Annual Review of Economics

Large Language Models: An Applied Econometric Framework

Jens Ludwig,[1,2] Sendhil Mullainathan,[2,3,4] and Ashesh Rambachan[3]

[1]Harris School of Public Policy, University of Chicago, Chicago, Illinois, USA

[2]National Bureau of Economic Research, Cambridge, Massachusetts, USA

[3]Department of Economics, Massachusetts Institute of Technology, Cambridge, Massachusetts, USA; email: asheshr@mit.edu

[4]Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, Cambridge, Massachusetts, USA

Annu. Rev. Econ. 2026. 18:283–316

The *Annual Review of Economics* is online at economics.annualreviews.org

https://doi.org/10.1146/annurev-economics-120925-105620

JEL codes: C10, C52

Keywords

large language models, prediction, estimation, leakage, validation

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.

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; they are 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 data set, this reflects the LLM's out-of-sample performance if and only if there is no overlap between the LLM's training data set and the researcher's data set. This can be violated because many LLMs are trained on intentionally obscured data sets. Computer scientists have found that LLMs are often trained on common benchmark evaluations. We find that LLMs appear to be trained on economic data sets (see also Glasserman & Lin 2023, Sarkar & Vafa 2024, Lopez-Lira et al. 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, Grattafiori 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 instead to 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: collecting a small validation sample and using 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 & Sepanski 1995, Chen et al. 2005, Schennach 2016) and extended in machine learning (e.g.,Wang et al. 2020, Angelopoulos et al. 2023, Battaglia et al. 2024, Egami et al. 2024, Carlson & 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 & 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 experiments (e.g., Park et al. 2023, Horton 2023, Manning et al. 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: ensuring no training leakage for prediction and collecting 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 data sets, and algorithms that resist formal characterization.

[1]Many resources are available about the technical foundations of LLMs. Readers are referred to, for example, Chang et al. (2024), Minaee et al. (2025), and Zhao et al. (2025). For overviews aimed at economists, readers are referred to Korinek (2023), 2024), Ash et al. (2024), and Dell (2025).

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 Data Set

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

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 data set is summarized by the vector $\mathbf{d}$, whose elements d_σ are sampling indicators for whether the researcher collected string σ. 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 1 (Congressional legislation). Consider descriptions of bills introduced in the US 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 2 (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 the following questions: 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 to use the collected text pieces to tackle two types of empirical analyses. The first is a prediction problem: predicting the linked variable Y_r using the associated text piece r. For example, can we predict stock returns from news headlines? The

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

second is an estimation problem: estimating 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 an LLM as any mapping from possible training data sets $\mathbf{t}$ to mappings between strings, where $\widehat{m}(\cdot; \mathbf{t}) : \Sigma^* \to \Sigma^*$ is its text generator when trained on data set $\mathbf{t}$ and $\widehat{m}(\sigma; \mathbf{t})$ is the LLM's response when prompted by string σ.

This definition has a key implication: The LLM is actually two distinct algorithms. First is a training algorithm that takes any data set $\mathbf{t}$ and learns the mapping between strings. Second, the text generator $\widehat{m}(\cdot; \mathbf{t})$ is the output of the training algorithm and of 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 ensure 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 and others by algorithm builders (often invisibly to researchers).

2.2.1. Interpreting the text generator. The text generator $\widehat{m}(\cdot; \mathbf{t})$ 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, White et al. 2023, Chen et al. 2024, Wei et al. 2024). Alternative prompt engineering strategies can be cast as alternative specifications of the text generator $\widehat{m}(\cdot; \mathbf{t})$.

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]

2.2.2. 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 RLHF aligns output with

[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 comply reliably. Our text generator abstraction takes further postprocessing steps as given, and $\widehat{m}(\cdot; \mathbf{t})$ represents the resulting final output.

human preferences. More recent developments RLVR, wherein models are trained on tasks with objectively correct answers. Crucially, the training data set $\mathbf{t}$ in our framework encompasses all strings used in both pretraining and posttraining 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 $\hat{Y}_r = \widehat{m}(r; \mathbf{t})$. 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 $\hat{V}_r = \widehat{m}(r; \mathbf{t})$ 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 data sets 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. 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 the social sciences (Hendrycks et al. 2021). 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 that 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 et al. 2024b, Dreyfuss & Raux 2025).

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. 2024), and LLMs struggle with logical puzzle variations (Nezhurina et al. 2025). 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., Nanda et al. 2023, jylin04 et al. 2024, Li et al. 2024, Nikankin et al. 2025). The evidence so far is at best mixed and at worst suggests that LLMs may not learn generalizable world models. For example, Vafa et al. (2024a) trained a generative sequence model on turn-by-turn driving data from 12.6 million taxi rides in New York City. While the model 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, and we cannot 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, $\widehat{Y}_r = \widehat{m}(r; \mathbf{t})$.[5] For some loss function $\ell(y, \tilde{y})$, the researcher calculates the sample average loss

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

[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.

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 $\mathbf{t} = (t_{\sigma_1}, \ldots, t_{\sigma_{|\Sigma^*|}})'$ denote the sampling indicators summarizing the LLM's realized training data set, all research contexts $Q(\cdot) \in \mathcal{Q}$ satisfy (*a*) for all values $\mathbf{d}$, $Q(\mathbf{D} = \mathbf{d}, \mathbf{T} = \mathbf{t}) = \Pi_{\sigma \in \Sigma^*} Q(D_\sigma = d_\sigma, T_\sigma = t_\sigma)$; and (*b*) $\mathbb{E}_Q\left[\sum_{r \in \mathcal{R}} D_r\right] = \mathbb{E}_Q\left[\sum_{r \in \mathcal{R}} D_r \mid \mathbf{T} = \mathbf{t}\right]$.

Assumption 1*a* states that sampling across strings is independent but not necessarily identically distributed. Assumption 1*b* 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 data set 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 defined by the research context $Q(\cdot)$:

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

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

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

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

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 $\widehat{m}(\cdot; \mathbf{t}) \in \mathcal{M}$.

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 $\widehat{m}(\cdot; \mathbf{t})$ with guarantee $\mathcal{M}$ generalizes in research context $Q(\cdot)$ if, for all text generators satisfying the guarantee $\widehat{m}(\cdot) \in \mathcal{M}$,

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

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 that only the guarantee $\mathcal{M}$ is satisfied.

Example 3 (Congressional legislation). Consider researchers predicting whether a bill will pass either the House or Senate, Y_r, using only its text description r. The LLM generates predictions $\widehat{m}(r; \mathbf{t})$ 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 cryptocurrency regulation or artificial intelligence.

Example 4 (Financial news headlines). Consider researchers predicting a company's realized returns Y_r from a financial news headline r. The LLM generates predictions $\widehat{m}(r; \mathbf{t})$ 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 $\widehat{m}(\cdot)$,

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

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

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

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—that is, 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 3.4 discusses how researchers can prevent training leakage through careful choices of model and prediction exercise.

3.3. Some Evidence of Training Leakage

Training leakage is a well-documented problem in computer science. LLMs' training data sets frequently contain examples from popular benchmark evaluations (Sainz et al. 2023, Golchin & Surdeanu 2024, LM Contamination Index 2024). This has generated skepticism about evaluating LLMs on any publicly available data (e.g., Ravaut et al. 2025). However, this evidence focuses on computer science applications. Might economic prediction problems face similar risks?

