Large Language Models: An Applied Econometric Framework*

Jens Ludwig Sendhil Mullainathan Ashesh Rambachan†

December 6, 2024

Abstract

Large language models (LLMs) are being used in economics research to form predictions, label text, simulate human responses, generate hypotheses, and even produce data for times and places where such data don't exist. While these uses are creative, are they valid? When can we abstract away from the inner workings of an LLM and simply rely on their outputs? We develop an econometric framework to answer this question. Our framework distinguishes between two types of empirical tasks. Using LLM outputs for *prediction problems* (including hypothesis generation) is valid under one condition: no "leakage" between the LLM's training dataset and the researcher's sample. Using LLM outputs for *estimation problems* to automate the measurement of some economic concept (expressed by some text or from human subjects) requires an additional assumption: LLM outputs must be as good as the gold standard measurements they replace. Otherwise estimates can be biased, even if LLM outputs are highly accurate but not perfectly so. We document the extent to which these conditions are violated and the implications for research findings in illustrative applications to finance and political economy. We also provide guidance to empirical researchers. The only way to ensure no training leakage is to use open-source LLMs with documented training data and published weights. The only way to deal with LLM measurement error is to collect validation data and model the error structure. A corollary is that if such conditions can't be met for a candidate LLM application, our strong advice is: don't.

*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 also thank audiences at UIUC, UT Austin, and the University of Chicago "AI in Social Science" Conference. All interpretations and any errors are our own.

†Ludwig: University of Chicago and NBER. Mullainathan: Massachusetts Institute of Technology and NBER. Rambachan: Massachusetts Institute of Technology.

1 Introduction

How should large language models (LLMs) be used in economics research?[1] Since they are easy-to-use, general-purpose tools, researchers are already using LLM outputs to form predictions, label text, answer surveys, simulate humans in experiments, generate hypotheses, and even produce data for time periods and places for which such data don't currently exist.[2] These uses are no doubt creative, but are they valid? How can we assess whether empirical research using LLM outputs is leading to correct inferences?

Consider how we normally answer this question for a more familiar econometric procedure like 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 possible 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 must defend.

Appropriately using LLM outputs requires the same sort of contract: what assumptions about LLMs must be defended in order to draw some particular inference? Yet at present, no such contracts exist. In fact, producing such a contract is challenging given how econometrics typically models data procedures. Our contracts for OLS are based on knowing the exact operations of OLS on a data matrix. In contrast, LLMs are extraordinarily complex machine learning models involving many layers of interactivity and billions of parameters. Their training datasets and architecture (among many other details) are proprietary—that is, intentionally hidden from the user.[3] They are also a diverse, dynamic set of commercial products. Different families of LLMs make different choices along each of these dimensions. The state-of-the-art is ever-evolving: new LLMs are released constantly. Deriving guarantees from the exact operations of these models, like we do for OLS, is simply not feasible.

[1] Good reviews of machine learning more broadly and its applications within economics include Varian (2014); Mullainathan and Spiess (2017); Athey (2018); Gentzkow, Kelly and Taddy (2019); Dell (2024).

[2] For example, generating data on, say, inflation expectations or consumer sentiment during pre-Revolutionary America. There are many adjacent uses which we do not consider here, such as the use of LLMs as computational tools like others that occupy our workflow: search engines, and coding copilots. Korinek (2023, 2024) provide a useful description of all the ways LLMs can be used in this way. Our focus instead is on the uses of LLMs not in day-to-day workflow but directly in empirical analysis.

[3] 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).

The goal of this paper is to provide an econometric framework that provides valid LLM contracts despite the complexity of these models. 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 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 would like to know how well can stock prices be predicted using the text of 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 training leakage, meaning 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* 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 to researchers. This is a well-known challenge in computer science, where researchers worry that LLMs have increasingly been trained on the benchmark datasets commonly used to evaluate LLM performance. 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 headlines and Congressional bills.[4]

Even though it can be violated, we argue that "no training leakage" can serve as a credible assumption in many applications since it can be controlled by the researcher's choice of large language model. But that requires using 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). Researchers can thereby mechanically enforce that the no-training-leakage condition is satis-

[4]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.

fied since it is known that these models have not been trained on any strings past a documented date.

Our second result relates to *estimation problems*, in which the researcher wants to measure some economic concept (e.g., positive or negative financial news, hawkish attitudes, or policy topic) expressed by some 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? The researcher has access to a resource-intensive, "gold-standard" process for measuring the economic concept. For instance, with enough time and effort, the text could be reliably labeled for the concept by the researcher or trained staff. Because the labeling of the text is often prohibitively costly or time-consuming, the researcher hopes to automate this process—to use the LLM to label text for the concepts.

In order for an LLM to successfully automate the existing measurement process, we demonstrate the key condition is that there must be no measurement error in the LLM's outputs. The plug-in estimate using the LLM's labels recovers the target estimate if and only if the LLM's labels reproduce the existing gold-standard measurement process.

This no-measurement-error assumption is simply not credible. A rapidly growing body of research documents the brittleness of LLMs across computer science benchmarks — what seem like small modifications to a task (choice of LLM or specific prompt, for example) leads to substantial changes in LLM performance. We present new results showing the implications of this brittleness for downstream estimation of economic parameters. For the applications of relating financial news headlines to stock prices, and relating the policy topic of a Congressional bill to characteristics of the bill's sponsor, we show that variability in the LLM's outputs across models and prompts leads to large changes in magnitude, statistical significance and even sign of resulting regression coefficients.

Our third result is a constructive solution for how to use LLMs for estimation problems in light of these challenges. Since LLMs are brittle and therefore not a general-purpose technology for producing economic labels, researchers will need some *context-dependent* playbook for using them in estimation problems. Simply calculating overall accuracy measures of the LLM labels is not sufficient because it is not just the magnitude of the measurement error that matters (sometimes reported in existing studies), but also its covariance with the other economic variables in the regression (unfortunately rarely if ever reported). Precisely because we have little insight into their design, it is difficult to articulate reasonable statistical assumptions on the properties of the LLM's measurement error.

We therefore borrow from an idea long known in economics: the only solution to measurement error is to collect validation data containing gold standard measurements and explicitly model the LLM's measurement error. 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., 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); Egami et al.

(2024). We illustrate the value of validation 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 debiasing LLM outputs using a validation 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 empirically demonstrate this performance in finite samples with Monte Carlo simulations using data on Congressional bills.

The implication is that for estimation problems, LLM outputs are not substitutes for gold standard labels. When done correctly, LLM outputs instead serve to amplify a small validation sample, enabling the researcher to draw correct inferences at lower cost. A corollary is that researchers should not use LLM outputs for estimation problems in applications where such validation data *cannot* be collected.

Our final result is to show that this framework is powerful enough to help economists carry out valid inference not just for familiar empirical tasks like prediction and estimation, but for more novel ones as well. For example, we argue that the use of LLMs for hypothesis generation has the same structure as other prediction problems. That means our framework can clarify what assumptions are required for the valid use of LLMs for this purpose—no training leakage. Or consider the use of LLMs to simulate human subject responses. We note the similarity of this application to other estimation problems, and hence the vital importance of collecting validation data (e.g., running experiments, collecting surveys, etc. from at least some real subjects).

Notice that 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. After all, isn't it important to understand the trick of, say, positional encoding and all the innards of the transformer architecture? Economists, however, *already* black-box important upstream elements of familiar econometric procedures — we just often don't recognize it. For example, few researchers could describe the specific numerical implementation of OLS in their preferred statistical software (in fact many of us have not realized these implementations differ across software programs). We are nonetheless happy to proceed with the returned estimate $\widehat{\beta}$ so long as we know they satisfy a key property, in this case of having minimized the sum of squared residuals. Our empirical framework provides a similar set of properties for LLMs that allows us to use their outputs in empirical research.

LLMs represent a remarkable new technological advance. They purport to be a general purpose technology, one that is both a substitute for intelligence and a source of ground truth. Their impressive performance on certain activities—as well as a great deal of hype—lures us in to treating their output as if it were indeed ground truth. Their general purpose makes us feel we should be able to apply them everywhere. But those temptations, understandable as they may be, must be resisted.

Our over-arching argument is that there shouldn't be a separate "VIP lane" for LLMs into our

economics research. The bridge from statistical output to conclusions about the world should be carefully guarded. The cardinal sin of empirical research is to use a data procedure without clearly stating our assumptions and defending them. No one would claim that some instrumental variables estimate is a local average treatment effect without defending the monotonicity assumption (Imbens and Angrist, 1994; Angrist, Imbens and Rubin, 1996), or present some panel-data regression without defending its standard error calculation (Bertrand, Duflo and Mullainathan, 2004) - much less fail to present a standard error at all. Yet analogous practices are rampant with LLMs. While LLMs *seem* different, just like any other procedure in our empirical toolkit they require rigorous econometric contracts clarifying when and how economists can make valid inferences based on them.

