Amundi
Investment Solutions

Trust must be earned

Investment Institute

WORKING PAPER #187 | MARCH 2026

Out of the Black Box: Uncertainty Quantification for LLMs via Conditional Probabilities

ESSEC
BUSINESS SCHOOL

ESSEC-Amundi Chair on Asset & Risk Management

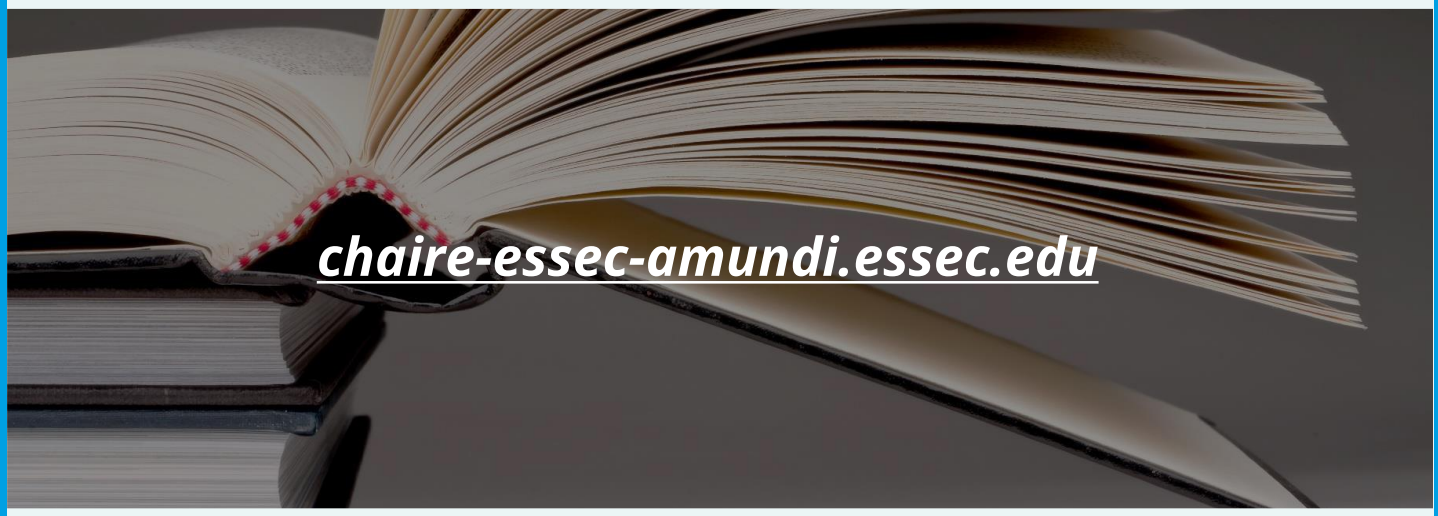
WORKING PAPER #187 | **MARCH 2026**

The Chair was created in 2016 with the objective to encourage and stimulate scientific cooperation between the research teams of ESSEC and Amundi on topics linked to asset and risk management with a special emphasis on responsible (ESG) investment. The Chair sponsors research projects, organizes an annual workshop and a series of webinars on asset management and ESG issues, prepares a series of research letters on ESG related issues for the Amundi research team and management and offers a series of research seminars for Amundi collaborators. Finally, it sponsors several scholarships for ESSEC Finance PhD students working in areas closely related to the Chair. As part of our collaboration with ESSEC, we are pleased to share this working paper authored by ESSEC researchers. This is an ad hoc release, outside our usual publications, reflecting the partnership agreement between our institutions. The papers cover key topics in machine learning and portfolio allocation.

