from a human" (pg. 1). Other studies have used LLMs to simulate human subjects in economics experiments (Manning, Zhu and Horton,

[12] Within the economics literature, discussion about how artificial intelligence more generally can be used for hypothesis generation and scientific discovery can be found in Fudenberg and Liang (2019), Ludwig and Mullainathan (2024), Agrawal, McHale and Oettl (2024) and Mullainathan and Rambachan (2024).

2024; Mei et al., 2024), in marketing (Brand, Israeli and Ngwe, 2023), in finance (Bybee, 2024), in political science (Argyle et al., 2023), and in computer science (Aher, Arriaga and Kalai, 2023).

Using LLMs as in-silico subjects can be mapped into our framework by reinterpreting the text pieces $r \in \mathcal{R}$ and the measurement process defining the economic concept $V_r := f^*(r)$. Each text piece r is now a particular experimental design or survey instrument and the economic concept V_r is the average or modal response of a human subject.[13] If the researcher were to collect human subject responses $V_r := f^*(r)$ on a collection of experimental designs (i.e., $r \in \mathcal{R}$ such that $D_r = 1$), she would be satisfied to calculate possible downstream parameter estimates (Equation 4).

While we could in principle collect responses on any experimental design or survey instrument, it would be costly to do so. Since LLMs could be prompted with the same experimental design or survey instrument, the researcher may instead collect the LLM's response $\widehat{m}(r;t) = \widehat{V}_r$. Viewing the LLM as a computational model of human behavior, the researcher reports the plug-in parameter estimate (Equation 5) using the LLM's responses.

To make this concrete, consider two categories of choice experiments in behavioral economics, both aiming to understand how well existing models describe the risky choices made by people.

Example: Testing anomalies for theories of risky choice In order to probe particular systematic violations of a model of risky choice, a long tradition in behavioral economics constructs "anomalies," small collections of menus of lotteries that highlight flaws in an existing model, such as the Allais Paradox (Allais, 1953) or the original Kahneman-Tversky choice experiments (Kahneman and Tversky, 1979). Researchers recruit human subjects to make choices V_r on these specific anomalies r, and researchers then calculate the extent to which these choices violate, for example, expected utility theory (e.g., Harless and Camerer, 1994). Could an LLM instead simulate human choices $\widehat{m}(r;t)$ on new anomalies we have yet to construct? ▲

Example: Large-scale risky choice experiments In order to compare alternative models of risky choice, recent work encourages behavioral economists to measure the predictive accuracy of alternative models of risky choice on a diverse collection of lottery choice problems (Erev et al., 2017; Fudenberg et al., 2022). In response to this challenge, Peterson et al. (2021) recruited nearly 15,000 respondents on Amazon Mechanical Turk (MTurk) to make over one million choices V_r from menus of risky lotteries r, producing the "Choices13K" dataset. This impressive feat of data collection was quite costly both in terms of effort and dollars. Could an LLM instead simulate the choices of these MTurk workers $\widehat{m}(r;t)$? ▲

Viewing in-silico studies through the lens of our estimation framework then implies that re-

[13] Of course, we could instead interpret each text piece as being associated with some random variable that summarizes the distribution of possible human subject responses on any experimental design or survey instrument. Our discussion continues to apply at the expense of more cumbersome notation.

searchers need the condition that there is no measurement error in the LLM's outputs (Proposition 2). It implies that the LLM precisely reproduces the behavior of the target pool of human subjects on the researcher's collection of experimental designs or survey questions.

Is this assumption credible? There are studies showing LLMs can seemingly reproduce results in published experiments or surveys. However, just as with benchmark evaluations of LLMs, for every such success, there seems to be a counterexample — LLM responses to psychology experiments appear to produce *more* falsely significant findings than human subjects (Cui, Li and Zhou, 2024), LLM responses cannot accurately reproduce the responses of human subjects on opinion polls (Santurkar et al., 2023; Boelaert et al., 2024), and LLM responses on economic reasoning tasks can be sensitive to prompt engineering (Raman et al., 2024). Beyond this evidence of brittleness in the LLM's performance, there is also the second requirement for using an LLM for estimation problems: no training leakage. The results from existing published experiments almost certainly enter into the LLM's training corpus. But presumably what we are interested in is not so much whether an LLM can memorize and regurgitate some existing result it has already seen so much as simulate human behavior in response to entirely new experiment and survey designs.

Fortunately viewing human subject simulation as an estimation problem also implies a practical fix for researchers. Just as we could collect a small validation sample to de-bias LLM labels for "standard" estimation problems (e.g., Does the ideology of a Congressional bill's sponsor help explain the bill's policy topic?), the same classic solution can be applied to the use of LLMs for simulating human responses. One implication is that we simply *cannot* avoid running our experiment or survey on at least some sample of real human subjects.

Notice that the value of the LLM itself will then depend on the nature of the research application. Some economics experiments are about testing anomalies, or even a single anomaly (for example the Allais Paradox). In that case the research context $Q(\cdot)$ might contain as few as one design r, and the economic concept V_r we wish the LLM to be able to reproduce is the minimum number of human subject responses we think we would need for a reliable answer to "how do people tend to choose given this menu of options r?" The only way we could model the LLM's measurement error in approximating that quantity would be to collect enough human subject responses to calculate the target quantity V_r on that very design. But then why use the LLM at all?

As a consequence, LLMs as human subject simulators are likely to only add value to experimental studies where the research context contains a large number of designs, r. In that setting, collecting human subject responses V_r on a small subset of designs r allows us to model the LLM's error Δ_r and correct the LLM outputs on the remaining designs. For example, we could collect LLM responses on all menus in the Choices13K dataset and only collect human subject responses on a small validation sample, drastically reducing the cost of data collection. Once again, in-silico subjects can only serve to amplify, rather than fully replace, human subjects in such experimental studies.

7 Conclusion

Machine learning tools are radically expanding the scope of empirical research in economics. We now move beyond estimating average causal effects to learning personalized treatment effects (e.g., Athey and Imbens, 2017; Wager and Athey, 2018). We use unstructured data, such as satellite images, to infer outcomes at high-frequencies and granular scales (e.g., Donaldson and Storeygard, 2016; Rambachan, Singh and Viviano, 2024). We tackle prediction policy problems (Kleinberg et al., 2015, 2018) and develop tools for hypothesis generation (Fudenberg and Liang, 2019; Ludwig and Mullainathan, 2024; Mullainathan and Rambachan, 2024). In this context, LLMs are the latest machine learning tools to enter our empirical toolkit.

While powerful and easy to use, LLMs are also brittle, which has consequences for empirical research using LLM outputs. As we showed, naive uses of LLM outputs distort both downstream predictions and parameter estimates. To understand this problem and identify constructive solutions, we have developed here an econometric framework for studying LLMs. Rather than making assumptions about how they are designed, we treat LLMs as black boxes and provided conditions that the LLM's outputs must satisfy in order to be used in prediction and estimation problems. Altogether our results suggest that the excitement around the empirical uses of LLMs is warranted, provided researchers guard against training leakage by using open-source LLMs in prediction problems and collect benchmark data in estimation problems.

The need for such econometric contracts pervades every aspect of our efforts to incorporate machine learning and other modern computing tools into economics research. While such contracts are familiar to empirical researchers, their development will need to be done differently for this new world of machine learning. Traditionally, econometrics provided contracts for empirical tools that we designed ourselves. Today, economists increasingly sit on the *consumer* side of new AI models, and we are blind to most of what happens in their production. As we have argued in this paper (using LLMs as an example), it is nonetheless still possible to write meaningful contracts for AI tools that enable economists to use these tools to push the empirical frontier rigorously forward.

References

Abadie, Alberto, Susan Athey, Guido W. Imbens, and Jeffrey M. Wooldridge. 2020. "Sampling-Based versus Design-Based Uncertainty in Regression Analysis." *Econometrica*, 88(1): 265–296.

Adler, E Scott, and John Wilkerson. 2020. *Congressional Bills Project, NSF 00880066 and 00880061.* http://congressionalbills.org/download.html (accessed July 5, 2024).

Aenlle, Miguel. 2020. *Daily Financial News for 6000+ Stocks.* https://www.kaggle.com/datasets/miguelaenlle/massive-stock-news-analysis-db-for-nlpbacktests (accessed August 1, 2024).

Agrawal, Ajay, John McHale, and Alexander Oettl. 2024. "Artificial intelligence and scientific discovery: A model of prioritized search." *Research Policy*, 53(5): 104989.

Aher, Gati, Rosa I. Arriaga, and Adam Tauman Kalai. 2023. "Using large language models to simulate multiple humans and replicate human subject studies." *ICML'23*. JMLR.

Allais, Maurice. 1953. "Le comportement de l'homme rationnel devant le risque: critique des postulats et axiomes de l'école américaine." *Econometrica: journal of the Econometric Society*, 503–546.

Angelopoulos, Anastasios N., Stephen Bates, Clara Fannjiang, Michael I. Jordan, and Tijana Zrnic. 2023. "Prediction-powered inference." *Science*, 382(6671): 669–674.

Argyle, Lisa P., Ethan C. Busby, Nancy Fulda, Joshua R. Gubler, Christopher Rytting, and David Wingate. 2023. "Out of One, Many: Using Language Models to Simulate Human Samples." *Political Analysis*, 31(3): 337–351.

Ash, Elliott, and Stephen Hansen. 2023. "Text Algorithms in Economics." *Annual Review of Economics*, 15: 659–688.

Athey, Susan. 2018. "The impact of machine learning on economics." *The economics of artificial intelligence: An agenda*, 507–547.

Athey, Susan, and Guido W. Imbens. 2017. "The econometrics of randomized experiments." In *Handbook of economic field experiments*. Vol. 1, 73–140. Elsevier.

Baker, Scott R., Nicholas Bloom, and Steven J. Davis. 2016. "Measuring Economic Policy Uncertainty*." *The Quarterly Journal of Economics*, 131(4): 1593–1636.

Balloccu, Simone, Patrícia Schmidtová, Mateusz Lango, and Ondrej Dusek. 2024. "Leak, Cheat, Repeat: Data Contamination and Evaluation Malpractices in Closed-Source LLMs." 67–93. Association for Computational Linguistics.

Barrie, Christopher, Alexis Palmer, and Arthur Spirling. 2024. "Replication for Language Models Problems, Principles, and Best Practice for Political Science." https://arthurspirling.org/documents/BarriePalmerSpirling_TrustMeBro.pdf.

Batista, Rafael, and James Ross. 2024. “Words that Work: Using Language to Generate Hypotheses.” *Available at SSRN.*

Battaglia, Laura, Timothy Christensen, Stephen Hansen, and Szymon Sacher. 2024. “Inference for Regression with Variables Generated by AI or Machine Learning.”

Berglund, Lukas, Meg Tong, Max Kaufmann, Mikita Balesni, Asa Cooper Stickland, Tomasz Korbak, and Owain Evans. 2023. “The Reversal Curse: LLMs trained on ”A is B” fail to learn ”B is A”.” *arXiv preprint arXiv:2309.12288.*

Beta Suite by WRDS. 2024. Provided by Wharton Research Data Services. https://wrds-www.wharton.upenn.edu/pages/grid-items/beta-suite-wrds (accessed August 1, 2024).

Biemer, Paul P, Robert M Groves, Lars E Lyberg, Nancy A Mathiowetz, and Seymour Sudman. 2013. *Measurement errors in surveys.* Vol. 548, John Wiley & Sons.

Boelaert, Julien, Samuel Coavoux, Etienne Ollion, Ivaylo D Petev, and Patrick Präg. 2024. “How do Generative Language Models Answer Opinion Polls?” https://osf.io/preprints/socarxiv/r2pnb.

Bohannon, Molly. 2023. “Lawyer Used ChatGPT In Court—And Cited Fake Cases. A Judge Is Considering Sanctions.” *Forbes*, June 8. https://www.forbes.com/sites/mollybohannon/2023/06/08/lawyer-used-chatgpt-in-court-and-cited-fake-cases-a-judge-is-considering-sanctions (accessed December 5, 2024).

Bound, John, and Alan B. Krueger. 1991. “The Extent of Measurement Error in Longitudinal Earnings Data: Do Two Wrongs Make a Right?” *Journal of Labor Economics*, 9(1): 1–24.

Bound, John, Charles Brown, and Nancy Mathiowetz. 2001. “Chapter 59 - Measurement Error in Survey Data.” In *Handbook of Econometrics.* Vol. 5, , ed. James J. Heckman and Edward Leamer, 3705–3843. Elsevier.

Bound, John, Charles Brown, Greg J. Duncan, and Willard L. Rodgers. 1994. “Evidence on the Validity of Cross-Sectional and Longitudinal Labor Market Data.” *Journal of Labor Economics*, 12(3): 345–368.

Brand, James, Ayelet Israeli, and Donald Ngwe. 2023. “Using LLMs for Market Research.” *Harvard Business School Marketing Unit Working Paper No. 23-062, Available at SSRN.*

Bubeck, Sébastien, Varun Chandrasekaran, Ronen Eldan, Johannes Gehrke, Eric Horvitz, Ece Kamar, Peter Lee, et al. 2023. “Sparks of Artificial General Intelligence: Early experiments with GPT-4.” *arXiv preprint arXiv:2303.12712.*

Bybee, Leland. 2024. “The Ghost in the Machine: Generating Beliefs with Large Language Models.” https://lelandbybee.com/files/LLM.pdf.

Chen, Banghao, Zhaofeng Zhang, Nicolas Langrené, and Shengxin Zhu. 2024. “Unleashing the potential of prompt engineering in Large Language Models: a comprehensive review.” *arXiv preprint arXiv:2310.14735.*

Cheng, Jeffrey, Marc Marone, Orion Weller, Dawn Lawrie, Daniel Khashabi, and Benjamin Van Durme. 2024. “Dated Data: Tracing Knowledge Cutoffs in Large Language Models.” *arXiv preprint arXiv:2403.12958.*

Chen, Xiaohong, Han Hong, and Elie Tamer. 2005. “Measurement error models with auxiliary data.” *The Review of Economic Studies*, 72(2): 343–366.

Cui, Ziyan, Ning Li, and Huaikang Zhou. 2024. “Can AI Replace Human Subjects? A Large-Scale Replication of Psychological Experiments with LLMs.” *arXiv preprint arXiv:2409.00128.*

Dahl, Matthew, Varun Magesh, Mirac Suzgun, and Daniel E Ho. 2024. “Large Legal Fictions: Profiling Legal Hallucinations in Large Language Models.” *Journal of Legal Analysis*, 16(1): 64–93.

Dell’Acqua, Fabrizio, Edward McFowland III, Ethan R. Mollick, Hila Lifshitz-Assaf, Katherine Kellogg, Saran Rajendran, Lisa Krayer, François Candelon, and Karim R. Lakhani. 2023. “Navigating the Jagged Technological Frontier: Field Experimental Evidence of the Effects of AI on Knowledge Worker Productivity and Quality.” *Harvard Business School Technology & Operations Mgt. Unit Working Paper No. 24-013, The Wharton School Research Paper, Available at SSRN.*

Dell, Melissa. 2024. “Deep Learning for Economists.” *arXiv preprint arXiv:2407.15339.*

Donaldson, Dave, and Adam Storeygard. 2016. “The View from Above: Applications of Satellite Data in Economics.” *Journal of Economic Perspectives*, 30(4): 171–98.

Donoho, David. 2024. “Data Science at the Singularity.” *Harvard Data Science Review*, 6(1).

Dreyfuss, Bnaya, and Raphael Raux. 2024. “Human Learning about AI Performance.” *arXiv preprint arXiv:2406.05408.*

Dubey, Abhimanyu, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, et al. 2024. “The Llama 3 Herd of Models.” *arXiv preprint arXiv:2407.21783.*

Durvasula, Maya M., Sabri Eyuboglu, and David M. Ritzwoller. 2024. “Distilling Data from Large Language Models: An Application to Research Productivity Measurement.” *arXiv preprint arXiv:2405.08030.*

Egami, Naoki, Musashi Hinck, Brandon M. Stewart, and Hanying Wei. 2024. “Using imperfect surrogates for downstream inference: design-based supervised learning for social science applications of large language models.” *NIPS ’23.* Curran Associates Inc.

Erev, Ido, Ert Eyal, Ori Plonsky, Doron Cohen, and Oded Cohen. 2017. “From anomalies to forecasts: Toward a descriptive model of decisions under risk, under ambiguity, and from experience.” *Psychological Review*, 124(4): 369–409.

Fraser, Colin (`@colin_fraser`). 2024*a*. “Claude still can’t solve the impossible one farmer one sheep one boat problem `https://pbs.twimg.com/media/GQiieRPXwAAjiqB`.” *X*, June 20. https://x.com/colin_fraser/status/1803870308908048695 (accessed December 5, 2024).

Fraser, Colin (@colin_fraser). 2024*b*. "It's dumb :(https://pbs.twimg.com/media/GXTdblvbgAERvSF." *X*, September 12. https://x.com/colin_fraser/status/1834334418007457897 (accessed December 5, 2024).

Fudenberg, Drew, and Annie Liang. 2019. "Predicting and understanding initial play." *American Economic Review*, 109(12): 4112–4141.

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

Gentzkow, Matthew, Bryan Kelly, and Matt Taddy. 2019. "Text as data." *Journal of Economic Literature*, 57(3): 535–574.

Glasserman, Paul, and Caden Lin. 2023. "Assessing Look-Ahead Bias in Stock Return Predictions Generated By GPT Sentiment Analysis." *arXiv preprint arXiv:2309.17322*.

Golchin, Shahriar, and Mihai Surdeanu. 2024. "Time Travel in LLMs: Tracing Data Contamination in Large Language Models." *arXiv preprint arXiv:2308.08493*.

Hansen, Anne Lundgaard, and Sophia Kazinnik. 2024. "Can ChatGPT Decipher Fedspeak?" *Available at SSRN*.

Hansen, Stephen, Peter John Lambert, Nicholas Bloom, Steven J Davis, Raffaella Sadun, and Bledi Taska. 2023. "Remote Work across Jobs, Companies, and Space." National Bureau of Economic Research Working Paper 31007.

Han, Sukjin. 2024. "Mining Causality: AI-Assisted Search for Instrumental Variables." *arXiv preprint arXiv:2409.14202*.

Harless, David W., and Colin F. Camerer. 1994. "The Predictive Utility of Generalized Expected Utility Theories." *Econometrica*, 62(6): 1251–1289.

Hendrycks, Dan, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2020. "Measuring massive multitask language understanding." *arXiv preprint arXiv:2009.03300*.

Horton, John J. 2023. "Large Language Models as Simulated Economic Agents: What Can We Learn from Homo Silicus?" *arXiv preprint arXiv:2301.07543*.

Jacovi, Alon, Avi Caciularu, Omer Goldman, and Yoav Goldberg. 2023. "Stop Uploading Test Data in Plain Text: Practical Strategies for Mitigating Data Contamination by Evaluation Benchmarks." 5075–5084. Association for Computational Linguistics.

Jones, Bryan D., Frank R. Baumgartner, Sean M. Theriault, Derek A. Epp, Cheyenne Lee, and Miranda E. Sullivan. 2023. "Policy Agendas Project: Codebook."

Kahneman, Daniel, and Amos Tversky. 1979. "Prospect theory: An analysis of decision under risk." *Econometrica*, 47(2): 363–391.