2 Evaluating General-Purpose Technologies: the LLM Conundrum

Why has our research community been willing to create a VIP lane for LLMs straight into our 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 reasons turn out to be problematic. The domains on which LLMs perform well turns out to be quite 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. In what follows we give a sense for how truly odd the behavior of LLMs can be, and consequently how far they are from being able to perfectly substitute for human intelligence, much less serve as a sort of "super intelligence."

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 — 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 simply 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). Even more common is the evaluation of LLMs by their designers 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 a 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 that could accomplish these feats yet fail on related (or even simpler) tasks, we tend to generalize the performance of a 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 to this effect 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 its 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 its implications for LLM usage.[5]

[5]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

This research, in other words, confirms that people engage in a naive heuristic of anthropomorphic generalization, which is surely one important reason we are all so willing to treat the labels of LLMs as ground-truth and plug them into our research pipelines.

2.2 Impressive Feats, But Also Puzzling Examples

This anthropomorphic generalization of the LLM's capacity is common, understandable (given their seemingly human interface), and deeply problematic. Recent work in computer science shows the capabilities of LLMs are brittle: For every example of impressive performance there is 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, a 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 limited to trivial application domains, as was discovered by the lawyer who used a 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 unavailable 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 can perform on other math problems. For example, the same LLM that can reliably solve (9/5) x + 32 cannot solve (7/5) x + 31; a 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. A 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 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

seemingly-similar difficulty levels that the models handle poorly—what the authors call AI's "jagged technological frontier" (Dell'Acqua et al., 2023).

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. A LLM trained on "A is B" will not know "B is A." For example, a 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*).

Despite this accumulating body of evidence, outputs from *these* tools are being uncritically incorporated into a growing number of empirical studies. This sets up the problem statement that we seek to address. Since LLMs are easy-to-use yet brittle tools for processing text, under what conditions can researchers safely plug-in the outputs of LLMs into a research pipeline? Our argument, in short, is that "I played around with it and it seemed to do well" is *not* a valid justification for widespread use of such models within economics research. To provide a more constructive alternative, we develop next an econometric framework to explore this question formally.

3 An Empirical Framework for LLMs

A key feature of modern LLMs is that we typically interact with them as black boxes: even though key choices in their design and the exact contents of their training datasets are unknown to us, we nonetheless use them to generate responses given prompts. We are willing to use these LLMs as a generic tool for processing almost any text—a true general purpose technology—given their impressive performance on benchmark evaluations. But does this heuristic make any sense?

In this section, we develop a framework that helps clarify what properties a 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 $\sigma \in \Sigma^*$, and 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 also expresses some economic concept V_r for which it is costly in terms of either time or money to obtain measurements. A key assumption (often left implicit) in much empirical research is that there is *some* procedure that could in principle be applied to each text piece to collect what the researcher would be willing to consider a "gold standard" measurement of the economic concept; that is, there exists some mapping such that $V_r = f^*(r)$ for all text pieces $r \in \mathcal{R}$.

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.[6] In other applications, the researcher, motivated by findings in psychology (e.g., Biemer et al., 2013; Kahneman, Sibony and Sunstein, 2021), will worry that human annotations of some text are noisy proxies for the economic concept, but is

[6]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).

willing to accept an average from some minimum number of labelers as a reliable measure.

No matter how the status quo procedure for collecting measurements is defined, we face a *text processing problem*: measuring the economic concept V_r requires processing each text piece r, and in many applications it may be prohibitively costly or time-consuming to do so. Absent a solution to this text processing problem, the researcher does not observe V_r.

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 $\widehat{m}(\cdot;t)\colon \Sigma^* \to \Sigma^*$ 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 a 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 the LLM 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 an economics research pipeline.

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 a 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 a 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. A 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 exists alternative "inference algorithms" that may be used, such as greedy

decoding or beam search. Since alternative inference algorithms may affect a LLM's outputs, alternative choices correspond to alternative text generators in our framework.

Finally, the training algorithm captures all aspects of the design and training process of a 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 can 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 scientific 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 use our framework to establish the key condition under which the use of a 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, and it places meaningful limits on the types of prediction problems that researchers can tackle using LLMs. 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 — far more seriously than has typically been the case in social science research to date.

4.1 The Researcher's Prediction Problem

Suppose the researcher prompts a LLM, trained on some unknown training dataset t, to form predictions of the linked variable based on the text pieces, $\widehat{Y}_r = \widehat{m}(r;t)$.[7] For some non-negative loss 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 $t = \left(t_{\sigma_1},\ldots,t_{\sigma_{|\Sigma^*|}}\right)'$ denote 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 researcher's and the LLM's sampling process over strings is independent but not 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 let $q_\sigma^T(t_\sigma) = Q(T_\sigma = t_\sigma)$ and $q_\sigma^D = Q(D_\sigma = 1)$.

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

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 sample average loss converges to

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

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 a 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 $Q(\cdot) \in \mathcal{Q}$.

In words, 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. Notice that this is an "out-of-sample" inferential goal in the prediction problem. 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 Representatives or Senate? Think carefully.`

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 selects a 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 a 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 $\widehat{m}(\cdot)$,*

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

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*

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

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).

What is the intuition behind this result? 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 that have captured the public's imagination, 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).[8] 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 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 (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 makes 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 A10 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 actually much simpler: 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

[8]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).

based on only the first 50% of its text, following research in computer science such as Golchin and Surdeanu (2024) (see Appendix Figure A11 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 perhaps 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 (Glasserman and Lin, 2023). Despite applying this prompt engineering strategy to the our sample of congressional bills (see Appendix Figure A10-A11 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 (this now being the key importance of that observation). The training algorithm involves months of computing time to estimate millions (or in many cases billions) of parameters. The text generator that results, $\widehat{m}(\cdot;t)$, will then by definition depend on the entire training dataset upon which the training algorithm was run. Prompting a 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 A12 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's 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 A12 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. Altogether, our findings on memorization of financial news headlines and this existing work indicates 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. Fortunately our econometric framework provides some guidance for how (and how not) to use LLMs productively in prediction problems.

First and foremost, 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.[9] As

[9]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

an example, the Llama family of models (Touvron et al., 2023; Dubey et al., 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 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.[10] 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 *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 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 involve building tools to make predictive inferences

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.

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

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 some economic concept V_r expressed by each collected text piece r in order to estimate some downstream parameter of interest. The researcher is interesting in having a LLM automate the collection of gold standard measurements, so that the model's outputted labels of the economic concept of interest could be plugged into the economist's estimation problem.

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 be as good as the gold standard everywhere—for every dataset and for every task. This assumption, stated so directly, is as a conceptual matter unlikely to be true with vanishingly small odds. Yet it is often implicit in many current research uses of LLMs within economics.

As an empirical matter, the assumption of no measurement error contradicts the 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. Specifically, in both of our empirical applications (financial headlines and Congressional legislation), we show that seemingly minor implementation choices such as which LLM to use or how to prompt the model affects not only the labels the LLM assigns to a given piece of text, but also substantially affects 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 sub-sample of gold-standard validation data to model the LLM's measurement 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, which winds up ruling out applications in which validation data cannot be collected.

5.1 The Researcher's Estimation Problem

To illustrate what guarantee is needed for an economist to use a 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 observed the economic concept V_r associated with each collected text piece r, she would be satisfied to calculate the parameter

estimate

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

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, the researcher does not observe V_r and so $\widehat{\theta}^*$ cannot be calculated directly.

To solve the text processing problem, suppose the researcher prompted a 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

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

We would like to draw conclusions about the target estimate $\widehat{\theta}^*$ defined using the economic concept V_r based on the feasible estimate $\widehat{\theta}$ defined using the LLM's labels $\widehat{m}(r;t)$. What conditions must the LLM satisfy in order for this workflow to be valid?

To answer this question, we must (as with prediction problems) 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

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

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

With this in hand, we introduce the following definition. Given a 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 $\mathcal{M}$ is a *valid measure* of the economic concept V_r 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 $\theta \in \Theta$,

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

The LLM $\widehat{m}(\cdot;t)$ with guarantee $\mathcal{M}$ is a *general-purpose technology for estimation* if it is a valid measure of the economic concept V_r 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 $\mathcal{M}$ is a valid measure of the economic concept V_r 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 a LLM to be a valid measure of the economic concept for 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*

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

The second 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,*

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

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

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

where $\delta(r)$ is defined in the proof.

The guarantee $\mathcal{M}(Q,\delta)$ bounds the measurement error of the LLM 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 is a valid measure of the economic concept. 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 for the economic concept that lead to non-trivial bias for the downstream economic parameters we seek to estimate — 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 is a valid measure of the economic concept.