3.3.1. 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 (Adler & Wilkerson 2020, Wilkerson et al. 2023) contains descriptions r for over 400,000 bills proposed in Congress, along with information about 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 $\widehat{Y}_r = \widehat{m}(r; \mathbf{t})$ based on each bill's description r by prompting GPT-4o (see **Supplemental 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 (see left panel of **Supplemental 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 data set.

To evaluate training leakage, we prompt the model to complete each bill's text description based only on the first half of its text, following research in computer science such as Golchin & Surdeanu's (2024) (see **Supplemental 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 also appears to have memorized them. **Supplemental 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 (see left panel of **Supplemental 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 **Supplemental Figures 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 (see right panel of **Supplemental 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 (see right panel of **Supplemental Table A2**).

3.3.2. Assessing training leakage in financial news headlines. We consider another domain relevant for economists: financial markets. Existing work (e.g., Glasserman & Lin 2023, Lopez-Lira & Tang 2025) 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 four million headlines for 6,000 publicly traded companies from 2009 to 2020.

We sampled 10,000 financial news headlines from 2019, and we prompted GPT-4o to complete each financial news headline based on only 50% of its text (see **Supplemental Figure A15** for the specific prompt). GPT-4o reproduced 60 financial news headlines exactly as they were written in the publicly available data set, indicating that GPT-4o was likely trained on these headlines and memorized them. **Supplemental Figure A2** provides two examples of successful completions. On all other headlines, word embeddings of the model's completions are close to those of the original headlines (see left panel of **Supplemental Table A3**).

We again explore whether explicitly incorporating date restrictions into the LLM prompt moderates this evidence of training leakage (see **Supplemental 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.

Sarkar & 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 & Lin 2023, Lopez-Lira et al. 2025), this indicates substantial risk of training leakage in finance applications.

3.4. 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 $\frac{q^{T|\mathcal{D}}(t_r)}{q^{T}(t_r)} - 1$, that is, on how much learning that the researcher sampled text piece r updates 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

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

different ways. We illustrate this by considering the threat of training leakage and how to manage it across several examples.

Example 5 (Look-ahead 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, Grattafiori 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 τ and constructs an evaluation sample using documents published after τ, then $q_r^T(t_r)$ mechanically equals zero, since that information 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 are undisclosed, and they may be continuously fine-tuned. Researchers in natural language processing now regularly caution against sending test data to the application programming interfaces (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 worryingly, Barrie et al. (2024) illustrates that 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 6 (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 7 (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. The researcher draws a random sample to form their evaluation set. In this case, the researcher's sampling process is independent of the LLM's training data: Their 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 nonrandom treatment assignment resurrects possible confounding in causal inference, nonrandom sampling may resurrect training leakage.

[7] Balloccu et al. (2024) estimate that 263 benchmarks may have been inadvertently leaked to OpenAI through use of the chat interface and API, and Cheng et al. (2024) question the validity of publicly stated knowledge cutoffs in closed LLMs.

Example 8 (Prediction on the complete population). As a final example, suppose the researcher collected all Congressional bill descriptions from 1973 to 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.

More broadly, these examples illustrate a general principle: Preventing 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 discussed in Section 2.3, 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: collecting a small validation sample and using 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 they collected the economic concept V_r on each text piece r, they 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, $\widehat{V}_r := \widehat{m}(r; \mathbf{t})$, and plugs in the LLM's labels

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

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 **Supplemental Section C.1**), we can characterize the behavior of both estimators. The target moment condition converges to, at any parameter value $\theta \in \Theta$,

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

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

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

As in the prediction problem, conditioning on the training data set 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 $\widehat{m}(\cdot; \mathbf{t})$ 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 $\theta \in \Theta$,

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

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

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

The second term captures training leakage, that is, the potential overlap between the LLM's training data set and the researcher's evaluation sample. As discussed in Section 3.4, 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 $\Delta_r = \widehat{m}(r; \mathbf{t}) - V_r$ 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 δ 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 $\mathcal{M}(Q, \delta)$, that is, the collection of text generators satisfying $\left\|\widehat{m}(\cdot) - f^*(\cdot)\right\|_{\infty,Q} = \max_{r \in \mathcal{R}: q_r^D > 0} |\widehat{m}(r; \mathbf{t}) - f^*(r)| \leq \delta$. 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 and 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, θ. 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, (*a*) for all $\theta \in \Theta$ and $\widehat{m}(\cdot) \in \mathcal{M}(Q, \delta)$ satisfying no training leakage,

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

However, (*b*) 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,

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

Lemma 3*a* shows that guarantee $\mathcal{M}(Q, \delta)$ bounds the LLM's error for the moment condition. However, this is not enough for the LLM to automate the existing measurement. Lemma 3*b* shows that 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 $\widehat{m}(\cdot; \mathbf{t})$ 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 $\widehat{m}(\cdot; \mathbf{t})$ satisfies the guarantee $\mathcal{M}(Q, 0)$ for all research contexts $Q(\cdot)$.

4.3. 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.

4.3.1. 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,

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

or the independent variable,

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

The bias depends on how the LLM's error $\Delta_r = \widehat{m}(r;\mathbf{t}) - V_r$ 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 $r \in \mathcal{R}$.

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

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

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.

4.3.2. Assessing measurement error in financial news headlines. We return to the financial news headlines data set, 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 (Wharton Research Data Services 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, $\widehat{V}_r := \widehat{m}(r;\mathbf{t})$. 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 nonignorable 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

Table 1 Variation in point estimates across large language models and prompting strategies on financial news headlines

Point estimates	Return horizon		
	1 day	5 days	10 days
Mean	−0.677	−0.290	0.289
Median	−0.221	0.141	0.637
5th percentile	−2.475	−1.787	−1.584
95th percentile	0.079	0.465	1.186
Sample average	0.045	0.316	0.588

Positive labels

Point estimates	Return horizon		
	1 day	5 days	10 days
Mean	1.219	1.027	1.088
Median	0.862	1.082	1.156
5th percentile	0.092	−0.201	−0.126
95th percentile	3.474	2.325	2.505
Sample average	0.045	0.316	0.588

Negative labels

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 according to whether it expresses 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 4.3 for discussion.

its reply, either as filling-in-the-blanks text or as a structured JavaScript Object Notation (JSON) object. The prompts are provided in **Supplemental Section E**.

We further vary these base prompts in two ways, motivated by popular prompt engineering strategies (Liu et al. 2023, White et al. 2023, Chen et al. 2024, Wei 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 levels. **Figure 1** and **Table 1** summarize the results. Simply changing the prompt or model yields markedly different estimates. Many prompt–model combinations produce different directions and magnitudes of the relationship between the positive/negative label and the realized returns.[8]

4.3.3. 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 a measure of ideological positioning based on roll-call votes known as the DW1 score. 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 **Supplemental Section E** for the specific prompts we used).

[8] **Supplemental 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.