Kahneman, Daniel, Olivier Sibony, and Cass R Sunstein. 2021. *Noise: A flaw in human judgment.* Hachette UK.

Kalai, Adam Tauman, and Santosh S. Vempala. 2024. "Calibrated Language Models Must Hallucinate." *STOC 2024*, 160–171. Association for Computing Machinery.

Kleinberg, Jon, Himabindu Lakkaraju, Jure Leskovec, Jens Ludwig, and Sendhil Mullainathan. 2018. "Human decisions and machine predictions." *The quarterly journal of economics*, 133(1): 237–293.

Kleinberg, Jon, Jens Ludwig, Sendhil Mullainathan, and Ziad Obermeyer. 2015. "Prediction policy problems." *American Economic Review*, 105(5): 491–495.

Korinek, Anton. 2023. "Generative AI for economic research: Use cases and implications for economists." *Journal of Economic Literature*, 61(4): 1281–1317.

Korinek, Anton. 2024. "Generative AI for Economic Research: LLMs Learn to Collaborate and Reason." National Bureau of Economic Research Working Paper 33198.

Lee, Lung-fei, and Jungsywan H. Sepanski. 1995. "Estimation of Linear and Nonlinear Errors-in-Variables Models Using Validation Data." *Journal of the American Statistical Association*, 90(429): 130–140.

Lewis, Martha, and Melanie Mitchell. 2024. "Using Counterfactual Tasks to Evaluate the Generality of Analogical Reasoning in Large Language Models." *arXiv preprint arXiv:2402.08955*.

Liu, Pengfei, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. 2023. "Pre-train, Prompt, and Predict: A Systematic Survey of Prompting Methods in Natural Language Processing." *ACM Comput. Surv.*, 55(9).

Li, Xinran, and Peng Ding. 2017. "General Forms of Finite Population Central Limit Theorems with Applications to Causal Inference." *Journal of the American Statistical Association*, 112(520): 1759–1769.

LM Contamination Index. 2024. https://hitz-zentroa.github.io/lm-contamination (accessed December 5, 2024).

Lopez-Lira, Alejandro, and Yuehua Tang. 2024. "Can ChatGPT Forecast Stock Price Movements? Return Predictability and Large Language Models." *arXiv preprint arXiv:2304.07619*.

Ludwig, Jens, and Sendhil Mullainathan. 2024. "Machine learning as a tool for hypothesis generation." *The Quarterly Journal of Economics*, 139(2): 751–827.

Manning, Benjamin S., Kehang Zhu, and John J. Horton. 2024. "Automated Social Science: Language Models as Scientist and Subjects." *arXiv preprint arXiv:2404.11794*.

McCoy, R. Thomas, Shunyu Yao, Dan Friedman, Mathew D. Hardy, and Thomas L. Griffiths. 2024. "Embers of Autoregression: Understanding Large Language Models Through the Problem They are Trained to Solve." *Proceedings of the National Academy of Sciences*, 121(41): e2322420121.

Mei, Qiaozhu, Yutong Xie, Walter Yuan, and Matthew O. Jackson. 2024. "A Turing test of whether AI chatbots are behaviorally similar to humans." *Proceedings of the National Academy of Sciences*, 121(9): e2313925121.

Mitchell, Melanie. 2023. "How do we know how smart AI systems are?" *Science*, 381(6654): eadj5957.

Mullainathan, Sendhil, and Ashesh Rambachan. 2024. "From predictive algorithms to automatic generation of anomalies." National Bureau of Economic Research Working Paper 32422.

Mullainathan, Sendhil, and Jann Spiess. 2017. "Machine learning: an applied econometric approach." *Journal of Economic Perspectives*, 31(2): 87–106.

Nezhurina, Marianna, Lucia Cipolina-Kun, Mehdi Cherti, and Jenia Jitsev. 2024. "Alice in Wonderland: Simple Tasks Showing Complete Reasoning Breakdown in State-Of-the-Art Large Language Models." *arXiv preprint arXiv:2406.02061.*

OpenAI. 2023. "GPT-4 Technical Report." *arXiv preprint arXiv:2303.08774.*

Peterson, Joshua C., David D. Bourgin, Mayank Agrawal, Daniel Reichman, and Thomas L. Griffiths. 2021. "Using large-scale experiments and machine learning to discover theories of human decision-making." *Science*, 372(6547): 1209–1214.

Raman, Narun, Taylor Lundy, Samuel Amouyal, Yoav Levine, Kevin Leyton-Brown, and Moshe Tennenholtz. 2024. "STEER: Assessing the Economic Rationality of Large Language Models." *arXiv preprint arXiv:2402.09552.*

Rambachan, Ashesh, and Jonathan Roth. 2024. "Design-Based Uncertainty for Quasi-Experiments." *arXiv preprint arXiv:2008.00602.*

Rambachan, Ashesh, Rahul Singh, and Davide Viviano. 2024. "Program Evaluation with Remotely Sensed Outcomes." *arXiv preprint arXiv:2411.10959.*

Ravaut, Mathieu, Bosheng Ding, Fangkai Jiao, Hailin Chen, Xingxuan Li, Ruochen Zhao, Chengwei Qin, Caiming Xiong, and Shafiq Joty. 2024. "How Much are Large Language Models Contaminated? A Comprehensive Survey and the LLMSanitize Library." *arXiv preprint arXiv:2404.00699.*

Sainz, Oscar, Jon Campos, Iker García-Ferrero, Julen Etxaniz, Oier Lopez de Lacalle, and Eneko Agirre. 2023. "NLP Evaluation in trouble: On the Need to Measure LLM Data Contamination for each Benchmark." 10776–10787. Association for Computational Linguistics.

Santurkar, Shibani, Esin Durmus, Faisal Ladhak, Cinoo Lee, Percy Liang, and Tatsunori Hashimoto. 2023. "Whose opinions do language models reflect?" *ICML'23*. JMLR.

Sarkar, Suproteem. 2024. "StoriesLM: A Family of Language Models With Time-Indexed Training Data." *Available at SSRN.*

Sarkar, Suproteem K, and Keyon Vafa. 2024. "Lookahead bias in pretrained language models." *Available at SSRN.*

Schennach, Susanne M. 2016. "Recent advances in the measurement error literature." *Annual Review of Economics*, 8(1): 341–377.

Si, Chenglei, Diyi Yang, and Tatsunori Hashimoto. 2024. "Can LLMs Generate Novel Research Ideas? A Large-Scale Human Study with 100+ NLP Researchers." *arXiv preprint arXiv:2409.04109.*

Srivastava, Aarohi, Abhinav Rastogi, Abhishek Rao, Abu Awal Md Shoeb, Abubakar Abid, Adam Fisch, Adam R Brown, et al. 2022. "Beyond the imitation game: Quantifying and extrapolating the capabilities of language models." *arXiv preprint arXiv:2206.04615.*

Szempruch, Daniel E. 2023. "Generative AI Model Hallucinations: The Good, The Bad, and The Hilarious." *LinkedIn*, March 16. `https://www.linkedin.com/pulse/generative-ai-model-hallucinations-good-bad-hilarious-szempruch` (accessed December 5, 2024).

Touvron, Hugo, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, et al. 2023. "Llama 2: Open Foundation and Fine-Tuned Chat Models." *arXiv preprint arXiv:2307.09288.*

Vafa, Keyon, Ashesh Rambachan, and Sendhil Mullainathan. 2024. "Do Large Language Models Generalize the Way People Expect? A Benchmark for Evaluation." *arXiv preprint arXiv:2406.01382.*

Varian, Hal R. 2014. "Big data: New tricks for econometrics." *Journal of economic perspectives*, 28(2): 3–28.

Wager, Stefan, and Susan Athey. 2018. "Estimation and inference of heterogeneous treatment effects using random forests." *Journal of the American Statistical Association*, 113(523): 1228–1242.

Wang, Siruo, Tyler H. McCormick, and Jeffrey T. Leek. 2020. "Methods for correcting inference based on outcomes predicted by machine learning." *Proceedings of the National Academy of Sciences*, 117(48): 30266–30275.

Wei, Jason, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. 2024. "Chain-of-Thought Prompting Elicits Reasoning in Large Language Models." *NIPS '22.* Curran Associates Inc.

Wei, Yanhao, and Nikhil Malik. 2023. "Unstructured Data, Econometric Models, and Estimation Bias."

White, Jules, Quchen Fu, Sam Hays, Michael Sandborn, Carlos Olea, Henry Gilbert, Ashraf Elnashar, Jesse Spencer-Smith, and Douglas C. Schmidt. 2023. "A Prompt Pattern Catalog to Enhance Prompt Engineering with ChatGPT." *arXiv preprint arXiv:2302.11382.*

Wilkerson, John, E. Scott Adler, Bryan D. Jones, Frank R. Baumgartner, Guy Freedman, Sean M. Theriault, Alison Craig, Derek A. Epp, Cheyenne Lee, and Miranda E. Sullivan. 2023. *Policy Agendas Project: Congressional Bills.* https://minio.la.utexas.edu/compagendas/datasetfiles/US-Legislative-congressional_bills_19.3_3_3.csv (accessed July 5, 2024).

Wu, Zhaofeng, Linlu Qiu, Alexis Ross, Ekin Akyürek, Boyuan Chen, Bailin Wang, Najoung Kim, Jacob Andreas, and Yoon Kim. 2024. "Reasoning or Reciting? Exploring the Capabilities and Limitations of Language Models Through Counterfactual Tasks." 1819–1862. Association for Computational Linguistics.

Xu, Ruonan. 2020. "Potential outcomes and finite-population inference for M-estimators." *The Econometrics Journal*, 24(1): 162–176.

Xu, Ziwei, Sanjay Jain, and Mohan Kankanhalli. 2024. "Hallucination is Inevitable: An Innate Limitation of Large Language Models." *arXiv preprint arXiv:2401.11817*.

Zhou, Yangqiaoyu, Haokun Liu, Tejes Srivastava, Hongyuan Mei, and Chenhao Tan. 2024. "Hypothesis Generation with Large Language Models." *arXiv preprint arXiv:2404.04326*.

Zong, Yongshuo, Tingyang Yu, Ruchika Chavhan, Bingchen Zhao, and Timothy Hospedales. 2024. "Fool Your (Vision and) Language Model with Embarrassingly Simple Permutations." Vol. 235 of *Proceedings of Machine Learning Research*, 62892–62913. PMLR.

Zou, Andy, Zifan Wang, Nicholas Carlini, Milad Nasr, J. Zico Kolter, and Matt Fredrikson. 2023. "Universal and Transferable Adversarial Attacks on Aligned Language Models." *arXiv preprint arXiv:2307.15043*.

Main Figures and Tables

Original Bill: to amend title xviii of the social security act to distribute additional information to medicare beneficiaries to prevent health care fraud and for other purposes

GPT-4o: to amend title xviii of the social security act to distribute additional information to medicare beneficiaries to prevent health care fraud and for other purposes

Original Bill: a bill to amend the comprehensive environmental response compensation and liability act of 1980 to promote the cleanup and reuse of brownfields to provide financial assistance for brownfields revitalization to enhance state response programs and for other purposes

GPT-4o: a bill to amend the comprehensive environmental response compensation and liability act of 1980 to promote the cleanup and reuse of brownfields to provide financial assistance for brownfields revitalization to enhance state response programs and for other purposes

Figure 1: Two examples of GPT-4o completions that exactly match original descriptions of congressional legislation.

Notes: On 10,000 randomly sampled congressional bills, we prompted GPT-4o to complete the description of the congressional bill based on 50% of its text. See Section 4.3.1.

Original Headline: piper jaffray maintains overweight on activision blizzard raises price target to 62

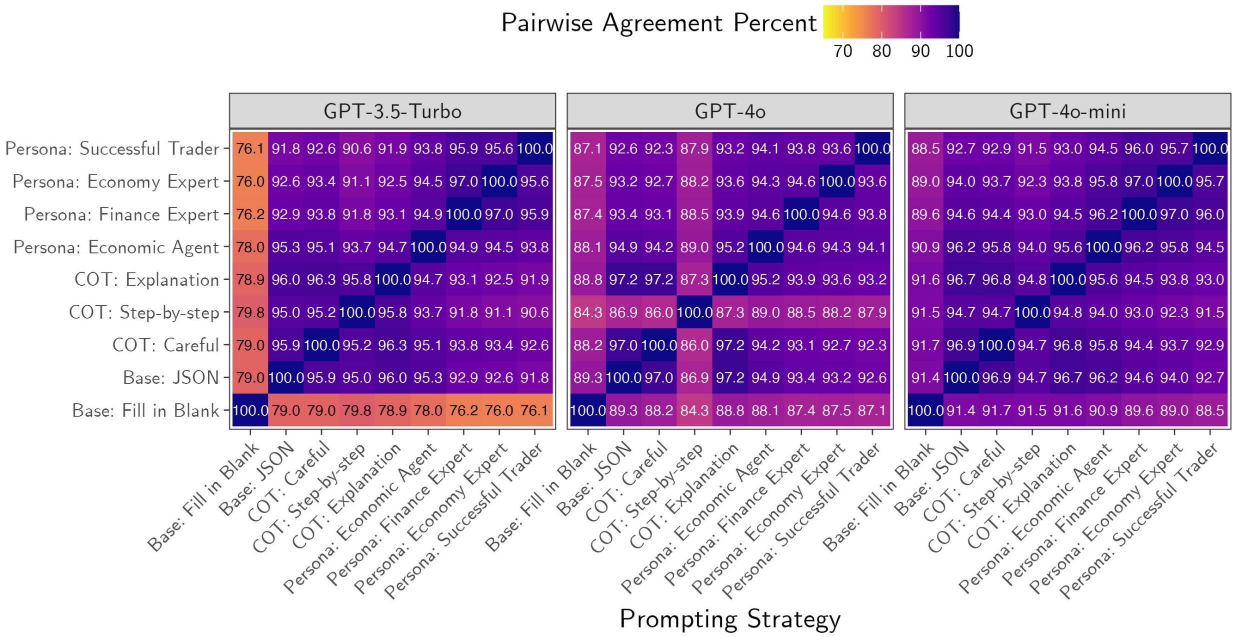
GPT-4o: piper jaffray maintains overweight on activision blizzard raises price target to 62

Original Headline: sinclair completes acquisition of regional sports networks from disney

GPT-4o: sinclair completes acquisition of regional sports networks from disney

Figure 2: Two examples of GPT-4o completions that exactly match original financial news headlines.

Notes: On 10,000 randomly sampled financial news headlines from 2019, we prompted GPT-4o to complete the financial news headline based on 50% of its text. See Section 4.3.2.



GPT-3.5-Turbo

	Base: Fill in Blank	Base: JSON	COT: Careful	COT: Step-by-step	COT: Explanation	Persona: Economic Agent	Persona: Finance Expert	Persona: Economy Expert	Persona: Successful Trader
Persona: Successful Trader	76.1	91.8	92.6	90.6	91.9	93.8	95.9	95.6	100.0
Persona: Economy Expert	76.0	92.6	93.4	91.1	92.5	94.5	97.0	100.0	95.6
Persona: Finance Expert	76.2	92.9	93.8	91.8	93.1	94.9	100.0	97.0	95.9
Persona: Economic Agent	78.0	95.3	95.1	93.7	94.7	100.0	94.9	94.5	93.8
COT: Explanation	78.9	96.0	96.3	95.8	100.0	94.7	93.1	92.5	91.9
COT: Step-by-step	79.8	95.0	95.2	100.0	95.8	93.7	91.8	91.1	90.6
COT: Careful	79.0	95.9	100.0	95.2	96.3	95.1	93.8	93.4	92.6
Base: JSON	79.0	100.0	95.9	95.0	96.0	95.3	92.9	92.6	91.8
Base: Fill in Blank	100.0	79.0	79.0	79.8	78.9	78.0	76.2	76.0	76.1

GPT-4o

	Base: Fill in Blank	Base: JSON	COT: Careful	COT: Step-by-step	COT: Explanation	Persona: Economic Agent	Persona: Finance Expert	Persona: Economy Expert	Persona: Successful Trader
Persona: Successful Trader	87.1	92.6	92.3	87.9	93.2	94.1	93.8	93.6	100.0
Persona: Economy Expert	87.5	93.2	92.7	88.2	93.6	94.3	94.6	100.0	93.6
Persona: Finance Expert	87.4	93.4	93.1	88.5	93.9	94.6	100.0	94.6	93.8
Persona: Economic Agent	88.1	94.9	94.2	89.0	95.2	100.0	94.6	94.3	94.1
COT: Explanation	88.8	97.2	97.2	87.3	100.0	95.2	93.9	93.6	93.2
COT: Step-by-step	84.3	86.9	86.0	100.0	87.3	89.0	88.5	88.2	87.9
COT: Careful	88.2	97.0	100.0	86.0	97.2	94.2	93.1	92.7	92.3
Base: JSON	89.3	100.0	97.0	86.9	97.2	94.9	93.4	93.2	92.6
Base: Fill in Blank	100.0	89.3	88.2	84.3	88.8	88.1	87.4	87.5	87.1

GPT-4o-mini

	Base: Fill in Blank	Base: JSON	COT: Careful	COT: Step-by-step	COT: Explanation	Persona: Economic Agent	Persona: Finance Expert	Persona: Economy Expert	Persona: Successful Trader
Persona: Successful Trader	88.5	92.7	92.9	91.5	93.0	94.5	96.0	95.7	100.0
Persona: Economy Expert	89.0	94.0	93.7	92.3	93.8	95.8	97.0	100.0	95.7
Persona: Finance Expert	89.6	94.6	94.4	93.0	94.5	96.2	100.0	97.0	96.0
Persona: Economic Agent	90.9	96.2	95.8	94.0	95.6	100.0	96.2	95.8	94.5
COT: Explanation	91.6	96.7	96.8	94.8	100.0	95.6	94.5	93.8	93.0
COT: Step-by-step	91.5	94.7	94.7	100.0	94.8	94.0	93.0	92.3	91.5
COT: Careful	91.7	96.9	100.0	94.7	96.8	95.8	94.4	93.7	92.9
Base: JSON	91.4	100.0	96.9	94.7	96.7	96.2	94.6	94.0	92.7
Base: Fill in Blank	100.0	91.4	91.7	91.5	91.6	90.9	89.6	89.0	88.5

Figure 3: Variation in pairwise agreement between large language model labels across prompting strategies on financial news headlines.

Notes: On financial news headlines from 2019, we prompt GPT-3.5-Turbo, GPT-4o, and GPT-4o-mini to label each headline for whether it expressed positive, negative or uncertain news about the respective company using alternative prompting strategies. For each pair of prompting strategies, we calculate the fraction of financial news headlines that receive the same label by the two prompting strategies, separately by large language model. See Section 5.3.2 for discussion.

