NBER WORKING PAPER SERIES

# LARGE LANGUAGE MODELS: AN APPLIED ECONOMETRIC FRAMEWORK

Jens Ludwig
Sendhil Mullainathan
Ashesh Rambachan

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

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

This paper was supported by the Center for Applied Artificial Intelligence at the University of Chicago and the Altman Family Fund at MIT. We thank Haya Alsharif and Janani Sekar for excellent research assistance. We thank Peter Bergman, Tim Christensen, Oeindrila Dube, Larry Katz, Lindsey Raymond, Suproteem Sarkar, Andrei Shleifer, and David Yanagizawa-Drott as well as numerous seminar audiences and conference participants for helpful comments. We also thank the editor and three reviewers for their valuable feedback. Wharton Research Data Services (WRDS) was used in preparing this paper. This service and the data available constitute valuable intellectual property and trade secrets of WRDS and/or its third-party suppliers. All interpretations and any errors are our own. The views expressed herein are those of the authors and do not necessarily reflect the views of the National Bureau of Economic Research.

NBER working papers are circulated for discussion and comment purposes. They have not been peer-reviewed or been subject to the review by the NBER Board of Directors that accompanies official NBER publications.

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

## ABSTRACT

Large language models (LLMs) enable researchers to analyze text at unprecedented scale and minimal cost. Researchers can now revisit old questions and tackle novel ones with rich data. We provide an econometric framework for realizing this potential in two empirical uses. For prediction problems – forecasting outcomes from text – valid conclusions require "no training leakage" between the LLM's training data and the researcher's sample, which can be enforced through careful model choice and research design. For estimation problems – automating the measurement of economic concepts for downstream analysis – valid downstream inference requires combining LLM outputs with a small validation sample to deliver consistent and precise estimates. Absent a validation sample, researchers cannot assess possible errors in LLM outputs, and consequently seemingly innocuous choices (which model, which prompt) can produce dramatically different parameter estimates. When used appropriately, LLMs are powerful tools that can expand the frontier of empirical economics.

Jens Ludwig
University of Chicago
Harris School of Public Policy
and NBER
jludwig@uchicago.edu

Ashesh Rambachan
Massachusetts Institute of Technology
asheshr@mit.edu

Sendhil Mullainathan
Massachusetts Institute of Technology
Department of Electrical Engineering
and Computer Science,
and Department of Economics
and NBER
sendhil.mullainathan@gmail.com

# 1 Introduction

Large language models (LLMs) enable economists to process text at unprecedented scale and minimal cost, unlocking questions previously out of reach: predicting stock returns from earnings call transcripts, measuring partisan polarization in social media posts, backcasting historical consumer sentiment from newspapers, or simulating survey responses cheaply. By making these questions – and countless others – tractable, LLMs offer transformative potential for empirical research.

To realize this potential, we must answer a practical question: how should LLM outputs be incorporated into empirical workflows? Can they be plugged in "as-is," or do our estimation strategies require adjustment to use these powerful tools?

This question is challenging because LLMs resist traditional statistical analysis. LLMs are complex, often proprietary, constantly evolving, and trained on sprawling, heterogeneous corpora that defy tractable modeling – extraordinary engineering achievements accomplished without our usual statistical foundations. Existing evaluation methods have been remarkably effective for developing better models but offer limited guidance for how any given LLM will perform on a new task.

We develop an econometric framework that addresses these complexities and provides practical guidance for empirical research using LLM outputs, focusing on two empirical uses: prediction problems and estimation problems. For each use, we clarify what assumptions and practices ensure valid conclusions.

In prediction problems, researchers use collected text to predict some economic outcome (e.g., predicting stock prices from corporate earnings calls). Extracting meaning from text requires modeling the complex structure of language. LLMs, having already learned this structure from enormous training corpora, can possibly serve as the foundation upon which economists can build in prediction problems — either by directly prompting an LLM to make a prediction or by using its representations as features in a predictor.

Using LLMs in prediction problems requires one condition: "no training leakage." If a researcher forms predictions using an LLM and evaluates those predictions on an evaluation dataset, this reflects the LLM's out-of-sample performance if and only if there is no overlap between the LLM's training dataset and the researcher's dataset. This can be violated because many LLMs are trained on intentionally obscured datasets. Computer scientists have found that LLMs are often trained on common benchmark evaluations. We find that LLMs appear to be trained on economic datasets; see also Glasserman and Lin (2023); Sarkar and Vafa (2024); Lopez-Lira, Tang and Zhu (2025) among others.

We provide guidance on enforcing no training leakage, clarifying that it requires careful

attention to both the choice of model and research design. For example, when the goal is to predict future, unseen documents (such as tomorrow's financial news), open-source LLMs with fixed, published weights (e.g., Touvron et al., 2023; Dubey et al., 2024) or time-stamped training data (e.g., Sarkar, 2024; He et al., 2025) can be paired with evaluation samples constructed only from documents published after the model's publication date or training cutoff.

In estimation problems, researchers estimate relationships between economic concepts expressed in text (e.g., partisanship in social media posts) and downstream parameters (e.g., the causal effect of a policy change). There is *some* resource-intensive procedure for measuring the economic concept; if it could be scaled, we would use these measurements and report the resulting estimates. Often, this would involve the researcher carefully reading each text piece themselves. We would like to instead use an LLM to economize on measurement costs. How can we use LLM outputs for valid downstream inference?

We highlight a constructive solution borrowing from an old idea in econometrics: collect a small validation sample and use it to *empirically* correct for possible LLM errors. We illustrate how researchers can incorporate validation data in the context of linear regression. Debiasing LLM outputs preserves our usual econometric guarantees of consistency and asymptotic normality. This has been well-studied in econometrics (e.g., Lee and Sepanski, 1995; Chen, Hong and Tamer, 2005; Schennach, 2016) and extended in machine learning, such as Wang, McCormick and Leek (2020), Angelopoulos et al. (2023), Egami et al. (2024), Battaglia et al. (2024), and Carlson and Dell (2025). We illustrate that incorporating debiased LLM outputs can substantially improve the precision of downstream estimates – resulting in estimates that are often more precise than using validation data alone. Used correctly, imperfect LLM outputs serve not as substitutes but as amplifiers of validation samples, allowing researchers to achieve tighter standard errors.

Absent a validation sample, researchers cannot assess the magnitude or pattern of errors in LLM outputs—and therefore cannot evaluate their impact on downstream parameter estimates. We demonstrate this problem empirically: absent a validation sample, seemingly innocuous choices—which LLM to use, how to phrase the prompt—lead to dramatically different parameter estimates in applications to finance and political economy, with coefficients varying in magnitude, sign, and significance. In estimation problems, researchers have no choice but to invest effort and collect validation samples when using LLMs.

Finally, this framework is flexible enough to account for creative uses of LLMs in economics. We argue that using LLMs for hypothesis generation (e.g., Ludwig and Mullainathan, 2024) can be cast as a form of prediction problem, so that the key assumption required is again no training leakage. LLM simulation of human subjects in surveys or ex-

periments (e.g., Park et al., 2023; Horton, 2023; Manning, Zhu and Horton, 2024; Park et al., 2024) can be thought of as an estimation problem, and so in-silico subjects can amplify – but not fully replace – a small validation sample of actual human responses.

Our framework clarifies when and how researchers can harness LLMs in empirical research. The requirements are straightforward: ensure no training leakage for prediction and collect small validation samples for estimation. We provide a checklist based on our framework in Section 6. These simple practices unlock the transformative potential of LLMs for empirical research.

# 2 An Econometric Framework for Large Language Models

LLMs are complex architectures trained on vast corpora through multi-stage processes: pretraining (learning to predict next tokens), instruction fine-tuning, reinforcement learning from human feedback (RLHF), and reinforcement learning from verifiable rewards (RLVR). Reasoning models employ test-time computation to further improve performance. The field evolves rapidly, with new architectures, training procedures, and capabilities emerging regularly.[1]

This complexity creates a fundamental challenge for econometric analysis. We typically study statistical procedures by articulating assumptions about the data-generating process and combining them with a computational understanding of the procedure. This approach is at present intractable for LLMs. They have billions of parameters, proprietary training datasets, and algorithms that resist formal characterization.

We adopt a different strategy based on how economists actually use these tools. We treat LLMs as black boxes – prompting them or extracting embeddings without characterizing internal mechanisms – and identify conditions they must satisfy for valid empirical use. We focus on two applications: prediction problems and estimation problems. By black-boxing their inner workings, our approach provides guidance that is robust to inevitable changes in architectures and training procedures.

## 2.1 Setting and the Researcher's Dataset

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

[1] Many resources are available about the technical foundations of large language models. See, for example, Chang et al. (2024), Minaee et al. (2025), and Zhao et al. (2025). For overviews aimed at economists, see Korinek (2023), Dell (2024), Korinek (2024), and Ash, Hansen and Muvdi (2024).

For any empirical question, only some strings are economically relevant. Denote these as $\mathcal{R} \subseteq \Sigma^*$ with elements $r \in \mathcal{R}$ that we refer to as text pieces. The researcher's dataset is summarized by the vector $d$ whose elements $d_\sigma$ are sampling indicators for whether the researcher collected string $\sigma$. The researcher only collects economically relevant text pieces, and so $d_\sigma = 0$ for all $\sigma \in \Sigma^* \setminus \mathcal{R}$.

Each text piece $r$ can be linked to observable economic variables $(Y_r, W_r)$, which are economic outcomes $Y_r$ that might be influenced by the text or candidate covariates $W_r$ that might influence the text.

**Example: Congressional legislation** Consider descriptions of bills introduced in the United States Congress. Each text piece $r \in \mathcal{R}$ is a bill's description such as "`A bill to revise the boundary of Crater Lake National Park in the State of Oregon`." The economic outcome $Y_r$ might be whether the bill passed its originating chamber. The covariate $W_r$ 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 $r \in \mathcal{R}$ is a 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 window after the headline's publication date, while the covariate $W_r$ could be the company's past fundamentals. ▲

Each text piece $r$ expresses some economic concept $V_r := f^*(r)$, where $f^*(\cdot)$ is the measurement procedure the researcher would use if time and resources were no constraint. Typically these measurements are what the researcher themselves or an appropriate domain expert would produce by carefully reading each text piece.[2] In defining the economic concept, researchers must answer: what exactly am I measuring from the text, and how would I label it if resources were no constraint? Moving forward, we assume a measurement procedure $f^*(\cdot)$ exists that the researcher would be satisfied to use *if* it could be scaled.

This creates a text processing problem: measuring the economic concept requires processing each text piece $r$, which may be prohibitively costly at scale. Absent a solution to this text processing problem, the researcher cannot collect $V_r$ on all text pieces.

In settings like this, researchers would like use the collected text pieces to tackle two types of empirical analyses. The first is a *prediction problem*: predict the linked variable $Y_r$ using the associated text piece $r$. For example, can we predict stock returns from news

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

headlines? The second is an *estimation problem*: estimate some parameter that relates the economic concept $V_r$ to the linked variables $(Y_r, W_r)$. If bill descriptions $r$ express the policy topic $V_r$ of the bill (e.g., whether it is related to foreign affairs or health policy), how do policy topics relate to the party affiliation of the bill's sponsor $W_r$?

LLMs are general-purpose and easy-to-use models for processing text, so researchers would like to use them in prediction and estimation problems. We next introduce LLMs into our framework.

## 2.2 Incorporating Large Language Models in Empirical Research

To capture how we often interact with LLMs as black boxes — generating responses from prompts without knowing their design nor training data — 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 trained on dataset $t$ and $\widehat{m}(\sigma; t)$ is the LLM's response when prompted by string $\sigma$.

This definition has a key implication: the LLM is actually two distinct algorithms. First, a training algorithm that takes any dataset $t$ and learns the mapping between strings. Second, the text generator $\widehat{m}(\cdot; t)$ is the output of the training algorithm and what users interact with.

Our analysis will not depend on how exactly these algorithms work. Since the state of the art is constantly evolving, we should expect the exact implementation of training algorithms and text generators to change. Furthermore, alternative LLMs may differ in their implementations. Studying LLMs at this level of abstraction will provide interpretable conditions for empirical research and ensures the durability of our analysis. Researchers will have conditions under which any model's output can be incorporated into empirical research.

Our framework captures all key LLM design choices — some made by researchers, others by algorithm builders (often invisibly to researchers).

**Interpreting the Text Generator** The text generator $\widehat{m}(\cdot; t)$ captures all choices that influence how an LLM generates responses once trained. This includes prompt engineering strategies that materially affect the quality of responses (e.g., Liu et al., 2023; Wei et al., 2024; White et al., 2023; Chen et al., 2024). Alternative prompt engineering strategies can be cast as alternative specifications of the text generator $\widehat{m}(\cdot; t)$.

The text generator captures other important choices governing generation. Parameters like temperature, top-$p$ sampling, or top-$k$ sampling control randomness in sampling from the LLM's probability distribution over next tokens. Reasoning models employ test-time computation, generating intermediate "thought tokens" before producing final outputs to

enhance performance on complex tasks.[3] 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 generates its most likely token. Our results extend naturally to stochastic text generators as well, but at the expense of more cumbersome notation.[4]

**Interpreting the Training Algorithm** The training algorithm captures all aspects of design and training that produce the text generator. This includes architectural choices (e.g., parameter count, attention layers, context window), the pretraining objective (typically next-token prediction), and optimization details. LLMs undergo multiple training stages beyond pretraining. Instruction fine-tuning trains models to follow user instructions, and reinforcement learning from human feedback (RLHF) aligns output with human preferences. More recent developments include reinforcement learning from verifiable rewards (RLVR), where models are trained on tasks with objectively correct answers. Crucially, the training dataset $t$ in our framework encompasses all strings used in pretraining *and* post-training stages.

So how do researchers use LLMs in empirical research? In prediction problems, researchers often prompt an LLM with each text piece $r$ to generate a prediction $\hat{Y}_r = \widehat{m}(r;t)$. For instance, they might prompt a model to predict whether a stock return will be positive. In estimation problems, researchers prompt an LLM to generate labels $\hat{V}_r = \widehat{m}(r;t)$ for the economic concept. For example, they might prompt a model to classify a bill's policy topic.

## 2.3 The Challenge of Evaluating Large Language Models

At some level, the viability of using LLMs for prediction and estimation problems hinges on the quality of LLM outputs: how large are possible errors $Y_r - \widehat{Y}_r$ and $V_r - \widehat{V}_r$ in any application? A natural starting point is to understand how computer science approaches this evaluation problem.

Computer scientists adapted methods from supervised learning. In supervised learning, the "common task framework" (Donoho, 2024) builds benchmark datasets like ImageNet for image classification and the Netflix Prize for recommender systems and evaluates competing algorithms on these benchmarks. Since LLMs aspire to be general-purpose technologies useful across all tasks, computer scientists have built increasingly diverse benchmarks.

[3]For many reasoning models, users cannot control generation parameters like temperature. Consequently, there is inherent randomness that cannot be eliminated.

[4]A practical challenge is ensuring LLM outputs are formatted appropriately for empirical analysis. Researchers often prompt LLMs to return structured outputs, but LLMs may not reliably comply. Our text generator abstraction takes further post-processing steps as given, and $\widehat{m}(\cdot;t)$ represents the resulting final output.

For example, the "Beyond-the-Imitation-Game benchmark" (BIG-bench) collects problems across 204 tasks (Srivastava et al., 2022), while the "Massive Multitask Language Understanding" (MMLU) benchmark spans 57 categories from logic to social sciences (Hendrycks et al., 2020). Other benchmarks assess specialized capabilities: SWE-bench for coding ability (Jimenez et al., 2024), GSM8K for mathematical reasoning (Cobbe et al., 2021), and standardized exams such as the SAT, GRE, and AP tests. Modern LLMs perform remarkably well on these benchmarks, and this impressive performance forms much of the quantitative basis for current enthusiasm.

Can economists use these benchmark evaluations to reason about how LLMs will perform on specific empirical tasks? The answer is more complicated than it might initially appear.

### 2.3.1 The Limits of Benchmarks

We do not intrinsically care about benchmark performance itself — after all, who deploys LLMs to solve SAT problems? We hope to *generalize* from benchmarks to new economically relevant tasks.

This generalization happens intuitively. We assume an LLM that aces AP chemistry must, like a person, handle many chemistry-related tasks. Evidence for "anthropomorphic generalization" suggests that people apply similar generalization heuristics to LLMs as they do to humans when predicting performance across tasks (Vafa, Rambachan and Mullainathan, 2024; Dreyfuss and Raux, 2024).

Yet this intuition is misleading. LLMs exhibit what Mancoridis et al. (2025) call "potemkin understanding" — performing well on benchmarks without grasping underlying concepts. This manifests as remarkable brittleness: performance is sensitive to seemingly minor details that would not affect human performance. Consider several examples from an accumulating body of evidence. While humans who master one math problem typically solve easier variations, LLMs have not readily generalized this way. An LLM that could reliably solve $(9/5)x + 32$ could not solve $(7/5)x + 31$ (McCoy et al., 2024). An LLM correctly defines an ABAB rhyming scheme but then generates poems violating its own definition (Mancoridis et al., 2025). An LLM trained on "A is B" will not know "B is A" (Berglund et al., 2023) and LLMs struggle with logical puzzle variations (Nezhurina et al., 2024). Brittleness extends to presentation: LLMs answer multiple-choice questions correctly but fail when answer order is permuted (Zong et al., 2024), succeed at programming with 0-based indexing but fail with 1-based indexing (Wu et al., 2024), and show inconsistent performance on similar spatial reasoning tasks (Mitchell, 2023).

LLM errors align poorly with human intuitions precisely because we expect them to understand and misunderstand as humans do. Dell'Acqua et al. (2023) aptly characterize

this as AI's "jagged frontier" – a landscape where performance on similar tasks varies unpredictably. This jagged frontier captures why benchmark performance cannot be intuitively generalized to specific economic applications.

### 2.3.2 Do Large Language Models have World Models?

Perhaps this is too pessimistic, and this brittleness is superficial – noise around a deeper structural understanding embedded within LLMs. LLMs are trained on massive corpora containing rich information about reality, followed by extensive reinforcement learning. Perhaps LLMs have successfully learned "world models" – internal, structured representations of how the world works – that would enable reliable generalization despite occasional errors.

This hypothesis is an active research area focused on settings where researchers can both infer the LLM's implicit world model and compare it to the truth (e.g., Li et al., 2024; Nanda, Lee and Wattenberg, 2023; jylin04 et al., 2024; Nikankin et al., 2025). The evidence so far is at best mixed and at worst suggests LLMs may not learn generalizable world models. For example, Vafa et al. (2024) trained a generative sequence model on turn-by-turn driving data from 12.6 million NYC taxi rides. While it predicts the next turn between locations with high accuracy, the authors reverse-engineered the model's implicit map of Manhattan's street grid: the implicit map bears no resemblance to the actual grid. High predictive accuracy did not require learning the underlying spatial structure. Vafa et al. (2025) trained a generative sequence model on planetary orbits. Despite accurately predicting planetary movements, the model reveals no understanding of Newtonian physics. Rather than learning the gravitational law, it relies on task-specific heuristics that fail to generalize beyond its training distribution. Even advanced reasoning models exhibited similar failures.

Taken together, the evidence suggests that LLMs not only make errors, but these errors cannot be reliably predicted. They fail in ways misaligned with how humans generalize across tasks, nor can we assume these errors are noise around accurate world models. This creates our challenge. Since LLMs are impressive yet brittle tools, how should researchers incorporate their outputs into empirical research?

# 3 Prediction with Large Language Models

We begin with prediction problems – using text to predict economic outcomes. LLMs' extensive training makes them natural candidates for this task, having already learned rich representations of language. Valid inference in prediction problems hinges on one condition: no leakage between the model's training data and the researcher's evaluation sample. While training leakage plagues benchmark evaluations in computer science, economists can manage it through careful model choice and research design.