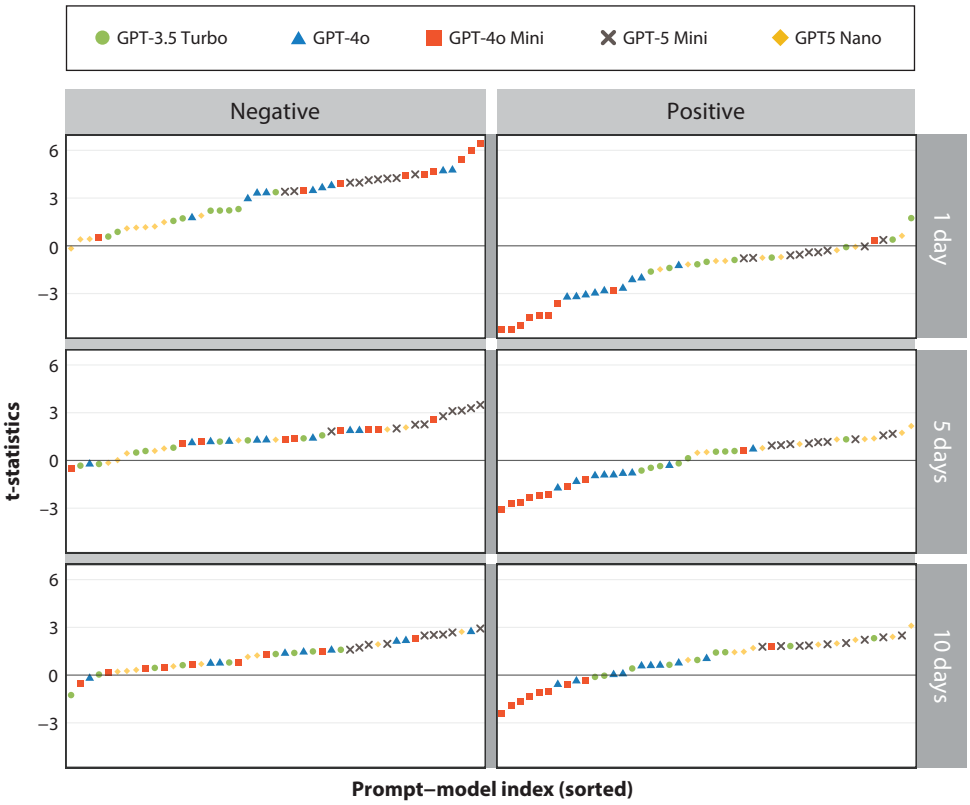

Figure 1

Variation in t-statistics for realized returns across large language models and prompting strategies on financial news headlines. 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 according to 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 4.3 for discussion.

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

[9]Supplemental 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.

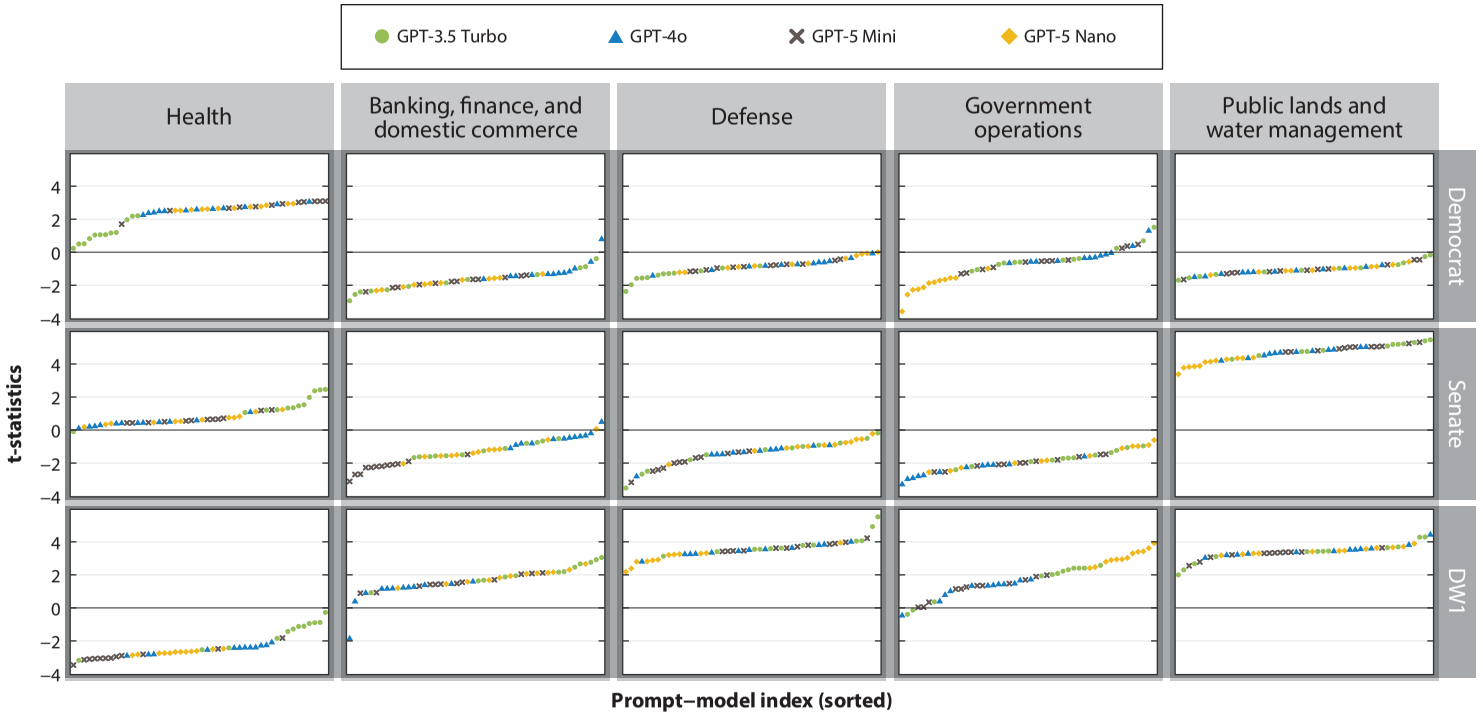

Figure 2

Variation in t-statistics across large language models and prompting strategies on Congressional legislation. On 10,000 Congressional bills, we prompt GPT-3.5 Turbo, GPT-4o, GPT-5 Mini, and GPT-5 Nano to label each description according to 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 (a measure of ideological positioning based on roll-call votes) of the bill's sponsor. In each subplot, the t-statistics are sorted in ascending order for clarity. See Section 4.3 for discussion.

Table 2 Variation in point estimates across large language models and prompting strategies on Congressional bills

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, and domestic commerce	DW1	0.013	0.013	0.007	0.022	0.127
Banking, finance, and domestic commerce	Democrat	−0.010	−0.010	−0.015	−0.004	0.127
Banking, finance, and domestic commerce	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 and water management	DW1	0.027	0.027	0.020	0.030	0.238
Public lands and water management	Democrat	−0.006	−0.007	−0.010	−0.003	0.238
Public lands and water management	Senate	0.032	0.032	0.025	0.036	0.238

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 according to 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 $1\{V_r = v\}$. DW1 refers to a measure of ideological positioning based on roll-call votes. See Section 4.3 for discussion.

4.4. 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 debias the plug-in estimate based on the LLM labels $\widehat{V}_r$. The virtues of validation data have been understood for decades (Bound & Krueger 1991, Bound et al. 1994, Lee & Sepanski 1995). This approach has been recently revived in machine learning to handle the types of mismeasured covariates and outcomes produced by modern machine learning models (see, e.g., Wang et al. 2020, Angelopoulos et al. 2023, Egami et al. 2024, Carlson & Dell 2025).