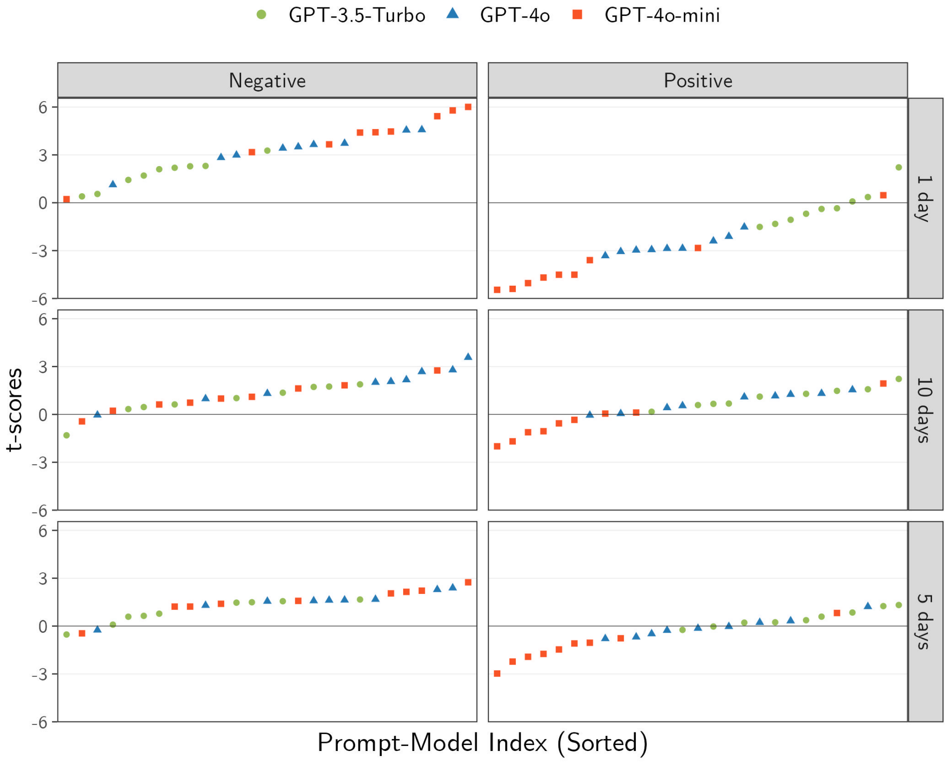


Figure 4: Variation in t-statistics for realized returns across large language models and prompting strategies on financial news headlines.

Notes: On financial news headlines from 2019, we prompt GPT-3.5-Turbo, GPT-4o-mini, and GPT-4o to label each headline for whether it expressed positive, negative or uncertain news about the respective company using alternative prompting strategies. For each model m and prompt p, we regress the realized returns of each stock within 1 day, 5 days or 10 days of the headline's publication date on each large language model's labels $\widehat{V}_r^{m,p}$, the large language model's assessed magnitude denoted $S_r^{m,p}$ and their interaction, controlling for lagged realized returns. We separately report the t-statistics associated with the regression coefficients on whether the headline is labeled as positive or negative news (standard errors are two-way clustered at the date and company level). In each subplot, the t-statistics are sorted in ascending order for clarity. See Section 5.3.2 for discussion.

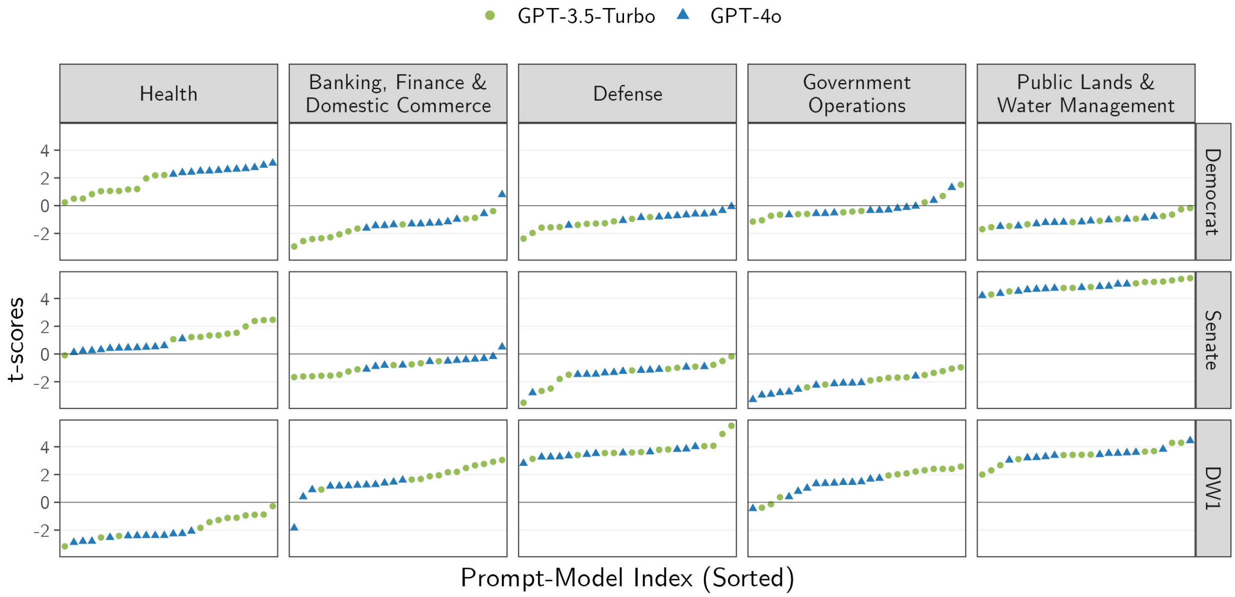


Figure 5: Variation in t-statistics across large language models and prompting strategies on congressional legislation.

Notes: On 10,000 Congressional bills, we prompt GPT-3.5-Turbo and GPT-4o to label each description for its policy topic area using alternative prompting strategies. For each model m and prompt p, we regress $\widehat{V}_r^{m,p}$ on the linked covariate W_r, where $\widehat{V}_r^{m,p}$ are indicators for the policy topic of the bill (i.e., Health, Banking, Finance and Domestic Commerce, Defense, Government Operations and Public Lands and Water Management) and the covariates W_r are whether the bill's sponsor was a Democrat, whether the bill originated in the Senate, and the DW1 score of the bill's sponsor. In each subplot, the t-score estimates were sorted in ascending order for clarity. See Section 5.3.3 for discussion.

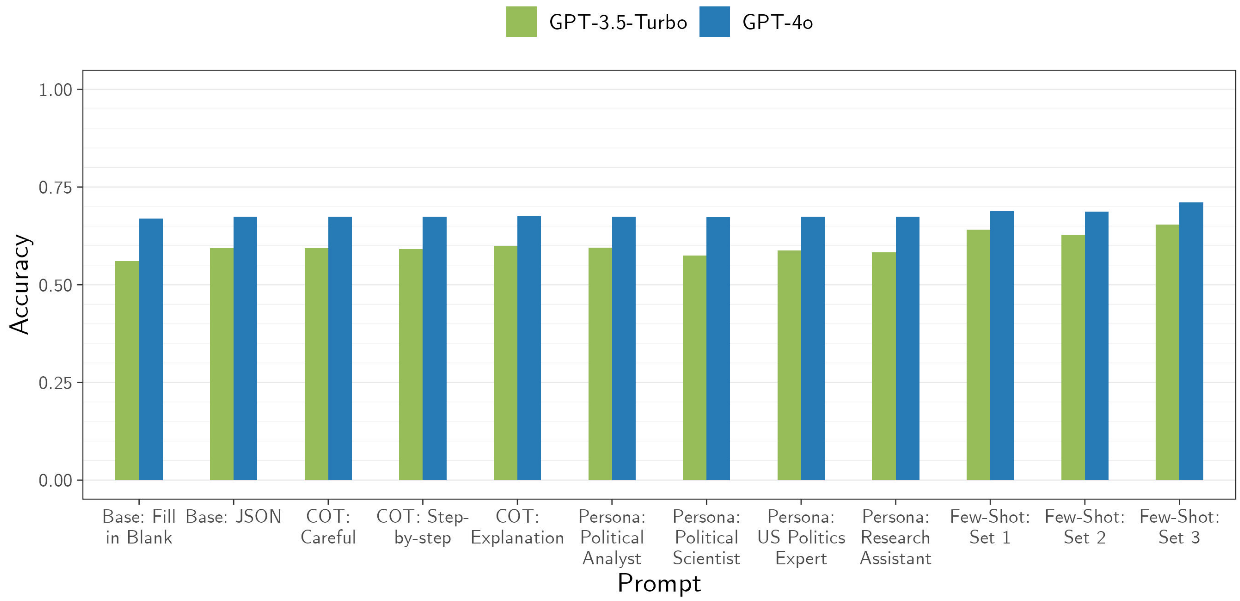


Figure 6: Accuracy of large language model labels of bill topic across model and prompt variation.

Notes: On 10,000 Congressional bills, we prompt GPT-3.5-Turbo and GPT-4o to label each description for its policy topic area using alternative prompting strategies. For each combination of model m and prompt p, we calculate the accuracy of the labels $\widehat{V}_r^{m,p}$ for the ground-truth label V_r. See Section 5.4.2 for discussion.

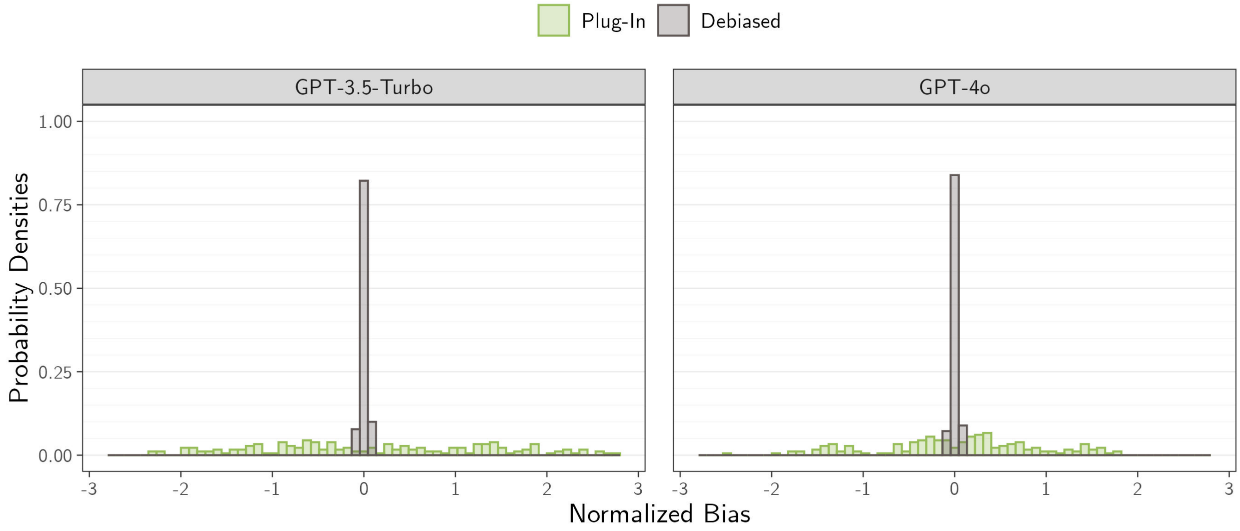


Figure 7: Normalized bias of the plug-in regression and bias-corrected regression across Monte Carlo simulations based on congressional legislation.

Notes: The normalized bias reports the average bias of the plug-in regression coefficient $\widehat{\beta}$ and the debiased coefficient $\widehat{\beta}^{debiased}$ for the target regression coefficient divided by their respective standard deviations across simulations. We summarize the distribution of normalized bias and coverage across regression specifications, choice of large language model and prompting strategies. For each combination of model topic V_r, linked covariate W_r, large language model m and prompting strategy p, we randomly sample 5,000 Congressional bills and calculate the plug-in regression coefficient $\widehat{\beta}$ and the bias-corrected regression coefficient $\widehat{\beta}^{debiased}$ based on a 5% validation sample. See Section 5.4.2 for discussion.

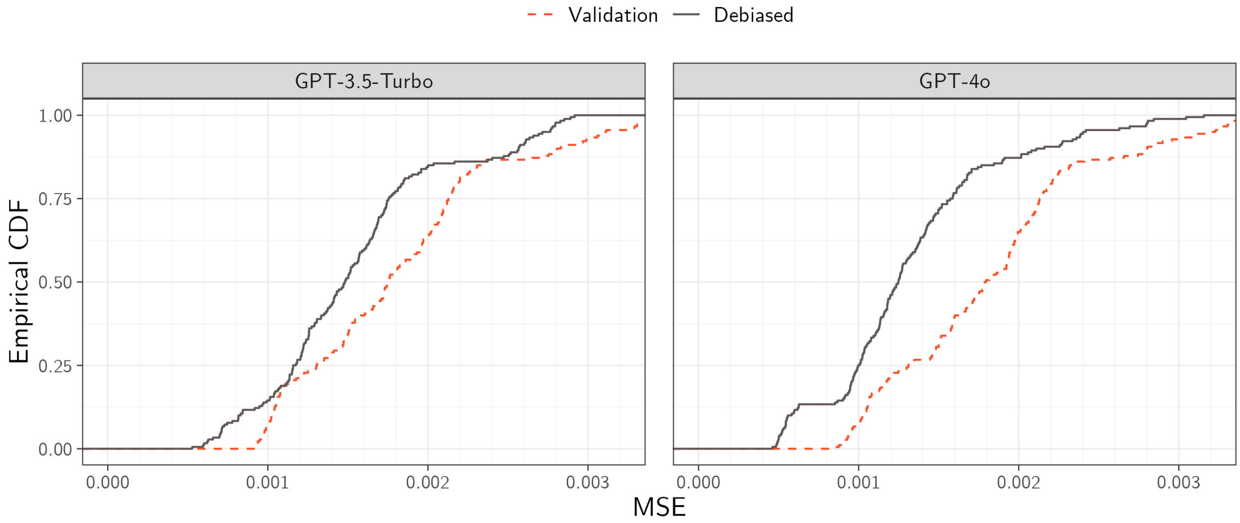


Figure 8: Cumulative distribution function of mean square error for the bias-corrected estimator against validation-sample only estimator.

Notes: For each combination of model topic V_r, covariate W_r, large language model m and prompting strategy p, we randomly sample 5,000 Congressional bills and calculate the bias-corrected regression coefficient using a 5% validation sample and the validation-sample only regression coefficient. We calculate the mean square error of $\widehat{\beta}^{debiased}$ and $\widehat{\beta}^{*}$ for the target regression, and we average the results over 1,000 simulations. We summarize the distribution of average mean square error across regression specifications, choice of large language model and prompting strategies. See Section 5.4.2 for discussion.

	Accuracy	TPR	FPR
House	0.912	0.198	0.031
Senate	0.925	0.225	0.031

(a) Base prompt

	Accuracy	TPR	FPR
House	0.644	0.691	0.359
Senate	0.695	0.711	0.306

(b) Prompt with date restriction

Table 1: Accuracy, true positive rate (TPR), and false positive rate (FPR) of GPT-4o's predictions on Congressional legislation.

Notes: We prompt GPT-4o to predict whether 10,000 randomly selected Congressional bills would pass the Senate or the House based on its text description. This table reports the accuracy, true positive rate (TPR), and false positive rate (FPR) of GPT-4o's predictions. Table (a) provides results for the base prompt, and Table (b) provides results for the base prompt with the additional date restriction. See Section 4.3.1 for discussion.

	Return Horizon		
Point Estimates	1 day	5 days	10 days
Mean	-0.984	-0.321	0.401
Median	-1.042	-0.130	0.536
5th Percentile	-2.892	-1.962	-1.387
95th Percentile	0.108	0.762	1.780
Sample Average	0.051	0.313	0.584

(a) Positive Labels

	Return Horizon		
Point Estimates	1 day	5 days	10 days
Mean	1.620	1.324	1.471
Median	1.304	1.329	1.489
5th Percentile	0.236	-0.180	-0.162
95th Percentile	3.468	2.759	3.119
Sample Average	0.051	0.313	0.584

(b) Negative Labels

Table 2: Variation in point estimates across large language models and prompting strategies on financial news headlines.

Notes: On financial news headlines from 2019, we prompt GPT-3.5-Turbo, GPT-4o-mini, and GPT-4o to label each headline for whether it expressed positive, negative or uncertain news about the respective company using alternative prompting strategies. For each model m and prompt p, we regress the realized returns of each stock within 1-day of the headline's publication date on each large language model's labels $\widehat{V}_r^{m,p}$, the large language model's assessed magnitude denoted $S_r^{m,p}$ and their interaction, controlling for lagged realized returns. See Section 5.3.2 for discussion.

Policy Topic	Covariate	*Point Estimates*				Sample
		Mean	Median	5%	95%	Average
Health	DW1	-0.015	-0.018	-0.022	-0.006	0.150
Health	Democrat	0.010	0.012	0.002	0.016	0.150
Health	Senate	0.005	0.005	0.001	0.013	0.150
Banking, Finance & Domestic Com.	DW1	0.012	0.012	0.004	0.024	0.127
Banking, Finance & Domestic Com.	Democrat	-0.009	-0.008	-0.017	-0.002	0.127
Banking, Finance & Domestic Com.	Senate	-0.005	-0.005	-0.010	-0.001	0.127
Defense	DW1	0.025	0.024	0.019	0.043	0.204
Defense	Democrat	-0.006	-0.006	-0.013	-0.002	0.204
Defense	Senate	-0.008	-0.006	-0.019	-0.003	0.204
Government Operations	DW1	0.012	0.013	-0.003	0.020	0.281
Government Operations	Democrat	-0.001	-0.002	-0.006	0.008	0.281
Government Operations	Senate	-0.013	-0.013	-0.021	-0.007	0.281
Public Lands & Water Management	DW1	0.026	0.027	0.016	0.031	0.238
Public Lands & Water Management	Democrat	-0.007	-0.007	-0.010	-0.002	0.238
Public Lands & Water Management	Senate	0.032	0.032	0.027	0.036	0.238

Table 3: Variation in point estimates across large language models and prompting strategies on Congressional bills.

Notes: On 10,000 Congressional bills, we prompt GPT-3.5-Turbo and GPT-4o to label each Congressional bill for its policy topic using alternative prompting strategies. For each model m and prompt p, we regress an indicator for whether the large language model labeled a particular policy topic $1\{\widehat{V}_r^{m,p} = v\}$, focusing on Health, Banking, Finance & Domestic Commerce, Defense, Government Operations, and Public Lands & Water Management, on alternative covariates W_r (whether the bill's sponsor was a Democrat, whether the bill originated in the Senate, and the DW1 score of the bill's sponsor). For comparison, the final column ("Sample Average") reports the fraction of all Congressional bills assigned to the policy topic $1\{V_r = v\}$. See Section 5.3.3 for discussion.

	Median	5%	95%
Normalized Bias			
Plug-In	-0.023	-1.899	2.211
Debiased	0.001	-0.055	0.066
Coverage			
Plug-In	0.820	0.381	0.945
Debiased	0.930	0.910	0.945

(a) GPT-3.5-turbo

	Median	5%	95%
Normalized Bias			
Plug-In	0.084	-1.411	1.514
Debiased	0.001	-0.055	0.054
Coverage			
Plug-In	0.920	0.637	0.954
Debiased	0.927	0.902	0.945

(b) GPT-4o

Table 4: Summary statistics for normalized bias and coverage across Monte Carlo simulations based on Congressional legislation.