## 3.1 The Researcher's Prediction Problem

Suppose the researcher prompts an LLM to predict the linked variable from text pieces, $\widehat{Y}_r = \widehat{m}(r;t)$.[5] For some loss function $\ell(y, \tilde{y})$, the researcher calculates the sample average loss

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

where $N = \sum_r D_r$ is the number of text pieces collected. We would like to draw conclusions about the predictability of $Y_r$ from the text piece $r$ based on the LLM's sample average loss. When is this valid?

Researchers face two sources of uncertainty about LLMs: What data was the LLM trained on? How does it generate output from text? We introduce two objects to formalize these uncertainties. The research context formalizes what we know (and do not know) about the LLM's training data. The LLM's guarantee summarizes high-level properties about how the model behaves without requiring exact knowledge of its internal workings. Together, these will allow us to derive interpretable conditions for using LLMs in prediction problems.

The *research context* $Q(\cdot) \in \mathcal{Q}$ is a joint distribution over the sampling indicators $(D, T)$. It encodes two distinct features. The sampling distribution over $D$ defines the researcher's out-of-sample prediction problem – the collection of text pieces over which they want to evaluate the LLM's predictive performance. The sampling distribution over $T$ captures the researcher's uncertainty about the LLM's training data. We make a technical assumption about the collection of research contexts $\mathcal{Q}$.

**Assumption 1.** Letting $t = (t_{\sigma_1}, \ldots, t_{\sigma_{|\Sigma^*|}})'$ denote the sampling indicators summarizing the LLM's realized training dataset, all research contexts $Q(\cdot) \in \mathcal{Q}$ satisfy: (i) For all values $d$, $Q(D = d, T = t) = \Pi_{\sigma\in\Sigma^*} Q(D_\sigma = d_\sigma, T_\sigma = t_\sigma)$; and (ii) $\mathbb{E}_Q[\sum_{r\in\mathcal{R}} D_r] = \mathbb{E}_Q[\sum_{r\in\mathcal{R}} D_r \mid T = t]$.

Assumption 1(i) states that sampling across strings is independent but not necessarily identically distributed. Assumption 1(ii) states that the researcher's expected sample size does not depend on the LLM's training corpus. Let $q_\sigma^{T|D}(t_\sigma) = Q(T_\sigma = t_\sigma \mid D_\sigma = 1)$ denote the conditional probability the string is sampled by the LLM's training dataset given that it is sampled by the researcher. The marginal probabilities are $q_\sigma^T(t_\sigma) = Q(T_\sigma = t_\sigma)$ and $q_\sigma^D = Q(D_\sigma = 1)$.

Our goal is to assess the LLM's predictive performance over the population of text pieces

[5] Researchers may use an 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 equally applies to evaluating the performance of a prediction function using LLM embeddings as features.

defined by the research context $Q(\cdot)$:

$$\mathbb{E}_Q\left[\sum_{r\in\mathcal{R}} D_r \ell(Y_r, \widehat{m}(r;t))\right]. \tag{2}$$

This is summarizes whether the model's predictive performance on the broader target population. As the number of economically relevant text pieces grows large (see Appendix C.1), we can characterize what the sample average loss converges to. Conditional on the LLM's realized training dataset,

$$\frac{1}{N}\sum_{r\in\mathcal{R}} D_r \ell(Y_r, \widehat{m}(r;t)) - \frac{1}{\mathbb{E}_Q[\sum_{r\in\mathcal{R}} D_r \mid T=t]}\mathbb{E}_Q\left[\sum_{r\in\mathcal{R}} D_r \ell(Y_r, \widehat{m}(r;t)) \mid T=t\right] \xrightarrow{p} 0. \tag{3}$$

Conditioning on the training dataset $t$ treats the LLM $\widehat{m}(\cdot;t)$ as a fixed mapping, avoiding strong assumptions about how the LLM was trained (e.g., how would it behave over counterfactual training datasets?). The sample average loss recovers our target quantity in Equation (2) if $\mathbb{E}_Q\left[\sum_{r\in\mathcal{R}} D_r \ell(Y_r, \widehat{m}(r;t)) \mid T=t\right] = \mathbb{E}_Q\left[\sum_{r\in\mathcal{R}} D_r \ell(Y_r, \widehat{m}(r;t))\right]$.

Recall the second source of uncertainty: researchers do not know precisely how the LLM generates outputs from text. Yet researchers often observe other high-level information about the LLM's behavior, such as performance on benchmarks (e.g., "achieves 88.7% on MMLU" or "scores 1520 on the SAT"). We formalize this through a *guarantee* $\mathcal{M}$, a collection of possible text generators that captures what the researcher knows about the LLM's behavior. The researcher only knows that $\widehat{m}(\cdot;t) \in \mathcal{M}$.

Given a guarantee $\mathcal{M}$ and a research context $Q(\cdot)$, we define when this workflow is valid for prediction problems.

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

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

This definition formalizes an out-of-sample prediction goal. The LLM generalizes if its sample average loss reflects its predictive performance on the target population. Under Definition 1, the researcher's workflow in the prediction problem is justified for *any* LLM satisfying the guarantee $\mathcal{M}$. The researcher can draw conclusions based on the sample average loss knowing only the guarantee $\mathcal{M}$ is satisfied.

**Example: Congressional legislation** Consider researchers predicting whether a bill passes either house of Congress $Y_r$ using only its text description $r$. The LLM generates

predictions $\widehat{m}(r;t)$ using a specific prompting strategy. Each researcher calculates the sample average loss of the LLM's predictions on their own collected sample of Congressional bills. Different researchers face different prediction problems, formalized by alternative research contexts $Q(\cdot)$. One researcher might assess whether the LLM can predict outcomes for future legislation by sampling bills after a cutoff date, such as all bills from 2025 onward. Another might evaluate whether the LLM's predictions generalize to novel policy domains, sampling bills on as cryptocurrency regulation or artificial intelligence policy. ▲

**Example: Financial news headlines** Consider researchers predicting a company's realized returns $Y_r$ from a financial news headline $r$. The LLM generates predictions $\widehat{m}(r;t)$ using a specific prompting strategy. Each researcher calculates the sample average loss on their collected headlines. Different researchers again face different out-of-sample prediction problems, formalized by alternative research contexts $Q(\cdot)$. One researcher might assess whether the LLM predicts returns during future market conditions, sampling headlines from 2025 onward. Another might evaluate whether predictions generalize across company types, focusing on small-cap technology firms or international equities. ▲

## 3.2 Training Leakage as a Threat to Prediction

Using LLMs in prediction problems requires one condition: no training leakage between the LLM's training data and the researcher's evaluation sample.

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

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

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

$$\mathcal{M}(Q) = \left\{\widehat{m}(\cdot):\ -\mathbb{E}_Q\left[\sum_{r\in\mathcal{R}} D_r \left(\frac{q_r^{T|D}(t_r)}{q_r^{T}(t_r)} - 1\right) \ell(Y_r, \widehat{m}(r))\right] = 0\right\}. \tag{4}$$

The no-training leakage condition (Equation 4) captures the extent to which overlap between the LLM's training data and the researcher's sample covaries with the LLM's predictive performance. It has an intuitive interpretation as omitted variable bias. The term $\frac{q_r^{T|D}(t_r)}{q_r^{T}(t_r)} - 1$ measures how learning that the researcher sampled text piece $r$ updates beliefs about whether $r$ appeared in the LLM's training data. When this correlation is positive – text pieces in the researcher's sample are more likely to have been in the training data – and the LLM performs well on such pieces, the overall bias term is positive and the sample average loss overstates

true predictive performance. Uncertainty over what entered into the LLM's training data acts like an omitted variables bias in the prediction problem.

This bias arises from both the researcher's choice of out-of-sample prediction and our beliefs about what entered into the LLM's training data. Section 3.3 discusses how researchers can prevent training leakage through careful choices of model and prediction exercise.

### 3.2.1 Some Evidence of Training Leakage

Training leakage is a well-documented problem in computer science. LLMs' training datasets frequently contain examples from popular benchmark evaluations (Sainz et al., 2023; Golchin and Surdeanu, 2024; LM Contamination Index, 2024). This has generated skepticism about evaluating LLMs on any publicly available data (e.g., Ravaut et al., 2024). But this evidence focuses on CS applications. Might economic prediction problems face similar risks?

**Assessing Training Leakage in Congressional Legislation** We assess training leakage in an empirical setting relevant for economists studying politics and political economy: congressional legislation. The Congressional Bills Project (Wilkerson et al., 2023; Adler and Wilkerson, 2020) contains descriptions $r$ for over 400,000 bills proposed in Congress, along with whether each passed the House or Senate $Y_r$. We sample $10,000$ bills introduced from 1973 to 2016 and test whether passage can be predicted from text descriptions alone — a potentially challenging task given the strategic dynamics of congressional voting. Among these bills, only 7.4% pass the House and 6.0% pass the Senate.

We generate predictions $\widehat{Y}_r = \widehat{m}(r;t)$ based on each bill's description $r$ by prompting GPT-4o (see Appendix Figure A13 for the specific prompt). 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 Appendix Table A1). What drives the model's ability to accurately predict whether a Congressional bill will pass the House or the Senate based only on its description? The answer is that the text of congressional legislation is likely included in its training dataset.

To evaluate training leakage, we prompt the model to complete each bill's text description based on only the first half of its text, following research in computer science such as Golchin and Surdeanu (2024) (see Appendix Figure A14 for the specific prompt). If a model reproduces the text exactly, it has likely seen it during training. This is not a necessary condition for training leakage; models may benefit from exposure to a text piece without memorization. Perfect reproduction nonetheless offers compelling evidence.

On 344 bills, GPT-4o completes the bill's text description *exactly* as it is written, indicating that not only was GPT-4o likely trained on these text pieces but it appears to have memorized them. Appendix Figure A1 provides two examples of successful completions. On

the other bills, GPT-4o's completed bill descriptions are close to the original bill descriptions. 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 A2).

One might wonder whether this training leakage could be addressed through prompt engineering — for example, by commanding the models to not pay attention to any information past a certain date. Applying this prompt engineering strategy to our sample of Congressional bills (see Appendix Figure A13 and A14 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 Appendix Table A1). GPT-4o still exactly completes nearly the same number of bill descriptions as without the prompt engineering (330 versus 344), and the word embeddings of its completed descriptions remain quite close on average to those of the originals (right panel of Appendix Table A2).

**Assessing Training Leakage in Financial News Headlines** We consider another domain relevant for economists: financial markets. Existing work (e.g., Glasserman and Lin, 2023; Lopez-Lira and Tang, 2024) found that LLMs predict stock returns accurately from news headlines. We test whether this reflects training leakage using publicly available headline data (Aenlle, 2020) covering nearly 4 million headlines for 6,000 publicly traded companies from 2009-2020.

We sampled 10,000 financial news headlines from 2019, and we prompt GPT-4o to complete each financial news headline based on only 50% of its text (see Appendix Figure A15 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. Appendix Figure A2 provides two examples of successfully completions. On all other headlines, word embeddings of the model's completions are close to those of the original headlines (left panel of Appendix Table A3).

We again explore whether explicitly incorporating date restrictions into the LLM prompt moderate this evidence of training leakage (see Appendix Figure A15 for the associated prompts).[6] Surprisingly, this appears to make the problem *worse*; GPT-4o now reproduces 73 headlines exactly, and word embeddings of GPT-4o's completions with the date restriction are still on average close to those of the original headlines.

[6]See also Wongchamcharoen and Glasserman (2025). Another prompting strategy – entity masking – aims to prevent memorized information by masking identifiers in prompts (Glasserman and Lin, 2023; Sarkar and Vafa, 2024; Engelberg et al., 2025). However, implementation details matter, and its wider applicability is unclear. In our congressional legislation example, it is not obvious what should be masked.

Sarkar and Vafa (2024) provide additional evidence of training leakage in finance: prompting Llama 2 to predict risks from September-November 2019 earnings calls, the LLM mentions Covid-19 in over 25% of cases—a form of "look-ahead bias" from training on future information. Combined with our findings and other evidence (Glasserman and Lin, 2023; Lopez-Lira, Tang and Zhu, 2025), this indicates substantial risk of training leakage in finance applications.

### 3.3 Practical Guidance for Prediction Problems

The no-training-leakage condition (Equation 4) may seem abstract, but it provides concrete guidance for empirical practice. Researchers must consider what population they are predicting on, how their evaluation sample relates to it, and how their evaluation sample relates to the LLM's training corpus.

To make this concrete, recall that the bias term in Proposition 1 depends on $\frac{q^{T|D}(t_r)}{q^T(t_r)} - 1$ – how much does learning that the researcher sampled text piece $r$ update our beliefs about whether $r$ was in the training data. No training leakage requires this term to be zero, or at least uncorrelated with the LLM's performance. Different prediction problems and model choices achieve this in different ways. We illustrate this by considering the threat of training leakage and how to manage it across several examples.

**Example: Lookahead Bias and Time-Stamped Models** Consider a researcher who wants to predict stock returns from future financial news headlines. If the researcher evaluates on recent headlines from 2024 using an LLM trained on data through 2025, the leakage term is almost certainly positive: both the researcher and the LLM builders had access to 2024 headlines. Any predictive success may reflect training leakage.

The solution in this case is to use open-source LLMs with fixed published weights, such as the Llama family (Touvron et al., 2023; Dubey et al., 2024) and others (e.g., BLOOM, OLMo, etc.), or time-stamped training data (e.g., Sarkar, 2024; He et al., 2025). If the researcher uses a model with weights published on date $\tau$ and constructs an evaluation sample using documents published after $\tau$, then $q_r^T(t_r) = 0$ mechanically since they could not have been in the training data. This eliminates training leakage by design.

Closed models like the GPT family from OpenAI or the Claude family from Anthropic do not permit this solution. Their training data is undisclosed, and they may be continuously fine-tuned. 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).[7] Most

[7] Balloccu et al. (2024) estimates that 263 benchmarks 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

worryingly, 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. ▲

**Example: Prediction on Confidential Documents** Suppose a researcher wants to predict, for example, whether administrative case notes are predictive of some decision, like pretrial release. In examples like this, the target population consists of confidential documents that were never publicly released. Since these documents never entered any public corpus, we can again credibly argue that $q_r^T(t_r) = 0$ for all documents since they were never available for training. This reasoning hinges on credibly claiming the documents' provenance. ▲

**Example: Random Sampling from a Known Corpus** Consider a researcher with a complete corpus of economics papers published between 2000 and 2020 who wants to predict citation counts from abstracts. She draws a random sample to form her evaluation set. In this case, the researcher's sampling process is independent of the LLM's training data: her random number generator knows nothing about which papers OpenAI included in their training data. This implies $q_r^{T|D}(t_r) = q_r^T(t_r)$ – learning that a paper was randomly selected provides no information about whether it was trained on — so the leakage term equals zero.

The logic is familiar from causal inference: randomization eliminates omitted variable bias by making treatment assignment independent of potential outcomes. Here, random sampling eliminates training leakage by making the researcher's evaluation sample independent of the LLM's training data. This can be applied whenever the researcher draws a genuinely random sample from a well-defined corpus before using an LLM. The converse is also important: just as non-random treatment assignment resurrects possible confounding in causal inference, non-random sampling may resurrect training leakage. ▲

**Example: Prediction on the Complete Population** As a final example, suppose the researcher collected all Congressional bill descriptions from 1973–2016 and wants to understand what textual features predict passage. The researcher seeks to descriptively characterize patterns in this specific historical corpus, not to make predictions on some population of legislation. When the researcher observes the entire population, the prediction problem changes character. There is no sampling, and hence no inference to a broader population. In this case, $q_r^D = 1$ and the leakage condition is satisfied trivially: we are not asking whether performance generalizes beyond what we observe. The complete-population case licenses only descriptive conclusions about the collected corpus. ▲

"knowledge cutoffs" in closed LLMs.

More broadly, these examples illustrate a general principle: no training leakage is ultimately about *both* research design and model properties. Researchers must be explicit about their target population, their sampling procedure, and how that procedure relates to possible training corpora. When in doubt, the safest approach combines open-source models with published weights or documented training data and evaluation samples constructed to mechanically exclude the model's training corpus.

# 4 Estimation with Large Language Models

In estimation problems, the researcher measures economic concepts $V_r$ expressed in text pieces $r$ to estimate downstream parameters. The measurement $f^*(\cdot)$ is costly to scale across thousands or millions of text pieces. This is where LLMs enter as potential substitutes.

Using LLM outputs in plug-in estimation requires a strong assumption: the LLM must reproduce the existing measurement everywhere. As Section 2.3 discussed, LLM performance varies across similar tasks, and benchmark evaluations provide little guidance on new applications. We demonstrate this empirically: seemingly minor choices – which LLM, which prompt – substantially affect downstream parameter estimates in applications to finance and political economy. These choices change the magnitude, significance, and even sign of estimated parameters. The solution: collect a small validation sample and use it to debias the LLM's outputs.

## 4.1 The Researcher's Estimation Problem

The researcher specifies a parameter $\theta \in \Theta$ and a moment condition $g(\cdot)$ that identifies this parameter. If she collected the economic concept $V_r$ on each text piece $r$, she would calculate

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

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

The researcher prompts an LLM to measure the economic concept on each text piece, $\widehat{V}_r := \widehat{m}(r; t)$, and plugs in the LLM's labels

$$\widehat{\theta} = \arg\min_{\theta\in\Theta} \frac{1}{N} \sum_{r\in\mathcal{R}} D_r g(\widehat{m}(r; t), W_r; \theta). \tag{6}$$

When can we draw conclusions about $\widehat{\theta}^*$ based on $\widehat{\theta}$?

We associate the researcher with a research context $Q(\cdot) \in \mathcal{Q}$ satisfying Assumption 1.

We 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 $r \in \mathcal{R}$, $\theta \in \Theta$, and values of the economic concept $v$.

As the number of economically relevant text pieces grows large (see Appendix C.1), we can characterize the behavior of both estimators. The target moment condition converges to, at any parameter value $\theta \in \Theta$,

$$\frac{1}{N} \sum_{r \in \mathcal{R}} D_r g(V_r, W_r; \theta) - \frac{1}{\mathbb{E}_Q[\sum_{r \in \mathcal{R}} D_r]} \mathbb{E}_Q \left[ \sum_{r \in \mathcal{R}} D_r g(V_r, W_r; \theta) \right] \xrightarrow{p} 0. \tag{7}$$

Conditional on the LLM's realized training dataset, the plug-in moment condition converges to, at any parameter value $\theta \in \Theta$,

$$\frac{1}{N} \sum_{r \in \mathcal{R}} D_r g(V_r, W_r; \theta) - \frac{1}{\mathbb{E}_Q[\sum_{r \in \mathcal{R}} D_r \mid T = t]} \mathbb{E}_Q \left[ \sum_{r \in \mathcal{R}} D_r g(\widehat{m}(r; t), W_r; \theta) \mid T = t \right] \xrightarrow{p} 0. \tag{8}$$

As in the prediction problem, conditioning on the training dataset simplifies analysis by treating the LLM as a fixed mapping.

Given an LLM with guarantee $\mathcal{M}$, the researcher would like to recover the moment condition defined using the economic concept.

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

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

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

The excitement around LLMs stems from their potential as "general-purpose technologies" – tools deployable across diverse applications without task-specific engineering (e.g., Eloundou et al., 2024). For estimation problems, this would mean reliably substituting for existing measurements across different economic concepts and research contexts. Definition 2 formalizes what this requires: the LLM must produce moment conditions matching the existing measurement procedure, regardless of which moment condition or population the researcher studies. If an LLM satisfies this property, researchers could confidently use its outputs for plug-in estimation knowing only the guarantee $\mathcal{M}$.