So Lemma 3 immediately implies that a LLM is a general-purpose technology for estimation if and only if it satisfies a strong guarantee that its outputs have no measurement error for the economic concept studied. 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 are as-good-as the gold standard measurement 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

Plug-in estimation requires the strong guarantee that the LLM's labels are as-good-as gold standard measurements of the economic concept V_r. 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 showed in Section 2, that intuition turns out to be deeply misleading. A growing body of research in computer science provides evidence documenting a variety of errors with LLMs. What remains unknown is the degree to which this brittleness matters in practice for economics research; is this an important concern for estimation problems, or effectively just a rounding error?

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 that 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 a 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 measurement 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 $r \in \mathcal{R}$.*

1. *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 $\beta = \beta^* + \lambda_{\Delta|W}$.*

2. *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 measurement error $\Delta_r = \widehat{m}(r;t) - V_r$ covaries with the linked variable W_r in the research context. Measurement error in the LLM's label 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 label.

To concretely illustrate why researchers must be appropriately skeptical of claims that the labels of LLMs are as-good-as ground truth in economic settings, we return to our earlier examples of financial news headlines and congressional legislation. If they are equivalent to ground truth

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 are ground truth.

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 now 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) (WRDS Research Team, 2023).

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 would obviously be a painstaking process.

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 a LLM): what

specific LLM to use? What prompt engineering strategy? Economists happy to treat LLM labels as ground truth should have their faith shaken if alternative answers to these practical questions lead to meaningfully different labels and downstream parameter estimates—this would indicate non-ignorable measurement error for this economic concept.

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 V_r; 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 JSON (JavaScript Object Notation) object. The exact wordings of our base prompts are provided in Appendix Figure A13.

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 A14 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 A15 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 67% 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 89.2% of headlines; whereas prompting GPT-4o to answer like an "expert in finance" versus an "expert in the economy" leads to agreement on 96.3% of headlines. Across models, there appears 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. In other words, 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.

We then separately regress the labeled economic concept — in this case, the policy topic

area 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 economists see a 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. We cannot simply plug the LLM's output into an economics research pipeline and "set it and forget it," assuming the implicit claims that LLMs are providing ground-truth output. Viral examples of the capabilities of LLMs and benchmark evaluations do not justify the strong guarantee needed for 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. Articulating statistical assumptions about the structure of measurement error in LLMs appears challenging — after all, these are complicated algorithms trained on unknown datasets. Yet a classic solution can still be applied: invest in collecting the gold standard label V_r on a small validation sample and use it to de-bias the plug-in estimate based on the LLM labels $\widehat{V}_r$. The virtues of validation data in dealing with measurement error is well-known – see, for example, classic work such as Bound and Krueger (1991); Bound et al. (1994); Bound, Brown and Mathiowetz (2001) and Lee and Sepanski (1995); Chen, Hong and Tamer (2005). It has been recently revived in machine learning; see Wang, McCormick and Leek (2020); Angelopoulos et al. (2023); Egami et al. (2024). The corollary of this classic solution is that with any application in which the researcher *cannot* collect validation data, the researcher really should not be using the LLM's labels for estimation problems.

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 De-Biasing 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 extends 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 the gold standard label V_r on a small validation sample. 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 $(r,\widehat{m}(r;t),V_r)$ 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 for purposes of reporting 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 bills 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 a 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 ground-truth labels on a validation sample, a natural question arises: why bother with the possibly mis-measured 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 gold standard label V_r and the measurement error Δ_r. Importantly, Equation (14) 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.[11] In this case, 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

[11]This phenomenon has been documented elsewhere 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.

substitutes for gold standard labels; instead, they can serve to cheaply amplify the validation data.

Of course 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 ground-truth labels V_r 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, 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).

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 ground-truth label V_r on average.

Figure 6 plots the accuracy of the LLM label $\widehat{V}_r^{m,p}$ for the ground-truth 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 ground-truth label V_r on 10% 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 ground-truth 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.

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 5-th percentile yields a bias that is fully -1.86 standard deviations away from the target regression coefficient and the 95th percentile is fully 2.18 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 95% confidence interval obtained from just plugging in the language model's label includes the true parameter just 81% of the time; the 5^{th} percentile confidence interval includes the true parameter just 38% of the time. In contrast the 95% confidence interval from the de-biased label covers the true parameter value around 95% of the time the vast majority of the time. 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 some gold standard labels, 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, we show that these precision gains 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, 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 lead 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 highlight that LLM outputs are not substitutes for gold standard labels. But rather, when used correctly, LLM outputs can amplify a small validation sample, enabling the researcher to draw correct and precise inferences at a lower cost.

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, hypothesis generation bears striking resemblance to our analysis of the prediction problem. In assessing whether LLMs are good *tools* for hypothesis generation, we would like to generalize its quality in a particular research context $Q(\cdot)$ (some setting in which we would like to generate hypotheses), making 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 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, 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, 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

[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).

text pieces $r \in \mathcal{R}$ and their associated economic concept V_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 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 (i.e., there exists some mapping $V_r = f^*(r)$ for all $r \in \mathcal{R}$), 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 $\widehat{m}(r;t)$? ▲

By mapping the use of LLMs for human subject simulation into our framework, the parallel to estimation problems is clear. Viewing in silico studies through the lens of our estimation framework then implies that researchers need the condition that there is no measurement error in the LLM's

[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.

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). And beyond this evidence of brittleness in the LLMs performance, there is also the second requirement for using a LLM for some estimation problem: no training leakage. The results from existing published experiments almost certainly enter into the LLMs training corpus. But presumably what we are interested in is not so much whether a 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 use of LLMs to simulate 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.

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.

Angrist, Joshua D., Guido W. Imbens, and Donald B. Rubin. 1996. "Identification of causal effects using instrumental variables." *Journal of the American statistical Association*, 91(434): 444–455.

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.

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.*

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.*

Bertrand, Marianne, Esther Duflo, and Sendhil Mullainathan. 2004. "How much should we trust differences-in-differences estimates?" *The Quarterly journal of economics*, 119(1): 249–275.

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 Rafael 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*.

Imbens, Guido W., and Joshua D. Angrist. 1994. "Identification and estimation of local average treatment effects." *Econometrica*, 62(2): 467–475.

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.

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.

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*.

WRDS Research Team. 2023. *Beta Suite by WRDS.* `https://wrds-www.wharton.upenn.edu/pages/grid-items/beta-suite-wrds` (accessed August 1, 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

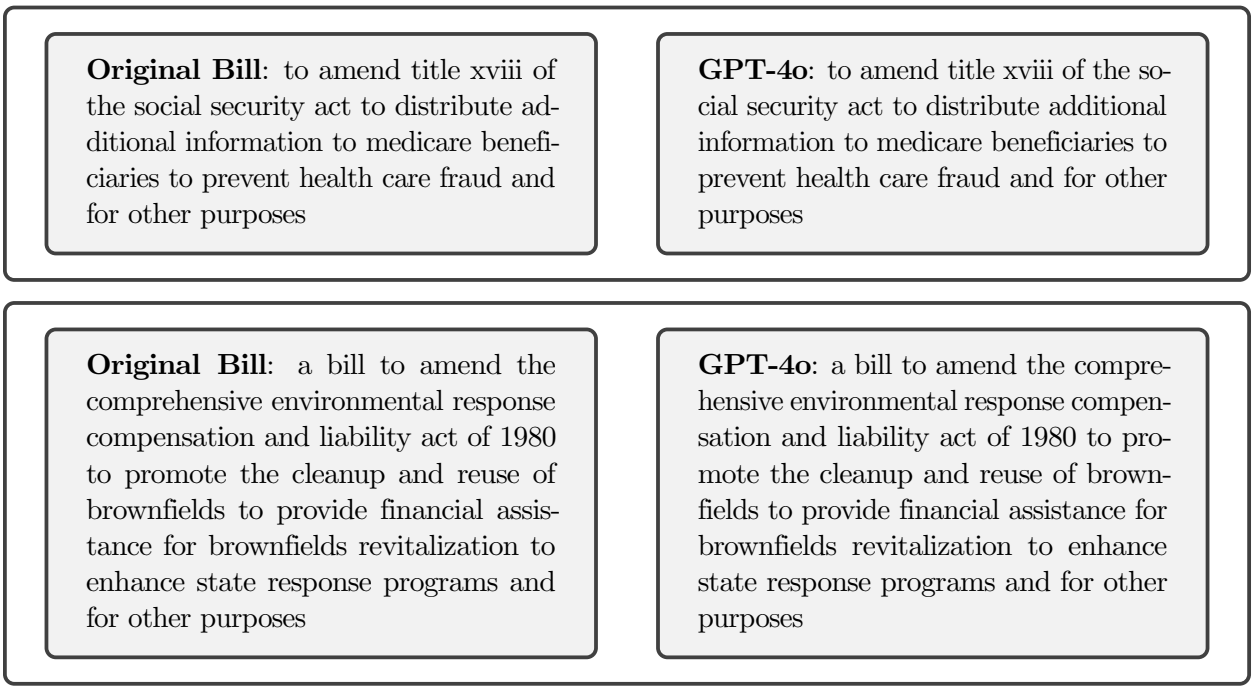


Figure 1: Two examples of GPT-4o completions that exactly match original descriptions of congressional legislation.