Notes: The normalized bias reports the average bias of the plug-in regression coefficient $\widehat{\beta}$ and the debiased coefficient $\widehat{\beta}^{debiased}$ for the target regression coefficient divided by their respective standard deviations across simulations. The coverage reports the fraction of simulations in which a 95% nominal confidence interval centered around the plug-in regression coefficient $\widehat{\beta}$ and the bias-corrected coefficient $\widehat{\beta}^{debiased}$ cover the target regression coefficient. We summarize the distribution of normalized bias and coverage across regression specifications, choice of large language model and prompting strategies. For each combination of model topic V_r, covariate W_r, large language model m and prompting strategy p, we randomly sample 5,000 Congressional bills and calculate the plug-in regression coefficient $\widehat{\beta}$ and the bias-corrected regression coefficient $\widehat{\beta}^{debiased}$ using a 5% validation sample. Results are averaged over 1,000 simulations. See Section 5.4.2 for discussion.

Large Language Models: An Applied Econometric Perspective

Online Appendix

Jens Ludwig & Sendhil Mullainathan & Ashesh Rambachan

A Appendix Figures and Tables

Metric	Average	Benchmark
Cosine similarity	0.830	0.379
Euclidean distance	0.536	1.110

(a) Base prompt

Metric	Average	Benchmark
Cosine similarity	0.830	0.379
Euclidean distance	0.536	1.110

(b) Prompt with date restriction

Table A1: Embedding distance between GPT-4o's completed bill descriptions and original bill descriptions.

Notes: This table calculates the cosine similarity and Euclidean distance between embeddings of GPT-4o's completed bill descriptions and embeddings of the original bill descriptions. We construct embeddings using OpenAI's text-embedding-3-small model. As a benchmark, we calculate the average cosine similarity and Euclidean distance between 10,000 randomly selected pairs of original bill descriptions. Table (a) provides results for the base prompt and Table (b) provides results for the base prompt with the additional date restriction. The results in Table (a) and Table (b) are the same up to 3 decimal places. See Section 4.3.1 for discussion.

Metric	Average	Benchmark
Cosine similarity	0.880	0.309
Euclidean distance	0.455	1.172

(a) Base prompt

Metric	Average	Benchmark
Cosine similarity	0.880	0.309
Euclidean distance	0.455	1.172

(b) Prompt with date restriction

Table A2: Embedding distance between GPT-4o's completed financial news headlines and original financial news headlines.

Notes: This table calculates the cosine similarity and Euclidean distance between embeddings of GPT-4o's completed financial news headlines and embeddings of the original financial news headlines. We construct embeddings using OpenAI's text-embedding-3-small model. As a benchmark, we calculate the average cosine similarity and Euclidean distance between 10,000 randomly selected pairs of original financial news headlines. Table (a) provides results for the base prompt and Table (b) provides results for the base prompt with the additional date restriction. The results in Table (a) and Table (b) are the same up to 3 decimal places. See Section 4.3.2 for discussion.

	Return Horizon		
Point Estimates	1 day	5 days	10 days
Mean	-2.184	-1.242	-0.052
Median	-2.302	-1.129	0.027
5th Percentile	-5.161	-2.868	-2.298
95th Percentile	0.465	-0.025	2.312
Sample Average	0.051	0.313	0.584

(a) Positive Labels

	Return Horizon		
Point Estimates	1 day	5 days	10 days
Mean	2.566	0.848	0.410
Median	2.783	1.999	2.384
5th Percentile	-1.318	-6.735	-9.395
95th Percentile	5.552	4.529	5.854
Sample Average	0.051	0.313	0.584

(b) Negative Labels

Table A3: Point estimates across large language models and prompting strategies on financial news headlines using confidence.

Notes: On financial news headlines from 2019, we prompt GPT-3.5-Turbo, GPT-4o-mini, and GPT-4o to label each headline for whether it expressed positive, negative or uncertain news about the respective company using alternative prompting strategies. For each model m and prompt p, we regress the realized returns of each stock within 1-day of the headline's publication date on each large language model's labels $\widehat{V}_r^{m,p}$, the large language model's assessed confidence denoted $C_r^{m,p}$ and their interaction, controlling for lagged realized returns. See Section 5.3.2 for discussion.

Original Bill: a bill to authorize the federal programs to prevent violence against women and for other purposes

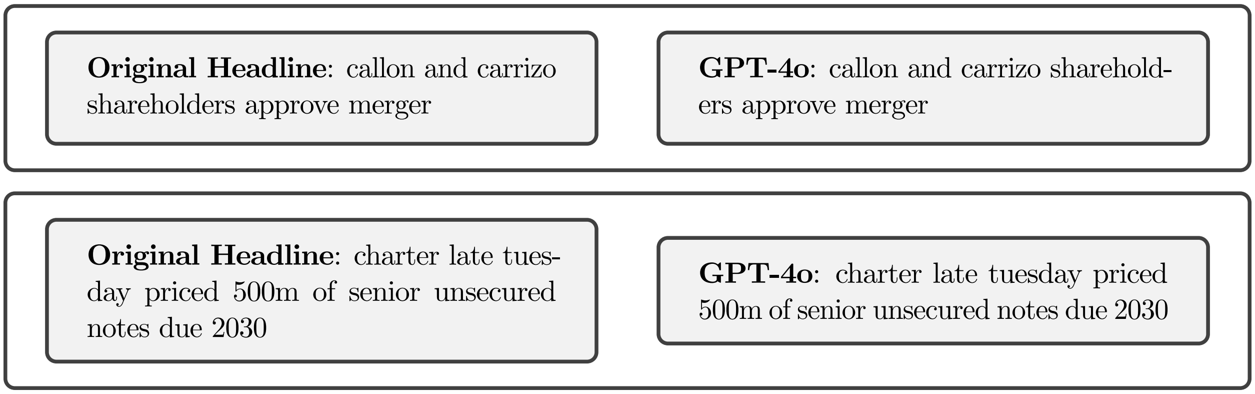
GPT-4o: a bill to authorize the federal programs to prevent violence against women and for other purposes

Original Bill: a bill to repeal the public utility holding company act of 1935 to enact the public utility holding company act of 1997 and for other purposes

GPT-4o: a bill to repeal the public utility holding company act of 1935 to enact the public utility holding company act of 1997 and for other purposes

Figure A1: Examples of GPT-4o completions with date restriction that exactly match original descriptions of Congressional legislation.

Notes: On 10,000 randomly sampled Congressional bills, we prompted GPT-4o to complete the description of the Congressional bill based on 50% of its text. The prompt included an explicit date restriction. See Section 4.3.1 for discussion.



Original Headline: callon and carrizo shareholders approve merger

GPT-4o: callon and carrizo shareholders approve merger

Original Headline: charter late tuesday priced 500m of senior unsecured notes due 2030

GPT-4o: charter late tuesday priced 500m of senior unsecured notes due 2030

Figure A2: Examples of GPT-4o completions with date restriction that exactly match original financial news headlines.

Notes: On 10,000 randomly sampled financial news headlines from 2019, we prompted GPT-4o to complete the financial news headline based on 50% of its text. The prompt included an explicit date restriction. See Section 4.3.2 for discussion.

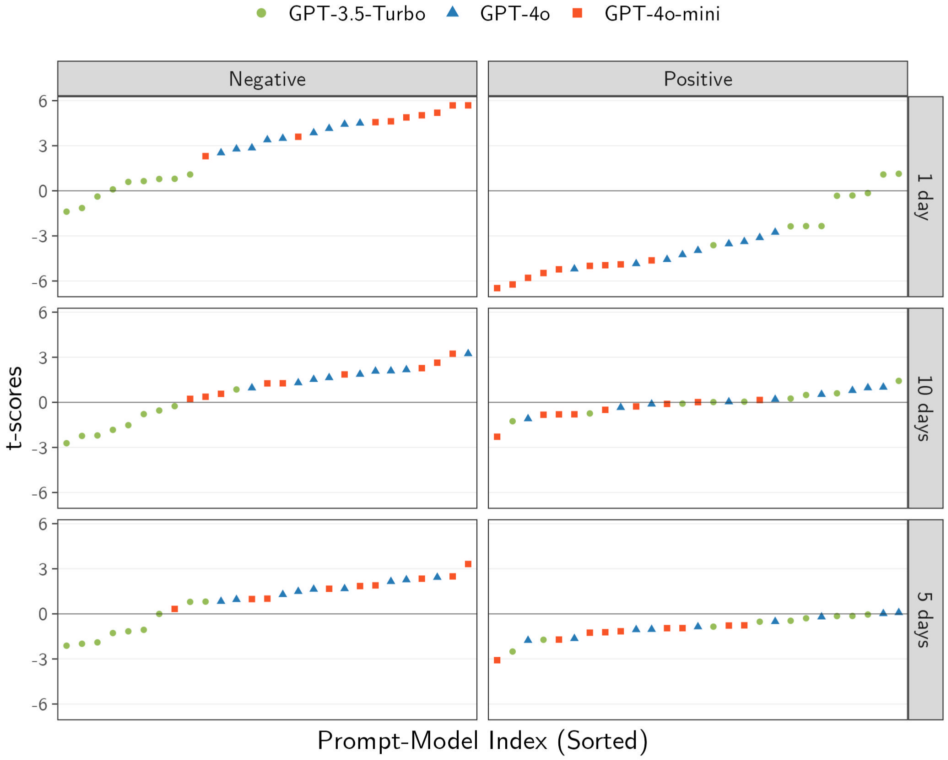
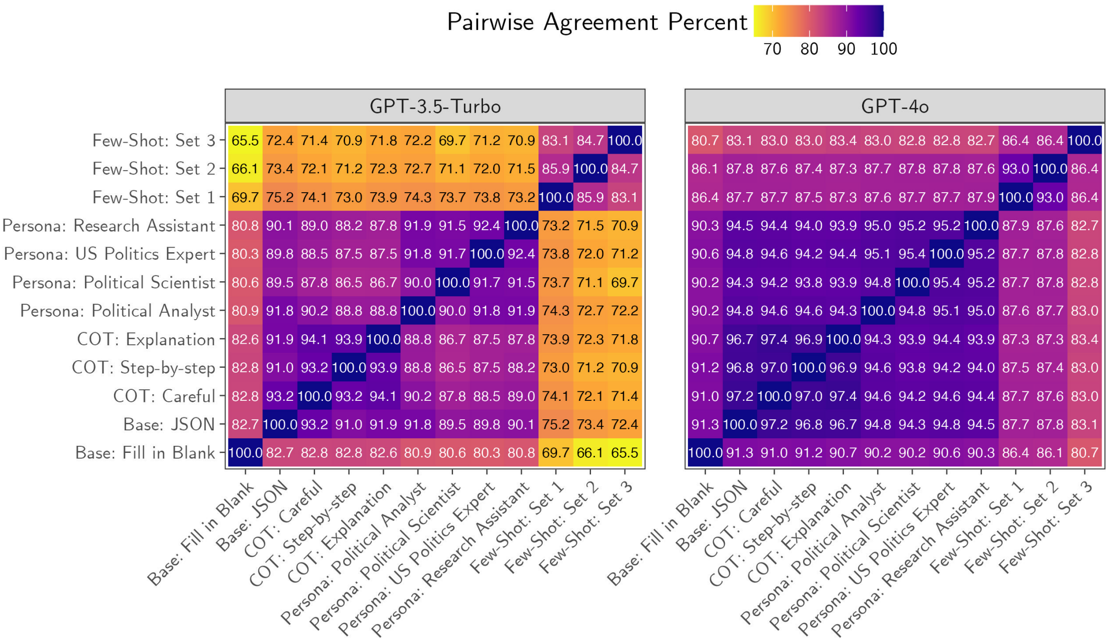


Figure A3: Variation in t-statistics realized returns across large language models and prompting strategies on financial news headlines using model's reported confidence.

Notes: On financial news headlines from 2019, we prompt GPT-3.5-Turbo, GPT-4o-mini, and GPT-4o to label each headline for whether it expressed positive, negative or uncertain news about the respective company using alternative prompting strategies. For each model m and prompt p, we regress the realized returns of each stock within 1 day, 5 days, and 10 days of the headline's publication date on each large language model's labels $\widehat{V}_r^{m,p}$, the large language model's reported confidence denoted $C_r^{m,p}$ and their interaction, controlling for lagged realized returns. We separately report the t-statistics associated with the regression coefficients on whether the headline is labeled as positive or negative news (standard errors are two-way clustered at the date and company level). In each subplot, the t-statistics are sorted in ascending order for clarity. See Section 5.3.2 for discussion.



GPT-3.5-Turbo

	Base: Fill in Blank	Base: JSON	COT: Careful	COT: Step-by-step	COT: Explanation	Persona: Political Analyst	Persona: Political Scientist	Persona: US Politics Expert	Persona: Research Assistant	Few-Shot: Set 1	Few-Shot: Set 2	Few-Shot: Set 3
Few-Shot: Set 3	65.5	72.4	71.4	70.9	71.8	72.2	69.7	71.2	70.9	83.1	84.7	100.0
Few-Shot: Set 2	66.1	73.4	72.1	71.2	72.3	72.7	71.1	72.0	71.5	85.9	100.0	84.7
Few-Shot: Set 1	69.7	75.2	74.1	73.0	73.9	74.3	73.7	73.8	73.2	100.0	85.9	83.1
Persona: Research Assistant	80.8	90.1	89.0	88.2	87.8	91.9	91.5	92.4	100.0	73.2	71.5	70.9
Persona: US Politics Expert	80.3	89.8	88.5	87.5	87.5	91.8	91.7	100.0	92.4	73.8	72.0	71.2
Persona: Political Scientist	80.6	89.5	87.8	86.5	86.7	90.0	100.0	91.7	91.5	73.7	71.1	69.7
Persona: Political Analyst	80.9	91.8	90.2	88.8	88.8	100.0	90.0	91.8	91.9	74.3	72.7	72.2
COT: Explanation	82.6	91.9	94.1	93.9	100.0	88.8	86.7	87.5	87.8	73.9	72.3	71.8
COT: Step-by-step	82.8	91.0	93.2	100.0	93.9	88.8	86.5	87.5	88.2	73.0	71.2	70.9
COT: Careful	82.8	93.2	100.0	93.2	94.1	90.2	87.8	88.5	89.0	74.1	72.1	71.4
Base: JSON	82.7	100.0	93.2	91.0	91.9	91.8	89.5	89.8	90.1	75.2	73.4	72.4
Base: Fill in Blank	100.0	82.7	82.8	82.8	82.6	80.9	80.6	80.3	80.8	69.7	66.1	65.5

GPT-4o

	Base: Fill in Blank	Base: JSON	COT: Careful	COT: Step-by-step	COT: Explanation	Persona: Political Analyst	Persona: Political Scientist	Persona: US Politics Expert	Persona: Research Assistant	Few-Shot: Set 1	Few-Shot: Set 2	Few-Shot: Set 3
Few-Shot: Set 3	80.7	83.1	83.0	83.0	83.4	83.0	82.8	82.8	82.7	86.4	86.4	100.0
Few-Shot: Set 2	86.1	87.8	87.6	87.4	87.3	87.7	87.8	87.8	87.6	93.0	100.0	86.4
Few-Shot: Set 1	86.4	87.7	87.7	87.5	87.3	87.6	87.7	87.7	87.9	100.0	93.0	86.4
Persona: Research Assistant	90.3	94.5	94.4	94.0	93.9	95.0	95.2	95.2	100.0	87.9	87.6	82.7
Persona: US Politics Expert	90.6	94.8	94.6	94.2	94.4	95.1	95.4	100.0	95.2	87.7	87.8	82.8
Persona: Political Scientist	90.2	94.3	94.2	93.8	93.9	94.8	100.0	95.4	95.2	87.7	87.8	82.8
Persona: Political Analyst	90.2	94.8	94.6	94.6	94.3	100.0	94.8	95.1	95.0	87.6	87.7	83.0
COT: Explanation	90.7	96.7	97.4	96.9	100.0	94.3	93.9	94.4	93.9	87.3	87.3	83.4
COT: Step-by-step	91.2	96.8	97.0	100.0	96.9	94.6	93.8	94.2	94.0	87.5	87.4	83.0
COT: Careful	91.0	97.2	100.0	97.0	97.4	94.6	94.2	94.6	94.4	87.7	87.6	83.0
Base: JSON	91.3	100.0	97.2	96.8	96.7	94.8	94.3	94.8	94.5	87.7	87.8	83.1
Base: Fill in Blank	100.0	91.3	91.0	91.2	90.7	90.2	90.2	90.6	90.3	86.4	86.1	80.7

Figure A4: Variation in pairwise agreement between large language model labels across prompting strategies on congressional legislation.

Notes: On 10,000 randomly sampled Congressional bills, we prompt GPT-3.5-turbo and GPT-4o to label the policy topic of each Congressional bill. For each pair of prompting strategies, we calculate the fraction of congressional bills that receive the same label, separately by large language model. See Section 5.3.3 for discussion.

B Proofs of Main Results

B.1 Proof of Lemma 1, Proposition 1, and Corollary 1

Proposition 1 is an immediate consequence of Lemma 1. To prove Lemma 1, observe that, for any $Q(\cdot)\in\mathcal{Q}$,

$$\mathbb{E}_Q[\sum_{r\in\mathcal{R}}D_r\ell(Y_r,\widehat{m}(r))]-\mathbb{E}_Q[\sum_{r\in\mathcal{R}}D_r\ell(Y_r,\widehat{m}(r))\,|\,T=t]=$$

$$\sum_{r\in\mathcal{R}}q_r^D\ell(Y_r,\widehat{m}(r))-\sum_{r\in\mathcal{R}}q_r^{D|T}(t_r)\ell(Y_r,\widehat{m}(r))=\sum_{r\in\mathcal{R}}\big(q_r^D-q_r^{D|T}(t_r)\big)\ell(Y_r,\widehat{m}(r)).$$

Under Assumption 1, for any text piece $r\in\mathcal{R}$, we can rewrite $q_r^D-q_r^{D|T}(t_r)$ as $q_r^D\left(1-\frac{q_r^{T|D}(t_r)}{q_r^T(t_r)}\right)$ by Bayes' rule. We therefore have

$$\mathbb{E}_Q[\sum_{r\in\mathcal{R}}D_r\ell(Y_r,\widehat{m}(r))]-\mathbb{E}_Q[\sum_{r\in\mathcal{R}}D_r\ell(Y_r,\widehat{m}(r))\,|\,T=t]=\sum_{r\in\mathcal{R}}q_r^D\left(1-\frac{q_r^{T|D}(t_r)}{q_r^T(t_r)}\right)\ell(Y_r,\widehat{m}(r)).$$

Lemma 1 then follows immediately.