## 4.2 Measurement Error as a Threat to Estimation

We clarify what guarantee $\mathcal{M}$ is necessary and sufficient for an LLM to automate the existing measurement. We 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 $\widehat{m}(\cdot)$, $\mathbb{E}_Q[\sum_{r\in\mathcal{R}} D_r g(\widehat{m}(r), W_r; \theta) \mid T = t] - \mathbb{E}_Q[\sum_{r\in\mathcal{R}} D_r g(V_r, W_r; \theta)]$ equals*

$$\left(\mathbb{E}_Q\left[\sum_{r\in\mathcal{R}} D_r g(\widehat{m}(r), W_r; \theta)\right] - \mathbb{E}_Q\left[\sum_{r\in\mathcal{R}} D_r g(V_r, W_r; \theta)\right]\right) + \mathbb{E}_Q\left[\sum_{r\in\mathcal{R}} D_r \left(\frac{q_r^{T|D}(t_r)}{q_r^T(t_r)} - 1\right) g(\widehat{m}(r), W_r; \theta)\right]. \tag{9}$$

The second term captures training leakage – potential overlap between the LLM's training dataset and the researcher's evaluation sample. As discussed in Section 3.3, training leakage can be controlled through appropriate choice of model and research context. For example, if the researcher collects all text pieces in their research context or randomly samples text pieces, then training leakage is mechanically zero.

We therefore focus on the first term, which captures possible LLM errors. Intuitively, LLM errors $\Delta_r = \widehat{m}(r; t) - V_r$ can bias parameter estimates if they correlate with the economic variables $W_r$. A natural question arises: what if we know the LLM is "pretty good" – say, within some $\delta$ of the true measurement everywhere? This is what we might be tempted to conclude from impressive benchmark evaluations and demonstrations of LLM capabilities. More precisely, suppose the LLM satisfies the guarantee $\mathcal{M}(Q, \delta)$ — the collection of text generators satisfying $\|\widehat{m}(\cdot) - f^*(\cdot)\|_{\infty,Q} = \max_{r\in\mathcal{R}:\, q_r^D>0} |\widehat{m}(r; t) - f^*(r)| \leq \delta$. Does this ensure valid plug-in estimation?

Unfortunately, knowing the guarantee $\mathcal{M}(Q, \delta)$ does not tell us about the exact pattern of errors — whether they relate to economic variables in ways that bias our estimates. We 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 $v, \theta$. Let $\mathcal{R}(g, Q)$ denote the collection of sensitive text pieces.

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

$$\left|\mathbb{E}_Q\left[\sum_{r\in\mathcal{R}} D_r g(\widehat{m}(r), W_r; \theta) \mid T = t\right] - \mathbb{E}_Q\left[\sum_{r\in\mathcal{R}} D_r g(V_r, W_r; \theta)\right]\right| \leq \overline{G}\delta. \tag{10}$$

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

*for $\delta(r)$ defined in the proof,*

$$\left|\mathbb{E}_Q\left[\sum_{r\in\mathcal{R}} D_r g(\widehat{m}(r), W_r; \theta) \mid T = t\right] - \mathbb{E}_Q\left[\sum_{r\in\mathcal{R}} D_r g(V_r, W_r; \theta)\right]\right| \geq \underline{G}\left(\sum_{r\in\mathcal{R}(g,Q)} |\delta(r)| q_r^D\right). \tag{11}$$

Lemma 3(i) shows guarantee $\mathcal{M}(Q, \delta)$ bounds the LLM's error for the moment condition. But this is not enough for the LLM to automate the existing measurement. Lemma 3(ii) shows text generators satisfying the guarantee $\mathcal{M}(Q, \delta)$ can still produce meaningful estimation errors. We cannot rule out that the LLM's errors correlate with economic variables in ways that bias estimates. This makes measurement error pernicious.

This leads to our main characterization. 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 its labels reproduce the existing measurement process everywhere. Anything less cannot guarantee valid plug-in estimation across all possible applications.

**Proposition 2.** *Suppose the LLM $\widehat{m}(\cdot; t)$ satisfies 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 $Q(\cdot)$.*

### 4.2.1 Some Evidence of Measurement Error

Plugging in LLM outputs requires that the model reproduces the target measurement process $f^*(\cdot)$. While their impressive performance on some tasks makes this assumption appealing, Section 2.3 showed such intuitions about LLMs are unreliable. No LLM achieves perfect accuracy on benchmark evaluations, and growing evidence documents their surprising errors. Does this matter for economics research? We next show that LLM errors substantially affect downstream parameter estimates in two empirical examples.

**Linear Regression with LLMs** Consider a researcher relating the economic concept $V_r$ and linked variables $W_r$ through linear regression. The researcher may use the LLM's labels as the dependent variable:

$$V_r = W_r'\beta^* + \epsilon_r, \text{ and } \widehat{m}(r; t) = W_r'\beta + \widetilde{\epsilon}_r. \tag{12}$$

or the independent variable:

$$W_r = V_r'\alpha^* + \nu_r, \text{ and } W_r = \widehat{m}(r; t)'\alpha + \widetilde{\nu}_r. \tag{13}$$

The bias depends on how the LLM's error $\Delta_r = \widehat{m}(r;t) - V_r$ varies across text pieces. These are well-known results, dating back to Bound et al. (1994).

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

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

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

When the economic concept is the dependent variable, the bias equals the best linear predictor of the LLM's errors given the covariates. When the economic concept is the independent variable, the bias has a more complex form involving attenuation and correlation between the error and the residual. Knowing the LLM is "accurate" provides no guarantee about these biases. What matters is whether errors correlate with economic variables in the specific regression.

**Assessing Measurement Error in Financial News Headlines** We return to the financial news headlines dataset, focusing on headlines published in 2019 for 6,000 publicly traded stocks. We observe each headline's text $r$, publication date and stock ticker. We merge realized returns at various horizons (1, 5, and 10 days) after publication and lagged returns before publication (Beta Suite by WRDS, 2024).

Financial news headlines express various economic concepts that we could measure and relate to realized stock returns. We focus on one: is the headline positive, negative or neutral news for the company? While we could read every single financial news headline published in 2019, this process would be painstaking. It is natural to use LLMs to solve this text processing problem.

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, $\widehat{V}_r := \widehat{m}(r;t)$. This requires making several practical choices: what specific LLM to use? What prompt engineering strategy? If alternative choices lead to different labels and downstream estimates, this would indicate non-ignorable errors in the LLM outputs.

We prompt GPT-3.5-Turbo, GPT-4o, GPT-4o-mini, GPT-5-mini, and GPT-5-nano to label each financial news headline $r$, using nine alternative prompts to each model. Our two base prompts provide the LLM with the text of the headline $r$ and ask it to label whether this news is positive, negative, or neutral for the company; we also ask the LLM to provide

its confidence and a magnitude in the label. Our two base prompts differ in how they ask the model to format its reply: filling-in-the-blanks text or a structured JavaScript Object Notation (JSON) object. The prompts are provided in Appendix E.

We further vary these base prompts in two ways, motivated by popular prompt engineering strategies (Liu et al., 2023; Wei et al., 2024; White et al., 2023; Chen et al., 2024). First, we request the LLM adopt one of four different "personas," like "knowledgeable economic agent" or "expert in finance." Second, we append three prompt modifiers that ask the LLM to "think carefully" or "think step-by-step" and provide an explanation for its answer. For each financial news headline $r$, we obtain labels $\widehat{V}_r^{m,p}$ associated with a collection of different LLMs $m$ and prompting strategies $p$.

For each model $m$ and prompt $p$, we regress the realized returns of each stock within 1-day, 5-days and 10-days after the headline's publication date $Y_r$ on the LLM's labels $\widehat{V}_r^{m,p}$, controlling for the model's reported magnitudes and lagged realized returns. We report 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 their associated t-statistics with standard errors clustered at the date and company level. Figure 1 and Table 1 summarize the results. Simply changing the prompt or model yields markedly different estimates: many prompt-model combinations that produce different directions and magnitudes of the relationship between the positive/negative label and realized returns.[8]

**Assessing Measurement Error in Congressional Legislation** We next return to the Congressional legislation data. For each bill, we observe the text of its description $r$ as well as a collection of 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 researcher might reasonably study how these variables shape the topic of each bill, $V_r$. Could we use an LLM to collect those labels?

We randomly select $10,000$ Congressional bills and separately prompt GPT-3.5-Turbo, GPT-4o, GPT-5-mini, and GPT-5-nano 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 for the specific prompts we used).

We regress the labeled economic concept $\widehat{V}_r^{m,p}$ — in this case, the policy topic of the bill — against linked covariates $W_r$, separately reporting the coefficients $\widehat{\beta}_{m,p}$ and their associated t-statistics. Figure 2 and Table 2 summarize the variation in the resulting estimates across models and prompts. For each combination of labeled policy topic and linked covariate

[8] Appendix Figure A3, shows substantial variation in the pairwise agreement in the labels produced. There appear to be no consistent patterns in which pairs of prompting strategies tend to have the most agreement.

in the Congressional bills dataset, we see substantial variability across different LLMs and prompting strategies.[9] Once again, simply changing the prompt or model yields remarkably different downstream estimates.

## 4.3 Practical Guidance with Validation Data

Given the evidence on LLM brittleness in Section 2.3, it is difficult to defend the assumption of no measurement error in LLM outputs for estimation problems. The solution is to collect measurements $V_r = f^*(r)$ on a small validation sample and use them to de-bias the plug-in estimate based on the LLM labels $\widehat{V}_r$. The virtues of validation data have been understood for decades (Bound and Krueger, 1991; Bound et al., 1994; Lee and Sepanski, 1995). It has been recently revived in machine learning to handle the types of mismeasured covariates and outcomes produced by modern ML models; see for example Wang, McCormick and Leek (2020), Angelopoulos et al. (2023), Egami et al. (2024) and Carlson and Dell (2025) among others.

We illustrate the value of validation data using the familiar case of linear regression. We review the mechanics of debiasing when the LLM label appears as the dependent variable, provide intuition about when it will work well using asymptotic arguments, and demonstrate finite sample performance using Congressional legislation. Our discussion is pedagogical; we refer the readers to the works above for more general settings.

**An Illustration with Linear Regression** 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 plug-in regression $\widehat{V}_r = W_r'\beta + \widetilde{\epsilon}_r$. Appendix C.2 considers when the economic concept $V_r$ is a covariate.

Proposition 3 established that the bias of the plug-in regression coefficient $\beta$ depends on how the LLM's error $\Delta_r = \widehat{m}(r;t) - V_r$ covaries with $W_r$. This suggests a natural solution: on a random subset of the researcher's dataset, collect $V_r$ using the existing measurement $f^*(\cdot)$. For example, the researcher reads and labels a subset of the text pieces themselves. We call text pieces on which the researcher observes $(r, W_r, \widehat{m}(r;t), V_r)$ the validation sample, and we refer to the remaining text pieces on which she observes $(r, W_r, \widehat{m}(r;t))$ as the primary sample.

In the validation sample, the researcher can estimate the bias $\widehat{\lambda}_{\Delta|W}$ by forming $\Delta_r = \widehat{m}(r;t) - V_r$ on the validation sample and regressing it on $W_r$. The validation sample can therefore be used for two purposes. First, the validation sample provides a target to optimize when selecting a model and prompt. For any choice of LLM $m$ and prompt $p$, the researcher

[9] Appendix Figure A4 calculates the pairwise agreement in the labels produced by alternative prompting strategies $p$. We again find substantial variation in the pairwise agreement in the labels produced.

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, and more importantly, the validation sample enables bias correction. Rather than reporting the plug-in coefficient $\widehat{\beta}$, the researcher reports

$$\widehat{\beta}^{debiased} = \widehat{\beta} - \widehat{\lambda}_{\Delta|W}. \tag{14}$$

This is exceedingly simple to implement: run two regressions – regress $\widehat{m}(r;t)$ on $W_r$ in the primary sample and regress $\Delta_r$ on $W_r$ in the validation sample – and subtract. Inference is equally straightforward: for example, researchers may bootstrap the primary and validation samples.

As discussed in Appendix C.2, the bias-corrected estimator has desirable theoretical properties. Consider a research context where the researcher randomly samples text pieces, allocating a fraction $\rho_p$ to the primary sample and a fraction $\rho_v$ to 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 $\beta^*$. $\widehat{\beta}^{debiased}$ is asymptotically normal with limiting variance given by

$$\sigma_W^{-4}\left(\frac{1-\rho_p}{\rho_p}\sigma^2_{\widehat{V}W} + 2\sigma_{\widehat{V}W}\sigma_{\Delta W} + \frac{1-\rho_v}{\rho_v}\sigma^2_{\Delta W}\right), \tag{15}$$

where $\sigma_W$ 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. 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 $\Delta_r$ across text pieces.

A natural question arises: if we collect validation data anyway, why bother with the possibly mismeasured LLM labels on the primary sample? The validation-only estimator $\widehat{\beta}^*$ is also consistent and asymptotically normal: its limiting variance is given by $\sigma_W^{-4}\left(\frac{1-\rho_v}{\rho_v}\sigma^2_{VW}\right)$. Comparing these expressions reveals that the bias-corrected estimator is more precise when:

$$\frac{1-\rho_p}{\rho_p}\sigma^2_{\widehat{V}W} + 2\sigma_{\widehat{V}W}\sigma_{\Delta W} \leq \frac{1-\rho_v}{\rho_v}\left(\sigma^2_{VW} - \sigma^2_{\Delta W}\right). \tag{16}$$

This comparison depends on the relative standard deviation of the existing measurement $V_r$ and the LLM's error $\Delta_r$. Equation (16) 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. Correctly incorporating imperfect LLM outputs can

result in tighter standard errors than ignoring the LLM altogether. This phenomenon has been documented in recent machine learning research, such as Angelopoulos et al. (2023) and associated work, that combines validation data with the outputs of machine learning models to estimate downstream parameters.

In other words, a free-lunch is possible: use the LLM to solve the text processing problem while delivering precise estimates of the target regression coefficient. Importantly, LLM outputs are not substitutes for existing measurements; instead, they amplify a small validation sample.

Finally, validation data provides another critical benefit: it addresses concerns about specification searching that arise from LLM brittleness. As seen, alternative choices of prompts and models can yield dramatically different downstream estimates, creating a severe $p$-hacking risk. With countless possible prompts and multiple competing LLMs, researchers could search across specifications until finding desired results. Provided the researcher collects validation data and debiases whatever LLM output they collect, alternative choices of prompting strategy and LLM target the same empirical quantity: the parameter $\theta$ defined using the researcher's existing measurement.

**Monte Carlo Simulations based on Congressional Legislation** The Congressional legislation data is well-suited to illustrate the value of validation data. The Congressional Bills Project trained teams of human annotators to label the description of each Congressional bill $r$ for its major policy topic area $V_r := f^*(r)$, describing whether the bill falls into one of twenty possible policy areas.[10] Given its widespread use, researchers are comfortable using these measurements in downstream analyses. Can an LLM automate this measurement procedure $f^*(\cdot)$?

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. On this random sample of $5,000$ bills, we calculate the plug-in coefficient $\widehat{\beta}$ by regressing $\widehat{V}_r^{m,p}$ on the linked variable $W_r$. We next randomly reveal the existing label $V_r$ on 250 (i.e. 5%) of our random sample of $5,000$ bills, which produces a validation sample. We then calculate the bias-corrected coefficient $\widehat{\beta}^{debiased}$. We repeat these steps for $1,000$ randomly sampled datasets. Across simulations, we calculate the average bias of the plug-in coefficient and the bias-corrected coefficient for the target regression $\beta^*$ associated with regressing the existing label $V_r$ on the linked variable $W_r$ on all $10,000$ bills. We repeat this exercise for each combination of bill topic $V_r$, linked covariate $W_r$, LLM $m$ (either GPT-3.5-Turbo, GPT-4o, GPT-5-mini and GPT-5-nano) and

[10]The Congressional Bills Project states that all annotators were trained for a full academic quarter before beginning this task (Jones et al., 2023).

prompting strategy $p$. Appendix D.1 varies the size of the validation sample, finding similar results even when the validation sample contains as few as 125 bills.

Figure 3 and the top panel of Table 3 compares the average bias of the plug-in coefficient $\widehat{\beta}$ and the bias corrected coefficient $\widehat{\beta}^{debiased}$ for the target regression $\beta^*$ (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. Using the validation sample yields estimates that are reliably unbiased — indeed, the bias-corrected regression coefficient performs remarkably well across all regression specifications and pairs of LLM and prompting strategy.

For each regression specification and pair of LLM and prompting strategy, Table 3 summarizes 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 $\beta^*$. The nominal 95% confidence interval for the bias-corrected regression coefficient has approximately correct coverage across all regression specifications, LLMs and prompting strategies. By contrast, plug-in estimation often suffers from severe coverage distortions.

Finally, we can use these data to answer the question: If we already collected measurements $V_r$ in a validation sample, what is the value of the LLM labels? For each regression specification, LLM and prompting strategy, we compare the average mean square error of the bias-corrected coefficient and the validation-sample only coefficient for the target regression $\beta^*$. Figure 4 plots the resulting distribution across all choices of bill topic $V_r$, covariate $W_r$, and pair of model-and-prompting strategy. The average mean square error of the validation-sample-only coefficient is always higher (less precisely estimated) compared to the bias-corrected coefficient. Substantial precision improvements from using LLM outputs are possible in finite samples.

Taken together, these results indicate that estimation is a promising use case for LLMs *if* the researcher collects a validation sample to correct for LLM errors. LLMs can then lower the cost of data collection and improve statistical precision, while preserving the familiar econometric guarantees we desire.

## 4.4 Estimation without Validation Data

When using LLMs in estimation problems, validation data enables correct inference by allowing the researcher to estimate and correct for LLM errors. But what if such data cannot be collected? Before considering possible paths forward, the researcher must make a judgment call: does there exist – even in principle – a measurement procedure $f^*(\cdot)$ that would produce labels $V_r$ the researcher would trust? This is not a statistical question, but a conceptual one

about the nature of the economic concept being studied.

**When Ground Truth Exists** Suppose the researcher believes there exists a "true" value of the economic concept for each text piece, even if measuring it is costly or time-consuming. Consider labeling Congressional bills' policy topics. The researcher might be confident that if they (or trained experts) carefully read each bill, they could reliably classify policy topics. The researcher may still not collect validation data despite believing ground truth exists – perhaps due to budget constraints, time pressure, or confidence in LLM accuracy. In this case, researchers might pursue one of two paths that might appear defensible but ultimately are not.

First, the researcher could argue there are no errors in the LLM's outputs $\widehat{m}(r;t)$, justifying plug-in estimation. This argument is difficult to defend at present. A researcher proceeding this way must somehow argue why an LLM's output exactly reproduces the economic concept, even though it is imperfect on benchmarks and why evidence on LLM brittleness does not apply.

Second, acknowledging that LLM outputs are imperfect, the researcher might write down a statistical model of its errors $\Delta_r = \widehat{m}(r;t) - f^*(r)$, just as we would in the measurement error literature. Consider a stylized model: for a given LLM $m$, the errors errors $\Delta_r^{m,p} = V_r - \widehat{V}_r^{m,p}$ across prompting strategies $p$ are independent. Such an assumption would suggest particular solutions: perhaps the researcher could use one prompting strategy as an instrument for another. For a given LLM $m$, its labels across prompting strategies $p$ are surely correlated with one another. Across language models $m$, there is surely substantial overlap in training datasets. Consequently, the labels $\widehat{V}_r^{m,p}$ are correlated (Kim et al., 2025). Taking a step back, our usual measurement error frameworks were not designed for this setting – where the measurement comes from an algorithm whose behavior we do not fully understand, applied to a quantity we do not observe.

Given these challenges, the practical answer is clear: invest effort and collect a small validation sample.