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.

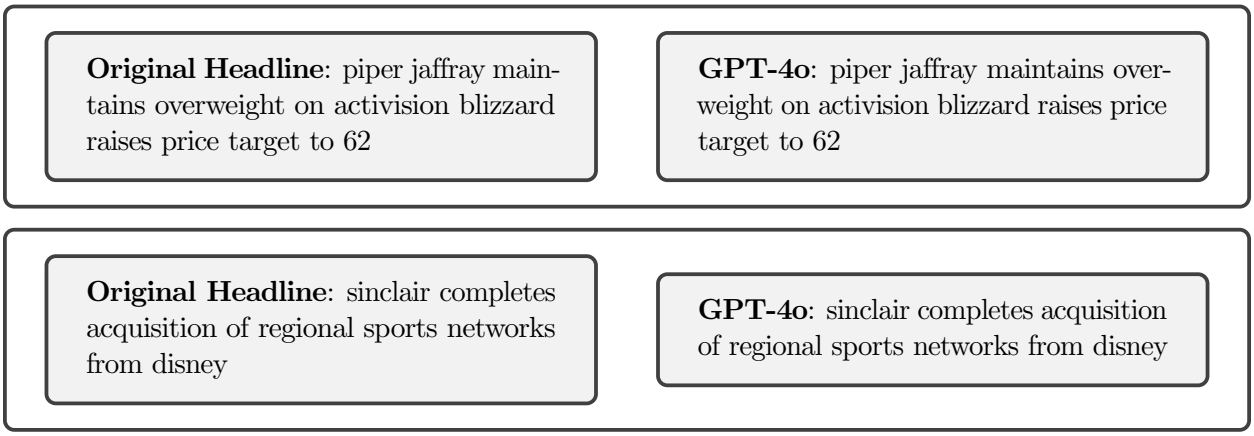


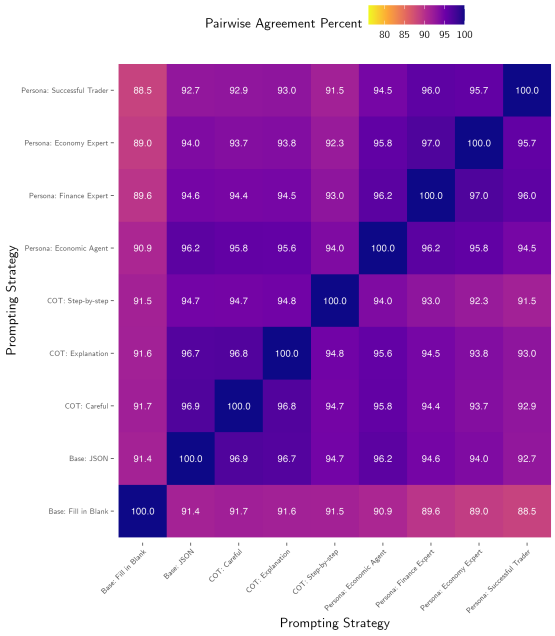
Figure 2: Examples of GPT-4o completions that exactly match original financial news headlines.

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.



(a) GPT-3.5-turbo

(b) GPT-4o



(c) GPT-4o-mini

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-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 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.

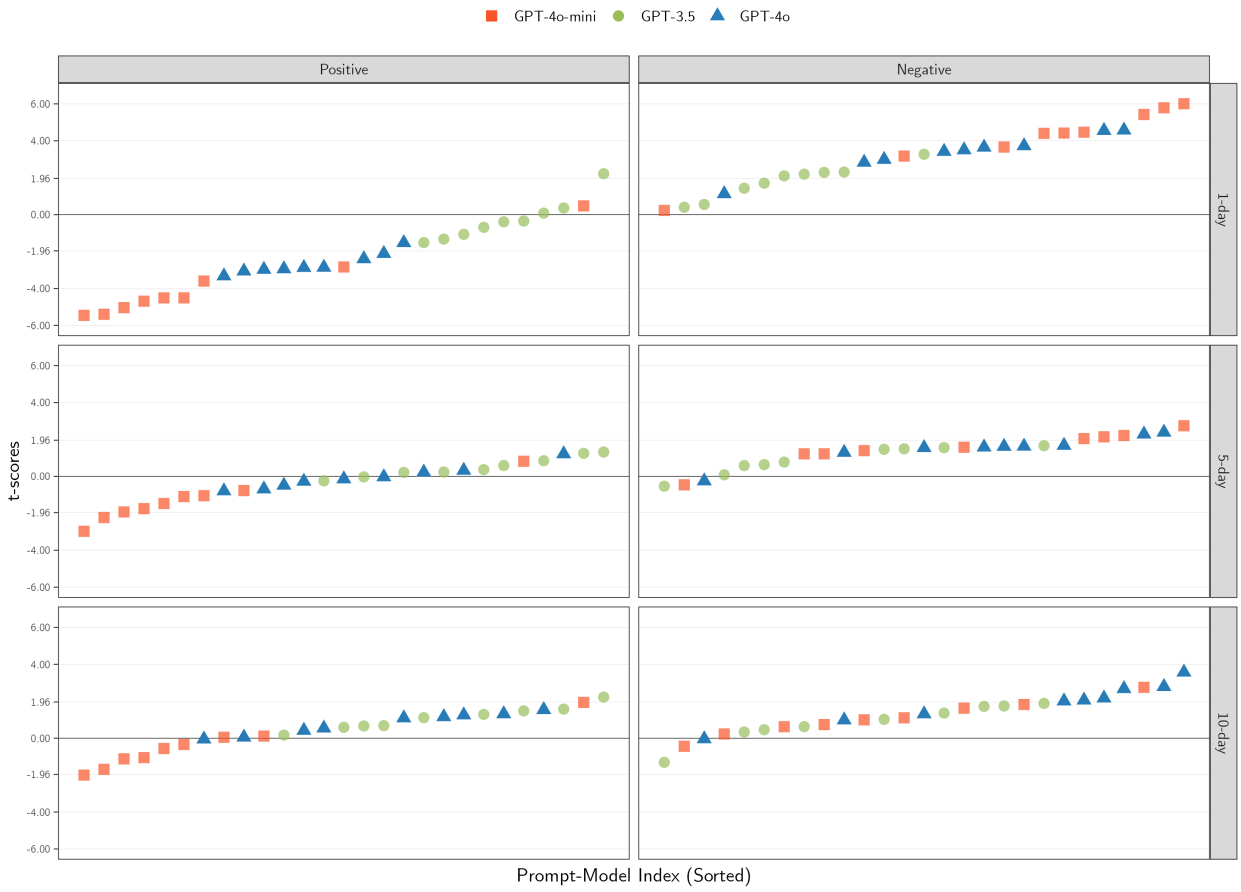


Figure 4: Variation in t-statistics for realized returns across large language models and prompting strategies on financial news headlines.

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.