Marie Brière
Head of Investor Intelligence & Academic Partnership (Amundi Investment Institute)
Co-head of the Chair

Elise Gourier
Associate Professor in Finance (ESSEC Business School)
Co-head of the Chair

Sofia Brito-Ramos
Professor of Finance (ESSEC Business School)
Co-head of the Chair



ESSEC BUSINESS SCHOOL

Amundi Investment Solutions

Trust must be earned

Out of the Black Box: Uncertainty Quantification for LLMs via Conditional Probabilities*

Hui Chen† Antoine Didisheim‡ Luciano Somoza§

February 1, 2026

Abstract

Autoregressive LLMs generate text by sampling from estimated probability distributions over the next token, conditional on preceding context. We leverage these conditional probabilities to construct an entropy-based measure of prediction uncertainty, which we term *inner confidence*. Predictions with higher inner confidence are systematically more accurate. To assess the measure's economic relevance, we use an LLM to predict daily stock returns based on firm-specific news and evaluate the performance of long-short portfolios built on these predictions. Conditioning on inner confidence significantly improves the performance: high-confidence predictions achieve Sharpe ratios roughly 20% higher than the unconditional benchmark, while low-confidence predictions yield no excess returns. By contrast, LLM's self-declared confidence exhibits strong biases and delivers no comparable gains.

*We thank Rohit Allena (discussant), Leland Bybee (discussant), Gerard Hoberg, Dongyihai Peng, Gordon Phillips, Simon Scheidegger, Hanqing Tian, Nitin Yadav, Terry Zhang, and participants at the AFA 2026, WFA 2025, NTU AI for Finance Summer School, and FIRN-UQ Asset Management Meeting for their valuable comments.

†MIT Sloan and NBER. Email: huichen@mit.edu

‡University of Melbourne. Email: antoine.didisheim@unimelb.edu.au

§ESSEC Business School. Email: somoza@essec.edu

> All [LLMs] are doing, after all, is predicting the next word needed to string out a response that will statistically satisfy a prompt.
>
> Geoffrey Hinton

Introduction

Recent breakthroughs in artificial intelligence, especially Large Language Models (LLMs), have drastically accelerated their adoption in financial markets and research.[1] As these tools are highly effective at simulating human interactions, their most natural use involves prompting and analyzing the generated text. For many applications, this approach is intuitive and sufficient. In this paper, however, we argue that for research in financial economics, reliance solely on surface-level outputs is often suboptimal. Indeed, generated text is the final product of two distinct components: a *neural language model* that produces a probability distribution over the next token given a sequence of prior tokens[2] (i.e., *inner probabilities*), and a *decoding strategy* that selects the next token (stochastically or deterministically) from the conditional distribution. Hence, focusing exclusively on the textual output presents two problems: first, the text is only as a point estimate of a much more complex and rich distribution, meaning that valuable information is discarded. Second, as the conditional distribution of any subsequent token depends heavily on both the prompt and previously generated tokens, this inherent inter-dependency can create systematic biases. To address these limitations, we argue that researchers should look beyond the generated text and directly analyze the model's underlying probability vectors.

We propose an entropy-based framework that leverages these inner probabilities for measuring the uncertainty of LLMs's outputs. Our approach is designed to be interpretable, semantics-aware, and feasible for large-scale empirical applications. While this framework generalizes to multi-token uncertainty measurement, we argue that in the majority of financial applications, it is preferable to restrict the LLM's output to a fixed set of discrete answers (e.g., "Yes" or "No"; "A", "B", or "C"). This constraint cheaply eliminates the issue of synonyms inflating entropy measures (where probability mass is dispersed across semantically identical but lexically distinct tokens) and allows for the construction of a highly interpretable confidence metric. Indeed, in the binary classification setting, we can measure *inner confidence* with the distance between the model's inner probabilities and a uniform

[1] See Lopez-Lira and Tang (2023); Bybee (2023); Cheng, Lin, and Zhao (2024); Babina, Fedyk, He, and Hodson (2024); Li, Shi, Xia, and Yang (2025); Acikalin, Caskurlu, Hoberg, and Phillips (2025); He, Lv, Manela, and Wu (2025), among others.

[2] A token can represent a single character such as "A" or "3", a fragment of a word like *ing* or *bel*, or even an entire word. Tokens serve as the basic units that AI models use to encode and processed text.

50/50 distribution. We show that this simple and intuitive distance statistic is strictly monotonic to the Shannon entropy of the predictive distribution.