Finally, to prove Corollary 1, suppose that $q_r^D\in\{0,1\}$ for all $r\in\mathcal{R}$. In this case,

$$\sum_{r\in\mathcal{R}}q_r^D\left(1-\frac{q_r^{T|D}(t_r)}{q_r^T(t_r)}\right)\ell(Y_r,\widehat{m}(r))=\sum_{r:\,q_r^D=1}\left(1-\frac{q_r^{T|D}(t_r)}{q_r^T(t_r)}\right)\ell(Y_r,\widehat{m}(r)).$$

Furthermore, for all r such that $q_r^D=1$, we have $q_r^{T|D}(t_r)=Q(T_r=t_r,D_r=1)$ equals $q_r^T(t_r)=Q(T_r=t_r)$ by the law of total probability. Consequently, $1-\frac{q_r^{T|D}(t_r)}{q_r^T(t_r)}=0$ for all r such that $q_r^D=1$ and no training leakage is satisfied. □

B.2 Proof of Lemma 2

To show this result, rewrite

$$\mathbb{E}_Q[\sum_{r\in\mathcal{R}}D_rg(\widehat{m}(r),W_r;\theta)\,|\,T=t]-\mathbb{E}_Q[\sum_{r\in\mathcal{R}}D_rg(V_r,W_r;\theta)]$$

as

$$\left(\mathbb{E}_Q[\sum_{r\in\mathcal{R}}D_rg(\widehat{m}(r),W_r;\theta)\,|\,T=t]-\mathbb{E}_Q[\sum_{r\in\mathcal{R}}D_rg(\widehat{m}(r),W_r;\theta)]\right)+$$
$$\left(\mathbb{E}_Q[\sum_{r\in\mathcal{R}}D_rg(\widehat{m}(r),W_r;\theta)]-\mathbb{E}_Q[\sum_{r\in\mathcal{R}}D_rg(V_r,W_r;\theta)]\right).$$

The result then follows by applying the same argument as the proof of Lemma 1 to rewrite the first term as $\mathbb{E}_Q[\sum_{r\in\mathcal{R}}D_r\left(\frac{q_r^{T|D}(t_r)}{q_r^T(t_r)}-1\right)g(\widehat{m}(r),W_r;\theta)]$. □

B.3 Proof of Lemma 3 and Proposition 2

Proposition 2 is an immediate consequence of Lemma 3. As a result, we focus on proving Lemma 3.

We first prove the claim in Equation (8). Consider any $Q(\cdot) \in \mathcal{Q}$ and $g(\cdot) \in \mathcal{G}$. Since no training leakage is satisfied, by Lemma 1, we may write

$$\mathbb{E}_Q[\sum_{r \in \mathcal{R}} D_r g(\widehat{m}(r), W_r; \theta) \,|\, T = t] - \mathbb{E}[\sum_{r \in \mathcal{R}} D_r g(V_r, W_r; \theta)] =$$

$$\mathbb{E}_Q[\sum_{r \in \mathcal{R}} D_r (g(\widehat{m}(r), W_r; \theta) - g(V_r, W_r; \theta))].$$

Defining $\Delta(r) = \widehat{m}(r) - f^*(r)$, the previous display can be further written as

$$\sum_{r \in \mathcal{R}} q_r^D (g(f^*(r) + \Delta(r), W_r; \theta) - g(f^*(r), W_r; \theta)) =$$

$$\sum_{r \in \mathcal{R}} q_r^D \frac{\partial g(\xi(t, W_r, \theta), W_r; \theta)}{\partial v} \Delta(r)$$

where the equality applies the mean value theorem for some $\xi(t, x; \theta)$ in between $f^*(r) + \Delta(r)$ and $f^*(r)$. It therefore follows that

$$\left| \mathbb{E}_Q[\sum_{r \in \mathcal{R}} D_r g(\widehat{m}(r), W_r; \theta) \,|\, T = t] - \mathbb{E}[\sum_{r \in \mathcal{R}} D_r g(V_t, W_r; \theta)] \right| \leq$$

$$\sum_{r \in \mathcal{R}} q_r^D \left| \frac{\partial g(\xi(t, W_r, \theta), W_r; \theta)}{\partial v} \right| |\Delta(r)| \leq \overline{G} \sum_{r \in \mathcal{R}} q_r^D |\Delta(r)|,$$

where the last inequality follows by Assumption 2. The result in Equation (8) is immediate following the definition of $\mathcal{M}(Q, \delta)$.

To prove Equation (9), consider any $Q(\cdot) \in \mathcal{Q}$ and $g(\cdot) \in \mathcal{G}$. Since no training leakage is satisfied, we can again write, for any $\widehat{m}(\cdot) \in \mathcal{M}(Q, \delta)$,

$$\left| \mathbb{E}_Q[\sum_{r \in \mathcal{R}} D_r g(\widehat{m}(r), W_r; \theta) \,|\, T = t] - \mathbb{E}_Q[\sum_{r \in \mathcal{R}} D_r g(V_r, W_r; \theta)] \right| =$$

$$\left| \sum_{r \in \mathcal{R}} q_t^D (g(\widehat{m}(r), W_r; \theta) - g(V_r, W_r; \theta)) \right|.$$

Again, defining $\Delta(r) = \widehat{m}(r) - f^*(r)$ and $\Delta(Q, \delta) = \{\Delta(r) \colon -\delta \leq \Delta(r) \leq \delta \text{ for } r \text{ with } q_r^D > 0\}$, we have that

$$\sup_{\widehat{m}(\cdot) \in \mathcal{M}(Q, \delta)} \left| \mathbb{E}_Q[\sum_{r \in \mathcal{R}} D_r g(\widehat{m}(r), W_r; \theta) \,|\, T = t] - \mathbb{E}_Q[\sum_{r \in \mathcal{R}} D_r g(f^*(r), W_r; \theta)] \right| =$$

$$\sup_{\Delta(\cdot)\in\Delta(Q,\delta)}\left|\sum_{r\in\mathcal{R}}q_r^D(g(f^*(r)+\Delta(r),W_r;\theta)-g(f^*(r),W_r;\theta))\right|$$

Consider the following choice of $\delta(r)$. Define $\tilde{\Delta}(r)=\arg\max_{-\delta\leq\tilde{\delta}\leq\delta}g(f^*(r)+\tilde{\delta},W_r;\theta)-g(f^*(r),W_r;\theta)$, and let $\delta(r)=\tilde{\Delta}(r)1\{g(f^*(r)+\tilde{\Delta}(r)\},W_r;\theta)-g(f^*(r),W_r;\theta)\geq 0\}$. This choice is feasible, and so it follows that

$$\sup_{\Delta(\cdot)\in\Delta(\delta,Q)}\left|\sum_{r\in\mathcal{R}}q_r^D(g(f^*(r)+\Delta(r),W_r;\theta)-g(f^*(r),W_r;\theta))\right|\geq$$

$$\sum_{r\in\mathcal{R}}q_r^D|g(f^*(r)+\delta(r),W_r;\theta)-g(f^*(r),W_r;\theta)|,$$

where we further used that the triangle inequality holds with equality when all terms in a summation are non-negative. By a similar argument as given in the proof of Equation (8), we can apply the mean value theorem and the definition of sensitive text pieces to obtain the lower bound

$$\sup_{\widehat{m}(\cdot)\in\mathcal{M}(\delta,Q)}\left|\mathbb{E}_Q[\sum_{r\in\mathcal{R}}D_rg(\widehat{m}(r),W_r;\theta)\,|\,T=t]-\mathbb{E}_Q[\sum_{r\in\mathcal{R}}D_rg(f^*(r),W_r;\theta)]\right|\geq\underline{G}\sum_{r\in\mathcal{R}_{g,Q}}\delta(r)q_r^D.$$

□

B.4 Proof of Proposition 3

To show (i), given that $q_r^T(t_r)=q_r^{T|D}(t_r)$ for all $r\in\mathcal{R}$, it follows that

$$\beta=\left(\sum_{r\in\mathcal{R}}q_r^DW_rW_r'\right)^{-1}\left(\sum_{r\in\mathcal{R}}q_r^DW_r\widehat{V}_r\right)\text{ and }\beta^*=\left(\sum_{r\in\mathcal{R}}q_r^DW_rW_r'\right)^{-1}\left(\sum_{r\in\mathcal{R}}q_r^DW_rV_r\right).$$

But, of course, since $\widehat{V}_r=V_r+\Delta_r$, it then follows that

$$\beta=\left(\sum_{r\in\mathcal{R}}q_r^DW_rW_r'\right)^{-1}\left(\sum_{r\in\mathcal{R}}q_r^DW_rV_r\right)+\left(\sum_{r\in\mathcal{R}}q_r^DW_rW_r'\right)^{-1}\left(\sum_{r\in\mathcal{R}}q_r^DW_r\Delta_r\right).$$

The result is then immediate from the definition of β^* and the best linear projection of Δ_r onto W_r.

To show (ii), since there is again no training leakage by assumption, it follows that

$$\beta=\left(\sum_{r\in\mathcal{R}}q_r^D\widehat{V}_r\widehat{V}_r'\right)^{-1}\left(\sum_{r\in\mathcal{R}}q_r^D\widehat{V}_rW_r\right)\text{ and }\beta^*=\left(\sum_{r\in\mathcal{R}}q_r^DV_rV_r'\right)^{-1}\left(\sum_{r\in\mathcal{R}}q_r^DV_rW_r\right).$$

But, of course, since $W_r=V_r'\beta^*+\epsilon_r$ for ϵ_r the residual from the best-linear projection, it then follows that

$$\beta=\left(\sum_{r\in\mathcal{R}}q_r^D\widehat{V}_r\widehat{V}_r'\right)^{-1}\left(\sum_{r\in\mathcal{R}}q_r^D\widehat{V}_rV_r'\right)\beta^*+\left(\sum_{r\in\mathcal{R}}q_r^D\widehat{V}_r\widehat{V}_r'\right)^{-1}\left(\sum_{r\in\mathcal{R}}q_r^D\widehat{V}_r\epsilon_r\right).$$

The result follows by the definition of the best linear projections of V_r onto $\widehat{V}_r$ and ϵ_r onto $\widehat{V}_r$ in the research context $Q(\cdot)$. □

C Additional Theoretical Results

In this section, we collect together additional theoretical results that are referenced in the main text.

C.1 Analyzing the Researcher's Sample Average Loss and Sample Moment Condition

C.1.1 The Researcher's Sample Average Loss

To tackle the prediction problem, the researcher calculates the sample average loss of the large language model's predictions on their collected dataset:

$$\frac{1}{N}\sum_{r\in\mathcal{R}}D_r\ell(Y_r,\widehat{m}(r;t))$$

where $N=\sum_{r\in\mathcal{R}}D_r$ is the number of text pieces collected by the researcher. Under Assumption 1(i), for all values d, $Q(D=d,T=t)=\Pi_{\sigma\in\Sigma^*}Q(D_\sigma=d_\sigma,T_\sigma=t_\sigma)$, and therefore $Q(T=t)=\Pi_{\sigma\in\Sigma^*}Q(T_\sigma=t_\sigma)$. We can then write $Q(D=d\,|\,T=t)=\Pi_{\sigma\in\Sigma^*}Q(D_\sigma=d_\sigma\,|\,T_\sigma=t_\sigma)$, and the researcher's sampling distribution over text pieces is also independent but not identically distributed over text pieces, conditional on the large language model's realized training dataset.

Consequently, we can re-interpret the researcher's sampling distribution over text pieces conditional on the large language model's realized training dataset as i.n.i.d sampling from the finite population of text pieces; and the researcher's sample average loss calculates the sample mean of the finite population characteristics $\ell(Y_r,\widehat{m}(r;t))$. Existing results on finite-population inference, such as those given in Abadie et al. (2020), Xu (2020) and Rambachan and Roth (2024), provide regularity conditions under which Equation (2) holds and

$$\frac{1}{N}\sum_{r\in\mathcal{R}}D_r\ell(Y_r;\widehat{m}(r;t))-\frac{1}{\mathbb{E}_Q\left[\sum_{r\in\mathcal{R}}D_r\,|\,T=t\right]}\mathbb{E}_Q\left[\sum_{r\in\mathcal{R}}D_r\ell(Y_r;\widehat{m}(r;t))\,|\,T=t\right]\xrightarrow{p}0,$$

as the number of text pieces grows large.

C.1.2 The Researcher's Sample Moment Condition

To tackle the estimation problem, recall that the researcher would like to calculate the sample moment function using the true economic concept:

$$\frac{1}{N}\sum_{r\in\mathcal{R}}D_rg(V_r,W_r;\theta),$$

where $N=\sum_{r\in\mathcal{R}}D_r$ is the number of text pieces collected by the researcher. Under Assumption 1(i), for all values d, $Q(D=d,T=t)=\Pi_{\sigma\in\Sigma^*}Q(D_\sigma=d_\sigma,T_\sigma=t_\sigma)$ and therefore $Q(D=d)=\Pi_{\sigma\in\Sigma^*}Q(D_\sigma=d_\sigma)$. We can therefore interpret the researcher's sampling distribution over text pieces as independent but not identically distributed sampling from the finite population; and the researcher's sample moment function calculates the sample mean of the finite population characteristic $g(V_r,W_r;\theta)$. As for the researcher's sample average loss, existing results in the finite-population literature imply that

$$\frac{1}{N}\sum_{r\in\mathcal{R}}D_r g(W_r,V_r;\theta)-\frac{1}{\mathbb{E}_Q\left[\sum_{r\in\mathcal{R}}D_r\right]}\mathbb{E}_Q\left[\sum_{r\in\mathcal{R}}D_r g(W_r,V_r;\theta)\right]\xrightarrow{p}0,$$

as the number of text pieces grow large.

Due to the text processing problem, the researcher instead constructs the large large language model's labels of the economic concept and calculates the plug-in, sample moment function:

$$\frac{1}{N}\sum_{r\in\mathcal{R}}D_r g(\widehat{m}(r;t),W_r;\theta).$$

By the same argument, we can interpret the researcher sampling distribution over text pieces conditional on the large language model's realized training dataset as i.n.i.d sampling from the finite population of text pieces; and the researcher's plug-in moment function then calculates the sample mean of the finite population characteristics $g(\widehat{m}(r;t),W_r;\theta)$. Existing results then provide regularity conditions under which Equation (6) holds and

$$\frac{1}{N}\sum_{r\in\mathcal{R}}D_r g(\widehat{m}(r;t),W_r;\theta)-\frac{1}{\mathbb{E}\left[\sum_{r\in\mathcal{R}}D_r\,|\,T=t\right]}\mathbb{E}\left[\sum_{r\in\mathcal{R}}D_r g(\widehat{m}(r;t),W_r;\theta)\,|\,T=t\right]\xrightarrow{p}0$$

as the number of text pieces grow large.

C.2 Analyzing the Asymptotic Distribution of Bias-Corrected Coefficient

In this section, we separately analyze the asymptotic distribution of the bias-corrected regression coefficient introduced in Section 5.4.1 in two separate cases: first, when the economic concept V_r is the dependent variable; and second, when the economic concept V_r is the independent variable.

C.2.1 Linear Regression with Large Language Model Labels as the Dependent Variable

As discussed in Section 5.4.1 of the main text, we now study the limiting distribution of the bias-corrected linear regression in which the researcher uses the economic concept as the dependent variable. It is convenient to now define the researcher's sampling indicator as taking three possible $D_r\in\{0,1,2\}$, where $D_r=0$ denotes the researcher does not sample the text piece r, $D_r=1$ denotes that the researcher samples the text piece in the primary sample and observes $(\widehat{m}(r;t),W_r)$, and $D_r=2$ denotes that the researcher samples the text piece in the validation sample and observes $(\widehat{m}(r;t),V_r,W_r)$. Altogether the researcher observes $(\widehat{m}(r;t),W_r)$ for all $r\in\mathcal{R}$ with $D_r=1$ and $(\widehat{m}(r;t),V_r,W_r)$ for all $r\in\mathcal{R}$ with $D_r=2$.

On the primary sample, the researcher calculate the plug-in regression coefficient

$$\widehat{\beta}=\left(\frac{1}{N_p}\sum_{r\in\mathcal{R}}1\{D_r=1\}W_rW_r'\right)^{-1}\left(\frac{1}{N_p}\sum_{r\in\mathcal{R}}1\{D_r=1\}W_r\widehat{m}(r;t)\right)$$

for $N_p=\sum_r 1\{D_r=1\}$ the size of the primary sample. On the validation sample, the researcher estimates the measurement error regression coefficient

$$\widehat{\lambda}_{\Delta|W}=\left(\frac{1}{N_v}\sum_{r\in\mathcal{R}}1\{D_r=2\}W_rW_r'\right)^{-1}\left(\frac{1}{N_v}\sum_{r\in\mathcal{R}}1\{D_r=2\}W_r\Delta_r\right)$$

for $N_v=\sum_r 1\{D_r=2\}$ the size of the validation sample. The bias-corrected regression coefficient is then given by $\widehat{\beta}^{debiased}=\widehat{\beta}-\widehat{\lambda}$. The researcher's validation-sample only regression coefficient is

$$\widehat{\beta}^{validation}=\left(\frac{1}{N_v}\sum_{r\in\mathcal{R}}1\{D_r=2\}W_rW_r'\right)^{-1}\left(\frac{1}{N_v}\sum_{r\in\mathcal{R}}1\{D_r=2\}W_rV_r\right).$$

As further notation, let $N=N_p+N_v$ denote the size of the researcher's dataset, N_R be the total number of text pieces, and let $N_o=N_R-N$ denote the number of text pieces that are not sampled by the researcher.

To derive the limiting distribution as the number of economically relevant text pieces N_R grows large, we make three simplifying assumptions. First, we assume that W_r is a scalar, which is not technically necessary but will simplify the resulting expressions. Second, we assume the large language model satisfies no training leakage as mentioned in the the main text. Third, we analyze a research context $Q(\cdot)$ in which the researcher randomly samples text pieces into their dataset and further randomly partitions the collected text pieces into the primary and validation sample. More formally, the text pieces are randomly sampled into three groups of size N_o,N_p,N_v respectively and the probability that the vector D takes a particular value d is given by $N_o!N_p!N_v!/N_R!$, where d satisfies $\sum_{r\in\mathcal{R}}1\{D_r=0\}=N_o$, $\sum_{r\in\mathcal{R}}1\{D_r=1\}=N_p$, $\sum_{r\in\mathcal{R}}1\{D_r=2\}=N_v$. Finally, we will assume there exists some finite constant $M>0$ such that $-M\leq W_r,V_r,\widehat{m}(r;t)\leq M$ for all $r\in\mathcal{R}$. The last two assumptions enable us to apply existing finite-population central limit theorem in deriving limiting distributions

We study the properties of the bias-corrected regression and the validation-sample only regression along a sequence of finite populations satisfying $N_R\to\infty$, $N_v/N_R=\rho_v>0$, $N_p/N_R=\rho_p>0$. Under these stated conditions, results in Li and Ding (2017) imply that $\frac{1}{N_p}\sum_{r\in\mathcal{R}}1\{D_r=1\}W_r^2-\frac{1}{N_R}\sum_{r\in\mathcal{R}}W_r^2\xrightarrow{p}0$ and $\frac{1}{N_v}\sum_{r\in\mathcal{R}}1\{D_r=2\}W_r^2-\frac{1}{N_R}\sum_{r\in\mathcal{R}}W_r^2\xrightarrow{p}0$. We therefore focus on analyzing the properties of $\frac{1}{N_p}\sum_{r\in\mathcal{R}}1\{D_r=1\}W_r\widehat{m}(r;t)$ and $\frac{1}{N_v}\sum_{r\in\mathcal{R}}1\{D_r=1\}W_r\Delta_r$.