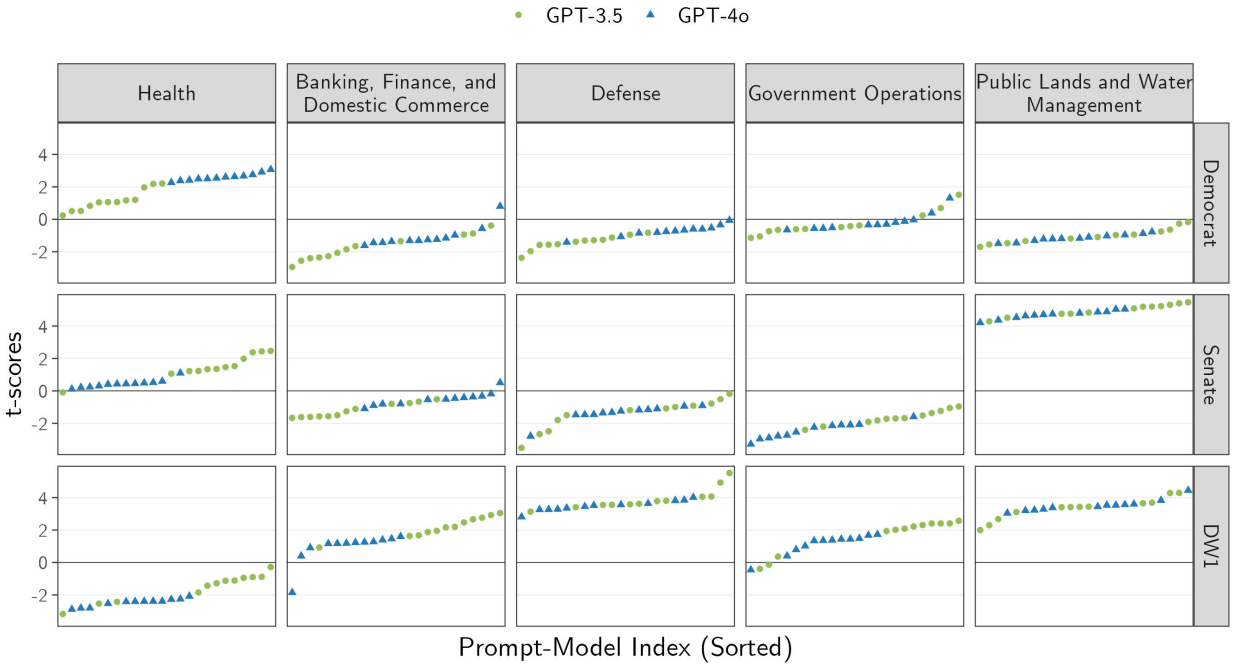


Figure 5: Variation in t-statistics across large language models and prompting strategies on congressional legislation.

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.

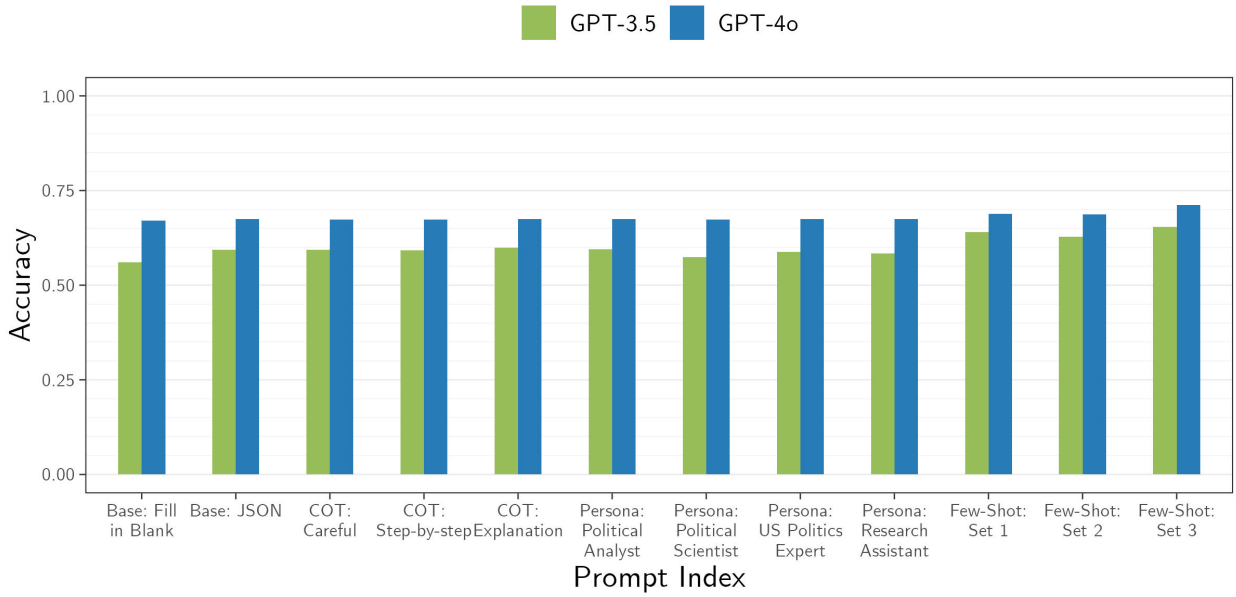


Figure 6: Accuracy of large language model labels of bill topic across model and prompt variation.

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 5.3.3 for discussion.

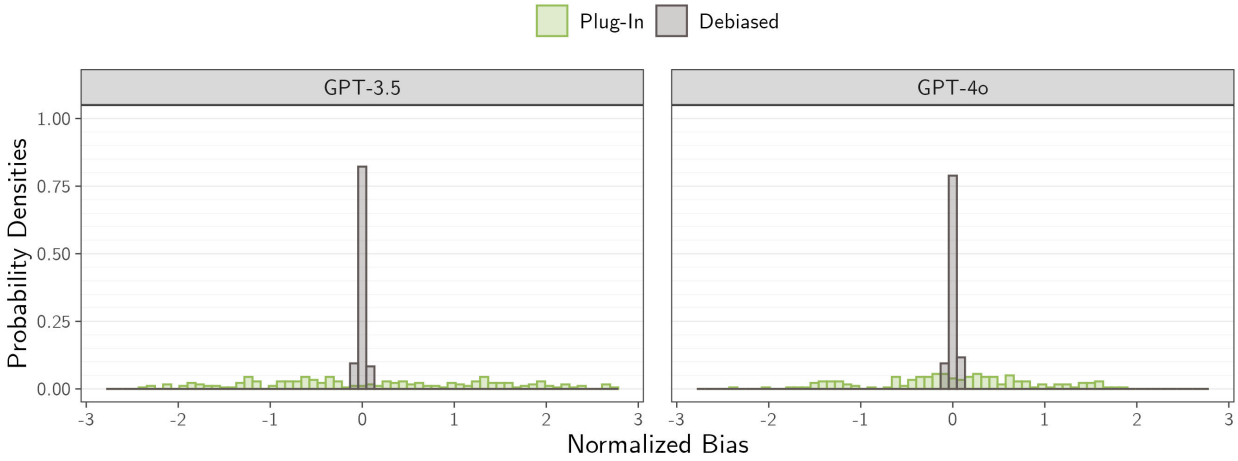


Figure 7: Normalized bias of the plug-in regression and bias-corrected regression across Monte Carlo simulations based on congressional legislation.

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 10% validation sample.

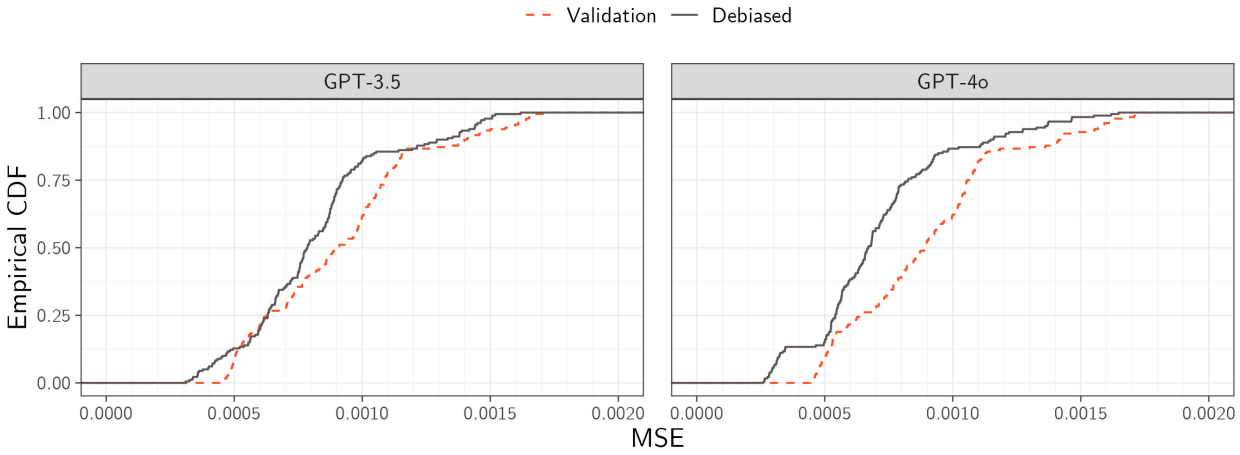


Figure 8: Cumulative distribution function of mean square error for the bias-corrected estimator against validation-sample only estimator.

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 10% 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.