**When Ground Truth is Undefined** Suppose the economic concept is sufficiently abstract or subjective that the researcher is uncomfortable articulating what "ground truth" would even mean. In this case, the researcher could define the object of study as the LLM's outputs themselves. The concept simply is, for example, whatever GPT-4o produces with a given prompt. This makes plug-in estimation valid by definition.

But the researcher is now in uncharted territory. This seemingly minor shift has important implications. When a new prompt engineering strategy is produced, should we publish a new paper? When OpenAI inevitably releases GPT-6, should we revisit all published work that used GPT-5? In Section 4.2.1, we found that variation across models and prompts can

be enormous—different prompts and models can even yield estimates with different signs. If 60% of model-prompt combinations suggest a positive relationship and 40% suggest negative, what have we learned?

More fundamentally, defining the LLM's outputs as the object of study sidesteps the hard – and economically interesting – question. We care about the underlying economic concept, not the LLM's outputs. When facing such conceptual ambiguity, the scientific response is to acknowledge it and probe from multiple angles – not to privilege one operationalization of the concept. This is especially true with LLMs, since we have found them to be brittle and unreliable on even well-defined tasks as discussed in Section 2.3.

What might a more systematic investigation look like? As an example, suppose we compared an LLM's outputs with measurements produced by the researcher or domain experts. How do they differ? Where do they agree and disagree? This work would help us articulate what exactly *is* the economic concept we are studying. By contrast, treating the LLM as ground truth bypasses this essential work: grappling with what the economic concept means and how best to measure it.

Altogether the researcher must ask themselves: Is it truly the case that no measurement procedure exists–even in principle–that they would trust? Or does one exist, but they are reluctant to invest in collecting even a small validation sample? These are fundamentally different situations. The former presents a genuine conceptual challenge deserving serious attention. The latter substitutes convenience for scientific rigor.

# 5 Novel Uses of Large Language Models

In addition to familiar prediction and estimation problems, LLMs enable exciting novel applications – simulating human subject responses or generating research hypotheses – that expand what we consider possible in empirical work. We offer one interpretation by mapping these applications into our framework. More broadly, these creative uses raise important questions about inferential goals, and we encourage further work to think rigorously about what researchers are ultimately trying to accomplish with such applications.

## 5.1 Human Subject Simulation

A growing body of research uses LLMs to simulate human subjects as "in-silico" subjects across economics (e.g., Horton, 2023; Manning, Zhu and Horton, 2024; Mei et al., 2024), marketing (e.g., Brand, Israeli and Ngwe, 2023), finance (e.g., Bybee, 2024), political science (e.g., Argyle et al., 2023), and computer science (e.g., Aher, Arriaga and Kalai, 2023).

This maps into our framework by reinterpreting text pieces $r \in \mathcal{R}$ and the existing measurement $f^*(\cdot)$. Each text piece $r$ represents an experimental design or survey instrument,

and $V_r$ might represent the average or model human response. If the researcher collected human responses $V_r$, she would calculate downstream parameter estimates (Equation 5). But collecting human responses is costly, so instead the researcher substitutes LLM responses $\widehat{m}(r;t) = \widehat{V}_r$ and reports the plug-in parameter estimate (Equation 6).

**Example: Testing Anomalies** To test violations of risky choice models, behavioral economists construct "anomalies" – lottery menus that highlighting flaws in the economic model, such as the Allais Paradox (Allais, 1953) or the Kahneman-Tversky choice experiments (Kahneman and Tversky, 1979). Could LLMs simulate human choices $\widehat{m}(r;t)$ on new anomalies? ▲

**Example: Large-scale Choice Experiments** To compare risky choice models, recent work measures predictive accuracy across diverse lottery problems (Erev et al., 2017; Fudenberg et al., 2022). Peterson et al. (2021) recruited nearly 15,000 MTurk respondents to make over one million choices $V_r$ from lottery menus $r$, producing the "Choices13K" dataset. Could LLMs simulate the Choices13K dataset? ▲

Viewing human subject simulation as an estimation problem implies in-silico subjects must exhibit no measurement error (Proposition 2) — the LLM must reproduce human subject behavior on the researcher's experiment or survey. While some studies show LLMs reproduce published experiments, counterexamples abound: LLM responses to psychology experiments appear to produce more falsely significant findings than human subjects (Cui, Li and Zhou, 2024), cannot accurately reproduce the responses of human subjects on opinion polls (Santurkar et al., 2023; Boelaert et al., 2024), and can be sensitive to prompt engineering on economic reasoning tasks (Raman et al., 2024). Moreover, estimation problems require no training leakage. Since published experiments likely enter LLM training corpora, the key question is whether LLMs can simulate behavior on entirely new experimental designs, not merely reproduce memorized results (Manning and Horton, 2025).

Viewing human subject simulation as an estimation problem also implies a practical fix: collect responses from at least some real human subjects. Consequently, in-silico subjects serve to amplify, rather than fully replace, human subjects. We refer the reader to recent work, such as Broska, Howes and van Loon (2025); Krsteski et al. (2025); Zhang et al. (2025), which provide further guidance on debiasing in-silico subjects.

## 5.2 Hypothesis Generation

Recent work uses LLMs to generate hypotheses across diverse applications: predicting user engagement from headlines (Batista and Ross, 2024; Zhou et al., 2024), suggesting instrumental variables (Han, 2024), proposing research ideas in natural language processing (Si,

Yang and Hashimoto, 2024), and generating interpretable hypotheses that summarize estimated relationships between variables and text (Modarressi, Spiess and Venugopal, 2025; Movva et al., 2025). We offer one interpretation viewing this as a type of prediction problem.

Researchers provide prompts $r$ containing one or more text pieces (e.g., a collection of headlines, a description of an empirical setting) and ask the LLM to generate a hypothesis $\widehat{m}(r;t) = \widehat{Y}_r$ that summarizes features or patterns in those texts (e.g., what drives engagement, what would be a valid instrument). The researcher evaluates average hypothesis quality $\frac{1}{N}\sum_{r\in\mathcal{R}} D_r \ell(\widehat{m}(r;t))$ – though this scoring rule $\ell(\cdot)$ may be implicit or informal – to determine whether the LLM is useful for hypothesis generation across different texts in the research context.

To assess whether the LLM is a useful tool for hypothesis generation in research context $Q(\cdot)$ – that is, whether the quality of its hypotheses generalizes across different texts in that context – we need no training leakage (Proposition 1). Has the LLM seen this prompt or setting before? If so, it may reproduce memorized hypotheses rather than demonstrate new capability to generate insights on novel texts. For example, an LLM trained on text describing distance-to-college as an instrument for education returns might reproduce this strategy, leading us to overestimate its ability to identify instruments in genuinely novel settings.

Hypothesis generation is an exciting frontier. Within economics, discussion about how machine learning and artificial intelligence are tools for hypothesis generation and scientific discovery can be found in Fudenberg and Liang (2019), Ludwig and Mullainathan (2024), Agrawal, McHale and Oettl (2024), Mullainathan and Rambachan (2024), and Mullainathan and Rambachan (2025). Further work is needed to clarify our inferential goals are in these settings – what constitutes a "good" hypothesis, and how we should evaluate LLM performance in generating them.

# 6 A Checklist for Empirical Research

To help use our framework in practice, we briefly summarize its key guidance for researchers incorporating LLM outputs in empirical work.

**Identify Your Problem: Prediction or Estimation?** The first step is to identify your problem: are you facing a prediction problem or an estimation?

In a prediction problem (Section 3), the researcher uses text pieces $r$ to predict some linked outcome $Y_r$ and evaluates the LLM's sample average loss $(1/N)\sum_{r\in R} D_r \ell(Y_r, \hat{m}(r;t))$. The researcher would like to understand whether this reflects the model's predictive performance.

In an estimation problem (Section 4), the researcher measures an economic concept

expressed in text pieces $r$ to estimate downstream parameters. There is a measurement procedure $f^*(\cdot)$ that could be applied (e.g., the researcher reading each text piece themselves) that would produce the concept $V_r := f^*(r)$ on all text pieces and the researcher would, for example, run a regression of $V_r$ on some linked covariates $W_r$. Since this is costly at scale, the researcher uses LLM outputs $\hat{V}_r := \hat{m}(r; t)$ as a substitute.

**For Prediction Problems, Ensure No Training Leakage** Valid conclusions in prediction problems require one condition: no training leakage between the LLM's training data and the researcher's dataset (Section 3.2).

Enforcing no training leakage is jointly determined by the prediction question and the model choice. Researchers must explicit about three key elements: (1) their target population – what collection of text pieces do they ultimately want to make predictions on? (2) their sampling procedure—how do they select evaluation samples from this population?; and (3) how does this sampling procedure relate to plausible training corpora used by the LLM? In Section 3.3, we illustrated how researchers can answer these questions in multiple common empirical settings, such as predicting on future documents, predicting on confidential documents, or constructing a random evaluation sample from a known corpus.

When in doubt, the safest approach combines open-source models with published weights or documented training data and evaluation samples constructed to mechanically exclude training data. Researchers should avoid relying on closed models like the GPT from OpenAI or the Claude from Anthropic families, since their training data is undisclosed and potentially continuously updated. Finally, prompt engineering strategies (e.g., "ignore information after date $\tau$") are not reliable solutions to training leakage, as our evidence demonstrates.

**For Estimation Problems, Collect Validation Data** In estimation problems, plugging in LLM outputs requires the strong assumption that the LLM reproduces the existing measurement procedure—an assumption that is difficult to defend given evidence on LLM brittleness in computer science (see Section 2.3 and Section 4.2).

The solution is straightforward: on a random sample of text pieces, apply the existing measurement procedure $f^*(\cdot)$ to collect some labels $V_r$. This validation sample allows the researcher to debias their downstream estimate for possible LLM errors. Section 4.3 provides a pedagogical treatment for linear regression when the LLM label appears as the dependent variable. We found that this approach delivers approximately unbiased estimates and confidence intervals with good coverage in finite samples. Moreover, the debiased estimator can be more precisely estimated and yield tighter standard errors than using the validation sample alone. In this sense, LLM outputs amplify rather than replace existing measurements.

In our simulations, validation samples with as few as 125-250 text pieces provided substantial benefits in terms of bias and coverage. The optimal size of the validation sample

depends on the trade-off between the cost of collecting measurements and the potential precision gains from expanding the validation sample. This should be approached as a design choice similar to determining sample size in surveys and experiments.

# 7 Conclusion

Machine learning and artificial intelligence expand 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 and digital traces, to infer outcomes at high-frequencies and granular scales (e.g., Donaldson and Storeygard, 2016; Blumenstock, Cadamuro and On, 2015; Rambachan, Singh and Viviano, 2024). We tackle prediction policy problems (Kleinberg et al., 2015, 2018; Mullainathan, 2025) and develop algorithms for hypothesis generation (Fudenberg and Liang, 2019; Ludwig and Mullainathan, 2024; Mullainathan and Rambachan, 2024). LLMs are the latest tools to enter empirical work.

By radically reducing the cost of analyzing vast text corpora, LLMs enable economists to tackle questions previously impossible due to scale or expense. Using large language models, researchers can predict market reactions from earnings calls, measure sentiment across historical newspapers, track partisan polarization in social media, and simulate human responses at minimal cost. Yet these are complex algorithms that seemingly resist traditional econometric analysis. Our framework shows how to harness LLMs despite their complexity. The straightforward practices we recommend unlock LLMs' transformative potential for empirical research.

# References


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

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

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

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

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

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

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

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

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

**Ash, Elliott, Stephen Hansen, and Yony Muvdi.** 2024. "Large Language Models in Economics." Centre for Economic Policy Research CEPR Discussion Paper 19479, London.

**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*.

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

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

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

**Blumenstock, Joshua, Gabriel Cadamuro, and Robert On.** 2015. "Predicting poverty and wealth from mobile phone metadata." *Science*, 350(6264): 1073–1076.

**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`.

**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, 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.*

**Broska, David, Michael Howes, and Austin van Loon.** 2025. "The Mixed Subjects Design: Treating Large Language Models as Potentially Informative Observations."

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

**Carlson, Jacob, and Melissa Dell.** 2025. "A Unifying Framework for Robust and Efficient Inference with Unstructured Data."

**Chang, Yupeng, Xu Wang, Jindong Wang, Yuan Wu, Linyi Yang, Kaijie Zhu, Hao Chen, Xiaoyuan Yi, Cunxiang Wang, Yidong Wang, Wei Ye, Yue Zhang, Yi Chang, Philip S. Yu, Qiang Yang, and Xing Xie.** 2024. "A Survey on Evaluation of Large Language Models." *ACM Trans. Intell. Syst. Technol.*, 15(3).

**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.

**Cobbe, Karl, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman.** 2021. "Training Verifiers to Solve Math Word Problems."

**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.*

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

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

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

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

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

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

**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.

**Eloundou, Tyna, Sam Manning, Pamela Mishkin, and Daniel Rock.** 2024. "GPTs are GPTs: Labor market impact potential of LLMs." *Science*, 384(6702): 1306–1308.

**Engelberg, Joseph, Asaf Manela, William Mullins, and Luka Vulicevic.** 2025. "Entity Neutering."

**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.

**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.

**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, 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*.

**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*.

**He, Songrun, Linying Lv, Asaf Manela, and Jimmy Wu.** 2025. "Chronologically Consistent Large Language Models."

**Horton, John J.** 2023. "Large language models as simulated economic agents: What can we learn from homo silicus?" National Bureau of Economic Research.

**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.

**Jimenez, Carlos E., John Yang, Alexander Wettig, Shunyu Yao, Kexin Pei, Ofir Press, and Karthik Narasimhan.** 2024. "SWE-bench: Can Language Models Resolve Real-World GitHub Issues?"

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

**jylin04, JackS, Adam Karvonen, and Can.** 2024. "OthelloGPT Learned a Bag of Heuristics." *LessWrong*.

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

**Kim, Elliot, Avi Garg, Kenny Peng, and Nikhil Garg.** 2025. "Correlated Errors in Large Language Models."

**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.

**Krsteski, Stefan, Giuseppe Russo, Serina Chang, Robert West, and Kristina Gligorić.** 2025. "Valid Survey Simulations with Limited Human Data: The Roles of Prompting, Fine-Tuning, and Rectification."

**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.

**Li, Kenneth, Aspen K. Hopkins, David Bau, Fernanda Viégas, Hanspeter Pfister, and Martin Wattenberg.** 2024. "Emergent World Representations: Exploring a Sequence Model Trained on a Synthetic Task."

**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*.

**Lopez-Lira, Alejandro, Yuehua Tang, and Mingyin Zhu.** 2025. "The Memorization Problem: Can We Trust LLMs' Economic Forecasts?"

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

**Mancoridis, Marina, Bec Weeks, Keyon Vafa, and Sendhil Mullainathan.** 2025. "Potemkin Understanding in Large Language Models." *arXiv preprint arXiv:2506.21521*.

**Manning, Benjamin S., and John J. Horton.** 2025. "General Social Agents."

**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.

**Minaee, Shervin, Tomas Mikolov, Narjes Nikzad, Meysam Chenaghlu, Richard Socher, Xavier Amatriain, and Jianfeng Gao.** 2025. "Large Language Models: A Survey."

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

**Modarressi, Iman, Jann Spiess, and Amar Venugopal.** 2025. "Causal Inference on Outcomes Learned from Text."

**Movva, Rajiv, Kenny Peng, Nikhil Garg, Jon Kleinberg, and Emma Pierson.** 2025. "Sparse Autoencoders for Hypothesis Generation."

**Mullainathan, Sendhil.** 2025. "Economics in the Age of Algorithms." *AEA Papers and Proceedings*, 115: 1–23.

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

**Mullainathan, Sendhil, and Ashesh Rambachan.** 2025. "Science in the Age of Algorithms." In *The Economics of Transformative AI*. Chapter 13. University of Chicago Press.

**Nanda, Neel, Andrew Lee, and Martin Wattenberg.** 2023. "Emergent Linear Representations in World Models of Self-Supervised Sequence Models."

**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*.

**Nikankin, Yaniv, Anja Reusch, Aaron Mueller, and Yonatan Belinkov.** 2025. "Arithmetic Without Algorithms: Language Models Solve Math With a Bag of Heuristics."

**Park, Joon Sung, Carolyn Q. Zou, Aaron Shaw, Benjamin Mako Hill, Carrie Cai, Meredith Ringel Morris, Robb Willer, Percy Liang, and Michael S. Bernstein.** 2024. "Generative Agent Simulations of 1,000 People."

**Park, Joon Sung, Joseph C. O'Brien, Carrie J. Cai, Meredith Ringel Morris, Percy Liang, and Michael S. Bernstein.** 2023. "Generative Agents: Interactive Simulacra of Human Behavior."

**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*.

**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*.

**Vafa, Keyon, Justin Y. Chen, Jon Kleinberg, Sendhil Mullainathan, and Ashesh Rambachan.** 2024. "Evaluating the World Model Implicit in a Generative Model." *Harvard Working Paper*.

**Vafa, Keyon, Peter G Chang, Ashesh Rambachan, and Sendhil Mullainathan.** 2025. "What has a foundation model found? using inductive bias to probe for world models." *arXiv preprint arXiv:2507.06952*.

**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*.

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

**Wongchamcharoen, Pattaraphon Kenny, and Paul Glasserman.** 2025. "Do Large Language Models (LLMs) Understand Chronology?"

**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.

**Zhang, Bohan, Jiaxuan Li, Ali Hortaçsu, Xiaoyang Ye, Victor Chernozhukov, Angelo Ni, and Edward Huang.** 2025. "Agentic Economic Modeling."

**Zhao, Wayne Xin, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, Yifan Du, Chen Yang, Yushuo Chen, Zhipeng Chen, Jinhao Jiang, Ruiyang Ren, Yifan Li, Xinyu Tang, Zikang Liu, Peiyu Liu, Jian-Yun Nie, and Ji-Rong Wen.** 2025. "A Survey of Large Language Models."

**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.

# Main Figures and Tables

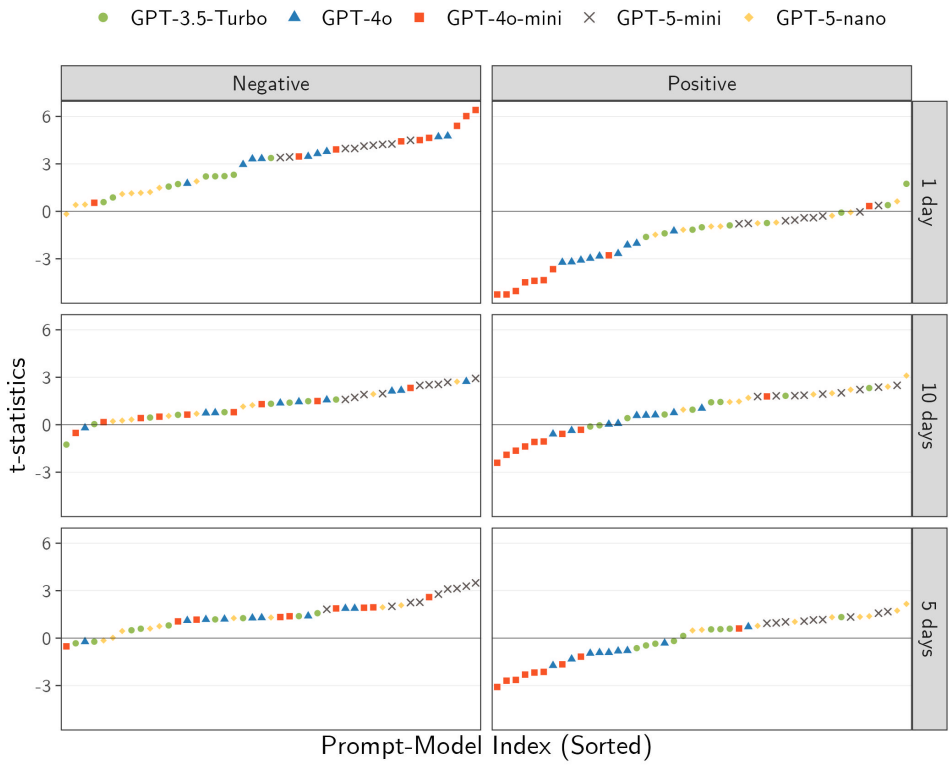


**Figure 1:** Variation in t-statistics for realized returns across large language models and prompting strategies on financial news headlines.