To test whether this measure of inner confidence carries economic meaning, we utilize a sample of 100,000 intra-day financial news items regarding individual firms alongside their associated contemporaneous stock returns. We prompt the LLM to classify each news item as positive or negative for the company's stock price, utilizing OpenAI's API to obtain both the generated token and the associated inner probabilities from GPT-4o. Accessing these probabilities is straightforward; most private and open-source LLMs provide direct access to the true inner probabilities at no extra cost, as they are standard outputs of the model. We use those probability to measure inner-confidence, and then measure the LLM's classification accuracy using the sign of the same-day intraday return as the ground truth.

We observe a clear relationship between the inner probabilities-based model confidence, referred to as *inner confidence* for brevity, and classification accuracy. Specifically, for predictions in the highest decile of inner confidence, the LLM achieves an accuracy rate of approximately 64.5%, compared to a below-50% accuracy rate in the lower confidence quintiles. A logit regression where the dependent variable is a binary indicator of correct classification, shows that these differences are statistically significant. This relationship and its magnitude persist even when isolating positive versus negative news, based on realized returns. This finding suggests that when the LLM assigns high inner confidence to a classification, it is much more likely to align with the actual market reaction. This result highlights the economic importance of inner probabilities as indicators of predictive accuracy.

To assess economic significance, we explore the use of inner confidence to improve portfolio construction. We follow the framework of Lopez-Lira and Tang (2023), and use overnight news from Reuters to build long-short intra-day portfolios. Specifically, we prompt the LLM to obtain a signal (positive or negative) and use inner-probabilities to build a measure of confidence on that signal. Next, using a 1-year rolling window, we estimate the optimal level of confidence below which removing low-confidence news maximizes the Sharpe ratio. Any news below the threshold is used to build a low-confidence long-short portfolio, and any news above the confidence threshold is used to build a high-confidence long-short portfolio. The portfolio based on the full sample serves as the baseline. The optimal confidence threshold removes on average 18% of the sample, and translates into economically significant results. In line with Lopez-Lira and Tang (2023), the full-sample baseline strategy produces a Sharpe ratio of 4.24. Strikingly, the Sharpe ratio of the low-confidence portfolio, constructed from the bottom 18% of the sample, is not statistically different from zero. This finding indicates that inner-confidence can identify instances where LLM forecasts lack economically significant

predictive power. Furthermore, the high-confidence portfolio significantly outperforms the baseline, achieving a Sharpe ratio of 5.15. This high value is consistent with previous research using similar setups (see, e.g., Chen, Kelly, and Xiu, 2022) and can largely be explained by the high turnover characteristic of intra-day portfolios. Further analysis shows that the elevated Sharpe ratios are primarily driven by short positions in stocks with low analyst coverage, low market capitalization, and low liquidity. Regardless of the scale, these results indicate that, in an apple-to-apples comparison, incorporating inner probabilities as a measure of confidence can substantially enhance LLM-based portfolios.

As a natural benchmark for inner confidence, we also examine *declared confidence*, obtained by prompting the model to generate a value between 0 and 1 representing its confidence. In contrast to inner confidence, a high-confidence portfolio constructed using declared confidence fails to outperform the baseline portfolios.

While our previous results highlight the economic significance of inner confidence, it is important to emphasize that our confidence measure should not be interpreted literally as a probability. The inner probabilities produced by LLMs are often highly concentrated, frequently nearing values close to 0 and 1. Our experiments show that even extremely small differences in inner-probabilities (e.g., from 0.99, to 0.999) often carry meaningful economic information. Hence, the inner-confidence metric should be understood as an ordinal indicator of predictive conviction rather than a cardinal metric.