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, suggest 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.

4.4.1. 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 $\widehat{V}_r = W_r'\beta + \widetilde{\epsilon}_r$. **Supplemental Section C.2** considers when the economic concept V_r is a covariate.

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

In the validation sample, the researcher can estimate the bias $\widehat{\lambda}_{\Delta|W}$ by forming $\Delta_r = \widehat{m}(r; \mathbf{t}) - V_r$ on the validation sample and regressing it on W_r. The validation sample can therefore be used for two purposes. First, it provides a target to optimize when selecting a model and prompt. For any choice of LLM m and prompt p, the researcher can estimate the bias of the plug-in regression $\widehat{\lambda}^{m,p}_{\Delta|W}$ associated with the labels $\widehat{V}^{m,p}_r$ and thereby select the combination that results in the smallest bias.

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

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

This is exceedingly simple to implement: It suffices to run two regressions—regressing $\widehat{m}(r; \mathbf{t})$ on W_r in the primary sample and regressing Δ_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 **Supplemental Section C.2**, the bias-corrected estimator has desirable theoretical properties. Consider a research context where the researcher randomly samples text pieces, allocating a fraction ρ_{p} to the primary sample and a fraction ρ_{v} to the validation sample. As the number of economically relevant text pieces grows large, the bias-corrected estimator $\widehat{\beta}^{\text{debiased}}$ is consistent for the target regression coefficient β^*. $\widehat{\beta}^{\text{debiased}}$ is asymptotically normal with limiting variance given by

$$\sigma_W^{-4}\left(\frac{1-\rho_{\text{p}}}{\rho_{\text{p}}}\sigma^2_{\widehat{V}W} + 2\sigma_{\widehat{V}W}\sigma_{\Delta W} + \frac{1-\rho_{\text{v}}}{\rho_{\text{v}}}\sigma^2_{\Delta W}\right), \tag{15.}$$

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

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 $\sigma_W^{-4}\left(\frac{1-\rho_{\text{v}}}{\rho_{\text{v}}}\sigma^2_{\widehat{V}W}\right)$. Comparing these expressions reveals that the bias-corrected estimator is more precise when

$$\frac{1-\rho_{\text{p}}}{\rho_{\text{p}}}\sigma^2_{\widehat{V}W} + 2\sigma_{\widehat{V}W}\sigma_{\Delta W} \leq \frac{1-\rho_{\text{v}}}{\rho_{\text{v}}}\left(\sigma^2_{\widehat{V}W} - \sigma^2_{\Delta W}\right). \tag{16.}$$

This comparison depends on the relative standard deviation of the existing measurement V_r and the LLM's error Δ_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—for example, by Angelopoulos et al. (2023) and in 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: using 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 provide another critical benefit: They address 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 the 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 θ defined using the researcher's existing measurement.

4.4.2. 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 according to its major policy topic area $V_r := f^*(r)$, describing whether the bill falls into one of 20 possible policy areas.[10] Given the widespread use of these labels, researchers are comfortable using these measurements in downstream analyses. Can an LLM automate the 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 LLM 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 $\widehat{\beta}^{\text{debiased}}$. We repeat these steps for 1,000 randomly sampled data sets. Across simulations, we calculate the average bias of the plug-in coefficient and the bias-corrected coefficient for the target regression β^* associated with regressing the existing label V_r on the linked variable W_r on all 10,000 bills. We repeat this exercise for each combination of bill topic V_r, linked covariate W_r, LLM m (GPT-3.5 Turbo, GPT-4o, GPT-5 Mini, or GPT-5 Nano), and prompting strategy p. **Supplemental Section 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** compare the average bias of the plug-in coefficient $\widehat{\beta}$ and the bias-corrected coefficient $\widehat{\beta}^{\text{debiased}}$ for the target regression β^* (normalized by their standard deviations) across possible combinations of bill topic V_r, linked covariate W_r, LLM m, and prompting strategy p. For most regression specifications and pairs of LLM and prompting strategy, the simple plug-in regression suffers from substantial biases. 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 β^*. The nominal 95%

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

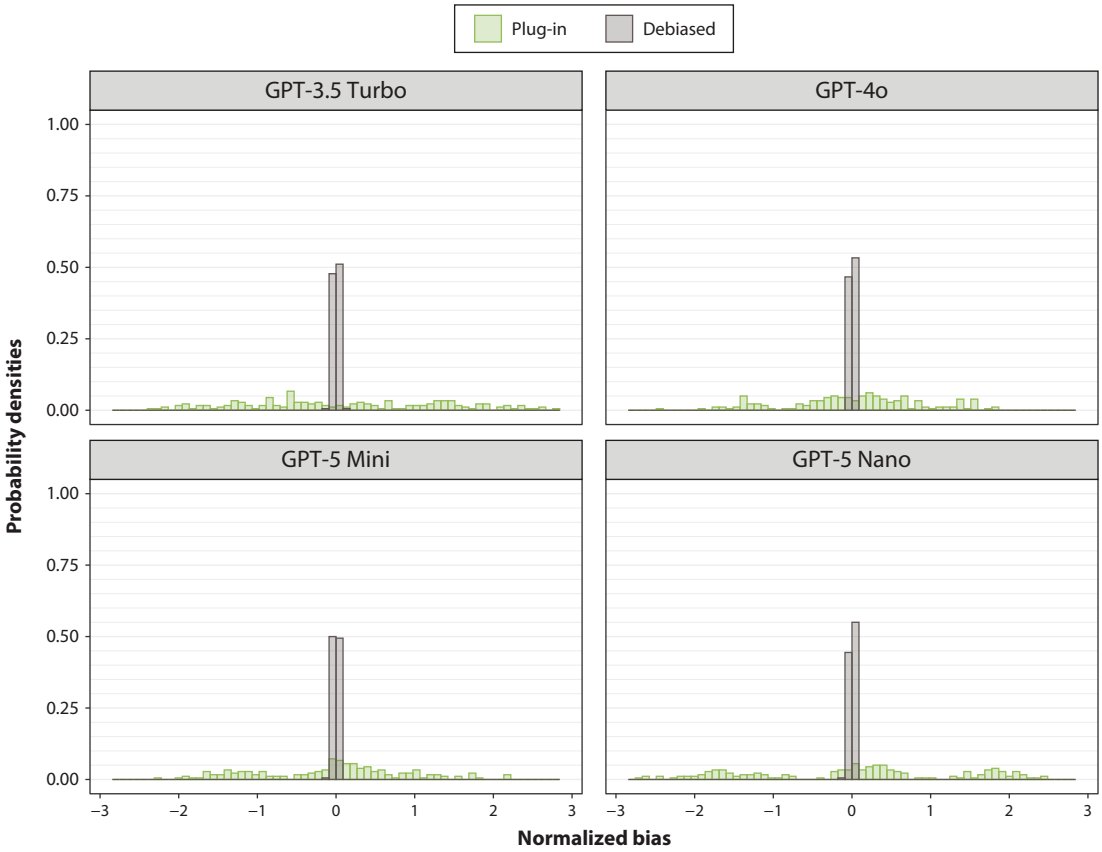

Figure 3

Normalized bias of the plug-in regression and bias-corrected regression across Monte Carlo simulations based on Congressional legislation. The normalized bias reports the average bias of the plug-in regression coefficient $\widehat{\beta}$ and the debiased coefficient $\widehat{\beta}^{\text{debiased}}$ 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 $\widehat{\beta}^{\text{debiased}}$ based on a 5% validation sample. See Section 4.4 for discussion.