*Notes*: On financial news headlines from 2019, we prompt GPT-3.5-Turbo, GPT-4o-mini, GPT-4o, GPT-5-mini, and GPT-5-nano 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 4.2.1 for discussion.

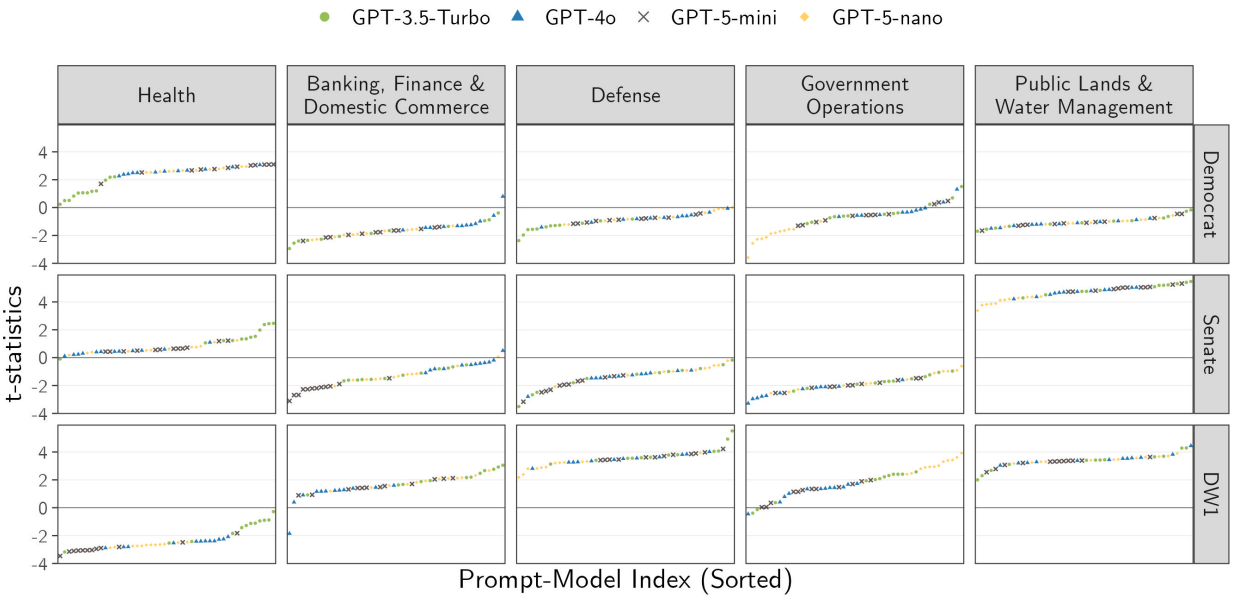


**Figure 2:** Variation in t-statistics across large language models and prompting strategies on congressional legislation.

*Notes*: On 10,000 Congressional bills, we prompt GPT-3.5-Turbo, GPT-4o, GPT-5-mini, and GPT-5-nano 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 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-statistics were sorted in ascending order for clarity. See Section 4.2.1 for discussion.

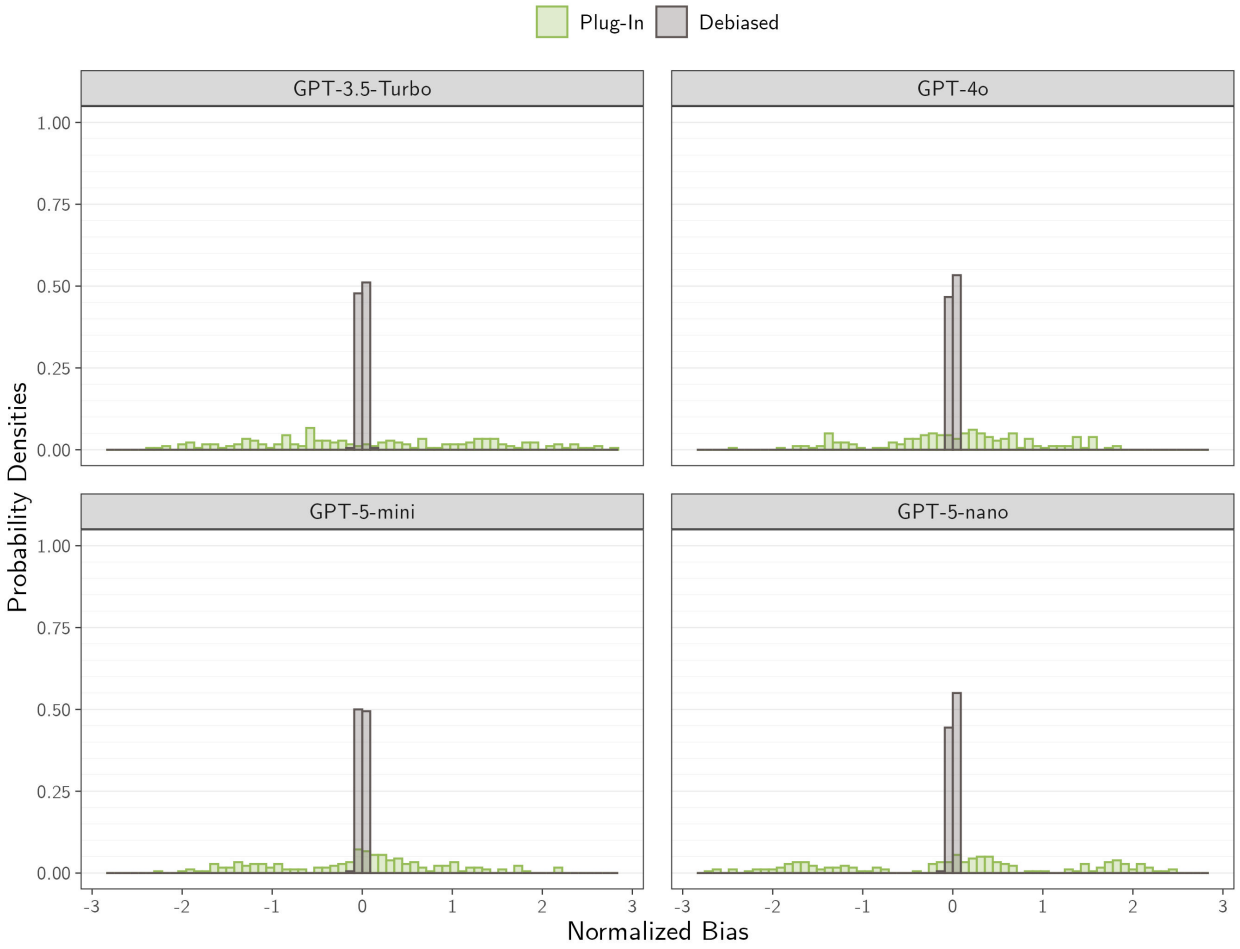


**Figure 3:** Normalized bias of the plug-in regression and bias-corrected regression across Monte Carlo simulations based on congressional legislation.

*Notes*: The normalized bias reports the average bias of the plug-in regression coefficient $\widehat{\beta}$ and the debiased coefficient $\widehat{\beta}^{debiased}$ for the target regression coefficient divided by their respective standard deviations across simulations. We summarize the distribution of normalized bias and coverage across regression specifications, choice of large language model and prompting strategies. For each combination of model topic $V_r$, linked covariate $W_r$, large language model $m$ and prompting strategy $p$, we randomly sample $5,000$ Congressional bills and calculate the plug-in regression coefficient $\widehat{\beta}$ and the bias-corrected regression coefficient $\widehat{\beta}^{debiased}$ based on a 5% validation sample. See Section 4.3 for discussion.

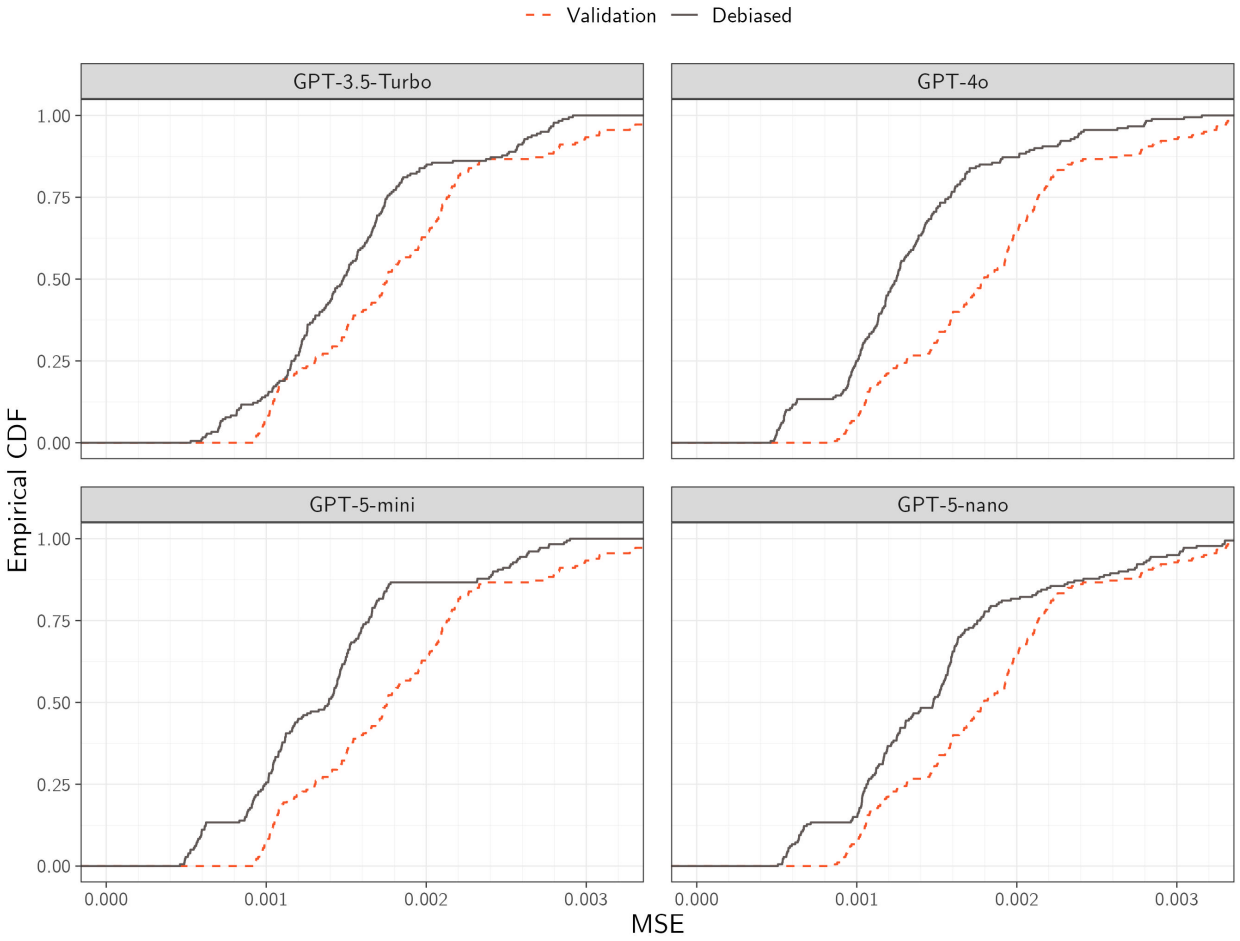


**Figure 4:** Cumulative distribution function of mean square error for the bias-corrected estimator against validation-sample only estimator.

*Notes*: For each combination of model topic $V_r$, covariate $W_r$, large language model $m$ and prompting strategy $p$, we randomly sample $5,000$ Congressional bills and calculate the bias-corrected regression coefficient using a 5% validation sample and the validation-sample only regression coefficient. We calculate the mean square error of $\widehat{\beta}^{debiased}$ and $\widehat{\beta}^{*}$ for the target regression, and we average the results over $1,000$ simulations. We summarize the distribution of average mean square error across regression specifications, choice of large language model and prompting strategies. See Section 4.3 for discussion.

| | *Return Horizon* | | |
|---|---|---|---|
| Point Estimates | 1 day | 5 days | 10 days |
| Mean | -0.677 | -0.290 | 0.289 |
| Median | -0.221 | 0.141 | 0.637 |
| $5^{\text{th}}$ Percentile | -2.475 | -1.787 | -1.584 |
| $95^{\text{th}}$ Percentile | 0.079 | 0.465 | 1.186 |
| Sample Average | 0.045 | 0.316 | 0.588 |

**(a)** Positive Labels

| | *Return Horizon* | | |
|---|---|---|---|
| Point Estimates | 1 day | 5 days | 10 days |
| Mean | 1.219 | 1.027 | 1.088 |
| Median | 0.862 | 1.082 | 1.156 |
| $5^{\text{th}}$ Percentile | 0.092 | -0.201 | -0.126 |
| $95^{\text{th}}$ Percentile | 3.474 | 2.325 | 2.505 |
| Sample Average | 0.045 | 0.316 | 0.588 |

**(b)** Negative Labels

**Table 1:** 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, GPT-4o, GPT-5-mini, and GPT-5-nano 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 4.2.1 for discussion.

| Policy Topic | Covariate | *Point Estimates* | | | | Sample |
|---|---|---|---|---|---|---|
| | | Mean | Median | 5% | 95% | Average |
| Health | DW1 | -0.018 | -0.020 | -0.023 | -0.006 | 0.150 |
| Health | Democrat | 0.012 | 0.014 | 0.003 | 0.016 | 0.150 |
| Health | Senate | 0.005 | 0.004 | 0.001 | 0.012 | 0.150 |
| Banking, Finance & Domestic Com. | DW1 | 0.013 | 0.013 | 0.007 | 0.022 | 0.127 |
| Banking, Finance & Domestic Com. | Democrat | -0.010 | -0.010 | -0.015 | -0.004 | 0.127 |
| Banking, Finance & Domestic Com. | Senate | -0.008 | -0.009 | -0.016 | -0.001 | 0.127 |
| Defense | DW1 | 0.022 | 0.022 | 0.015 | 0.034 | 0.204 |
| Defense | Democrat | -0.005 | -0.004 | -0.009 | 0.000 | 0.204 |
| Defense | Senate | -0.008 | -0.006 | -0.019 | -0.003 | 0.204 |
| Government Operations | DW1 | 0.015 | 0.015 | 0.000 | 0.028 | 0.281 |
| Government Operations | Democrat | -0.005 | -0.004 | -0.014 | 0.003 | 0.281 |
| Government Operations | Senate | -0.013 | -0.013 | -0.020 | -0.006 | 0.281 |
| Public Lands & Water Management | DW1 | 0.027 | 0.027 | 0.020 | 0.030 | 0.238 |
| Public Lands & Water Management | Democrat | -0.006 | -0.007 | -0.010 | -0.003 | 0.238 |
| Public Lands & Water Management | Senate | 0.032 | 0.032 | 0.025 | 0.036 | 0.238 |

**Table 2:** 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, GPT-4o, GPT-5-mini, and GPT-5-nano 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 $1\{\widehat{V}_r^{m,p} = v\}$ on alternative covariates $W_r$. 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 4.2.1 for discussion.

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

**(a)** GPT-3.5-turbo

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

**(b)** GPT-4o

| | Median | 5% | 95% |
|---|---|---|---|
| *Normalized Bias* | | | |
| Plug-In | 0.030 | -1.646 | 1.567 |
| Debiased | -0.002 | -0.052 | 0.054 |
| *Coverage* | | | |
| Plug-In | 0.906 | 0.589 | 0.953 |
| Debiased | 0.927 | 0.903 | 0.945 |

**(c)** GPT-5-mini

| | Median | 5% | 95% |
|---|---|---|---|
| *Normalized Bias* | | | |
| Plug-In | 0.168 | -2.079 | 2.092 |
| Debiased | 0.005 | -0.052 | 0.060 |
| *Coverage* | | | |
| Plug-In | 0.779 | 0.387 | 0.953 |
| Debiased | 0.930 | 0.906 | 0.946 |

**(d)** GPT-5-nano

**Table 3:** Summary statistics for normalized bias and coverage across Monte Carlo simulations based on Congressional legislation.

*Notes*: The normalized bias reports the average bias of the plug-in regression coefficient $\widehat{\beta}$ and the debiased coefficient $\widehat{\beta}^{debiased}$ for the target regression coefficient divided by their respective standard deviations across simulations. The coverage reports the fraction of simulations in which a 95% nominal confidence interval centered around the plug-in regression coefficient $\widehat{\beta}$ and the bias-corrected coefficient $\widehat{\beta}^{debiased}$ cover the target regression coefficient. We summarize the distribution of normalized bias and coverage across regression specifications, choice of large language model and prompting strategies. For each combination of model topic $V_r$, covariate $W_r$, large language model $m$ and prompting strategy $p$, we randomly sample $5,000$ Congressional bills and calculate the plug-in regression coefficient $\widehat{\beta}$ and the bias-corrected regression coefficient $\widehat{\beta}^{debiased}$ using a 5% validation sample. Results are averaged over $1,000$ simulations. See Section 4.3 for discussion.

# Large Language Models: An Applied Econometric Perspective

*Online Appendix*

Jens Ludwig & Sendhil Mullainathan & Ashesh Rambachan

# A Appendix Figures and Tables

| | 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 A1:** 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 3.2.1 for discussion.

| Metric | Average | Benchmark |
|---|---|---|
| Cosine similarity | 0.830 | 0.379 |
| Euclidean distance | 0.536 | 1.110 |

**(a)** Base prompt

| Metric | Average | Benchmark |
|---|---|---|
| Cosine similarity | 0.830 | 0.379 |
| Euclidean distance | 0.536 | 1.110 |

**(b)** Prompt with date restriction

**Table A2:** Embedding distance between GPT-4o's completed bill descriptions and original bill descriptions.

*Notes*: This table calculates the cosine similarity and Euclidean distance between embeddings of GPT-4o's completed bill descriptions and embeddings of the original bill descriptions. We construct embeddings using OpenAI's text-embedding-3-small model. As a benchmark, we calculate the average cosine similarity and Euclidean distance between 10,000 randomly selected pairs of original bill descriptions. Table (a) provides results for the base prompt and Table (b) provides results for the base prompt with the additional date restriction. The results in Table (a) and Table (b) are the same up to 3 decimal places. See Section 3.2.1 for discussion.

| Metric | Average | Benchmark |
|---|---|---|
| Cosine similarity | 0.880 | 0.309 |
| Euclidean distance | 0.455 | 1.172 |

**(a)** Base prompt

| Metric | Average | Benchmark |
|---|---|---|
| Cosine similarity | 0.880 | 0.309 |
| Euclidean distance | 0.455 | 1.172 |

**(b)** Prompt with date restriction

**Table A3:** Embedding distance between GPT-4o's completed financial news headlines and original financial news headlines.

*Notes*: This table calculates the cosine similarity and Euclidean distance between embeddings of GPT-4o's completed financial news headlines and embeddings of the original financial news headlines. We construct embeddings using OpenAI's text-embedding-3-small model. As a benchmark, we calculate the average cosine similarity and Euclidean distance between 10,000 randomly selected pairs of original financial news headlines. Table (a) provides results for the base prompt and Table (b) provides results for the base prompt with the additional date restriction. The results in Table (a) and Table (b) are the same up to 3 decimal places. See Section 3.2.1 for discussion.