Next, we examine the factors that drive the model's confidence, in order to provide intuition about the circumstances under which the model is more likely to generate a confident prediction. First, we use forward-looking momentum and volatility as a proxy of the type of news, i.e., a positive news on average translates into high return in the following month, and a hard-to-interpret news translates into higher volatility. We use abnormal trading volume on the day of the news as a proxy for market disagreement, the VIX as a measure of overall market uncertainty, and the total number of news items–both market-wide and firm-specific–as proxies for information shocks. The results align with economic intuition. The model tends to be more confident when the subsequent price path is directional and volatility remains low. Conversely, news items linked to higher future volatility and abnormal volume are associated with lower confidence. Second, we examine the relationship between model confidence and news subject. We rely on the published word weights in Bybee, Kelly, Manela, and Xiu (2024) to assign one of the 180 defined news topics to each news in our sample. Next we use a Lasso regression model to predict the inner confidence decile.[3] We gradually increase the penalty term of our Lasso regression to select exactly ten most informative categories,

[3]As small differences in inner confidence can translate into high economic significance, we transform the confidence into deciles computed on the full sample

revealing that the model exhibits highest confidence when processing stories about lawsuits or product development, and lowest confidence for items on corporate governance or foreign markets. Repeating the estimation at the stock level, for the top 10 most covered firms in our sample, we find results in line with economic intuition and news coverage in our sample. For example, Disney's higher determinant of low confidence is news about *mergers and acquisition* while Tesla's news associated with lowest confidence mention *space program*. Third, we study our prompt-based sentiment classification in a Bayesian framework. The LLM can be interpreted as holding a prior belief about the firm, receiving a news item as signal, and producing a posterior–the inner probability. This framework offers a useful way to evaluate the model's economic rationality: by altering its prior and measuring the impact on inner confidence. We revisit our first exercise using the 10,000 news items employed to nowcast firm returns, but modify the prompt. In the first variation, we state that analysts agree (or disagree) on the firm's general outlook. This adjustment mimics a shift in the model's prior belief about future returns, reflecting a lower (or higher) variance in prior expectations. In the second prompt, we state that analysts agree (or disagree) on the interpretation of a specific news item. This approach corresponds to altering the model's belief about the signal itself, implying a lower (or higher) variance in the perceived informativeness of the news. We include a placebo condition using a third prompt that adds words commonly associated with high or low human confidence.[4] In line with a Bayesian interpretation, the first two types of prompt conditioning shift the model's internal probabilities in the expected direction. When analysts agree on a news item's interpretation, the model's confidence in its classification rises, reflecting a proper Bayesian update. In contrast, confidence-related words added at random have no effect.

Having established the economic relevance and determinants of inner probabilities, we next argue that inner confidence can inform the debate regarding LLMs in finance. A natural area to apply inner probabilities as an unblackboxing tool is the recent concern about look-ahead bias in LLMs, where model outputs unintentionally reflect future information.Levy (2024); Sarkar and Vafa (2024); Ludwig, Mullainathan, and Rambachan (2025); Engelberg, Manela, Mullins, and Vulicevic (2025). Indeed, since inner probabilities reflect the model's confidence, it is natural to ask whether they can also serve as a diagnostic tool to detect when such bias might be present. We follow the approach of Didisheim, Fraschini, and Somoza (2025); Lopez-Lira, Tang, and Zhu (2025) and prompt the LLM to recall realized returns for individual assets without any additional context. This allows us to isolate the look-ahead bias without the confusion effect of added context (Glasserman and Lin, 2023).

[4]Positive noise conditioning: "Confidence, clarity, conviction, precision, certainty." Negative noise conditioning: "Doubt, anxiety, mistrust, uncertainty, hesitation."

In this clean setting, we uncover a strong link between the model's inner confidence and its implicit memory: when the LLM "remembers" the return, its confidence is systematically and significantly higher. Next, we examine the average *inner-confidence* of the models when analyzing financial news in the one-year periods before and after the learning cutoff. Notably, we find no statistically significant increase in *inner-confidence* after the learning cutoff. This result is consistent with the findings of Glasserman and Lin (2023), who report that the look-ahead bias of LLMs in financial news analysis is smaller than the effects induced by model confusion.

Finally, we further investigate declared confidence, which is an intuitive alternative to inner confidence. The method consists in directly querying the LLM about its own confidence and has been previously used in the literature (see, e.g. Bybee, 2023). Unlike inner confidence, which reflects the transformer's actual "belief," declared confidence represents, at best, the model's self-assessed certainty.