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 following 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 β^*. **Figure 4** plots the resulting distribution across all choices of bill topic V_r, covariate W_r, and pair of LLM 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

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

GPT-3.5 Turbo	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

GPT-4o	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

GPT-5 Mini	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

GPT-5 Nano	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

The normalized bias reports the average bias of the plug-in regression coefficient $\widehat{\beta}$ and the debiased coefficient $\widehat{\beta}^{\text{debiased}}$ 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 $\widehat{\beta}^{\text{debiased}}$ covers 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 $\widehat{\beta}^{\text{debiased}}$ using a 5% validation sample. Results are averaged over 1,000 simulations. See Section 4.4 for discussion.

cost of data collection and improve statistical precision while preserving the familiar econometric guarantees we desire.

4.5. Estimation without Validation Data

When using LLMs in estimation problems, validation data enable 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.

4.5.1. 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, the researcher 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 $\widehat{m}(r; \mathbf{t})$, 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.

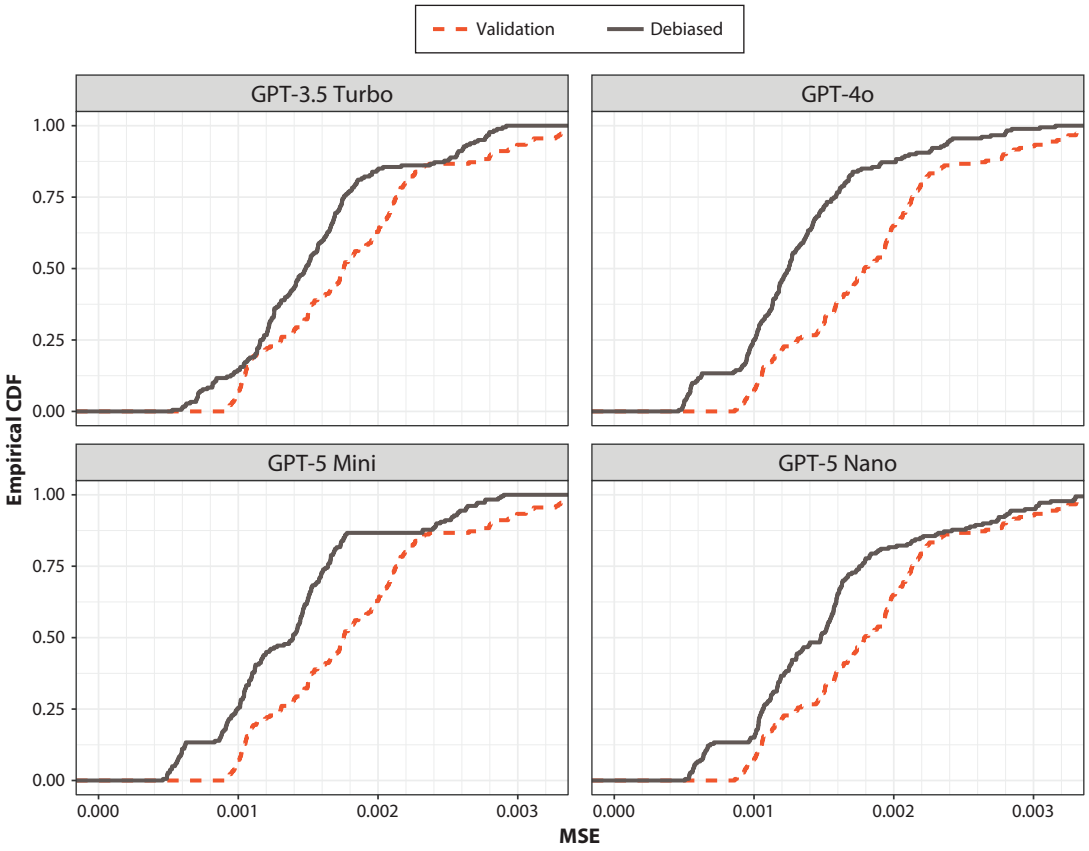

Figure 4

Cumulative distribution function (CDF) of mean square error (MSE) for the bias-corrected estimator against validation-sample-only estimator. 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 MSE of $\widehat{\beta}^{\text{debiased}}$ and $\widehat{\beta}^{*}$ for the target regression, and we average the results over 1,000 simulations. We summarize the distribution of average MSE across regression specifications, choice of large language model, and prompting strategies. See Section 4.4 for discussion.

Second, acknowledging that LLM outputs are imperfect, the researcher might write down a statistical model of errors $\Delta_r = \widehat{m}(r; \mathbf{t}) - f^*(r)$, just as we would in the measurement error literature. Consider a stylized model: For a given LLM m, the 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 data sets. 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 a setting like this in which 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: investing effort and collecting a small validation sample.

4.5.2. 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.

However, 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 4.3, 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 a negative one, 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 the economic concept we are studying is. 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 et al. 2024, Mei et al. 2024), marketing (e.g., Brand et al. 2023), finance (e.g., Bybee 2024), political science (e.g., Argyle et al. 2023), and computer science (e.g., Aher et al. 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,

they would calculate downstream parameter estimates (Equation 5). However, collecting human responses is costly, so the researcher instead substitutes LLM responses $\widehat{m}(r; \mathbf{t}) = \widehat{V}_r$ and reports the plug-in parameter estimate (Equation 6).

Example 9 (Testing anomalies). To test violations of risky choice models, behavioral economists construct anomalies—lottery menus that highlight flaws in the economic model, such as the Allais Paradox (Allais 1953) or the Kahneman-Tversky choice experiments (Kahneman & Tversky 1979). Could LLMs simulate human choices $\widehat{m}(r; \mathbf{t})$ on new anomalies?

Example 10 (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 data set. Could LLMs simulate the Choices13K data set?

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 et al. 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 the LLM training corpora, the key question is whether LLMs can simulate behavior on entirely new experimental designs and not merely reproduce memorized results (Manning & Horton 2025).

Viewing human subject simulation as an estimation problem also implies a practical fix: collecting 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 works, such as those by Broska et al. (2025), Krsteski et al. (2025), and 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 & Ross 2024, Zhou et al. 2024), suggesting instrumental variables (Han 2025), proposing research ideas in natural language processing (Si et al. 2024), and generating interpretable hypotheses that summarize estimated relationships between variables and text (Modarressi et al. 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 $\widehat{m}(r; \mathbf{t}) = \widehat{Y}_r$ 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 $\frac{1}{N}\sum_{r\in\mathcal{R}} D_r \ell(\widehat{m}(r; \mathbf{t}))$—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 a 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 has been advanced by Fudenberg & Liang (2019), Agrawal et al. (2024), Ludwig & Mullainathan (2024), Mullainathan & Rambachan (2024), and Mullainathan & Rambachan (2025). Further work is needed to clarify our inferential goals in these settings—what constitutes a good hypothesis and how we should evaluate LLM performance in generating it.

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.