	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		
	1 day	5 day	10 day
Point Estimates			
Mean	-0.98	-0.32	0.40
Median	-1.04	-0.13	0.54
5th Percentile	-2.89	-1.96	-1.38
95th Percentile	0.10	0.76	1.78
Sample Average	0.05	0.31	0.58

(a) Positive Labels

	Return Horizon		
	1 day	5 day	10 day
Point Estimates			
Mean	1.62	1.32	1.47
Median	1.30	1.33	1.49
5th Percentile	0.23	-0.18	-0.162
95th Percentile	3.46	2.75	3.119
Sample Average	0.05	0.31	0.58

(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 Average
		Mean	Median	5%	95%	
Health	Democrat	0.010	0.012	0.002	0.016	0.071
Health	Senate	0.005	0.005	0.001	0.013	0.071
Health	DW1	-0.015	-0.018	-0.022	-0.006	0.071
Banking	Democrat	-0.009	-0.008	-0.017	-0.002	0.060
Banking	Senate	-0.005	-0.005	-0.010	-0.001	0.060
Banking	DW1	0.012	0.012	0.004	0.024	0.060
Defense	Democrat	-0.006	-0.006	-0.013	-0.002	0.097
Defense	Senate	-0.008	-0.006	-0.019	-0.003	0.097
Defense	DW1	0.025	0.024	0.019	0.043	0.097
Government Operations	Democrat	-0.001	-0.002	-0.006	0.008	0.134
Government Operations	Senate	-0.013	-0.013	-0.021	-0.007	0.134
Government Operations	DW1	0.012	0.013	-0.003	0.020	0.134
Public Lands	Democrat	-0.007	-0.007	-0.010	-0.002	0.114
Public Lands	Senate	0.032	0.032	0.027	0.036	0.114
Public Lands	DW1	0.026	0.027	0.016	0.031	0.114

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 ("Banking"), Defense, Government Operations, and Public Lands & Water Management ("Public Lands"), 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	-0.005	-1.863	2.179
Debiased	-0.005	-0.053	0.049
Coverage			
Plug-In	0.812	0.383	0.949
Debiased	0.941	0.927	0.952

(a) GPT-3.5-turbo

	Median	5%	95%
Normalized Bias			
Plug-In	0.059	-1.447	1.507
Debiased	0.000	-0.066	0.050
Coverage			
Plug-In	0.919	0.642	0.952
Debiased	0.941	0.926	0.953

(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 10% 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

Distance	Average	Benchmark
Cosine similarity	0.830	0.379
Euclidean distance	0.536	1.110

(a) Base prompt

Distance	Average	Benchmark
Cosine similarity	0.829	0.379
Euclidean distance	0.538	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. See Section 4.3.1 for discussion.

Distance	Average	Benchmark
Cosine similarity	0.880	0.309
Euclidean distance	0.455	1.172

(a) Base prompt

Distance	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. See Section 4.3.2 for discussion.

	Return Horizon		
	1 day	5 day	10 day
Point Estimates			
Mean	-2.18	-1.24	-0.05
Median	-2.3	-1.13	0.03
25th Percentile	-3.08	-1.85	-0.98
75th Percentile	-1.38	-0.56	0.75
Sample Average	0.05	0.31	0.58

(a) Positive Labels

	Return Horizon		
	1 day	5 day	10 day
Point Estimates			
Mean	2.57	0.85	0.41
Median	2.78	2.00	2.38
25th Percentile	1.05	0.35	-1.3
75th Percentile	4.05	3.38	3.63
Sample Average	0.05	0.31	0.58

(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.

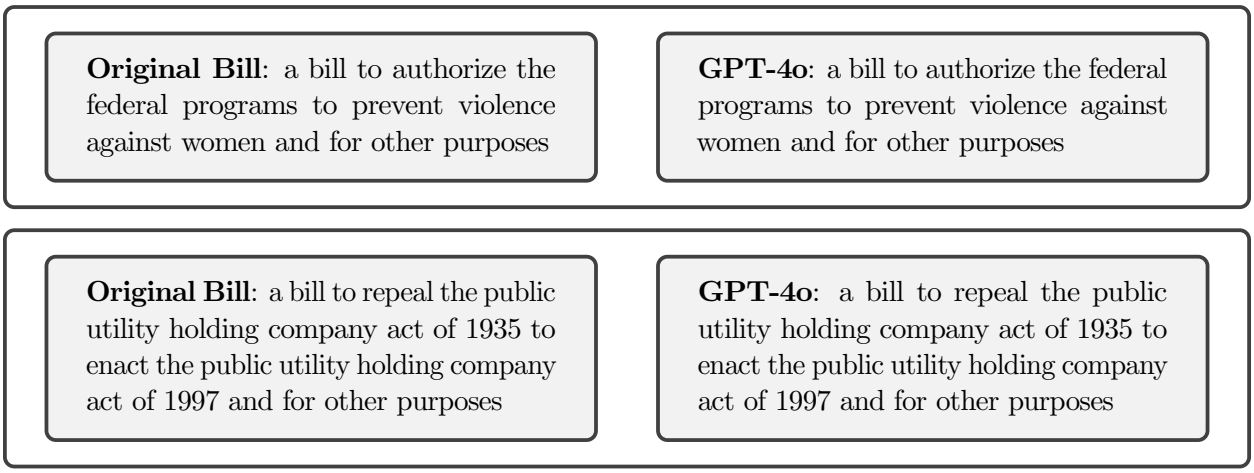


Figure A1: Examples of GPT-4o completions with date restriction that exactly match original descriptions of congressional legislation.

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: report comcast might sell hulu stake to disney

GPT-4o: report comcast might sell hulu stake to disney

Original Headline: euronet worldwide shares are trading lower after da davidson downgraded the stock from buy to neutral

GPT-4o: euronet worldwide shares are trading lower after da davidson downgraded the stock from buy to neutral

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.

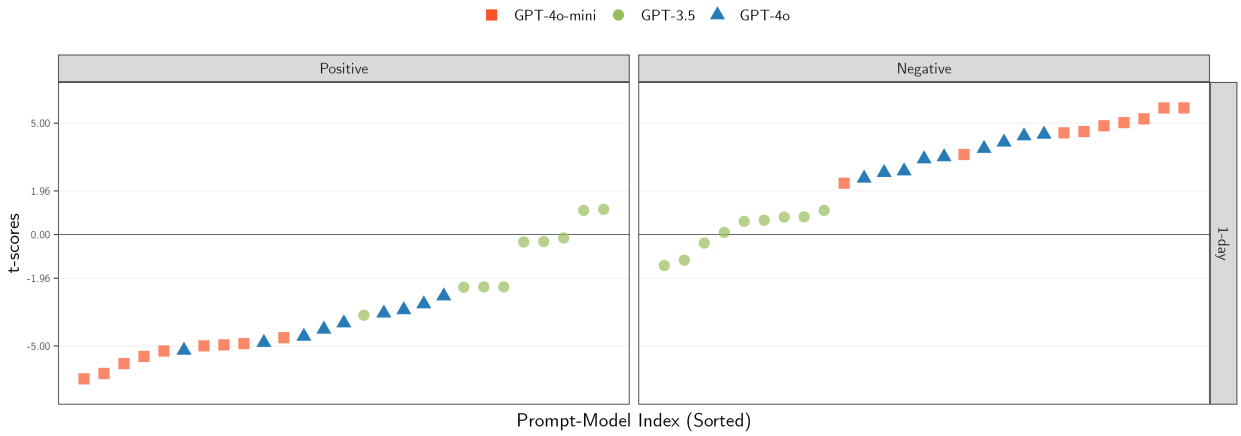


Figure A3: Variation in t-statistics realized returns across large language models and prompting strategies on financial news headlines using model's reported confidence.

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.