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

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

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

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

**Figure A1:** 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 3.2.1.

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

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

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

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

**Figure A2:** Two examples of GPT-4o completions that exactly match original financial news headlines.

*Notes*: On 10,000 randomly sampled financial news headlines from 2019, we prompted GPT-4o to complete the financial news headline based on 50% of its text. See Section 3.2.1.

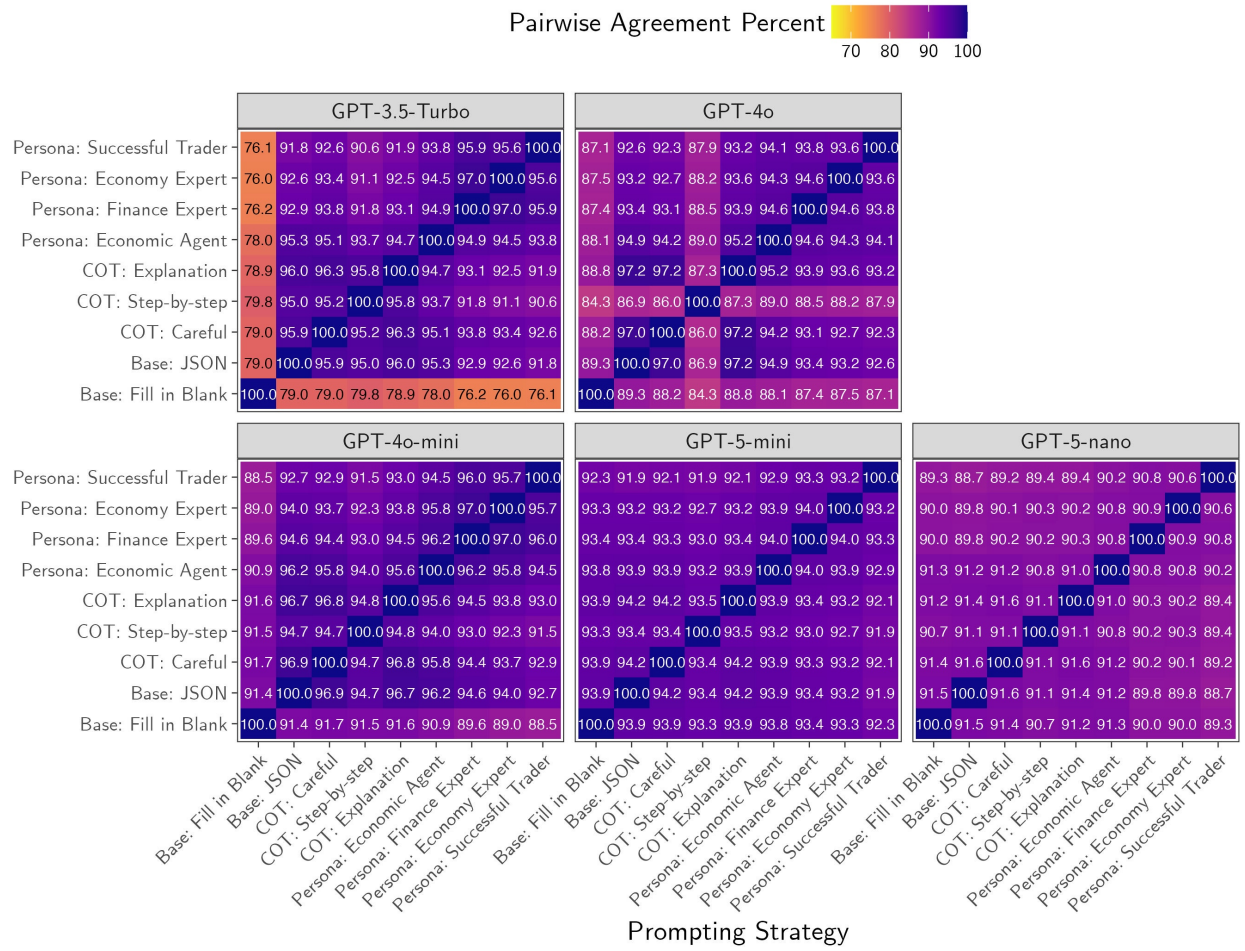


**Figure A3:** Variation in pairwise agreement between large language model labels across prompting strategies on financial news headlines.

*Notes*: On financial news headlines from 2019, we prompt GPT-3.5-Turbo, GPT-4o, GPT-4o-mini, GPT-5-mini, and GPT-5-nano 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 4.2.1 for discussion.

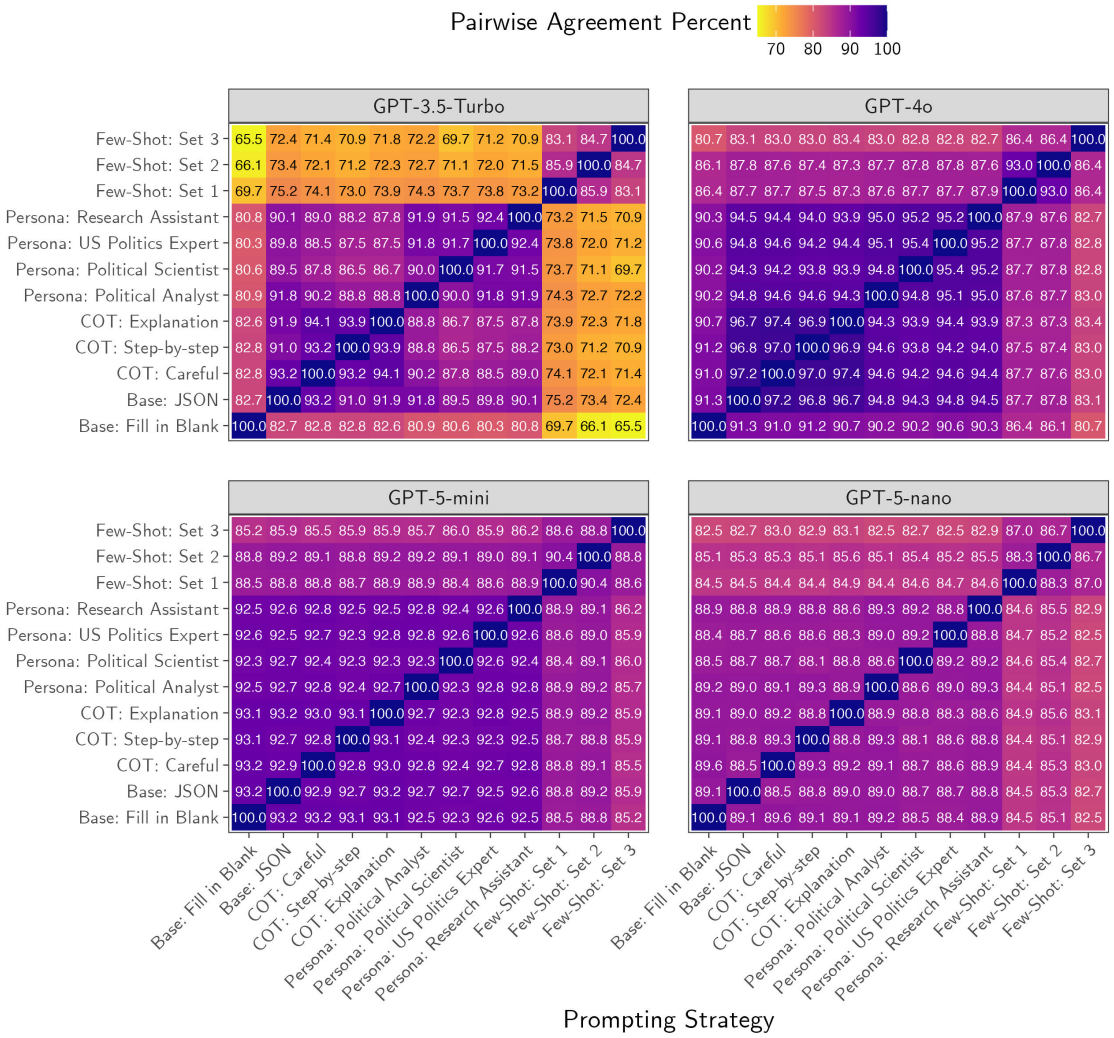


**Figure A4:** Variation in pairwise agreement between large language model labels across prompting strategies on congressional legislation.

*Notes*: On 10,000 randomly sampled Congressional bills, we prompt GPT-3.5-turbo, GPT-4o, GPT-5-mini, and GPT-5-nano 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 4.2.1 for discussion.

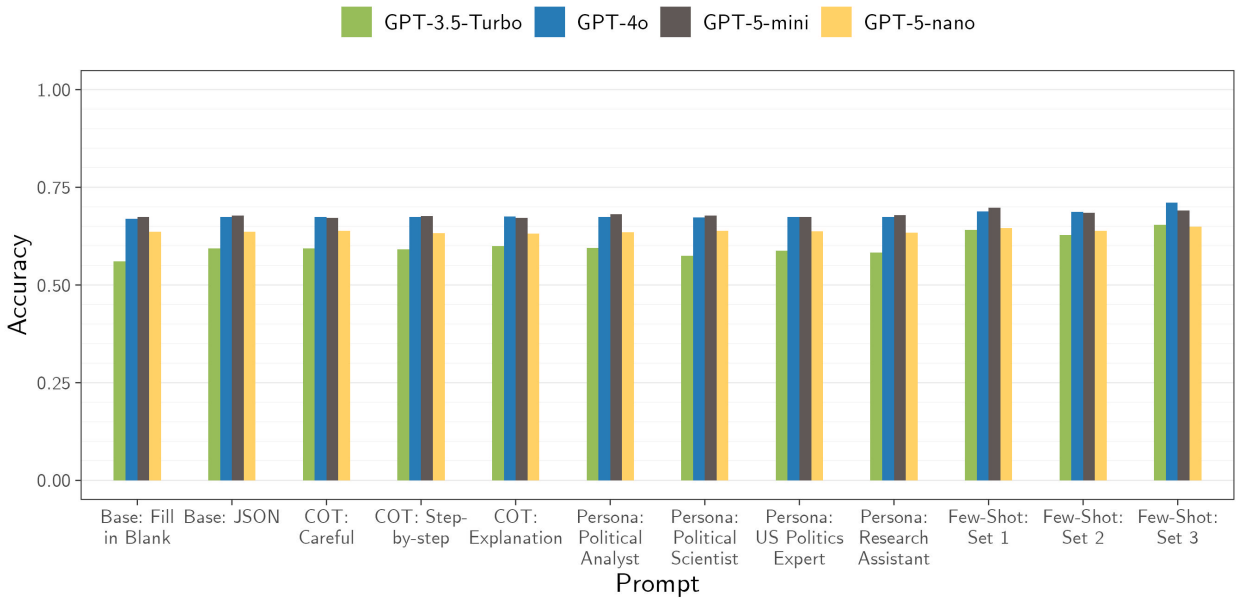


**Figure A5:** Accuracy of large language model labels of bill topic across model and prompt variation.

*Notes*: On 10,000 Congressional bills, we prompt GPT-3.5-Turbo, GPT-4o, GPT-5-mini, and GPT-5-nano 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 4.2.1 for discussion.

# B Proofs of Main Results

## B.1 Proof of Lemma 1 and Proposition 1

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

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

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

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

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

Lemma 1 then follows immediately. □

## B.2 Proof of Lemma 2

To show this result, rewrite

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

as

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

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 (10). 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

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

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

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

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

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

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

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

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

where the last inequality follows by Assumption 2. The result in Equation (10) is immediate following the definition of $\mathcal{M}(Q, \delta)$.

To prove Equation (11), 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 $\widehat{m}(\cdot) \in \mathcal{M}(Q, \delta)$,

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

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

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

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

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

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

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

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

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 (10), we can apply the mean value theorem and the definition of sensitive text pieces to obtain the lower bound

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

□

## B.4 Proof of Proposition 3

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

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

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

$$\beta = \left(\sum_{r\in\mathcal{R}} q_r^D W_r W_r'\right)^{-1} \left(\sum_{r\in\mathcal{R}} q_r^D W_r V_r\right) + \left(\sum_{r\in\mathcal{R}} q_r^D W_r W_r'\right)^{-1} \left(\sum_{r\in\mathcal{R}} q_r^D W_r \Delta_r\right).$$

The result is then immediate from the definition of $\beta^*$ and the best linear projection of $\Delta_r$ onto $W_r$.

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

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

But, of course, since $W_r = V_r'\beta^* + \epsilon_r$ for $\epsilon_r$ 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 $\epsilon_r$ 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:

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

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 (3) holds and

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

as the number of text pieces grows large.

### C.1.2 The Researcher's Sample Moment Condition

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

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

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

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

as the number of text pieces grow large.

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

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

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

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

as the number of text pieces grow large.

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

In this section, we separately analyze the asymptotic distribution of the bias-corrected regression coefficient introduced in Section 4.3 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 4.3, we 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 calculates the plug-in regression coefficient

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

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

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

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

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

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

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

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

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

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

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

for

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

and $\sigma_W^2$ the limit of $\frac{1}{N}\sum_{r\in\mathcal{R}} W_r^2$. This delivers Equation (15) given in Section 4.3 of the main text. By a similar argument, we can show that the validation-sample only regression coefficient has a limiting distribution given by

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

for $\Omega^{validation} = \sigma_W^{-4}\frac{1-\rho_V}{\rho_V}\sigma_{WV}^2$, as stated in Section 4.3 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

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

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

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

On the validation sample, the researcher separately calculates

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

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

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

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

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

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

# D Additional Monte Carlo Simulations based on Congressional Legislation

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

## D.1 Varying the Size of the Validation Data

In Section 4.3 of the main text, we evaluated the performance of the plug-in regression coefficient against the bias-corrected estimator using a 5% validation sample. We explore how performance varies as we vary the size of the validation sample.

For a given bill topic $V_r$, covariate $W_r$, and pair of large language model and prompting strategy, we randomly draw a sample of $5,000$ bills. On this random sample, we first calculate the plug-in regression coefficient $\widehat{\beta}$. We next randomly reveal the ground-truth label $V_r$ on 2.5% (125 bills), 5% (250 bills), 10% (500 bills), 25% (1250 bills), and 50% (2500 bills) of the random sample of $5,000$ bills. We then calculate the bias-corrected coefficient $\widehat{\beta}^{debiased}$ on each validation sample. We repeat these steps for $1,000$ randomly sampled datasets, and we calculate the average bias of these alternative estimates for the target regression $\beta^*$ 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 $m$, and prompting strategy $p$. This allows us to summarize how the plug-in regression performs against the bias-corrected

regression across a wide variety of possible regression specifications, choices of large language model and prompting strategies.

Appendix Figure A6 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 Appendix Tables A4-A7 report summary statistics for labels produced by each model 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 Appendix Tables A4-A7 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, Appendix Figure A7 compares the mean square error of the bias-corrected coefficient versus the validation-sample only estimate of the target regression as we vary the size of the validation sample. The bias-corrected coefficient obtains noticeable improvements in mean square error for the validation proportions equal to 2.5%, 5% and 10%. The bias-corrected coefficient performs similarly to the validation-sample only estimator for validation proportions equal to 25%, although it is likely unrealistic that the researcher would collect such large validation samples in an empirical application.

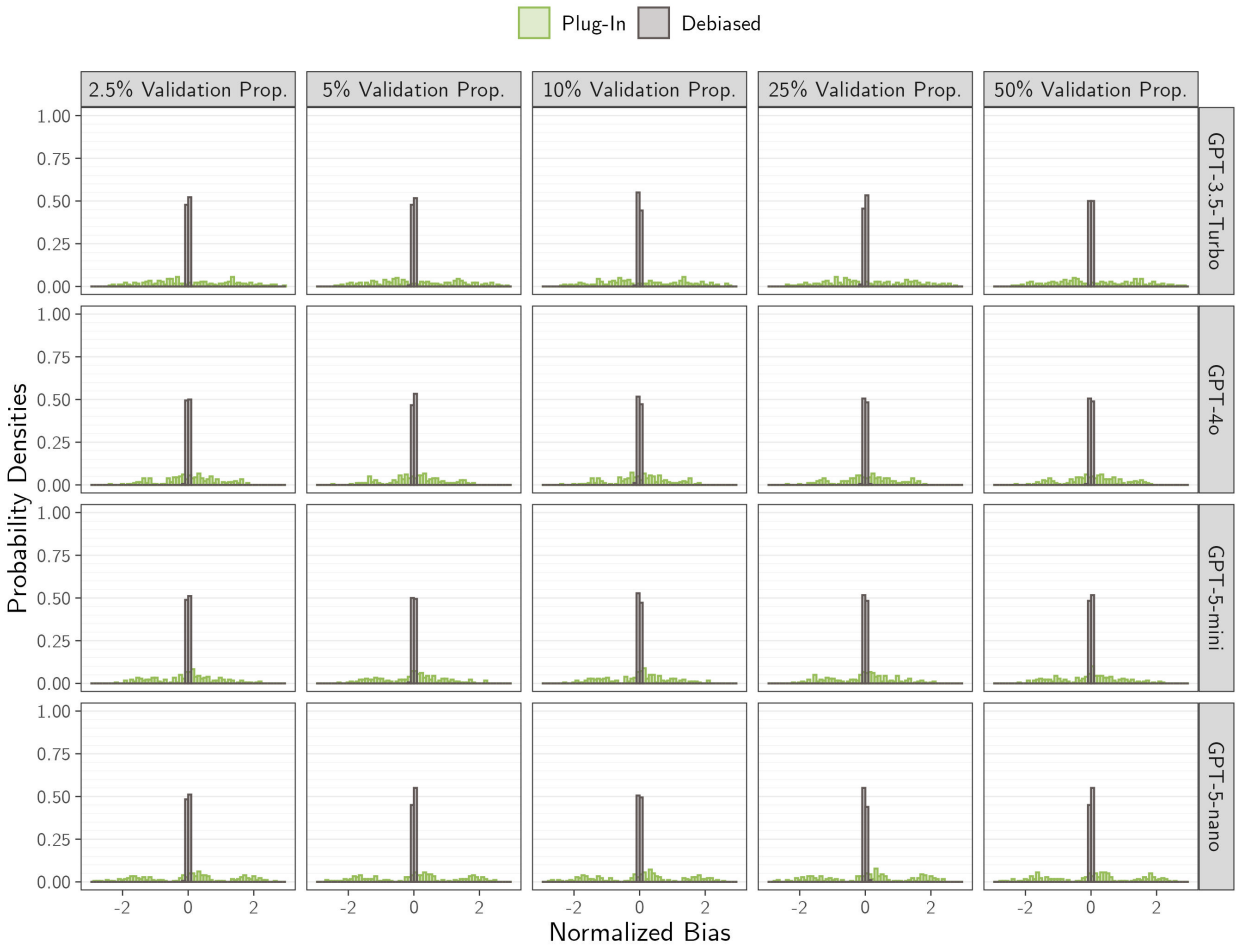


**Figure A6:** Normalized bias of the plug-in regression and bias-corrected regression across Monte Carlo simulations based on congressional legislation as the validation sample size varies.