Towards this, let us define $X_r=W_r\widehat{m}(r;t)$ and $Z_r=W_r\Delta_r$ as convenient shorthand. We then write $\bar{X}_p=\frac{1}{N_p}\sum_{r\in\mathcal{R}}1\{D_r=1\}X_r$ and $\bar{Z}_v=\frac{1}{N_p}\sum_{r\in\mathcal{R}}1\{D_r=2\}Z_r$. Define the finite population quantities $\bar{X}_N=\frac{1}{N_R}\sum_{r\in\mathcal{R}}X_r$ and $\bar{Z}_N=\frac{1}{N_R}\sum_{r\in\mathcal{R}}Z_r$, and $\sigma^2_{X,N}=\frac{1}{N-1}\sum_{r\in\mathcal{R}}(X_r-\bar{X}_N)^2$, $\sigma^2_{Z,N}=\frac{1}{N-1}\sum_{r\in\mathcal{R}}(Z_r-\bar{Z}_N)^2$.

Proposition 2 in Li and Ding (2017) implies that

$$Var_Q((\bar{X}_p,\bar{Z}_v)')=N_R^{-1}\begin{pmatrix}\frac{1-\rho_p}{\rho_p}\sigma^2_{X,N} & \sigma_{X,N}\sigma_{Z,N}\\ \sigma_{X,N}\sigma_{Z,N} & \frac{1-\rho_v}{\rho_v}\sigma^2_{Z,N}\end{pmatrix}.$$

Consequently, provided $\sigma^2_{X,N}\to\sigma^2_X$ and $\sigma^2_{Z,N}\to\sigma^2_Z$ as $N_R\to\infty$, Theorem 5 in Li and Ding (2017) implies that

$$\sqrt{N_R}((\bar{X}_p,\bar{Z}_v)'-(\bar{X}_N,\bar{Z}_N)')\xrightarrow{d}N\left(0,\begin{pmatrix}\frac{1-\rho_p}{\rho_p}\sigma^2_X & -\sigma_X\sigma_Z\\ -\sigma_X\sigma_Z & \frac{1-\rho_v}{\rho_v}\sigma^2_Z\end{pmatrix}\right).$$

We can therefore characterize the limiting distribution of the bias-corrected regression coefficient by an application of Slutsky's theorem and the Delta method. In particular, the previous display implies that

$$\sqrt{N}_R\left(\widehat{\beta}^{debiased}-\beta^*\right)\xrightarrow{d}N(0,\Omega^{debiased})$$

for

$$\Omega^{debiased}=\sigma_W^{-4}\left(\frac{1-\rho_p}{\rho_p}\sigma^2_X+2\sigma_X\sigma_Z+\frac{1-\rho_v}{\rho_v}\sigma^2_Z\right).$$

and σ^2_W the limit of $\frac{1}{N}\sum_{r\in\mathcal{R}}W_r^2$. This delivers Equation (13) given in Section 5.4.1 of the main text. By a similar argument, we can show that the validation-sample only regression coefficient has a limiting distribution given by

$$\sqrt{N}_R\left(\widehat{\beta}^{validation}-\beta^*\right)\xrightarrow{d}N(0,\Omega^{validation})$$

for $\Omega^{validation}=\sigma_W^{-4}\frac{1-\rho_V}{\rho_V}\sigma^2_{WV}$, as stated in Section 5.4.1 of the main text.

C.2.2 Linear Regression with Large Language Model Labels as Covariates

We next discuss how the researcher using the economic concept as a covariate in a linear regression could bias correct their estimates using a small validation sample. Towards this, recall that the target regression and plug-in regression are given by

$$W_r=V_r'\alpha^*+\nu_r, \text{ and } W_r=\widehat{m}(r;t)'\alpha+\widetilde{\nu}_r.$$

The researcher again observes $(\widehat{m}(r;t),W_r)$ for all $r\in\mathcal{R}$ with $D_r=1$ and $(\widehat{m}(r;t),V_r,W_r)$ for all $r\in\mathcal{R}$ with $D_r=2$.

We will estimate the target regression using the validation sample and the primary sample in the following manner. On the primary sample, the researcher separately calculates

$$\widehat{\Sigma}^{primary}_{\hat{V}\hat{V}}=\frac{1}{N_p}\sum_{r\in\mathcal{R}}1\{D_r=1\}\widehat{m}(r;t)\widehat{m}(r;t)' \text{ and } \widehat{\Sigma}^{primary}_{\hat{V}W}=\frac{1}{N_p}\sum_{r\in\mathcal{R}}1\{D_r=1\}\widehat{m}(r;t)W_r.$$

On the validation sample, the researcher separately calculates

$$\widehat{\Lambda}^{validation}_{\hat{V}V}\frac{1}{N_v}\sum_{r\in\mathcal{R}}1\{D_r=2\}(\widehat{m}(r;t)\widehat{m}(r;t)'-V_rV_r') \text{ and } \widehat{\Lambda}^{validation}_{\Delta W}=\frac{1}{N_v}\sum_{r\in\mathcal{R}}1\{D_r=2\}\Delta_rW_r.$$

The bias-corrected regression coefficient is then given by

$$\widehat{\alpha}^{debiased}=\left(\widehat{\Sigma}^{primary}_{\hat{V}\hat{V}}-\widehat{\Lambda}^{validation}_{\hat{V}V}\right)^{-1}\left(\widehat{\Sigma}^{primary}_{\hat{V}W}-\widehat{\Lambda}^{validation}_{\Delta W}\right).$$

The researcher's validation-sample only regression coefficient is

$$\widehat{\alpha}^{validation}=\left(\frac{1}{N_v}\sum_{r\in\mathcal{R}}1\{D_r=2\}V_rV_r'\right)^{-1}\left(\frac{1}{N_v}\sum_{r\in\mathcal{R}}1\{D_r=2\}V_rW_r\right).$$

To analyze the limiting distribution as the number of economically relevant text pieces N_R grows large, we make the same simplifying assumptions as in Appendix Section C.2.1 with the modification that V_r is a scalar. By the same arguments, it can be shown that

$$\sqrt{N_R}\left(\widehat{\alpha}^{debiased}-\alpha^*\right)\xrightarrow{d}N(0,\Omega^{debiased})$$

for

$$\Omega^{debiased}=\sigma_V^{-4}\left(\frac{1-\rho_p}{\rho_p}\sigma^2_{\hat{V}W}+2\sigma_{\hat{V}W}\sigma_{\Delta W}+\frac{1-\rho_v}{\rho_v}\sigma^2_{\Delta W}\right),$$

where $\sigma^2_{\hat{V}W}$, for example, is the finite-population limit of the variance of $\widehat{V}_rW_r$ across text pieces and the remaining terms are defined analogously. It can be analogously shown that

$$\sqrt{N_R}\left(\hat{\alpha}^{validation}-\alpha^*\right)\xrightarrow{d}N(0,\Omega^{validation})$$

for $\Omega^{validation}=\sigma^4_{-V}\frac{1-\rho_v}{\rho_v}\sigma^2_{VW}$. Consequently, we can compare the limiting variances, and again observe that the bias-corrected regression coefficient has a smaller limiting variance if $\frac{1-\rho_p}{\rho_p}\sigma^2_{\widehat{V}W}+2\sigma_{\widehat{V}W}\leq\frac{1-\rho_v}{\rho_v}(\sigma^2_{VW}-\sigma^2_{\Delta W})$. This can be satisfied provided the LLM's errors in reproducing the existing measurement are sufficiently small.

D Additional Monte Carlo Simulations based on Congressional Legislation

In this section, we report additional Monte Carlo simulations based on the data from the Congressional Bills Project (Adler and Wilkerson, 2020; Wilkerson et al., 2023). We first illustrate how the performance of the bias-corrected regression coefficient varies with the size of the validation data. We further illustrate that the performance of the bias-corrected regression when the economic concept is used as a covariate in the linear regression, as described in Appendix C.2.2.

D.1 Varying the Size of the Validation Data

In Section 5.4 of the main text, we evaluated the performance of the plug-in regression coefficient against the bias-corrected estimator using a 5% validation sample. We explore how the performance of the bias-corrected regression coefficient varies as we vary the size of the validation sample.

As in the main text, for a given bill topic V_r, covariate W_r, and pair of large language model and prompting strategy, we randomly draw a sample of 5,000 bills. On this random sample, we first calculate the plug-in regression coefficient $\widehat{\beta}$. We next randomly reveal the ground-truth label V_r on 2.5% (125 bills), 5% (250 bills), 10% (500 bills), 25% (1250 bills), and 50% (2500 bills) of the random sample of 5,000 bills. We then calculate the bias-corrected coefficient $\widehat{\beta}^{debiased}$ on each validation sample. We repeat these steps for 1,000 randomly sampled datasets, and we calculate the average bias of these alternative estimates for the target regression β^* of the ground-truth concept V_r on the chosen covariate W_r on all 10,000 bills as well as the coverage of conventional confidence intervals. We repeat this exercise for each possible combination of bill topic V_r, linked covariate W_r, large language model m (either GPT-3.5-Turbo or GPT-4o), and prompting strategy p. This allows us to summarize how the plug-in regression performs against the bias-corrected regression across a wide variety of possible regression specifications, choices of large language model and prompting strategies.

Figure A5 illustrates the distribution of normalized bias across possible combinations of bill topic V_r, linked covariate W_r, large language model m, and prompting strategy p, as the size of the validation sample changes. The top panels of Table A4 and Table A5 report summary statistics for labels produced by GPT-3.5-Turbo and GPT-4o respectively. While we often see severe biases for the plug-in regression, by contrast the bias-corrected regression coefficient is on average equal to the target regression coefficient for all sizes of the validation sample.

The bottom panels of Table A4 and Table A5 provide summary statistics of the coverage of conventional confidence intervals for the target regression. We see substantial coverage distortions for the plug-in regression, whereas the bias-corrected regression delivers approximately correct coverage for all sizes of the validation sample.

Finally, Figure A6 compares the mean square error of the bias-corrected coefficient versus the validation-sample only estimate of the target regression as we vary the size of the validation sample. The bias-corrected coefficient obtains noticeable improvements in mean square error for the validation proportions equal to 2.5%, 5% and 10%. The bias-corrected coefficient performs similarly to the validation-sample only estimator for validation proportions equal to 25%, although it is likely unrealistic that the researcher would collect such large validation samples in an empirical application.

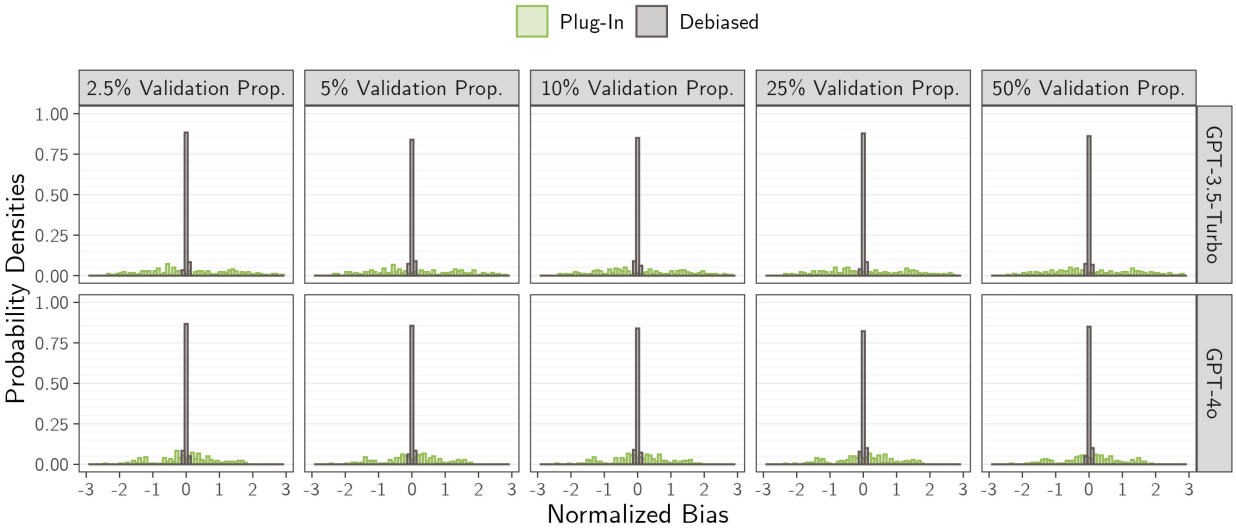


Figure A5: Normalized bias of the plug-in regression and bias-corrected regression across Monte Carlo simulations based on congressional legislation as the validation sample size varies.

Notes: The normalized bias reports the average bias of the plug-in regression coefficient $\widehat{\beta}$ and the bias-corrected coefficient $\widehat{\beta}^{debiased}$ for the target regression coefficient divided by their respective standard deviations across simulations. For each combination of model topic V_r, covariate W_r, large language model m and prompting strategy p, we randomly sample 5,000 Congressional bills and calculate the plug-in regression coefficient $\widehat{\beta}$ and the bias-corrected regression coefficient $\widehat{\beta}^{debiased}$. We vary the size of the validation sample over 2.5%, 5%, 10%, 25% and 50%. Results are averaged over 1,000 simulations. We summarize the distribution of normalized bias and coverage across regression specifications, choice of large language model and prompting strategies. See Appendix Section D.1 for discussion.

Validation Prop.	Median	5%	95%
Normalized Bias			
2.5%	-0.015	-1.907	2.159
5%	-0.023	-1.899	2.211
10%	-0.005	-1.863	2.179
25%	0.007	-1.837	2.264
50%	-0.004	-1.864	2.172
Coverage			
2.5%	0.806	0.381	0.946
5%	0.820	0.381	0.945
10%	0.812	0.383	0.949
25%	0.815	0.362	0.945
50%	0.816	0.369	0.950

(a) Plug-in regression

Validation Prop.	Median	5%	95%
Normalized Bias			
2.5%	0.003	-0.039	0.050
5%	0.001	-0.055	0.066
10%	-0.005	-0.053	0.049
25%	0.002	-0.044	0.053
50%	0.000	-0.050	0.049
Coverage			
2.5%	0.901	0.862	0.927
5%	0.930	0.910	0.945
10%	0.941	0.927	0.952
25%	0.946	0.934	0.957
50%	0.948	0.934	0.959

(b) Debiased regression

Table A4: Summary statistics for normalized bias and coverage for Monte Carlo simulations on congressional legislation for GPT-3.5-Turbo, varying the size of the validation sample

Notes: The normalized bias reports the average bias of the plug-in regression coefficient $\widehat{\beta}$ and the bias-corrected coefficient $\widehat{\beta}^{debiased}$ for the target regression coefficient divided by their respective standard deviations across simulations. The coverage reports the fraction of simulations in which a 95% nominal confidence interval centered around the plug-in regression coefficient $\widehat{\beta}$ and the bias-corrected coefficient $\widehat{\beta}^{debiased}$ cover the target regression coefficient β^*. For each combination of model topic V_r, covariate W_r, large language model m and prompting strategy p, we randomly sample 5,000 Congressional bills and calculate the plug-in regression coefficient $\widehat{\beta}$ and the bias-corrected regression coefficient $\widehat{\beta}^{debiased}$. We vary the size of the validation sample over 2.5%, 5%, 10%, 25% and 50%. Results are averaged over 1,000 simulations. We summarize the distribution of normalized bias and coverage across regression specifications, choice of large language model and prompting strategies. See Appendix Section D.1 for discussion.

Validation Prop.	Median	5%	95%
Normalized Bias			
2.5%	0.051	-1.456	1.540
5%	0.084	-1.411	1.514
10%	0.059	-1.447	1.507
25%	0.056	-1.422	1.463
50%	0.070	-1.441	1.510
Coverage			
2.5%	0.920	0.630	0.954
5%	0.920	0.637	0.954
10%	0.919	0.642	0.952
25%	0.920	0.625	0.954
50%	0.919	0.635	0.950

(a) Plug-in regression

Validation Prop.	Median	5%	95%
Normalized Bias			
2.5%	0.000	-0.058	0.046
5%	0.001	-0.055	0.054
10%	0.000	-0.066	0.050
25%	-0.001	-0.053	0.060
50%	-0.001	-0.045	0.059
Coverage			
2.5%	0.893	0.846	0.926
5%	0.927	0.902	0.945
10%	0.941	0.926	0.953
25%	0.946	0.934	0.958
50%	0.948	0.935	0.959

(b) Debiased regression

Table A5: Summary statistics for normalized bias and coverage for Monte Carlo simulations on congressional legislation, varying the size of the validation sample (GPT-4o)

Notes: The normalized bias reports the average bias of the plug-in regression coefficient $\widehat{\beta}$ and the debiased coefficient $\widehat{\beta}^{debiased}$ for the target regression coefficient divided by their respective standard deviations across simulations. The coverage reports the fraction of simulations in which a 95% nominal confidence interval centered around the plug-in regression coefficient $\widehat{\beta}$ and the debiased coefficient $\widehat{\beta}^{debiased}$ cover the target regression coefficient. For each combination of model topic V_r, covariate W_r, large language model m and prompting strategy p, we randomly sample 5,000 Congressional bills and calculate the plug-in regression $\widehat{V}_r^{m,p} = \alpha + \beta W_r$ and the debiased regression coefficient. We vary the size of the validation sample over 2.5%, 5%, 10%, 25% and 50%. Results are averaged over 1,000 simulations. See Appendix Section D.1 for discussion.

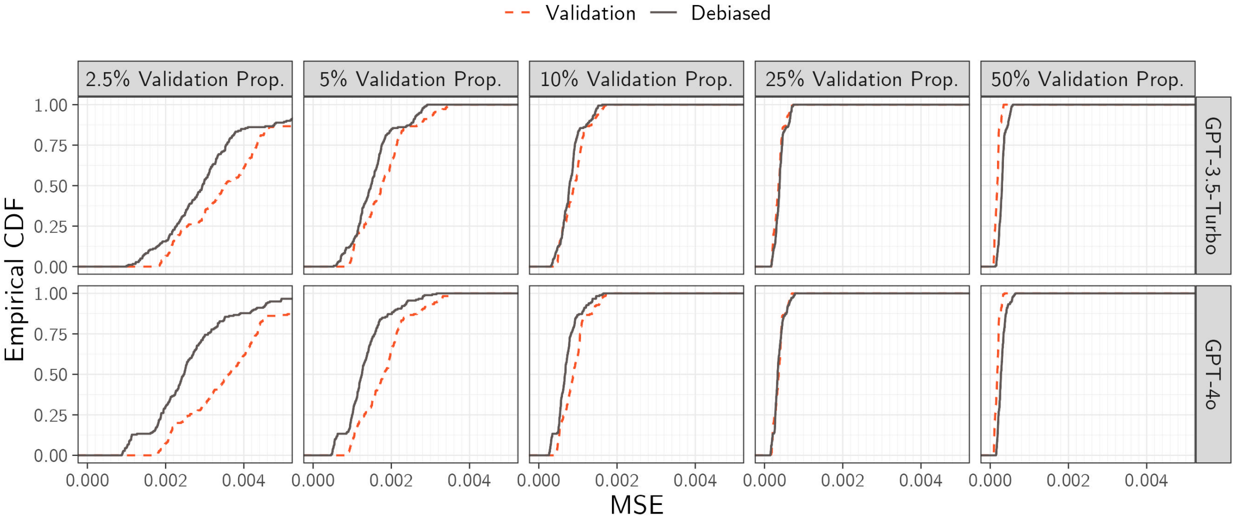


Figure A6: Cumulative distribution function of mean square error for the bias-corrected estimator against validation-sample only estimator, varying the size of the validation sample.