Our main experiment indicates that declared confidence does not yield economically significant effects in the portfolio construction exercise. We further demonstrate that declared confidence is substantially biased by the decoding strategy. Under standard decoding, each token is sampled from the conditional distribution implied by the neural network, where the distribution of a given token depends on both the prompt and all previously generated tokens. Consequently, generating declared confidence after the prediction can introduce bias, because it depends not only on the prompt but also on the initially generated token.

To illustrate, consider a simple example. Using a prompt P, we ask an LLM whether a piece of news is (A) positive or (B) negative. After generating a single token (A/B), the prompt P then asks the model to generate a number between 0 and 1 representing its confidence. Suppose the model assigns equal probability to the two outcomes, $p(\text{“A”} \mid P) = p(\text{“B”} \mid P) = 0.5$. Such a situation should correspond to low confidence; more importantly, the confidence should be identical regardless of whether the model randomly selects "A" or "B." However, because the declared confidence token D is conditional on both the prompt and the previously generated token, this need not hold. In particular, it may be that $p(D \mid P, \text{“A”}) \neq p(D \mid P, \text{“B”})$.

We show that for *GPT4o* these probabilities are not only different, but systematically biased toward higher declared confidence following a positive prediction, i.e., $p(D \mid P, \text{“A”}) > p(D \mid P, \text{“B”})$. This finding is robust to the sign of the realized return and to the underlying inner probabilities. In contrast, *inner confidence* is mechanically unaffected by the choice of the first token, since the latter is determined by the inner probabilities rather than influencing them.

Related literature: Our paper is related to the literature that studies and assesses applications of LLMs in economic and financial research. This literature is recent but fast growing (see e.g., Korinek, 2023; Eisfeldt and Schubert, 2024). More specifically, we contribute to two strands: the one using LLMs for classification purposes and the one using LLMs for simulating agent behavior.

While machine learning methods, like LDA, have been widely used in research for years (see e.g., Bybee et al., 2024; Allena, 2025), applications of LLMs are relatively more recent. Notably, Chang, Dong, Martin, and Zhou (2023) use LLM to identify earnings calls sentiment, Krockenberger, Saunders, Steffen, and Verhoff (2024) for covenant violations, and Caragea, Chen, Cojoianu, Dobri, Glandt, and Mihaila (2020) for patent classifications. We contribute to this growing literature by showing that inner probabilities can be used as a measure of classification accuracy and they are stable and easy to interpret.

LLMs are also finding applications in simulating agent behavior and expectations. Bybee (2023) introduces a survey of economic expectations formed by querying an LLMs. He finds the resulting expectations closely match major existing surveys. In the same spirit, Fedyk, Kakhbod, Li, and Malmendier (2024) investigates whether AI models accurately capture investment preferences across different demographics, shedding light on the models' ability to replicate nuanced human decision-making processes in financial contexts. Ashokkumar, Hewitt, Ghezae, and Willer (2024) tests the predictive capabilities of LLMs concerning the outcomes of social science experiments, finding strong results. Overall, this strand of the literature finds that LLMs perform well in capturing aggregate human choices and beliefs. Conversely, Ludwig and Mullainathan (2024) exploit the difference between human cognition and LLMs. They propose a systematic procedure to generate novel hypotheses about human behavior, which uses the capacity of machine learning algorithms to notice patterns that might not be obvious to humans.

In this context, we address the problem of interpretability of LLMs, which is a central theme in recent research. Korinek (2023) discusses the implications of generative AI in economics, emphasizing the necessity for methods that illuminate the internal reasoning processes of LLMs to enhance their reliability and acceptance in academic research. The black box nature of these models presents challenges for researchers aiming to understand and trust their outputs, especially when using them to simulate agent behavior.

1 LLM and Inner Probabilities

In this section, we briefly review the large language models (LLMs) and introduce the notion of inner probabilities. Colloquially, the term "LLM" refers to the entire process that maps a textual prompt into a generated response. This process comprises two distinct components. The first component is a *neural language model* that can be viewed as a probabilistic system. It produces conditional probability distributions over the next token given a preceding context. The second is a *decoding strategy*, which specifies how these conditional distributions are queried and transformed into a realized sequence of tokens through deterministic or stochastic selection rules.