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

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

**(a)** Plug-in regression

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

**(b)** Debiased regression

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

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

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

**(a)** Plug-in regression

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

**(b)** Debiased regression

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

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

| Validation Prop. | Median | 5% | 95% |
|---|---|---|---|
| *Normalized Bias* | | | |
| 2.5% | 0.062 | -1.612 | 1.566 |
| 5% | 0.030 | -1.646 | 1.567 |
| 10% | 0.058 | -1.624 | 1.545 |
| 25% | 0.040 | -1.569 | 1.533 |
| 50% | 0.040 | -1.619 | 1.572 |
| *Coverage* | | | |
| 2.5% | 0.907 | 0.570 | 0.954 |
| 5% | 0.906 | 0.589 | 0.953 |
| 10% | 0.909 | 0.558 | 0.954 |
| 25% | 0.900 | 0.585 | 0.955 |
| 50% | 0.905 | 0.583 | 0.954 |

**(a)** Plug-in regression

| Validation Prop. | Median | 5% | 95% |
|---|---|---|---|
| *Normalized Bias* | | | |
| 2.5% | 0.000 | -0.041 | 0.051 |
| 5% | -0.002 | -0.052 | 0.054 |
| 10% | -0.002 | -0.066 | 0.049 |
| 25% | -0.003 | -0.050 | 0.055 |
| 50% | 0.003 | -0.048 | 0.047 |
| *Coverage* | | | |
| 2.5% | 0.895 | 0.843 | 0.925 |
| 5% | 0.927 | 0.903 | 0.945 |
| 10% | 0.939 | 0.925 | 0.952 |
| 25% | 0.947 | 0.936 | 0.958 |
| 50% | 0.948 | 0.934 | 0.957 |

**(b)** Debiased regression

**Table A6:** Summary statistics for normalized bias and coverage for Monte Carlo simulations on congressional legislation for GPT-5-mini, varying the size of the validation sample.

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

| Validation Prop. | Median | 5% | 95% |
|---|---|---|---|
| *Normalized Bias* | | | |
| 2.5% | 0.151 | -2.072 | 2.113 |
| 5% | 0.168 | -2.079 | 2.092 |
| 10% | 0.141 | -2.017 | 2.116 |
| 25% | 0.160 | -2.058 | 2.085 |
| 50% | 0.171 | -2.021 | 2.123 |
| *Coverage* | | | |
| 2.5% | 0.771 | 0.379 | 0.952 |
| 5% | 0.779 | 0.387 | 0.953 |
| 10% | 0.781 | 0.361 | 0.951 |
| 25% | 0.780 | 0.390 | 0.951 |
| 50% | 0.785 | 0.377 | 0.956 |

**(a)** Plug-in regression

| Validation Prop. | Median | 5% | 95% |
|---|---|---|---|
| *Normalized Bias* | | | |
| 2.5% | 0.000 | -0.050 | 0.042 |
| 5% | 0.005 | -0.052 | 0.060 |
| 10% | -0.001 | -0.052 | 0.054 |
| 25% | -0.002 | -0.058 | 0.062 |
| 50% | 0.002 | -0.052 | 0.054 |
| *Coverage* | | | |
| 2.5% | 0.902 | 0.850 | 0.929 |
| 5% | 0.930 | 0.906 | 0.946 |
| 10% | 0.942 | 0.927 | 0.955 |
| 25% | 0.947 | 0.935 | 0.959 |
| 50% | 0.946 | 0.935 | 0.958 |

**(b)** Debiased regression

**Table A7:** Summary statistics for normalized bias and coverage for Monte Carlo simulations on congressional legislation for GPT-5-nano, varying the size of the validation sample.

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

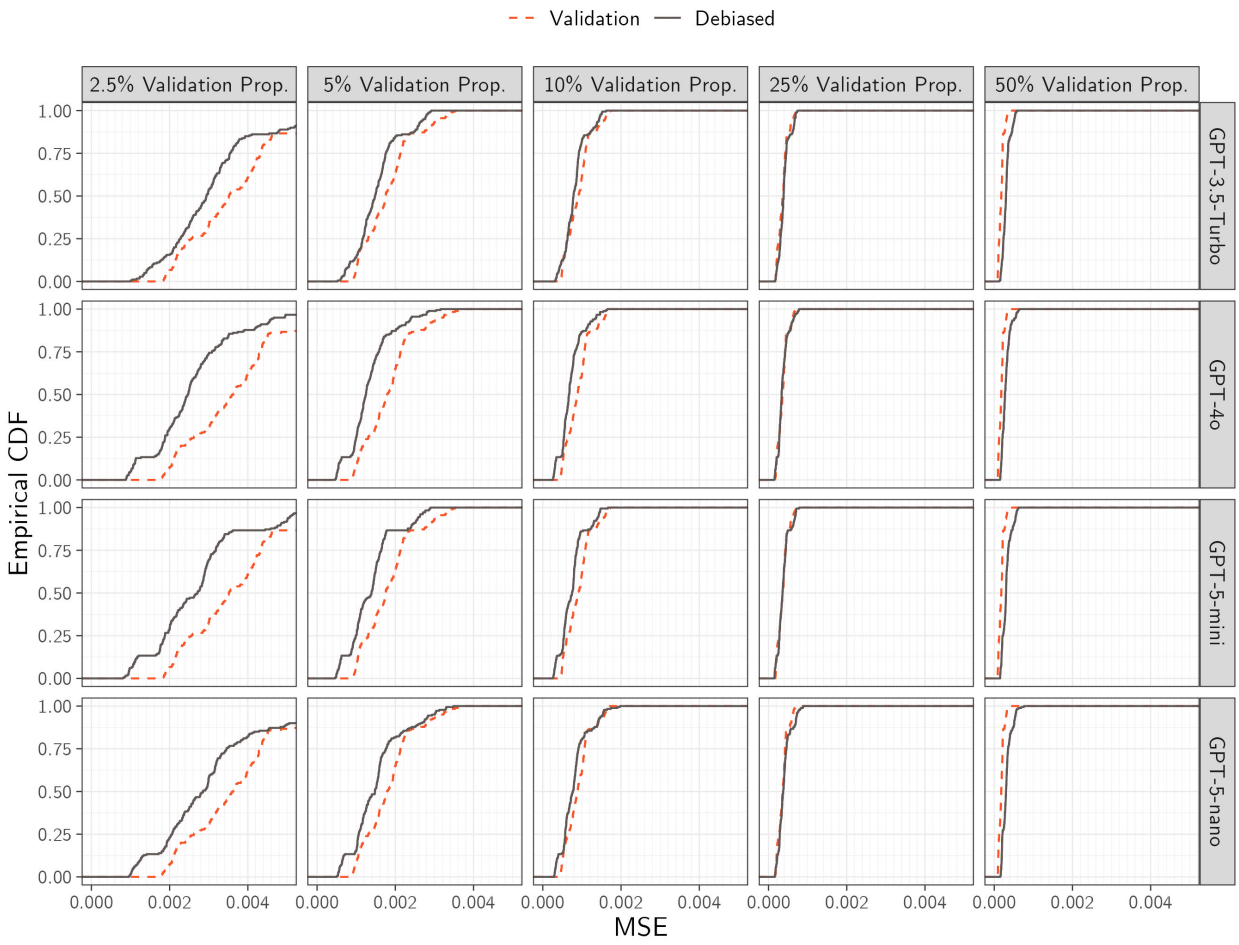


**Figure A7:** Cumulative distribution function of mean square error for the bias-corrected estimator against validation-sample only estimator, varying the size of the validation sample.

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

## 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.

We use the same 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 Appendix Figure A8, 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. 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. Appendix Table A8 summarizes the coefficient estimates across models and prompts for each choice of labeled policy topic and the covariate.

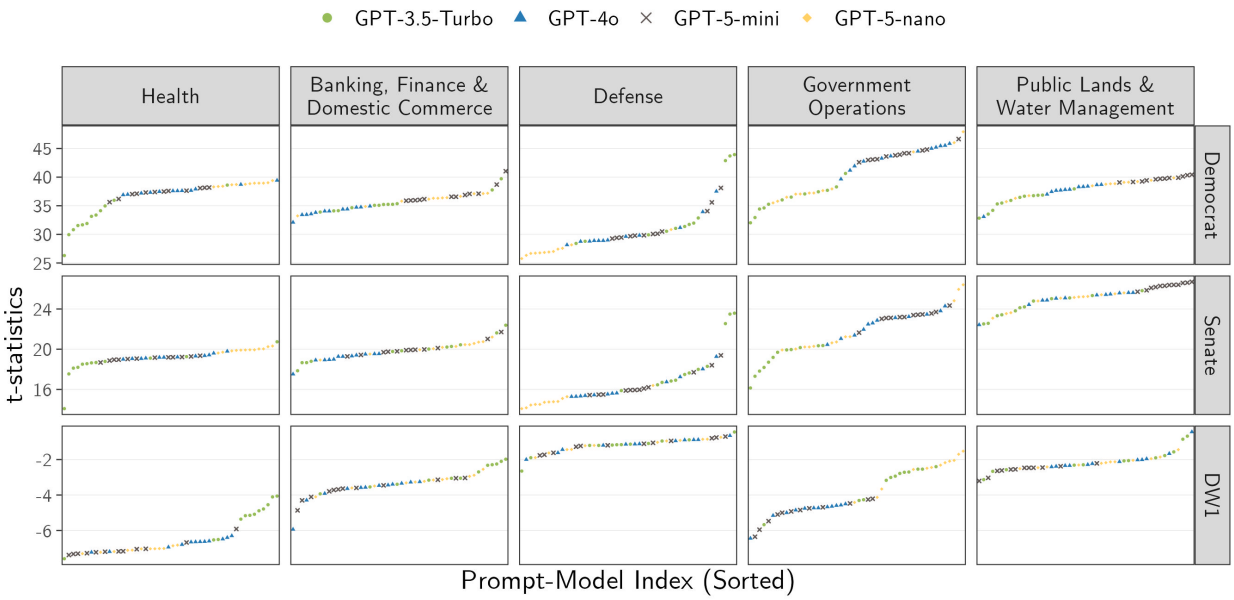


**Figure A8:** Variation in t-statistics across large language models and prompting strategies on congressional legislation, using the economic concept as a covariate.

*Notes*: On 10,000 Congressional bills, we prompt GPT-3.5-Turbo, GPT-4o, GPT-5-mini, and GPT-5-nano 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 $W_r$ on indicators $\widehat{V}_r^{m,p}$ for the large language model's labeled policy topic. In each subplot, the t-statistic estimates are sorted in ascending order for clarity. See Appendix D.2.

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

| Covariate | Policy Topic | *Point Estimates* | | | | Sample |
|---|---|---|---|---|---|---|
| | | Mean | Median | 5% | 95% | Average |
| DW1 | Health | -0.096 | -0.097 | -0.105 | -0.078 | -0.062 |
| DW1 | Banking, Finance & Domestic Com. | -0.043 | -0.043 | -0.053 | -0.029 | -0.062 |
| DW1 | Defense | -0.016 | -0.016 | -0.025 | -0.010 | -0.062 |
| DW1 | Government Operations | -0.043 | -0.046 | -0.064 | -0.024 | -0.062 |
| DW1 | Public Lands & Water Management | -0.025 | -0.026 | -0.033 | -0.013 | -0.062 |
| Democrat | Health | 0.643 | 0.646 | 0.617 | 0.655 | 0.604 |
| Democrat | Banking, Finance & Domestic Com. | 0.579 | 0.579 | 0.568 | 0.593 | 0.604 |
| Democrat | Defense | 0.588 | 0.588 | 0.577 | 0.602 | 0.604 |
| Democrat | Government Operations | 0.594 | 0.596 | 0.571 | 0.614 | 0.604 |
| Democrat | Public Lands & Water Management | 0.589 | 0.588 | 0.581 | 0.598 | 0.604 |
| Senate | Health | 0.331 | 0.327 | 0.320 | 0.359 | 0.317 |
| Senate | Banking, Finance & Domestic Com. | 0.299 | 0.298 | 0.282 | 0.313 | 0.317 |
| Senate | Defense | 0.293 | 0.294 | 0.276 | 0.308 | 0.317 |
| Senate | Government Operations | 0.292 | 0.292 | 0.282 | 0.305 | 0.317 |
| Senate | Public Lands & Water Management | 0.386 | 0.386 | 0.372 | 0.398 | 0.317 |

**Table A8:** 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, GPT-4o, GPT-5-mini, and GPT-5-nano 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 $W_r$ on indicators for whether the large language model labeled a particular policy topic $1\{\widehat{V}_r^{m,p} = v\}$. The final column ("Sample Average") reports the average of the linked variable $W_r$ across all Congressional bills. See Appendix D.2.

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 4.3 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 calculate the plug-in coefficients $\widehat{\beta}$ byregressing $W_r$ on $\widehat{V}_r^{m,p}$ (for $\widehat{V}_r^{m,p}$ a vector of indicators for the labeled policy topic). We next randomly reveal the ground-truth label $V_r$ on 5% of our random sample of $5,000$ bills, which produces a validation sample. We calculate the bias-corrected coefficients $\widehat{\beta}^{debiased}$ as described in Appendix C.2.2. We repeat these steps for $1,000$ randomly sampled datasets. We repeat this exercise for each possible combination of linked variable $W_r$, large language model $m$ (either GPT-3.5-turbo or GPT-4o) and prompting strategy $p$. This allows to summarize how the plug-in regression performs against the bias-corrected regression across a wide variety of possible regression specifications, choices of large language model and prompting strategies.

Appendix Figure A9 and Appendix Table A9 summarizes our results. The plug-in regression suffers from substantial biases for almost all combinations of linked variable $W_r$,

large language model $m$, and prompting strategy $p$. By contrast, using the validation sample for bias correction effectively eliminates these biases. Furthermore, the bottom panel of Appendix Table A9 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, Appendix Figure A10 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.

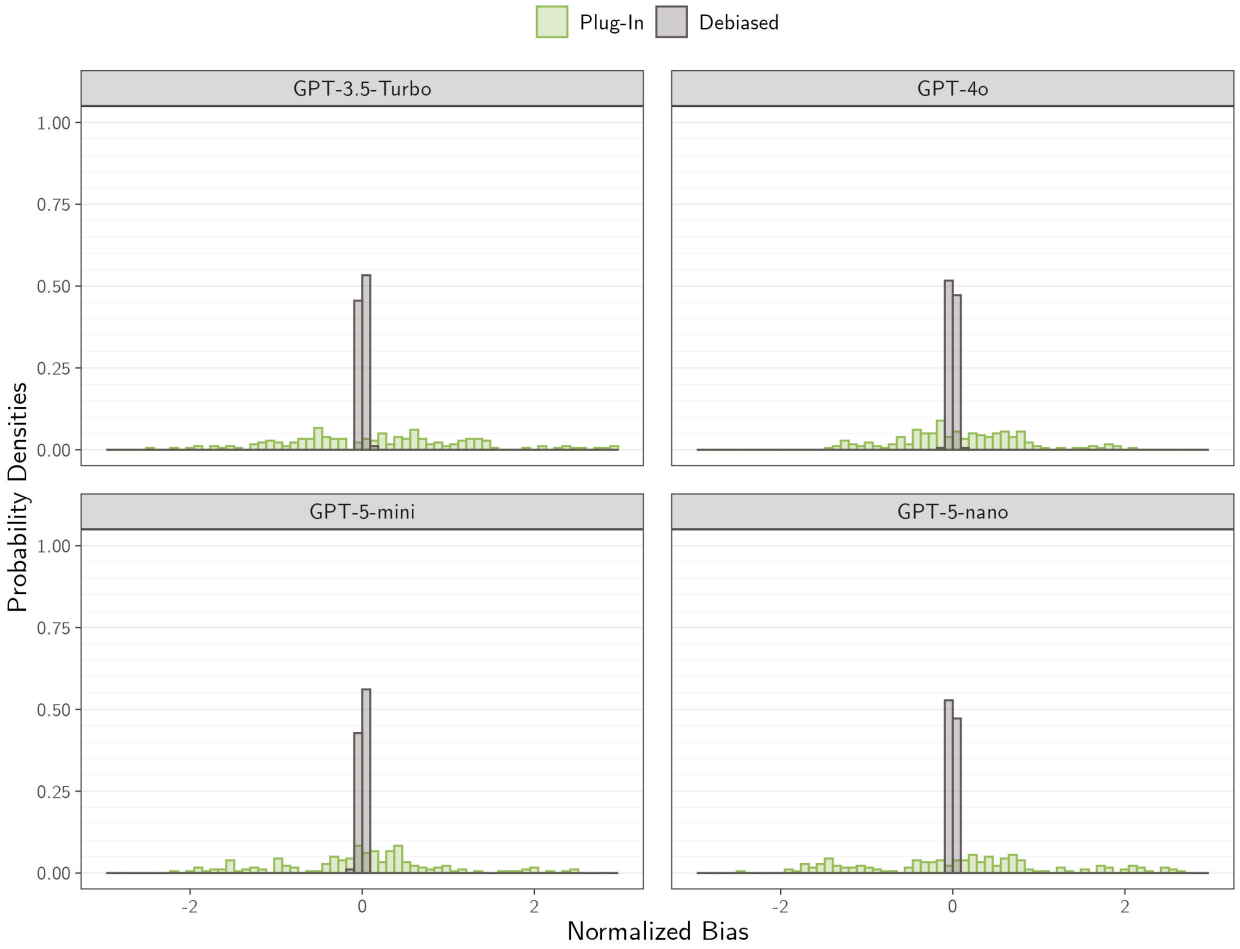


**Figure A9:** Normalized bias of the plug-in regression and bias-corrected regression using policy topic as a covariate across Monte Carlo simulations based on congressional legislation

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

| | Median | 5% | 95% |
|---|---|---|---|
| *Normalized Bias* | | | |
| Plug-In | 0.144 | -1.514 | 2.083 |
| Debiased | 0.003 | -0.051 | 0.060 |
| *Coverage* | | | |
| Plug-In | 0.901 | 0.363 | 0.950 |
| Debiased | 0.933 | 0.907 | 0.955 |

**(a)** GPT-3.5-Turbo

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

**(b)** GPT-4o

| | Median | 5% | 95% |
|---|---|---|---|
| *Normalized Bias* | | | |
| Plug-In | 0.034 | -1.615 | 1.828 |
| Debiased | 0.003 | -0.058 | 0.058 |
| *Coverage* | | | |
| Plug-In | 0.927 | 0.521 | 0.953 |
| Debiased | 0.932 | 0.906 | 0.955 |

**(c)** GPT-5-mini

| | Median | 5% | 95% |
|---|---|---|---|
| *Normalized Bias* | | | |
| Plug-In | 0.126 | -1.664 | 2.172 |
| Debiased | -0.002 | -0.064 | 0.058 |
| *Coverage* | | | |
| Plug-In | 0.900 | 0.405 | 0.950 |
| Debiased | 0.934 | 0.908 | 0.958 |

**(d)** GPT-5-nano

**Table A9:** Summary statistics for normalized bias and coverage for Monte Carlo simulations on congressional legislation using policy topic as a covariate.

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

### D.2.1 Varying the Size of the Validation Data:

We evaluated the performance of bias-correcting linear regression that use large language model labels as covariates using a 5% validation sample. We finally explore how the performance of the bias-corrected regression coefficient varies as we vary the size of the validation sample. We repeat our Monte Carlo simulations now varying the size of the validation sample by randomly revealing the measurements $V_r$ on 2.5% (125 bills), 5% (250 bills), 10% (500 bills), 25% (1250 bills), and 50% (2500 bills) of the random sample of $5,000$ bills. The results are summarized in Appendix Figure A11, Appendix Tables A10-A13 and Appendix Figure A12. We continue to find that the bias-corrected regression performs well in finite samples, even when the validation sample only contains 125 bills.