6.1. 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 $(1/N)\sum_{r\in R} D_r \ell(Y_r, \hat{m}(r;\mathbf{t}))$. 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 $\hat{V}_r := \hat{m}(r;\mathbf{t})$ as a substitute.

6.2. Ensure No Training Leakage in Prediction Problems

Valid conclusions in prediction problems require one condition: no training leakage between the LLM's training data and the researcher's data set (Section 3.2).

The absence of training leakage is jointly determined by the prediction question and the model choice. Researchers must be explicit about three key elements. The first is their target population: What collection of text pieces do they ultimately want to make predictions on? The second is their sampling procedure: How do they select evaluation samples from this population? And the third is the LLM's training provenance: How does this sampling procedure relate to plausible training corpora used by the LLM? In Section 3.4, 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 are undisclosed and potentially continuously updated. Finally, prompt engineering strategies (e.g., "ignore information after date τ") are not reliable solutions to training leakage, as our evidence demonstrates.

6.3. Collect Validation Data for Estimation Problems

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 Sections 2.3 and 4.2).

The solution is straightforward: on a random sample of text pieces, applying 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 4.4 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 can 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 & Imbens 2017, Wager & Athey 2018). We use unstructured data, such as satellite images and digital traces, to infer outcomes at high frequencies and granular scales (e.g., Blumenstock et al. 2015, Donaldson & Storeygard 2016, Rambachan et al. 2024). We tackle prediction policy problems (Kleinberg et al. 2015, 2018; Mullainathan 2025) and develop algorithms for hypothesis generation (Fudenberg & Liang 2019, Ludwig & Mullainathan 2024, Mullainathan & 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 LLMs, 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.

DISCLOSURE STATEMENT

The authors are not aware of any affiliations, memberships, funding, or financial holdings that might be perceived as affecting the objectivity of this review.

ACKNOWLEDGMENTS

This article 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.

LITERATURE CITED

Abadie A, Athey S, Imbens GW, Wooldridge JM. 2020. Sampling-based versus design-based uncertainty in regression analysis. *Econometrica* 88(1):265–96

Adler ES, Wilkerson J. 2020. Congressional Bills Project, NSF 00880066 and 00880061. *Congressional Bills*, accessed July 5, 2024

Aenlle M. 2020. *Daily Financial News for 6000+ Stocks*. Data Set, accessed Aug. 1, 2024

Agrawal A, McHale J, Oettl A. 2024. Artificial intelligence and scientific discovery: a model of prioritized search. *Res. Policy* 53(5):104989

Aher G, Arriaga RI, Kalai AT. 2023. Using large language models to simulate multiple humans and replicate human subject studies. In *ICML '23: Proceedings of the 40th International Conference on Machine Learning*. ACM

Allais M. 1953. Le comportement de l'homme rationnel devant le risque: critique des postulats et axiomes de l'école américaine. *Econometrica* 21:503–46

Angelopoulos AN, Bates S, Fannjiang C, Jordan MI, Zrnic T. 2023. Prediction-powered inference. *Science* 382(6671):669–74

Argyle LP, Busby EC, Fulda N, Gubler JR, Rytting C, Wingate D. 2023. Out of one, many: using language models to simulate human samples. *Political Anal.* 31(3):337–51

Ash E, Hansen S. 2023. Text algorithms in economics. *Annu. Rev. Econ.* 15:659–88

Ash E, Hansen S, Muvdi Y. 2024. Large language models in economics. Discuss. Pap. 19479, Centre for Economic Policy Research

Athey S, Imbens GW. 2017. The econometrics of randomized experiments. In *Handbook of Economic Field Experiments*, Vol. 1, ed. AV Banerjee, E Duflo. Elsevier

Balloccu S, Schmidtová P, Lango M, Dusek O. 2024. Leak, cheat, repeat: data contamination and evaluation malpractices in closed-source LLMs. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics*, Vol. 1: *Long Papers*, ed. Y Graham, M Purver. Association for Computational Linguistics

Barrie C, Palmer A, Spirling A. 2024. Replication for language models problems, principles, and best practice for political science. Work. Pap., New York University

Batista R, Ross J. 2024. *Words that work: Using language to generate hypotheses.* Unpublished manuscript

Battaglia L, Christensen T, Hansen S, Sacher S. 2024. Inference for regression with variables generated by ai or machine learning. Preprint, arXiv:2402.15585 [econ.EM]

Berglund L, Tong M, Kaufmann M, Balesni M, Stickland AC, et al. 2024. The reversal curse: LLMs trained on "A is B" fail to learn "B is A." Preprint, arXiv:2309.12288v4 [cs.CL]

Blumenstock J, Cadamuro G, On R. 2015. Predicting poverty and wealth from mobile phone metadata. *Science* 350(6264):1073–76

Boelaert J, Coavoux S, Ollion E, Petev ID, Präg P. 2024. How do generative language models answer opinion polls? Preprint, SocArXiv. **https://osf.io/preprints/socarxiv/r2pnb**

Bound J, Brown C, Duncan GJ, Rodgers WL. 1994. Evidence on the validity of cross-sectional and longitudinal labor market data. *J. Labor Econ.* 12(3):345–68

Bound J, Krueger AB. 1991. The extent of measurement error in longitudinal earnings data: Do two wrongs make a right? *J. Labor Econ.* 9(1):1–24

Brand J, Israeli A, Ngwe D. 2023. Using LLMs for market research. Work. Pap. 23-062, Harvard Business School Marketing Unit

Broska D, Howes M, van Loon A. 2025. The mixed subjects design: treating large language models as potentially informative observations. *Sociol. Methods Res.* 54(3):1074-109

Bybee L. 2024. The ghost in the machine: generating beliefs with large language models. Work. Pap., University of Chicago Booth School of Business

Carlson J, Dell M. 2025. A unifying framework for robust and efficient inference with unstructured data. Preprint, arXiv:2505.00282 [econ.EM]

Chang Y, Wang X, Wang J, Wu Y, Yang L, et al. 2024. A survey on evaluation of large language models. *ACM Trans. Intell. Syst. Technol.* 15(3):39

Chen B, Zhang Z, Langrené N, Zhu S. 2024. Unleashing the potential of prompt engineering in large language models: a comprehensive review. Preprint, arXiv:2310.14735v3

Chen X, Hong H, Tamer E. 2005. Measurement error models with auxiliary data. *Rev. Econ. Stud.* 72(2):343–66

Cheng J, Marone M, Weller O, Lawrie D, Khashabi D, Durme BV. 2024. Dated data: tracing knowledge cutoffs in large language models. Preprint, arXiv:2403.12958 [cs.CL]

Cobbe K, Kosaraju V, Bavarian M, Chen M, Jun H, et al. 2021. Training verifiers to solve math word problems. Preprint, arXiv:2110.14168 [cs.LG]

Cui Z, Li N, Zhou H. 2024. Can AI replace human subjects? A large-scale replication of psychological experiments with LLMs. Preprint, arXiv:2409.00128 [cs.CL]

Dell M. 2025. Deep learning for economists. *J. Econ. Lit.* 63(1):5–58

Dell'Acqua F, McFowland E III, Mollick ER, Lifshitz-Assaf H, Kellogg K, et al. 2023. Navigating the jagged technological frontier: field experimental evidence of the effects of ai on knowledge worker productivity and quality. Work. Pap. 24-013, The Wharton School of the University of Pennsylvania