Existing applications of LLMs in economics and finance typically focus on the realized textual output. By contrast, the central premise of this paper is that the model's internal conditional probabilities contain economically meaningful information not captured by the generated text. By analyzing these probabilities directly, we can effectively reduce the opacity introduced by the decoding strategy. In particular, the inner probabilities provide a principled way for quantifying the uncertainty associated with LLM-generated predictions.

We now briefly review the two core components of a large language model: the transformer neural network and the decoding strategy.

The network: The neural network architectures underlying LLMs are exceptionally large, reflecting the scale implied by the first "L" in the acronym. In practice, most state-of-the-art LLMs adopt a decoder-only transformer architecture designed for autoregressive generation. Within this framework, transformer layers (Vaswani, Shazeer, Parmar, Uszkoreit, Jones, Gomez, Kaiser, and Polosukhin, 2017) compute context-dependent representations using masked self-attention, which directs attention to different parts of an input text sequence and enables the model to capture long-range semantic relationships across the input. For example, the model can infer whether the term "bank" refers to a financial institution or a riverbank based on surrounding context. As information propagates through successive layers, the network constructs increasingly abstract internal representations of the evolving context, enabling it to generate coherent and contextually appropriate text to follow the prompt, thus achieving the goal of completing sentences, answering questions, performing reasoning tasks, etc.

Although LLMs may differ in architectural details, they are fundamentally characterized by their training objective. Modern LLMs are trained using causal language modeling, in which the model learns to predict the next token in a sequence conditional on all preceding tokens (Radford et al., 2019; Brown et al., 2020). During pre-training, contiguous segments

of text are sampled from a large corpus and tokenized. At each position in the sequence, the model observes only the preceding tokens, with future tokens masked to enforce the causal structure. The network outputs a probability distribution over the full vocabulary for the next token, and model parameters are estimated by minimizing the cross-entropy loss between the predicted distribution and the realized token, which is equivalent to maximum likelihood estimation.[5]

Let P denote a fixed prompt (context), and let $\mathcal{V}$ be the LLM's vocabulary of all possible tokens. An autoregressive LLM defines a conditional probability distribution over the next token, s, from a finite vocabulary $\mathcal{V}$. This conditional distribution is given by

$$p(s \mid P), \qquad s \in \mathcal{V}. \tag{1}$$

Here, p denotes the probability distribution produced by the neural network, such that $\sum_{s\in\mathcal{V}} p(s \mid P) = 1$. It is an estimate of the true but unknown distribution $q(s \mid P)$. For this reason, we refer to p as the LLM's *inner probabilities*.

The LLM vocabulary typically consists of between 50,000 (Radford et al., 2019) and 150,000 tokens (Dubey et al., 2024). The high dimensionality of the token space, combined with the normalization constraint (probabilities summing to one), naturally results in a high degree of sparsity in the conditional distribution p. This sparsity plays a central role in enabling tractable analysis of the model's inner probabilities. For any given context P, it often suffices to focus on a small subset of tokens in $\mathcal{V}$ with non-negligible predicted probabilities.

Decoding strategy: After training, the neural network does not autonomously generate text. It requires an algorithm to select tokens from the conditional probability distribution given the current context. For instance, given the prompt "The cat is on the," a well-trained model might assign high probability to "table" and negligible probability to unrelated words like "stock." The decoding strategy applies a specified selection rule to choose the next token, which is then appended to the existing context to form an updated context for predicting subsequent tokens. This iterative process continues until a stopping condition is met, such as reaching a maximum length or generating a special end-of-sequence token.

[5]This is equivalent to maximizing the likelihood of the masked tokens.