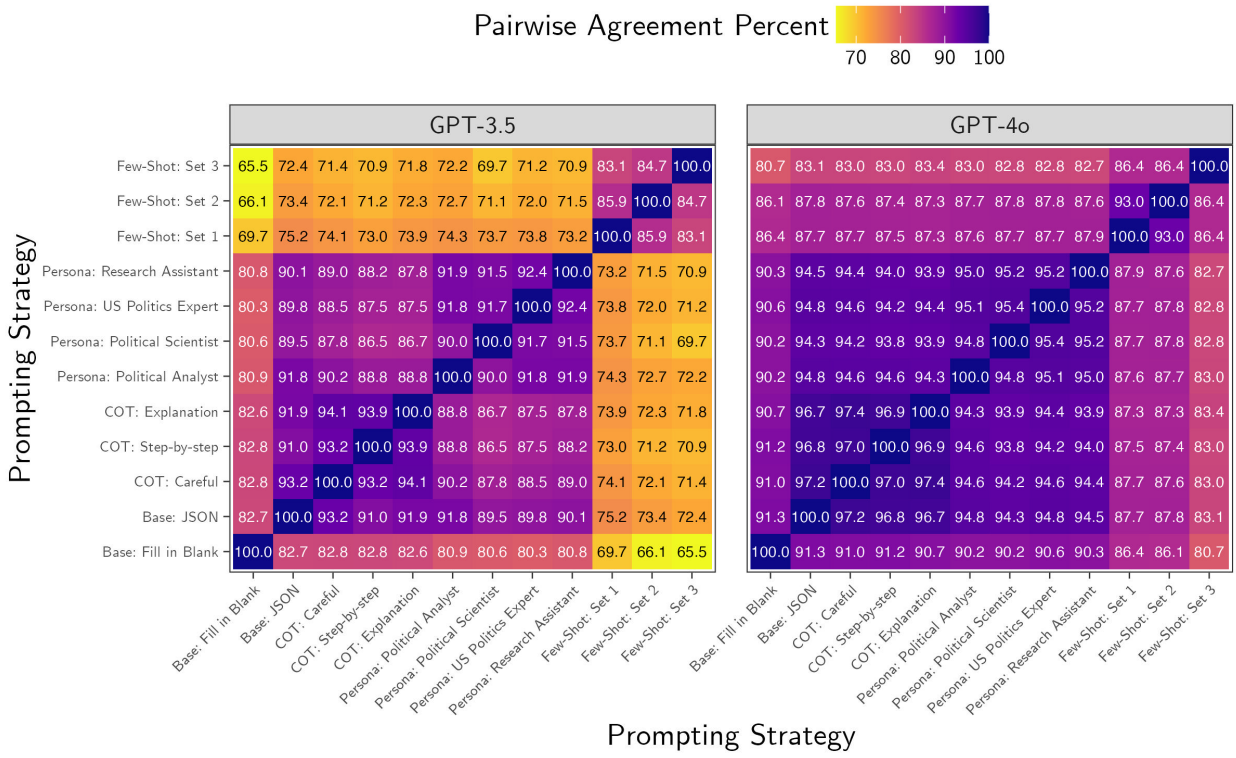


Figure A4: Variation in pairwise agreement between large language model labels across prompting strategies on congressional legislation.

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 $Q(\cdot) \in \mathcal{Q}$,

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

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

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

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

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,

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

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

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

as

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

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

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 $\widehat{m}(\cdot) \in \mathcal{M}(Q,\delta)$,

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

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

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 (1), 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 (2), 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

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

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 $Q(D=d \mid T=t) = \Pi_{\sigma\in\Sigma^*} Q(D_\sigma = d_\sigma \mid T_\sigma = t_\sigma)$, 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

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

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:

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

The bias-corrected regression coefficient is then given by

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

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

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

where $\sigma_{\hat{V}W}^2$, 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 $\Omega^{validation}=\sigma_{-V}^4\frac{1-\rho_v}{\rho_v}\sigma_{VW}^2$. Consequently, we can compare the limiting variances, and again observe that the bias-corrected regression coefficient has a smaller limiting variance if $\frac{1-\rho_p}{\rho_p}\sigma_{\hat{V}W}^2+2\sigma_{\hat{V}W}\leq\frac{1-\rho_v}{\rho_v}(\sigma_{VW}^2-\sigma_{\Delta W}^2)$. This can be satisfied provided the LLM outputs are sufficiently accurate for the gold-standard measurement.

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). 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 10% validation sample. We explore how the perfor-

mance 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 5%, 10%, 25%, and 50% of the random sample of 5,000 bills, producing validation samples of varying sizes. 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 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 5% and 10%. The bias-corrected coefficient performs similarly to the validation-sample only estimator for validation proportions equal to 25% and 50%.

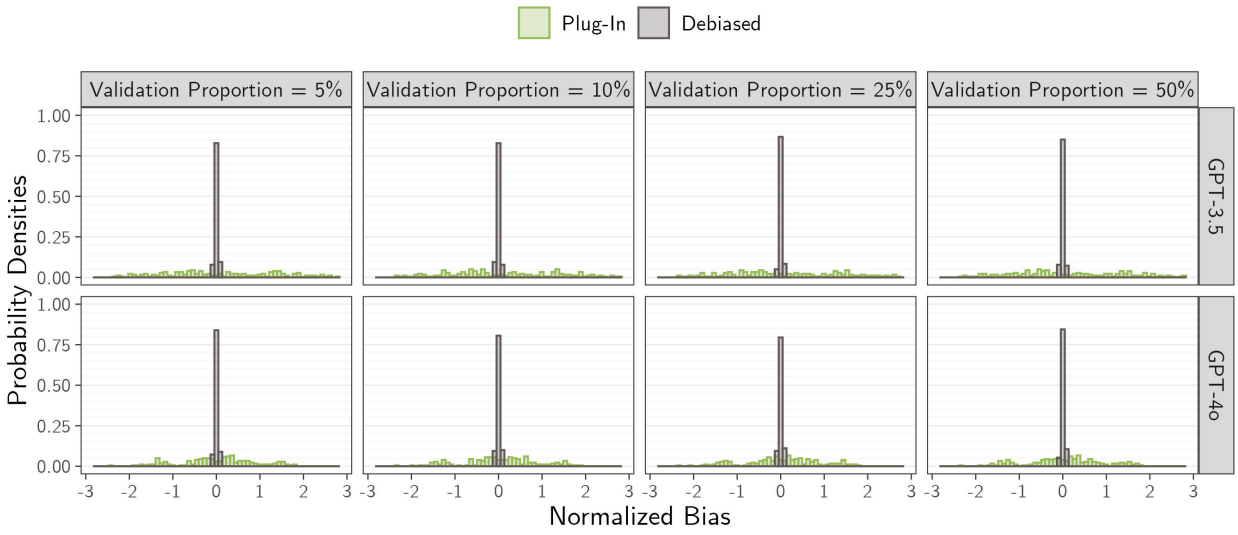


Figure A5: Normalized bias of the plug-in regression and bias-corrected regression across Monte Carlo simulations based on congressional legislation as the validation sample size varies.

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 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.

Validation Prop.	Median	5%	95%
Normalized Bias			
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			
5%	0.819	0.381	0.945
10%	0.812	0.383	0.949
25%	0.815	0.362	0.945
50%	0.815	0.369	0.950

(a) Plug-in regression

Validation Prop.	Median	5%	95%
Normalized Bias			
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			
5%	0.931	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 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			
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			
5%	0.921	0.637	0.954
10%	0.919	0.642	0.952
25%	0.921	0.625	0.954
50%	0.919	0.635	0.950

(a) Plug-in regression

Validation Prop.	Median	5%	95%
Normalized Bias			
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			
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 5%, 10%, 25% and 50%. Results are averaged over 1,000 simulations. See Appendix Section D.1 for discussion.

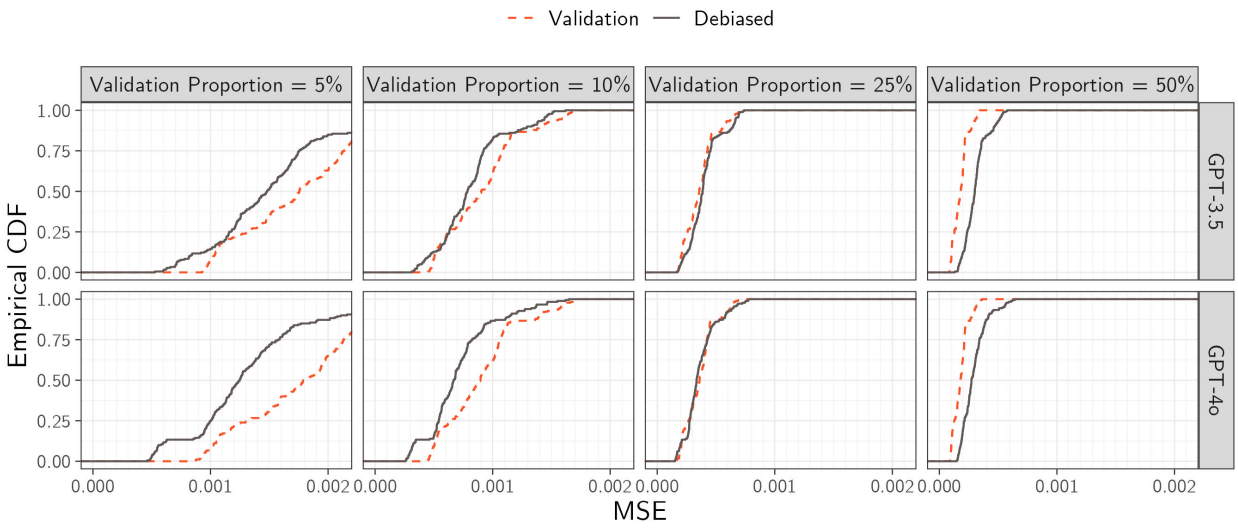


Figure A6: Cumulative distribution function of mean square error for the bias-corrected estimator against validation-sample only estimator, varying the size of the validation sample.

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 and the validation-sample only regression coefficient. 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 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.