Donaldson D, Storeygard A. 2016. The view from above: applications of satellite data in economics. *J. Econ. Perspect.* 30(4):171–98

Donoho D. 2024. Data science at the singularity. *Harv. Data Sci. Rev.* 6(1). **https://doi.org/10.1162/99608f92.b91339ef**

Dreyfuss B, Raux R. 2025. Human learning about AI. Preprint, arXiv:2406.05408 [econ.GN]

Egami N, Hinck M, Stewart BM, Wei H. 2024. Using imperfect surrogates for downstream inference: design-based supervised learning for social science applications of large language models. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*. Curran Associates

Eloundou T, Manning S, Mishkin P, Rock D. 2024. GPTs are GPTs: labor market impact potential of LLMs. *Science* 384(6702):1306–8

Engelberg J, Manela A, Mullins W, Vulicevic L. 2025. *Entity neutering.* Unpublished manuscript

Erev I, Eyal E, Plonsky O, Cohen D, Cohen O. 2017. From anomalies to forecasts: toward a descriptive model of decisions under risk, under ambiguity, and from experience. *Psychol. Rev.* 124(4):369–409

Fudenberg D, Liang A. 2019. Predicting and understanding initial play. *Am. Econ. Rev.* 109(12):4112–41

Fudenberg D, Liang A, Kleinberg J, Mullainathan S. 2022. Measuring the completeness of economic models. *J. Political Econ.* 130(4):956–90

Glasserman P, Lin C. 2023. Assessing look-ahead bias in stock return predictions generated by GPT sentiment analysis. Preprint, arXiv:2309.17322 [q-fin.GN]

Golchin S, Surdeanu M. 2024. Time travel in LLMs: tracing data contamination in large language models. Preprint, arXiv:2308.08493v3 [cs.CL]

Grattafiori A, Dubey A, Jauhri A, Pandey A, Kadian A, et al. 2024. The Llama 3 herd of models. Preprint, arXiv:2407.21783 [cs.AI]

Han S. 2025. Mining causality: AI-assisted search for instrumental variables. Preprint, arXiv:2409.14202v3 [econ.EM]

Hansen S, Lambert PJ, Bloom N, Davis SJ, Sadun R, Taska B. 2023. Remote work across jobs, companies, and space. NBER Work. Pap. 31007

He S, Lv L, Manela A, Wu J. 2025. Chronologically consistent large language models. Preprint, arXiv:2502.21206 [q-fin.GN]

Hendrycks D, Burns C, Basart S, Zou A, Mazeika M, et al. 2021. Measuring massive multitask language understanding. Preprint, arXiv:2009.03300 [cs.CY]

Horton JJ. 2023. Large language models as simulated economic agents: What can we learn from homo silicus? NBER Work. Pap. 31122

Jacovi A, Caciularu A, Goldman O, Goldberg Y. 2023. Stop uploading test data in plain text: practical strategies for mitigating data contamination by evaluation benchmarks. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, ed. H Bouamor, J Pino, K Bali. Association for Computational Linguistics

Jimenez CE, Yang J, Wettig A, Yao S, Pei K, et al. 2024. SWE-bench: Can language models resolve real-world github issues? Preprint, arXiv:2310.06770 [cs.CL]

Jones BD, Baumgartner FR, Theriault SM, Epp DA, Lee C, Sullivan ME. 2023. *Policy agendas project: Codebook*. Data Set, accessed March 1, 2026. **https://minio.la.utexas.edu/compagendas/codebookfiles/CodebookAP019.pdf**

jylin04, Jack S, Karvonen A, Can. 2024. OthelloGPT learned a bag of heuristics. *LessWrong*, July 2

Kahneman D, Tversky A. 1979. Prospect theory: an analysis of decision under risk. *Econometrica* 47(2):363–91

Kim E, Garg A, Peng K, Garg N. 2025. Correlated errors in large language models. Preprint, arXiv:2506.07962 [cs.CL]

Kleinberg J, Lakkaraju H, Leskovec J, Ludwig J, Mullainathan S. 2018. Human decisions and machine predictions. *Q. J. Econ.* 133(1):237–93

Kleinberg J, Ludwig J, Mullainathan S, Obermeyer Z. 2015. Prediction policy problems. *Am. Econ. Rev.* 105(5):491–95

Korinek A. 2023. Generative AI for economic research: use cases and implications for economists. *J. Econ. Lit.* 61(4):1281–317

Korinek A. 2024. Generative AI for economic research: LLMs learn to collaborate and reason. NBER Work. Pap. 33198

Krsteski S, Russo G, Chang S, West R, Gligorić K. 2025. Valid survey simulations with limited human data: the roles of prompting, fine-tuning, and rectification. Preprint, arXiv:2510.11408 [cs.CL]

Lee L, Sepanski JH. 1995. Estimation of linear and nonlinear errors-in-variables models using validation data. *J. Am. Stat. Assoc.* 90(429):130–40

Li K, Hopkins AK, Bau D, Viégas F, Pfister H, Wattenberg M. 2024. Emergent world representations: exploring a sequence model trained on a synthetic task. Preprint, arXiv:2210.13382 [cs.LG]

Li X, Ding P. 2017. General forms of finite population central limit theorems with applications to causal inference. *J. Am. Stat. Assoc.* 112(520):1759–69

Liu P, Yuan W, Fu J, Jiang Z, Hayashi H, Neubig G. 2023. Pre-train, prompt, and predict: a systematic survey of prompting methods in natural language processing. *ACM Comput. Surv.* 55(9):195

LM Contamination Index. 2024. *LM Contamination Index*. Database, accessed Dec. 5, 2024. **https://hitz-zentroa.github.io/lm-contamination**

Lopez-Lira A, Tang Y. 2025. Can ChatGPT forecast stock price movements? Return predictability and large language models. Preprint, arXiv:2304.07619v6 [q-fin.ST]

Lopez-Lira A, Tang Y, Zhu M. 2025. The memorization problem: Can we trust LLMs' economic forecasts? Preprint, arXiv:2504.14765 [q-fin.GN]

Ludwig J, Mullainathan S. 2024. Machine learning as a tool for hypothesis generation. *Q. J. Econ.* 139(2):751–827

Mancoridis M, Weeks B, Vafa K, Mullainathan S. 2025. Potemkin understanding in large language models. Preprint, arXiv:2506.21521 [cs.CL]

Manning BS, Horton JJ. 2025. General social agents. Preprint, arXiv:2508.17407 [econ.GL]

Manning BS, Zhu K, Horton JJ. 2024. Automated social science: language models as scientist and subjects. Preprint, arXiv:2404.11794 [econ.GN]

McCoy RT, Yao S, Friedman D, Hardy MD, Griffiths TL. 2024. Embers of autoregression: understanding large language models through the problem they are trained to solve. *PNAS* 121(41):e2322420121

Mei Q, Xie Y, Yuan W, Jackson MO. 2024. A Turing test of whether AI chatbots are behaviorally similar to humans. *PNAS* 121(9):e2313925121

Minaee S, Mikolov T, Nikzad N, Chenaghlu M, Socher R, et al. 2025. Large language models: a survey. Preprint, arXiv:2402.06196 [cs.CL]