Notes: For each combination of model topic V_r, covariate W_r, large language model m and prompting strategy p, we randomly sample 5,000 Congressional bills and calculate the bias-corrected regression coefficient $\widehat{\beta}^{debiased}$ and the validation-sample only regression coefficient $\widehat{\beta}^{*}$. We calculate the mean square error of $\widehat{\beta}^{debiased}$ and $\widehat{\beta}^{*}$ for the target regression. We vary the size of the validation sample over 2.5%, 5%, 10%, 25% and 50%, and we average the results over 1,000 simulations. We summarize the distribution of average mean square error across regression specifications, choice of large language model and prompting strategies. See Appendix Section D.1 for discussion.

D.2 Large Language Model Labels as Covariates

In this section, we extend our analysis using data from the Congressional Bills Project to explore the biases that can arise from using large language model labels as covariates in a linear regression and whether the resulting biases can be corrected using a small collection of validation data.

As a first step, we use the sample random sample of 10,000 Congressional bills from the main text, and we now regress alternative linked economic variables on dummy indicators for the large language model's labeled economic concept – in this case, the policy topic of the bill. For alternative dependent variables such as whether the bill's sponsor was a Democrat, whether the bill originated in the Senate, and the DW1 score of the bill's sponsor, we run the regression $W_r = \widehat{V}_r^{m,p}\beta + \epsilon$ for each possible pair of large language model m and prompting strategy p. In Figure A7, each row considers a different regression for a linked covariate W_r as the dependent variable, and each column plots the t-statistic for different large language model labels $\widehat{V}_r^{m,p}$ associated with alternative bill topics (e.g., Health, Banking, Finance and Domestic Commerce, Defense, Government Operations, and Public Lands and Water Management). For every combination of the linked variable W_r and policy topic area, we see substantial variation in the t-statistics across alternative large language models and prompting strategies. Table A6 summarizes the coefficient estimates across models and prompts for each choice of labeled policy topic and the covariate.

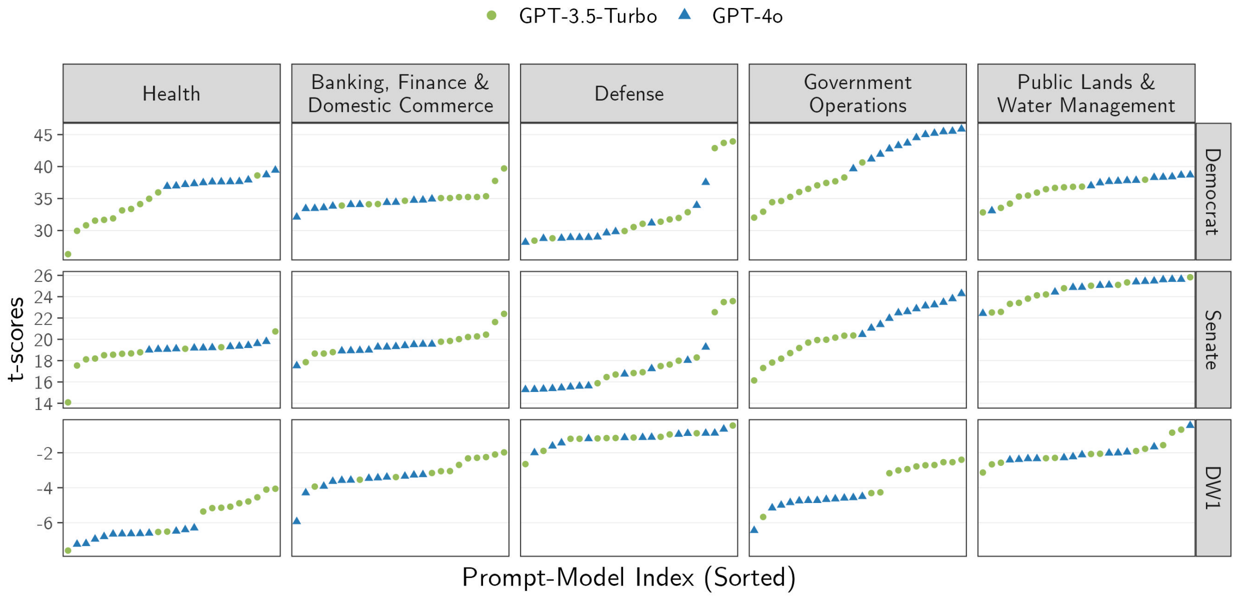


Figure A7: Variation in t-statistics across large language models and prompting strategies on congressional legislation, using the economic concept as a covariate.

Notes: On 10,000 Congressional bills, we prompt GPT-3.5-Turbo and GPT-4o to label each description for its policy topic area using alternative prompting strategies. For each model m and prompt p, we regress a linked variable W_r (whether the bill's sponsor was a Democrat, whether the bill originated in the Senate, and the DW1 score of the bill's sponsor) on indicators $\widehat{V}_r^{m,p}$ for the large language model's labeled policy topic (i.e., Health; Banking, Finance, and Domestic Commerce; Defense; Government Operations; Public Lands and Water Management; Other). In each subplot, the t-statistic estimates are sorted in ascending order for clarity. See Appendix Section D.2 for discussion.

We next explore whether these biases can be addressed by collecting a small validation sample

Covariate	Policy Topic	*Point Estimates*				Sample
		Mean	Median	5%	95%	Average
DW1	Health	-0.091	-0.095	-0.102	-0.076	-0.062
DW1	Banking, Finance & Domestic Com.	-0.043	-0.043	-0.056	-0.027	-0.062
DW1	Defense	-0.015	-0.016	-0.023	-0.009	-0.062
DW1	Government Operations	-0.047	-0.048	-0.066	-0.033	-0.062
DW1	Public Lands & Water Management	-0.024	-0.024	-0.033	-0.009	-0.062
Democrat	Health	0.636	0.642	0.614	0.651	0.604
Democrat	Banking, Finance & Domestic Com.	0.582	0.582	0.567	0.597	0.604
Democrat	Defense	0.586	0.586	0.576	0.598	0.604
Democrat	Government Operations	0.601	0.598	0.588	0.620	0.604
Democrat	Public Lands & Water Management	0.588	0.587	0.579	0.599	0.604
Senate	Health	0.334	0.331	0.319	0.364	0.317
Senate	Banking, Finance & Domestic Com.	0.304	0.305	0.293	0.314	0.317
Senate	Defense	0.295	0.295	0.280	0.308	0.317
Senate	Government Operations	0.290	0.291	0.280	0.301	0.317
Senate	Public Lands & Water Management	0.391	0.390	0.383	0.404	0.317

Table A6: Variation in point estimates across large language models and prompting strategies on Congressional bills, using the economic concept as a covariate.

Notes: On 10,000 Congressional bills, we prompt GPT-3.5-Turbo and GPT-4o to label each description for its policy topic area using alternative prompting strategies. For each model m and prompt p, we regress a linked variable W_r (whether the bill's sponsor was a Democrat, whether the bill originated in the Senate, and the DW1 score of the bill's sponsor) on indicators for whether the large language model labeled a particular policy topic $1\{\widehat{V}_r^{m,p} = v\}$, focusing on Health, Banking, Finance & Domestic Commerce, Defense, Government Operations, and Public Lands & Water Management. The final column ("Sample Average") reports the average of the linked variable W_r across all Congressional bills. See Appendix Section D.2 for discussion.

and implementing the bias-corrected procedure described in Appendix C.2.2 can address these issues. We leverage the same Monte Carlo simulation design as described in Section 5.4.2 of the main text. For each linked variable W_r and pair of large language model m and prompting strategy p, we randomly sample 5,000 bills from our dataset of 10,000 bills. On this random sample of 5,000 bills, we first calculate the plug-in coefficients $\widehat{\beta}$ from regressing W_r on $\widehat{V}_r^{m,p}$ (, for $\widehat{V}_r^{m,p}$ a vector of indicators for the labeled policy topic). We next randomly reveal the ground-truth label V_r on 5% of our random sample of 5,000 bills, which produces a validation sample. We calculate the bias-corrected coefficients $\widehat{\beta}^{debiased}$ as described in Appendix C.2.2. We repeat these steps for 1,000 randomly sampled datasets. We repeat this exercise for each possible combination of linked variable W_r, large language model m (either GPT-3.5-turbo or GPT-4o) and prompting strategy p. This allows to summarize how the plug-in regression performs against the bias-corrected regression across a wide variety of possible regression specifications, choices of large language model and prompting strategies.

Figure A8 and Table A7 summarizes our results. The plug-in regression suffers from substantial biases for almost all combinations of linked variable W_r, large language model m (either GPT-3.5-turbo and GPT-4o) and prompting strategy p. By contrast, using the validation sample for bias correction effectively eliminates these biases. Furthermore, the bottom panel of Table A7 further illustrates the coverage comparison between the plug-in regression and the bias-corrected estimator — while confidence intervals centered at the plug-in regression are significantly distorted, bias-correction restores nominal coverage.

Finally, Figure A9 compares the mean square error of the bias-corrected regression against directly estimating the target regression on the validation sample. For many regression specifications, choices of language model and prompting strategies, we again find that the MSE of the bias-corrected regression is smaller than that of the validation-sample only regression.

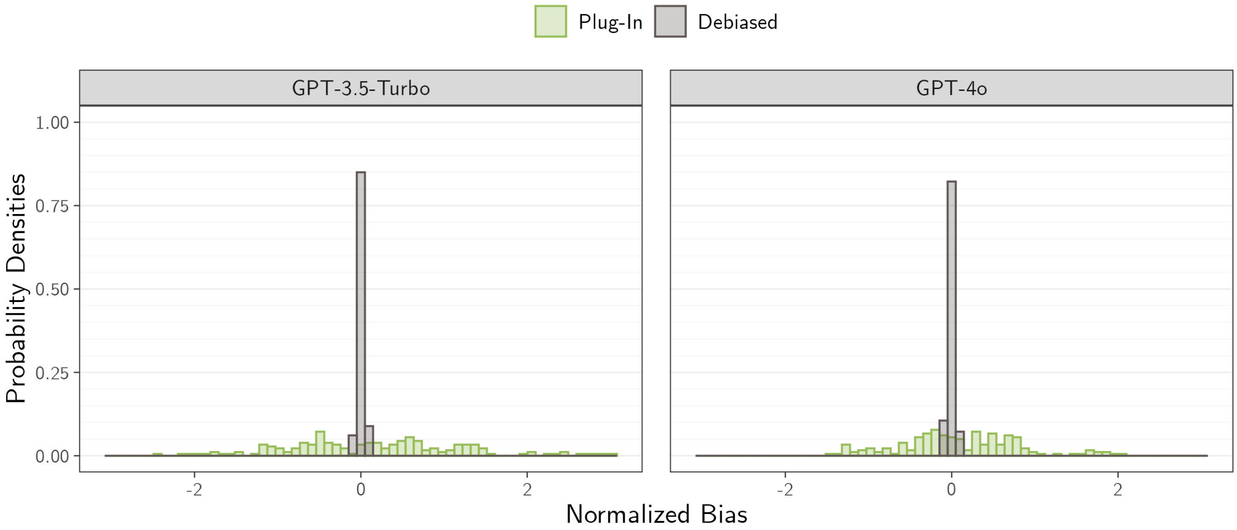


Figure A8: Normalized bias of the plug-in regression and bias-corrected regression using policy topic as a covariate across Monte Carlo simulations based on congressional legislation

Notes: The normalized bias reports the average bias of the plug-in regression coefficient $\widehat{\beta}$ and the bias-corrected coefficient $\widehat{\beta}^{debiased}$ for the target regression coefficient divided by their respective standard deviations across simulations. For each combination of left hand side variable W_r, large language model m and prompting strategy p, we randomly sample 5,000 Congressional bills and calculate the plug-in regression $\widehat{\beta}$ and bias-corrected coefficient $\widehat{\beta}^{*}$ with the policy topic as a covariate using a 5% validation sample. Results are averaged over 1,000 simulations. See Appendix Section D.2 for discussion.

D.2.1 Varying the Size of the Validation Data:

We evaluated the performance of bias-correcting linear regression that use large language model labels as covariates using a 5% validation sample. We finally explore how the performance of the bias-corrected regression coefficient varies as we vary the size of the validation sample. We repeat our Monte Carlo simulations now varying the size of the validation sample by randomly revealing the measurements V_r on 2.5% (125 bills), 5% (250 bills), 10% (500 bills), 25% (1250 bills), and 50% (2500 bills) of the random sample of 5,000 bills. The results are summarized in Figure A10, Table A8, Table A9 and Figure A11. We continue to find that the bias-corrected regression performs well in finite samples, even when the validation sample only contains 125 bills.

	Median	5%	95%
Normalized Bias			
Plug-In	0.144	-1.514	2.083
Debiased	0.002	-0.051	0.060
Coverage			
Plug-In	0.897	0.363	0.950
Debiased	0.933	0.911	0.956

(a) GPT-3.5-Turbo

	Median	5%	95%
Normalized Bias			
Plug-In	0.042	-1.146	1.558
Debiased	-0.003	-0.063	0.053
Coverage			
Plug-In	0.928	0.640	0.957
Debiased	0.930	0.900	0.952

(b) GPT-4o

Table A7: Summary statistics for normalized bias and coverage for Monte Carlo simulations on congressional legislation using policy topic as a covariate.

Notes: The normalized bias reports the average bias of the plug-in regression coefficient $\widehat{\beta}$ and the debiased coefficient $\widehat{\beta}^{debiased}$ for the target regression coefficient divided by their respective standard deviations across simulations. The coverage reports the fraction of simulations in which a 95% nominal confidence interval centered around the plug-in regression coefficient $\widehat{\beta}$ and the debiased coefficient $\widehat{\beta}^{debiased}$ cover the target regression coefficient. For each combination of left hand side variable W_r, large language model m and prompting strategy p, we randomly sample 5,000 Congressional bills and calculate the plug-in regression $\widehat{\beta}$ and bias-corrected coefficient $\widehat{\beta}^*$ with the policy topic as a covariate using a 5% validation sample. Results are averaged over 1,000 simulations. See Appendix Section D.2 for discussion.

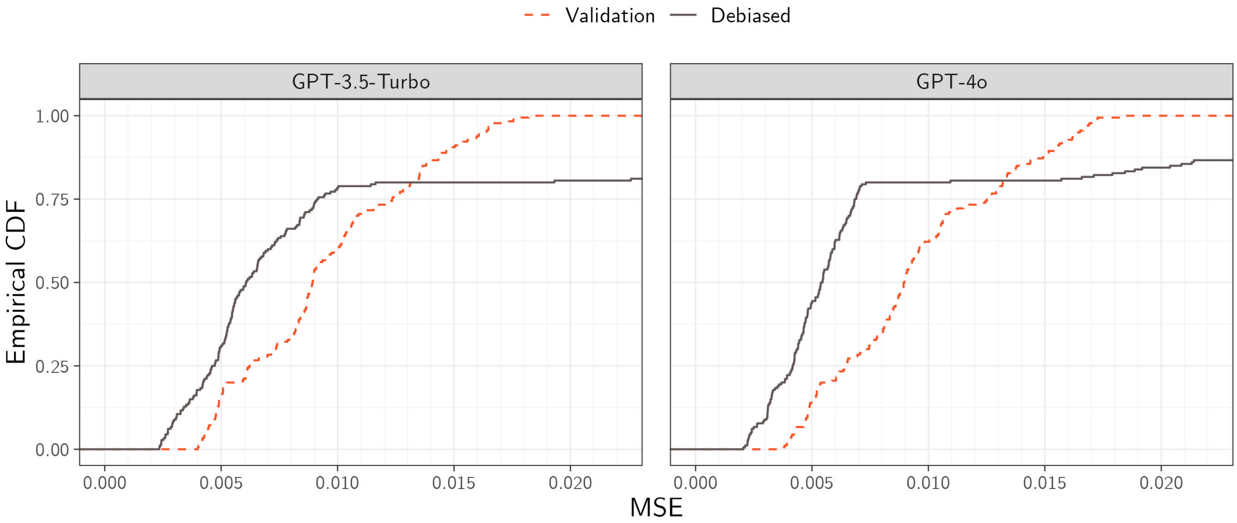


Figure A9: Cumulative distribution function of mean square error for the bias-corrected estimator against validation-sample only estimator using policy topic as a covariate.

Notes: For each combination of left hand side variable W_r, large language model m and prompting strategy p, we randomly sample 5,000 Congressional bills and calculate the plug-in regression $\widehat{\beta}$ and bias-corrected coefficient $\widehat{\beta}^*$ with the policy topic as a covariate using a 5% validation sample. We calculate the mean square error of $\widehat{\beta}^{debiased}$ and $\widehat{\beta}^*$ for the target regression β^*. Results are averaged over 1,000 simulations. We summarize the distribution of average mean square error across regression specifications, choice of large language model and prompting strategies. See Appendix Section D.2 for discussion.

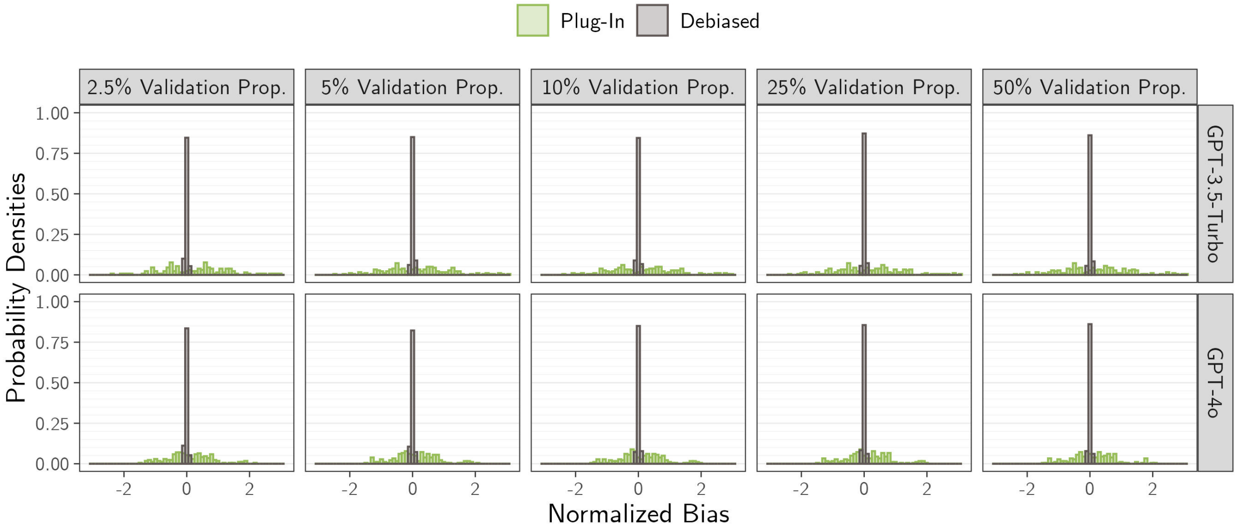


Figure A10: Normalized bias of the plug-in regression and bias-corrected regression using policy topic as a covariate as the validation sample size varies.