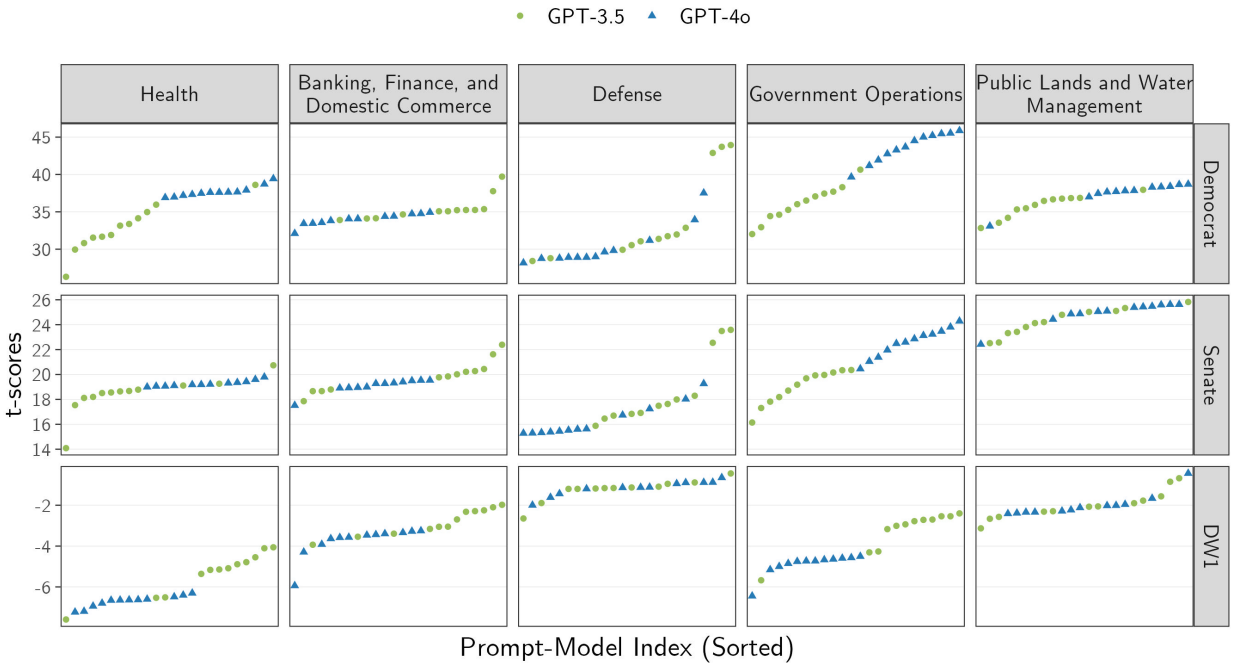


Figure A7: Variation in t-statistics across large language models and prompting strategies on congressional legislation, using the economic concept as a covariate.

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.

Covariate	Policy Topic	*Point Estimates*				Sample Average
		Mean	Median	5%	95%	
Democrat	Health	0.636	0.642	0.614	0.651	0.604
Democrat	Banking	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	0.588	0.587	0.579	0.599	0.604
Senate	Health	0.334	0.331	0.319	0.364	0.317
Senate	Banking	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	0.391	0.390	0.383	0.404	0.317
DW1	Health	-0.091	-0.095	-0.102	-0.076	-0.062
DW1	Banking	-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	-0.024	-0.024	-0.033	-0.009	-0.062

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 ("Banking"), Defense, Government Operations, and Public Lands & Water Management ("Public Lands") The final column ("Sample Average") reports the average of the linked variable W_r on all fraction of all Congressional bills. See Appendix Section 5.3.3 for discussion.

We next explore whether these biases can be addressed by collecting a small validation sample 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 10$ 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.

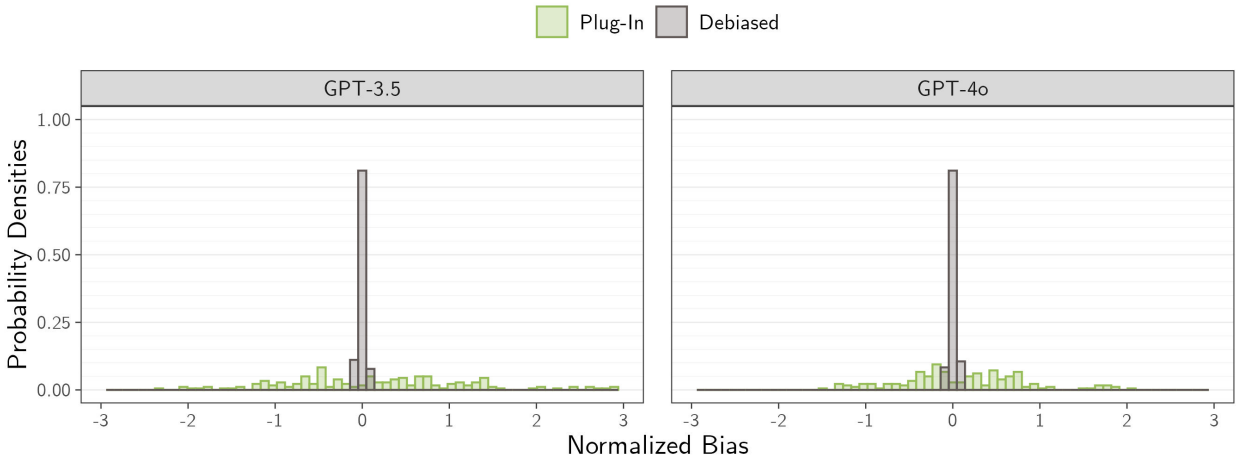


Figure A8: Normalized bias of the plug-in regression and bias-corrected regression using policy topic as a covariate across Monte Carlo simulations based on congressional legislation

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 10% validation sample. Results are averaged over 1,000 simulations. See Appendix Section D.2 for discussion.

	Median	5%	95%
Normalized Bias			
Plug-In	0.147	-1.437	2.099
Debiased	-0.002	-0.055	0.057
Coverage			
Plug-In	0.897	0.353	0.949
Debiased	0.941	0.927	0.954

(a) GPT-3.5-turbo

	Median	5%	95%
Normalized Bias			
Plug-In	0.024	-1.118	1.624
Debiased	-0.001	-0.059	0.055
Coverage			
Plug-In	0.927	0.642	0.952
Debiased	0.943	0.928	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 10% validation sample. Results are averaged over 1,000 simulations. See Appendix Section D.2 for discussion.

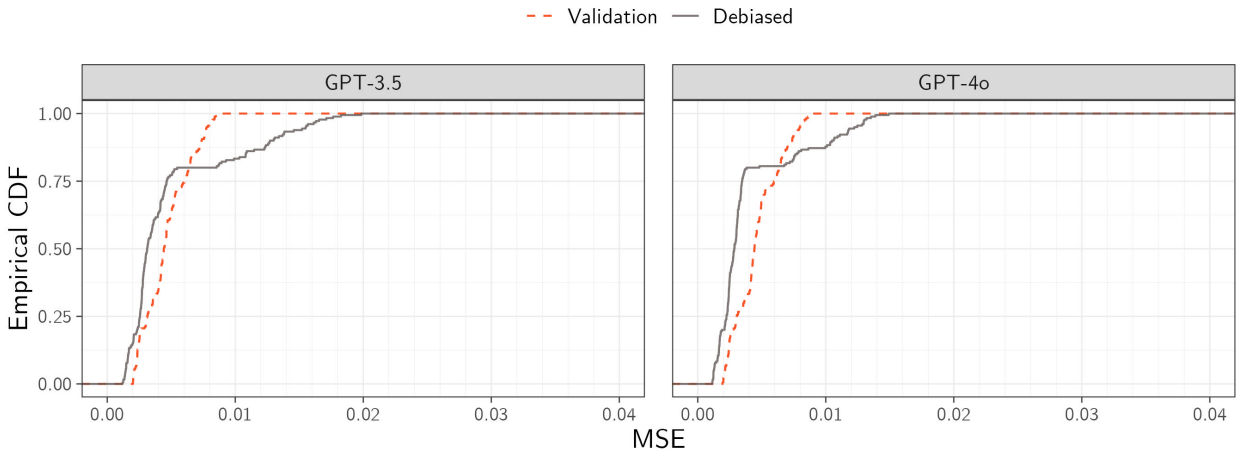


Figure A9: Cumulative distribution function of mean square error for the bias-corrected estimator against validation-sample only estimator using policy topic as a covariate.

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 10% 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.

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 A10: 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 A11: 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 A12: 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 A13: 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 A14: 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 fill-in-the-blanks output (Panel (a) of Appendix Figure A13). 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 A15: 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 how the base prompt with fill-in-the-blanks output (Panel (a) of Appendix Figure A13) asks for the model's response. 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 A16: 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 its 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 A17: 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 its major topic. Each persona modification is added to the beginning of the base prompt with JSON output (Panel (b) of Appendix Figure A16). 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 A18: 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 its major topic. Each chain-of-though modification alters how the base prompt with JSON output (Panel (b) of Appendix Figure A16) asks for the model's response. See Section 5.3.3 of the main text for further details.