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

Modarressi I, Spiess J, Venugopal A. 2025. Causal inference on outcomes learned from text. Preprint, arXiv:2503.00725 [econ.EM]

Movva R, Peng K, Garg N, Kleinberg J, Pierson E. 2025. Sparse autoencoders for hypothesis generation. Preprint, arXiv:2502.04382 [cs.CL]

Mullainathan S. 2025. Economics in the age of algorithms. *AEA Pap. Proc.* 115:1–23

Mullainathan S, Rambachan A. 2024. From predictive algorithms to automatic generation of anomalies. NBER Work. Pap. 32422

Mullainathan S, Rambachan A. 2025. Science in the age of algorithms. In *The Economics of Transformative AI*, ed. AK Agrawal, E Brynjolfsson, A Korinek. University of Chicago Press

Nanda N, Lee A, Wattenberg M. 2023. Emergent linear representations in world models of self-supervised sequence models. Preprint, arXiv:2309.00941 [cs.LG]

Nezhurina M, Cipolina-Kun L, Cherti M, Jitsev J. 2025. Alice in wonderland: simple tasks showing complete reasoning breakdown in state-of-the-art large language models. Preprint, arXiv:2406.02061v5 [cs.LG]

Nikankin Y, Reusch A, Mueller A, Belinkov Y. 2025. Arithmetic without algorithms: Language models solve math with a bag of heuristics. Preprint, arXiv:2410.21272 [cs.CL]

Park JS, O'Brien JC, Cai CJ, Morris MR, Liang P, Bernstein MS. 2023. Generative agents: Interactive simulacra of human behavior. Preprint, arXiv:2304.03442 [cs.HC]

Park JS, Zou CQ, Shaw A, Hill BM, Cai C, et al. 2024. Generative agent simulations of 1,000 people. Preprint, arXiv:2411.10109 [cs.AI]

Peterson JC, Bourgin DD, Agrawal M, Reichman D, Griffiths TL. 2021. Using large-scale experiments and machine learning to discover theories of human decision-making. *Science* 372(6547):1209–14

Raman N, Lundy T, Amouyal S, Levine Y, Leyton-Brown K, Tennenholtz M. 2024. STEER: assessing the economic rationality of large language models. Preprint, arXiv:2402.09552 [cs.CL]

Rambachan A, Roth J. 2024. Design-based uncertainty for quasi-experiments. Preprint, arXiv:2008.00602v6 [econ.EM]

Rambachan A, Singh R, Viviano D. 2024. Program evaluation with remotely sensed outcomes. Preprint, arXiv:2411.10959 [econ.EM]

Ravaut M, Ding B, Jiao F, Chen H, Li X, et al. 2025. How much are large language models contaminated? A comprehensive survey and the *LLMSanitize* library. Preprint, arXiv:2404.00699v5 [cs.CL]

Sainz O, Campos J, Garca-Ferrero I, Etxaniz J, de Lacalle OL, Agirre E. 2023. NLP evaluation in trouble: on the need to measure LLM data contamination for each benchmark. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, ed. H Bouamor, J Pino, K Bali. Association for Computational Linguistics

Santurkar S, Durmus E, Ladhak F, Lee C, Liang P, Hashimoto T. 2023. Whose opinions do language models reflect? In *Proceedings of the 40th International Conference on Machine Learning*. JMLR

Sarkar S. 2024. *StoriesLM: a family of language models with time-indexed training data.* Unpublished manuscript

Sarkar SK, Vafa K. 2024. *Lookahead bias in pretrained language models.* Unpublished manuscript

Schennach SM. 2016. Recent advances in the measurement error literature. *Annu. Rev. Econ.* 8:341–77

Si C, Yang D, Hashimoto T. 2024. Can LLMs generate novel research ideas? A large-scale human study with 100+ NLP researchers. Preprint, arXiv:2409.04109 [cs.CL]

Srivastava A, Rastogi A, Rao A, Shoeb AAM, Abid A, et al. 2022. Beyond the imitation game: quantifying and extrapolating the capabilities of language models. Preprint, arXiv:2206.04615 [cs.CL]

Touvron H, Martin L, Stone K, Albert P, Almahairi A, et al. 2023. Llama 2: open foundation and fine-tuned chat models. Preprint, arXiv:2307.09288 [cs.CL]

Vafa K, Chang PG, Rambachan A, Mullainathan S. 2025. What has a foundation model found? Using inductive bias to probe for world models. Preprint, arXiv:2507.06952 [cs.LG]

Vafa K, Chen JY, Kleinberg J, Mullainathan S, Rambachan A. 2024a. Evaluating the world model implicit in a generative model. Preprint, arXiv:2406.03689 [cs.CL]

Vafa K, Rambachan A, Mullainathan S. 2024b. Do large language models generalize the way people expect? A benchmark for evaluation. Preprint, arXiv:2406.01382 [cs.CL]

Wager S, Athey S. 2018. Estimation and inference of heterogeneous treatment effects using random forests. *J. Am. Stat. Assoc.* 113(523):1228–42

Wang S, McCormick TH, Leek JT. 2020. Methods for correcting inference based on outcomes predicted by machine learning. *PNAS* 117(48):30266–75

Wharton Research Data Services. 2024. *Beta Suite by WRDS*. Database, Wharton University of Pennsylvania, accessed Aug. 1, 2024. **https://wrds-www.wharton.upenn.edu/pages/grid-items/beta-suite-wrds**

Wei J, Wang X, Schuurmans D, Bosma M, Ichter B, et al. 2024. Chain-of-thought prompting elicits reasoning in large language models. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*. Curran Associates

White J, Fu Q, Hays S, Sandborn M, Olea C, et al. 2023. A prompt pattern catalog to enhance prompt engineering with ChatGPT. Preprint, arXiv:2302.11382 [cs.SE]

Wilkerson J, Adler ES, Jones BD, Baumgartner FR, Freedman G, et al. 2023. Policy agendas project: Congressional bills. *Comparative Agendas Project*, accessed July 5, 2024. **https://www.comparativeagendas.net/project/us/datasets**

Wongchamcharoen PK, Glasserman P. 2025. Do large language models (LLMs) understand chronology? Preprint, arXiv:2511.14214 [cs.AI]

Wu Z, Qiu L, Ross A, Akyürek E, Chen B, et al. 2024. Reasoning or reciting? Exploring the capabilities and limitations of language models through counterfactual tasks. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Vol. 1: *Long Papers*, ed. K Duh, H Gomez, S Bethard. Association for Computational Linguistics

Xu R. 2020. Potential outcomes and finite-population inference for m-estimators. *Econometr. J.* 24(1):162–76

Zhang B, Li J, Hortaçsu A, Ye X, Chernozhukov V, et al. 2025. Agentic economic modeling. Preprint, arXiv:2510.25743 [econ.EM]

Zhao WX, Zhou K, Li J, Tang T, Wang X, et al. 2025. A survey of large language models. Preprint, arXiv:2303.18223 [cs.CL]

Zhou Y, Liu H, Srivastava T, Mei H, Tan C. 2024. Hypothesis generation with large language models. Preprint, arXiv:2404.04326 [cs.AI]

Zong Y, Yu T, Chavhan R, Zhao B, Hospedales T. 2024. Fool your (vision and) language model with embarrassingly simple permutations. *Proc. Mach. Learn. Res.* 235:62892–913