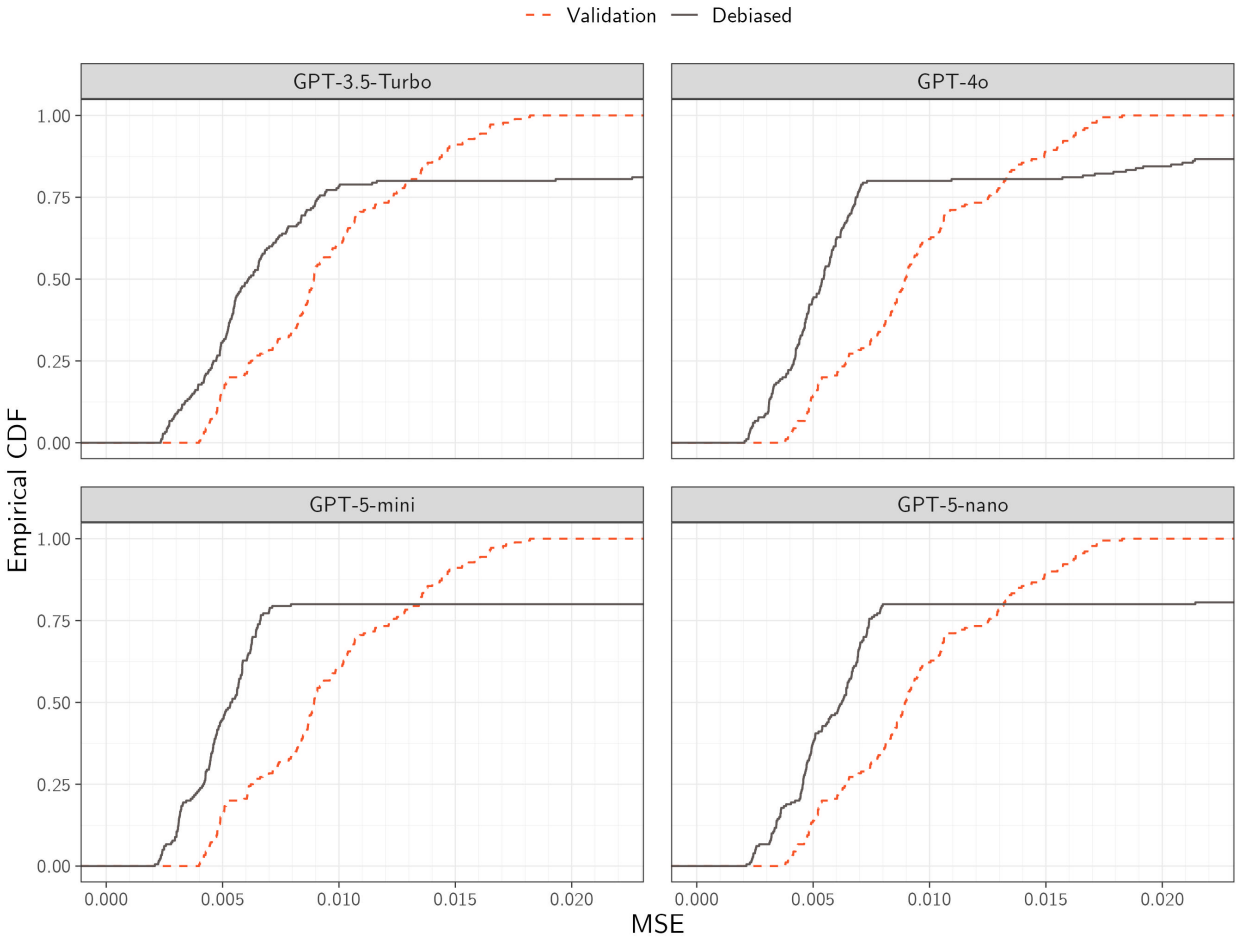


**Figure A10:** Cumulative distribution function of mean square error for the bias-corrected estimator against validation-sample only estimator using policy topic as a covariate.

*Notes*: For each combination of left hand side variable $W_r$, large language model $m$ and prompting strategy $p$, we randomly sample $5,000$ Congressional bills and calculate the plug-in regression $\widehat{\beta}$ and bias-corrected coefficient $\widehat{\beta}^*$ with the policy topic as a covariate using a 5% validation sample. We calculate the mean square error of $\widehat{\beta}^{debiased}$ and $\widehat{\beta}^*$ for the target regression $\beta^*$. 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 D.2.

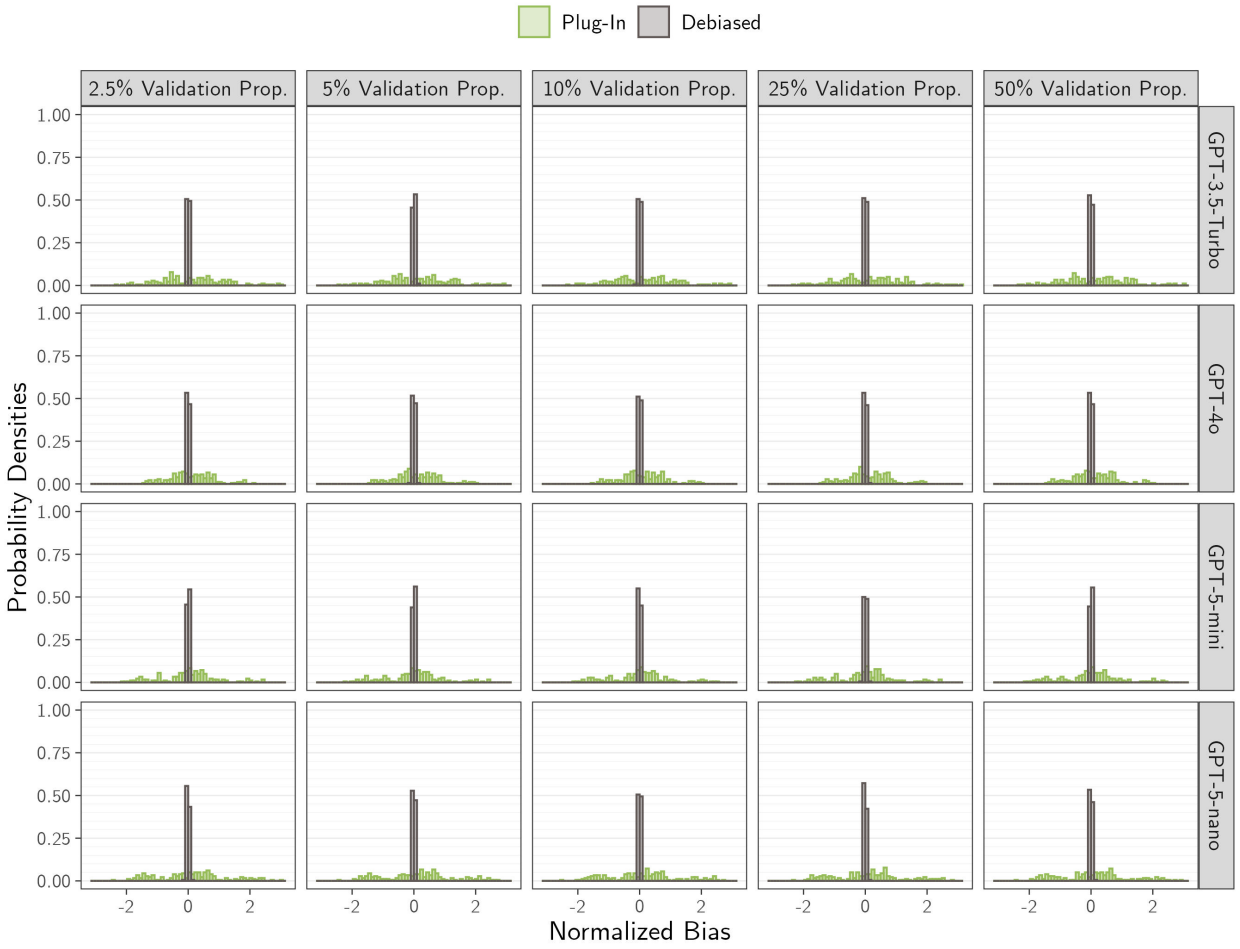


**Figure A11:** Normalized bias of the plug-in regression and bias-corrected regression using policy topic as a covariate as the validation sample size varies.

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

| Validation Prop. | Median | 5% | 95% |
|---|---|---|---|
| *Normalized Bias* | | | |
| 2.5% | 0.120 | -1.376 | 2.030 |
| 5% | 0.144 | -1.514 | 2.083 |
| 10% | 0.147 | -1.437 | 2.099 |
| 25% | 0.136 | -1.469 | 2.024 |
| 50% | 0.127 | -1.486 | 2.043 |
| *Coverage* | | | |
| 2.5% | 0.900 | 0.373 | 0.951 |
| 5% | 0.901 | 0.363 | 0.950 |
| 10% | 0.897 | 0.353 | 0.949 |
| 25% | 0.893 | 0.391 | 0.948 |
| 50% | 0.893 | 0.368 | 0.951 |

**(a)** Plug-in regression

| Validation Prop. | Median | 5% | 95% |
|---|---|---|---|
| *Normalized Bias* | | | |
| 2.5% | 0.000 | -0.057 | 0.050 |
| 5% | 0.003 | -0.051 | 0.060 |
| 10% | -0.002 | -0.055 | 0.057 |
| 25% | -0.001 | -0.053 | 0.052 |
| 50% | -0.002 | -0.055 | 0.056 |
| *Coverage* | | | |
| 2.5% | 0.909 | 0.868 | 0.959 |
| 5% | 0.933 | 0.907 | 0.955 |
| 10% | 0.941 | 0.927 | 0.954 |
| 25% | 0.946 | 0.935 | 0.958 |
| 50% | 0.947 | 0.936 | 0.959 |

**(b)** Debiased regression

**Table A10:** Summary statistics for normalized bias and coverage using policy topic as a covariate for GPT-3.5-Turbo, varying the size of the validation sample.

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

| Validation Prop. | Median | 5% | 95% |
|---|---|---|---|
| *Normalized Bias* | | | |
| 2.5% | 0.038 | -1.087 | 1.509 |
| 5% | 0.042 | -1.146 | 1.558 |
| 10% | 0.024 | -1.118 | 1.624 |
| 25% | 0.039 | -1.145 | 1.541 |
| 50% | 0.029 | -1.111 | 1.490 |
| *Coverage* | | | |
| 2.5% | 0.928 | 0.675 | 0.954 |
| 5% | 0.928 | 0.640 | 0.957 |
| 10% | 0.927 | 0.642 | 0.952 |
| 25% | 0.926 | 0.639 | 0.953 |
| 50% | 0.922 | 0.648 | 0.953 |

**(a)** Plug-in regression

| Validation Prop. | Median | 5% | 95% |
|---|---|---|---|
| *Normalized Bias* | | | |
| 2.5% | -0.003 | -0.065 | 0.052 |
| 5% | -0.003 | -0.063 | 0.053 |
| 10% | -0.001 | -0.059 | 0.055 |
| 25% | -0.002 | -0.062 | 0.053 |
| 50% | -0.003 | -0.057 | 0.052 |
| *Coverage* | | | |
| 2.5% | 0.904 | 0.860 | 0.948 |
| 5% | 0.930 | 0.900 | 0.952 |
| 10% | 0.943 | 0.928 | 0.952 |
| 25% | 0.947 | 0.933 | 0.957 |
| 50% | 0.949 | 0.935 | 0.959 |

**(b)** Debiased regression

**Table A11:** Summary statistics for normalized bias and coverage using policy topic as a covariate for GPT-4o, varying the size of the validation sample.

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

| Validation Prop. | Median | 5% | 95% |
|---|---|---|---|
| *Normalized Bias* | | | |
| 2.5% | 0.031 | -1.615 | 1.884 |
| 5% | 0.034 | -1.615 | 1.828 |
| 10% | 0.035 | -1.604 | 1.901 |
| 25% | 0.041 | -1.622 | 1.857 |
| 50% | 0.039 | -1.596 | 1.845 |
| *Coverage* | | | |
| 2.5% | 0.926 | 0.491 | 0.955 |
| 5% | 0.927 | 0.521 | 0.953 |
| 10% | 0.928 | 0.510 | 0.955 |
| 25% | 0.930 | 0.505 | 0.953 |
| 50% | 0.929 | 0.494 | 0.956 |

**(a)** Plug-in regression

| Validation Prop. | Median | 5% | 95% |
|---|---|---|---|
| *Normalized Bias* | | | |
| 2.5% | 0.005 | -0.049 | 0.065 |
| 5% | 0.003 | -0.058 | 0.058 |
| 10% | -0.004 | -0.052 | 0.059 |
| 25% | -0.001 | -0.050 | 0.052 |
| 50% | 0.003 | -0.054 | 0.055 |
| *Coverage* | | | |
| 2.5% | 0.904 | 0.852 | 0.958 |
| 5% | 0.932 | 0.906 | 0.955 |
| 10% | 0.940 | 0.926 | 0.953 |
| 25% | 0.947 | 0.935 | 0.958 |
| 50% | 0.947 | 0.935 | 0.958 |

**(b)** Debiased regression

**Table A12:** Summary statistics for normalized bias and coverage using policy topic as a covariate for GPT-5-mini, varying the size of the validation sample.

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

| Validation Prop. | Median | 5% | 95% |
|---|---|---|---|
| *Normalized Bias* | | | |
| 2.5% | 0.118 | -1.631 | 2.112 |
| 5% | 0.126 | -1.664 | 2.172 |
| 10% | 0.119 | -1.635 | 2.201 |
| 25% | 0.133 | -1.695 | 2.182 |
| 50% | 0.158 | -1.638 | 2.177 |
| *Coverage* | | | |
| 2.5% | 0.898 | 0.419 | 0.953 |
| 5% | 0.900 | 0.405 | 0.950 |
| 10% | 0.899 | 0.397 | 0.948 |
| 25% | 0.899 | 0.383 | 0.949 |
| 50% | 0.898 | 0.400 | 0.951 |

**(a)** Plug-in regression

| Validation Prop. | Median | 5% | 95% |
|---|---|---|---|
| *Normalized Bias* | | | |
| 2.5% | -0.003 | -0.052 | 0.057 |
| 5% | -0.002 | -0.064 | 0.058 |
| 10% | -0.001 | -0.047 | 0.053 |
| 25% | -0.004 | -0.052 | 0.053 |
| 50% | -0.003 | -0.057 | 0.052 |
| *Coverage* | | | |
| 2.5% | 0.905 | 0.858 | 0.966 |
| 5% | 0.934 | 0.908 | 0.958 |
| 10% | 0.943 | 0.928 | 0.956 |
| 25% | 0.946 | 0.935 | 0.957 |
| 50% | 0.948 | 0.936 | 0.959 |

**(b)** Debiased regression

**Table A13:** Summary statistics for normalized bias and coverage using policy topic as a covariate for GPT-5-nano, varying the size of the validation sample.

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

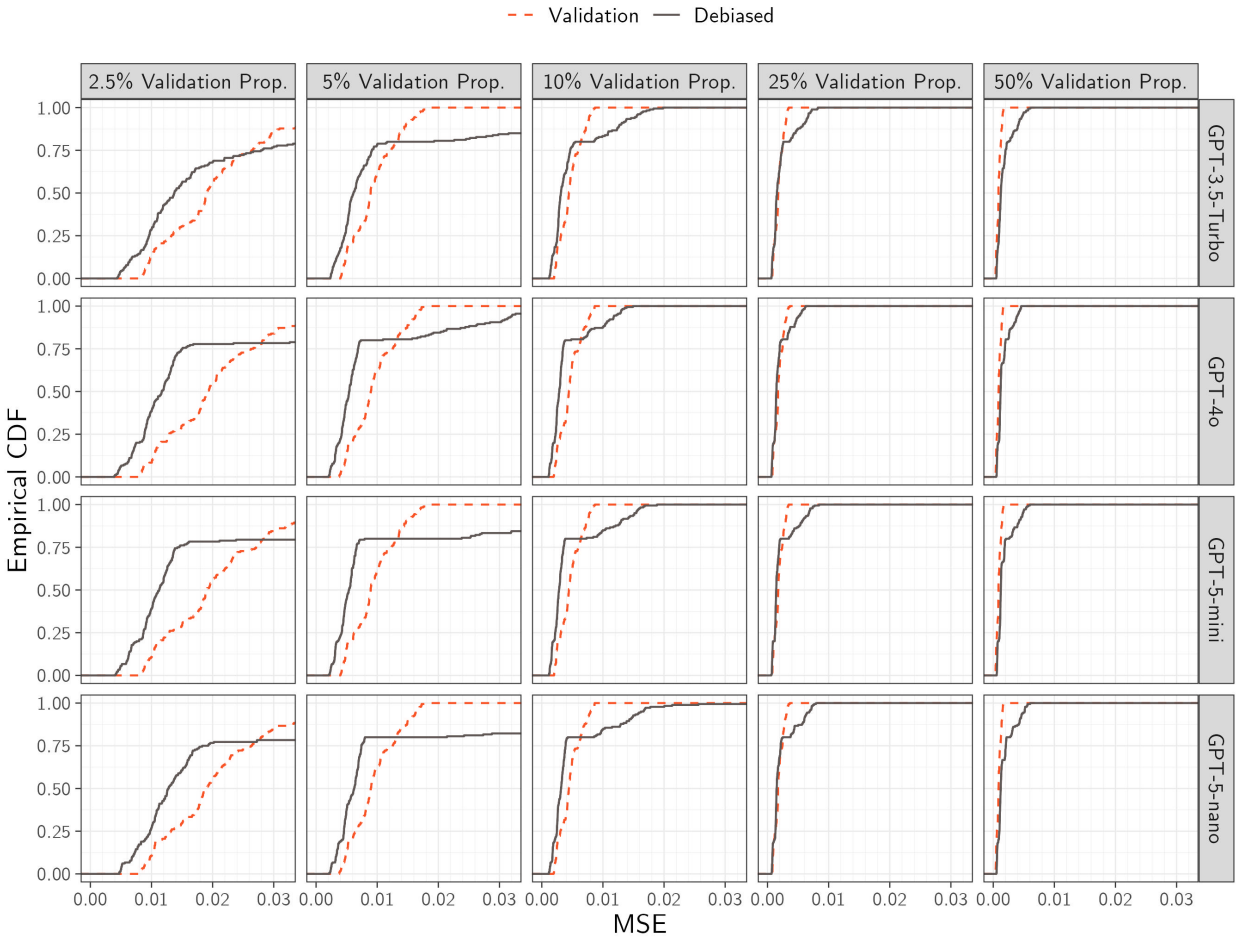


**Figure A12:** Cumulative distribution function of mean square error for the bias-corrected estimator against validation-sample only estimator using policy topic as the validation sample size varies.

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

# E Prompts for Congressional Bills and Financial News Headlines

**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 A13:** 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 3.2.1 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 A14:** 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 3.2.1 for further details.

**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 A15:** 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 3.2.1 for further details.

**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 A16:** 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, GPT-4o-mini, GPT-5-mini, and GPT-5-nano 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 4.2.1 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 A17:** 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, GPT-4o-mini, GPT-5-mini, and GPT-5-nano to label financial news headlines for whether they are positive, negative or neutral about the associated company. Each persona modification is added to the beginning of the base prompt with JSON output (Panel (a) of Appendix Figure A16). See Section 4.2.1 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 A18:** 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, GPT-4o-mini, GPT-5-mini, and GPT-5-nano to label financial news headlines for whether they are positive, negative or neutral about the associated company. Each chain-of-though modification alters the base prompt with JSON output (Panel (a) of Appendix Figure A16). See Section 4.2.1 for further details.

**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 A19:** 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, GPT-4o, GPT-5-mini, and GPT-5-nano to label the descriptions of 10,000 randomly drawn Congressional bills for their major topic. For each Congressional bill, we include the text of its description [description]. Figure (a) provides the base prompt with fill-in-the-blanks output, and Figure (b) provides the base prompt with JSON output. See Section 4.2.1 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 A20:** 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, GPT-4o, GPT-5-mini, and GPT-5-nano to label the descriptions of 10,000 randomly drawn Congressional bills for their major topic. Each persona modification is added to the beginning of the base prompt with JSON output (Panel (b) of Appendix Figure A19). See Section 4.2.1 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 A21:** 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, GPT-4o, GPT-5-mini, and GPT-5-nano to label the descriptions of 10,000 randomly drawn Congressional bills for their major topic. Each chain-of-though modification alters the base prompt with JSON output (Panel (b) of Appendix Figure A19). See Section 4.2.1 for further details.