Notes: The normalized bias reports the average bias of the plug-in regression coefficient $\widehat{\beta}$ and the bias-corrected coefficient $\widehat{\beta}^{debiased}$ for the target regression coefficient divided by their respective standard deviations across simulations. For each combination of left hand side variable W_r, large language model m and prompting strategy p, we randomly sample 5,000 Congressional bills and calculate the plug-in regression $\widehat{\beta}$ and bias-corrected coefficient $\widehat{\beta}^*$ with the policy topic as a covariate. We vary the size of the validation sample over 2.5%, 5%, 10%, 25% and 50%. We average the results over 1,000 simulations. Results are averaged over 1,000 simulations. See Appendix Section D.2 for discussion.

Validation Prop.	Median	5%	95%
Normalized Bias			
2.5%	0.326	-1.289	2.171
5%	0.144	-1.514	2.083
10%	0.147	-1.437	2.099
25%	0.136	-1.469	2.024
50%	0.127	-1.486	2.043
Coverage			
2.5%	0.900	0.360	0.951
5%	0.897	0.363	0.950
10%	0.897	0.353	0.949
25%	0.893	0.391	0.948
50%	0.893	0.368	0.951

(a) Plug-in regression

Validation Prop.	Median	5%	95%
Normalized Bias			
2.5%	-0.002	-0.058	0.051
5%	0.002	-0.051	0.060
10%	-0.002	-0.055	0.057
25%	-0.001	-0.053	0.052
50%	-0.002	-0.055	0.056
Coverage			
2.5%	0.909	0.867	0.960
5%	0.933	0.911	0.956
10%	0.941	0.927	0.954
25%	0.946	0.935	0.958
50%	0.947	0.936	0.959

(b) Debiased regression

Table A8: Summary statistics for normalized bias and coverage using policy topic as a covariate for GPT-3.5-Turbo, varying the size of the validation sample.

Notes: The normalized bias reports the average bias of the plug-in regression coefficient $\widehat{\beta}$ and the debiased coefficient $\widehat{\beta}^{debiased}$ for the target regression coefficient divided by their respective standard deviations across simulations. The coverage reports the fraction of simulations in which a 95% nominal confidence interval centered around the plug-in regression coefficient $\widehat{\beta}$ and the debiased coefficient $\widehat{\beta}^{debiased}$ cover the target regression coefficient. For each combination of left hand side variable W_r, large language model m and prompting strategy p, we randomly sample 5,000 Congressional bills and calculate the plug-in regression $\widehat{\beta}$ and bias-corrected coefficient $\widehat{\beta}^{*}$ with the policy topic as a covariate. We vary the size of the validation sample over 2.5%, 5%, 10%, 25% and 50%. Results are averaged over 1,000 simulations. See Appendix Section D.2 for discussion.

Validation Prop.	Median	5%	95%
Normalized Bias			
2.5%	0.015	-1.073	1.471
5%	0.042	-1.146	1.558
10%	0.024	-1.118	1.624
25%	0.039	-1.145	1.541
50%	0.029	-1.111	1.490
Coverage			
2.5%	0.928	0.681	0.955
5%	0.928	0.640	0.957
10%	0.927	0.642	0.952
25%	0.926	0.639	0.953
50%	0.922	0.648	0.953

(a) Plug-in regression

Validation Prop.	Median	5%	95%
Normalized Bias			
2.5%	-0.003	-0.065	0.050
5%	-0.003	-0.063	0.053
10%	-0.001	-0.059	0.055
25%	-0.002	-0.062	0.053
50%	-0.003	-0.057	0.052
Coverage			
2.5%	0.903	0.859	0.948
5%	0.930	0.900	0.952
10%	0.943	0.928	0.952
25%	0.947	0.933	0.957
50%	0.949	0.935	0.959

(b) Debiased regression

Table A9: Summary statistics for normalized bias and coverage using policy topic as a covariate for GPT-4o, varying the size of the validation sample.

Notes: The normalized bias reports the average bias of the plug-in regression coefficient $\widehat{\beta}$ and the debiased coefficient $\widehat{\beta}^{debiased}$ for the target regression coefficient divided by their respective standard deviations across simulations. The coverage reports the fraction of simulations in which a 95% nominal confidence interval centered around the plug-in regression coefficient $\widehat{\beta}$ and the debiased coefficient $\widehat{\beta}^{debiased}$ cover the target regression coefficient. For each combination of left hand side variable W_r, large language model m and prompting strategy p, we randomly sample 5,000 Congressional bills and calculate the plug-in regression $\widehat{\beta}$ and bias-corrected coefficient $\widehat{\beta}^*$ with the policy topic as a covariate. We vary the size of the validation sample over 2.5%, 5%, 10%, 25% and 50%. Results are averaged over 1,000 simulations. See Appendix Section D.2 for discussion.

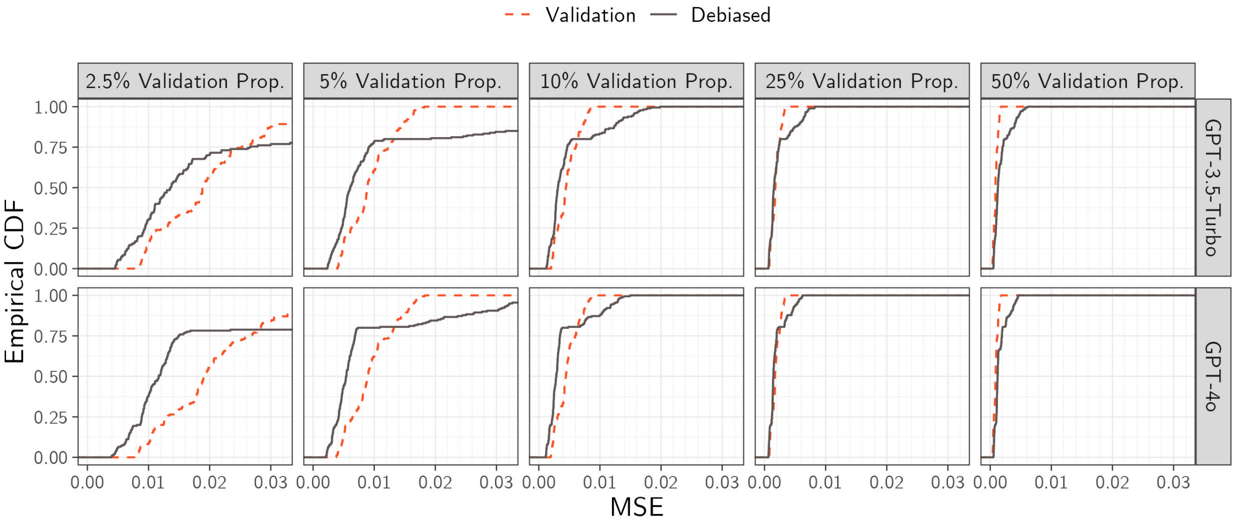


Figure A11: Cumulative distribution function of mean square error for the bias-corrected estimator against validation-sample only estimator using policy topic as the validation sample size varies.

Notes: For each combination of left hand side variable W_r, large language model m and prompting strategy p, we randomly sample 5,000 Congressional bills and calculate the plug-in regression $\widehat{\beta}$ and bias-corrected coefficient $\widehat{\beta}^*$ with the policy topic as a covariate. We calculate the mean square error of $\widehat{\beta}^{debiased}$ and $\widehat{\beta}^*$ for the target regression β^*. We vary the size of the validation sample over 2.5%, 5%, 10%, 25% and 50%, and we average the results over 1,000 simulations. We summarize the distribution of average mean square error across regression specifications, choice of large language model and prompting strategies. See Appendix Section D.2 for discussion.

E Prompts for Congressional Bills and Financial News Headlines

E.1 Prompts for Prediction on Congressional Legislation

```
User: Here is the summary for U.S. Congress bill [BillID]:

[Description]

To the best of your ability, please answer the following: 1. Is it likely that this bill will pass the Senate?
2. Is it likely that this bill will pass the House of Representatives?

Return the answers in the following JSON format:
{
"PassSLLM": 1 if the bill is likely to pass the Senate, otherwise 0,
"PassHLLM": 1 if the bill is likely to pass the House of Representatives, otherwise 0
}
```

(a) Base prompt

```
User: Here is the summary for U.S. Congress bill [BillID]:

[Description]

To the best of your ability, please answer the following, using information available up until [IntrDate]. Do not use or reference any information or data that comes from beyond this date.
1. Is it likely that this bill will pass the Senate?
2. Is it likely that this bill will pass the House of Representatives?

Return the answers in the following JSON format:
{
"PassSLLM": 1 if the bill is likely to pass the Senate, otherwise 0,
"PassHLLM": 1 if the bill is likely to pass the House of Representatives, otherwise 0
}
```

(b) Prompt with date restriction

Figure A12: Prompts used for prediction based on large language models with Congressional legislation.

Notes: This figure documents the prompts used for the prediction exercise based on congressional legislation. We prompt GPT-4o to predict whether 10,000 randomly selected congressional bills would pass the Senate or the House based on its text description. For each Congressional bill, we include its identifier [BillID], its text description [Description], and its introduction date [IntrDate] in the prompts. Figure (a) provides the base prompt, and Figure (b) provides the base prompt with the additional date restriction. See Section 4.3.1 of the main text for further details.

```
User: Here is the beginning of
the summary for U.S. Congress
bill [BillID]:

[Description]

To the best of your ability,
please complete the summary of
this bill.

Do not modify or paraphrase
the provided portion of the
bill summary and only complete
it starting from where it ends.
Only return the remaining part
of the bill summary in the following JSON format:
{
"DescriptionLLM": "[Remaining
summary text]"
}
```

(a) Base prompt

```
User: Here is the beginning of
the summary for U.S. Congress
bill [BillID]:

[Description]

To the best of your ability,
please complete the summary of
this bill using information
available up until [IntrDate].
Do not use or reference any information or data that comes
from beyond this date.

Do not modify or paraphrase
the provided portion of the
bill summary and only complete
it starting from where it ends.
Only return the remaining part
of the bill summary in the following JSON format:
{
"DescriptionLLM": "[Remaining
summary text]"
}
```

(b) Prompt with date restriction

Figure A13: Prompts used for text completion exercise based on large language models with Congressional legislation.

Notes: This figure documents the prompts used for the text completion exercise based on congressional legislation. We prompt GPT-4o to complete the description of 10,000 randomly selected congressional bills based on a segment of its text. For each Congressional bill, we include its identifier [BillID], the beginning of its text description [Description], and its introduction date [IntrDate] in the prompts. Figure (a) provides the base prompt, and Figure (b) provides the base prompt with the additional date restriction. See Section 4.3.1 of the main text for further details.

E.2 Prompts for Prediction on Financial News Headline

```
User: Here is the beginning of a news headline about [company_name] on [date]:

[headline]

To the best of your ability, please complete this headline.

Do not modify or paraphrase the provided portion. Only complete the headline starting from where it ends, and return the remaining part in the following JSON format:
{
"headline_llm": "[Remaining headline text]"
}
```

(a) Base prompt

```
User: Here is the beginning of a news headline about [company_name] on [date]:

[headline]

To the best of your ability, please complete this headline using information available up until [date]. Do not reference or use any data beyond this date.

Do not modify or paraphrase the provided portion. Only complete the headline starting from where it ends, and return the remaining part in the following JSON format:
{
"headline_llm": "[Remaining headline text]"
}
```

(b) Prompt with date restriction

Figure A14: Prompts used for text completion exercise based on large language models with financial news headlines.

Notes: This figure documents the prompts used for the text completion exercise based on financial news headlines. We prompt GPT-4o to complete 10,000 randomly selected financial news headline based on a segment of its text. For each financial news headline, we include the name of the company it is about [company_name], its publication date [date], and the beginning of its text [headline]. Figure (a) provides the base prompt, and Figure (b) provides the base prompt with the additional date restriction. See Section 4.3.2 of the main text for further details.

E.3 Prompts for Plug-In Estimation on Financial News Headlines

User: Here is a piece of news about [company_name]:

[headline]

Is this news positive, negative, or neutral about company?

Write your answer as:
_____ (fill in with one of positive/negative/neutral),
_____ (fill in with numerical value for confidence (0-1), 3 characters maximum),
_____ (fill in with numerical value for magnitude of positive/negative (0-1), 3 characters maximum)

(a) Base prompt with fill-in-the-blanks output

User: Here is a piece of news about [company_name]:

[headline]

Is this news positive, negative, or neutral about company?

Write your answer. Output a JSON object structured like:

```
{
"headline type": "positive" or "negative" or "neutral",
"confidence": 0-1 value of your confidence in the headline type,
"magnitude": 0-1 value for magnitude of positive or negative for the headline type
}
```

(b) Base prompt with JSON output

Figure A15: Base prompts for labeling financial news headlines with large language models.

Notes: This figure documents the base prompts used for labeling financial news headlines with large language models. We prompt GPT-3.5-Turbo, GPT-4o, and GPT-4o-mini to label financial news headlines for whether they are positive, negative or neutral about the associated company. For each financial news headline, we include the name of the company it is about [company_name] and the text of the headline [headline]. Figure (a) provides the base prompt with fill-in-the-blanks output, and Figure (b) provides the base prompt with JSON output. See Section 5.3.2 of the main text for further details.

User: `You are a knowledgeable economic agent.`

(a) Economic agent persona

User: `Answer this question as if you are an expert in finance.`

(b) Finance expert persona

User: `Answer this question as if you are an expert in the economy.`

(c) Economy expert persona

User: `Answer this question as if you were very knowledgeable about financial matters and in particular the stock market. So you are as knowledgeable as an analyst or trader at a very successful Wall Street Firm.`

(d) Successful trader persona

Figure A16: Persona modifications to the base prompt for labeling financial news headlines with large language models.

Notes: This figure documents the persona modifications to the base prompts for labeling financial news headlines with large language models. We prompt GPT-3.5-Turbo, GPT-4o, and GPT-4o-mini to label financial news headlines for whether they are positive, negative or neutral about the associated company. Each persona modification is added to the beginning of the base prompt with JSON output (Panel (a) of Appendix Figure A15). See Section 5.3.2 of the main text for further details.

User: Think carefully. Write your answer as:
_____ (fill in with one of positive/negative/neutral),
_____ (fill in with numerical value for confidence (0-1), 3 characters maximum),
_____ (fill in with numerical value for magnitude of positive/negative (0-1), 3 characters maximum)
_____ fill in with explanation (less than 25 words)

(a) Think carefully prompt

User: Please provide an explanation for your answer. Write your answer as:
_____ (fill in with one of positive/negative/neutral),
_____ (fill in with numerical value for confidence (0-1), 3 characters maximum),
_____ (fill in with numerical value for magnitude of positive/negative (0-1), 3 characters maximum)
_____ fill in with explanation (less than 25 words)

(b) Explanation prompt

User: Think step by step. Lay out each step. Write your answer as:
_____ (fill in with one of positive/negative/neutral),
_____ (fill in with numerical value for confidence (0-1), 3 characters maximum),
_____ (fill in with numerical value for magnitude of positive/negative (0-1), 3 characters maximum)
_____ fill in with explanation (less than 25 words)

(c) Think step by step prompt

Figure A17: Chain of thought modifications to the base prompt for labeling financial news headlines with large language models.

Notes: This figure documents the chain of thought modifications to the base prompts for labeling financial news headlines with large language models. We prompt GPT-3.5-Turbo, GPT-4o, and GPT-4o-mini to label financial news headlines for whether they are positive, negative or neutral about the associated company. Each chain-of-though modification alters the base prompt with JSON output (Panel (a) of Appendix Figure A15). See Section 5.3.2 of the main text for further details.

E.4 Prompts for Plug-In Estimation on Congressional Legislation

```
User: Here is a description of a bill introduced in the U.S. Congress:
[Description]

Please classify this description into one of the following categories:
1.  Macroeconomics
2.  Civil Rights, Minority Issues, and Civil Liberties
3.  Health
⋮
18.  International Affairs and Foreign Aid
19.  Government Operations
20.  Public Lands and Water Management

Write your answer as:
_____ (fill in with an integer from 1 to 20 that best represents the bill category),
_____ (fill in with confidence level in the bill classification as a number between 0 to 1 with 2 decimal places)
```

(a) Base prompt with fill-in-the-blanks output

```
User: Here is a description of a bill introduced in the U.S. Congress:
[Description]

Please classify this description into one of the following categories:
1.  Macroeconomics
2.  Civil Rights, Minority Issues, and Civil Liberties
3.  Health
⋮
18.  International Affairs and Foreign Aid
19.  Government Operations
20.  Public Lands and Water Management

Output a JSON object structured like:
{
"Category": an integer from 1 to 20 that best represents the bill category,
"Confidence": confidence level in the bill classification as a number between 0 to 1 with 2 decimal places
}
```

(b) Base prompt with JSON output

Figure A18: Base prompts for labeling the policy topic with large language models on Congressional legislation.

Notes: This figure documents the base prompts used for labeling the policy with large language models on congressional legislation. We prompt GPT-3.5-turbo and GPT-4o to label the descriptions of 10,000 randomly drawn Congressional bills for their major topic. For each Congressional bill, we include the text of its description [description]. Figure (a) provides the base prompt with fill-in-the-blanks output, and Figure (b) provides the base prompt with JSON output. See Section 5.3.3 of the main text for further details.

User: `You are a knowledgeable political analyst.`

(a) Political analyst persona

User: `Answer this question as if you are a political scientist that studies legislation in the United States Congress.`

(b) Political scientist persona

User: `Answer this question as if you are an expert in United States politics.`

(c) Expert in U.S. politics persona

User: `Answer this question as if you were a helpful research assistant for a political scientist.`

(d) Helpful research assistant persona

Figure A19: Persona modifications to the base prompt for labeling the policy topic with large language models on Congressional legislation.

Notes: This figure documents the persona modifications to the base prompts for measuring the policy topic with large language models on Congressional legislation. We prompt GPT-3.5-turbo and GPT-4o to label the descriptions of 10,000 randomly drawn Congressional bills for their major topic. Each persona modification is added to the beginning of the base prompt with JSON output (Panel (b) of Appendix Figure A18). See Section 5.3.3 of the main text for further details.

User: Think carefully. Output a JSON object structured like:
{
"Category": an integer from 1 to 20 that best represents the bill category,
"Confidence": confidence level in the bill classification as a number between 0 to 1 with 2 decimal places,
\Explanation": a one-sentence explanation of your bill category answer
}

(a) Think carefully prompt

User: Please provide an explanation for your answer. WOutput a JSON object structured like:
{
"Category": an integer from 1 to 20 that best represents the bill category,
"Confidence": confidence level in the bill classification as a number between 0 to 1 with 2 decimal places,
\Explanation": a one-sentence explanation of your bill category answer
}

(b) Explanation prompt

User: Think step by step. Lay out each step. Output a JSON object structured like:
{
"Category": an integer from 1 to 20 that best represents the bill category,
"Confidence": confidence level in the bill classification as a number between 0 to 1 with 2 decimal places,
\Explanation": a one-sentence explanation of your bill category answer
}

(c) Think step by step prompt

Figure A20: Chain of thought modifications to the base prompt for labeling the policy topic with large language models on Congressional legislation.

Notes: This figure documents the chain of thought modifications to the base prompts for labeling the policy topic with large language models on Congressional legislation. We prompt GPT-3.5-turbo and GPT-4o to label the descriptions of 10,000 randomly drawn Congressional bills for their major topic. Each chain-of-though modification alters the base prompt with JSON output (Panel (b) of Appendix Figure A18). See Section 5.3.3 of the main text for further details.