One possible decoding strategy selects the most probable token at each step:

$$s_1 = \arg\max_{s \in \mathcal{V}} p(s \mid P),$$
$$s_2 = \arg\max_{s \in \mathcal{V}} p(s \mid s_1, P),$$

and so on. Although intuitive, this deterministic decoding strategy has been shown to yield outputs that are repetitive and stylistically monotonous (Holtzman, Buys, Du, Forbes, and Choi, 2019). Consequently, modern LLMs typically employ stochastic sampling, for example, by selecting the next tokens according to their predicted probabilities. The degree of randomness is controlled by a temperature parameter that adjusts the distribution's sharpness: a lower temperature concentrates the distribution around the most probable tokens (with a temperature of zero for the deterministic case), while a higher temperature flattens the distribution, allowing for more diverse token selection.

It is worth emphasizing that the simplified explanation above masks the richness of real-world decoding strategies,[6] which add considerable complexity and opacity atop the probabilistic model. For example, rather than selecting individual tokens sequentially, *beam search* explores multiple candidate sequences simultaneously to find the one that maximizes the joint probability of the entire sequence, while dynamic filtering rules, such as top-k or nucleus sampling, truncate the probability distribution before sampling occurs. While these decoding strategies are commonly used in commercial LLMs, the specific algorithms and hyperparameters are rarely disclosed.

LLM uncertainty. Researchers are rightfully concerned with the opacity of LLMs and often refer to them as black boxes. The two distinct components of LLMs discussed above have important implications for understanding and quantifying uncertainty in their outputs. Just as deep neural networks are generally considered black-box models due to their complexity and scale, the neural language model component of LLMs shares this opacity. State-of-the-art models now contain billions of parameters, rendering the interpretation of individual neurons or parameters infeasible. Decoding strategies further exacerbate the opacity and introduce additional uncertainty in two ways, first due to the lack of transparency in the decoding algorithms themselves, and second due to a feedback loop, whereby each selected token becomes part of the context for subsequent predictions, altering future conditional distributions. As a result, randomness introduced by the decoding strategy can cascade and amplify the unpredictability of the final output.

[6] These include greedy decoding and beam search (Sutskever, Vinyals, and Le, 2014), as well as stochastic techniques such as top-k (Fan, Lewis, and Dauphin, 2018) and nucleus sampling (Holtzman et al., 2019).

Nevertheless, even in the absence of structural interpretability, the uncertainty associated with the neural language model's predictions is well-defined given our knowledge of the conditional probability distribution over the next token. In the next section, we show how these inner probabilities provide the foundation for an entropy-based framework for uncertainty quantification. This framework abstracts from the opaque decoding strategy and yields principled, interpretable measures of uncertainty for LLM-generated outputs.

2 Methodology

In this section, we develop an entropy-based framework for measuring the uncertainty of large language model (LLM) output, which is designed to be interpretable, semantics-aware, and feasible in large-scale empirical applications. We also use the framework to analyze the feedback loop embedded in a multi-token generation setting and discuss its implications for uncertainty quantification and prompt engineering. Finally, we compare the proposed entropy-based measures with alternative black-box approaches for uncertainty quantification.

2.1 Token-level Uncertainty

As discussed in the previous section, an autoregressive LLM produces a conditional probability distribution over the next token, s, based on the prompt P. This distribution is denoted by $p(s \mid P)$ in Eq. (1). Predicting the next token is effectively a classification problem over a large but finite set of discrete outcomes (tokens) from a finite vocabulary $\mathcal{V}$. A canonical measure of the predictive uncertainty in this setting is the Shannon entropy of the conditional distribution $p(s \mid P)$:

$$H_{\text{tok}}(P) = -\sum_{s \in \mathcal{V}} p(s \mid P) \log p(s \mid P). \tag{2}$$

Large values of $H_{\text{tok}}(P)$ indicate a diffuse predictive distribution (high uncertainty); small values indicate a concentrated predictive distribution (high confidence).

In practice, computing $H_{\text{tok}}(P)$ can be challenging. Modern LLM vocabularies are very large, often exceeding 10^5 tokens. For most prompts, however, the conditional probability distribution $p(s \mid P)$ is highly sparse: the vast majority of probability mass is concentrated on a relatively small set of tokens. Furthermore, closed-source LLMs typically only provide a limited number of (highest) token probabilities from the model API. Thus, it is more practical to compute a truncated entropy measure in place of $H_{\text{tok}}(P)$.

Specifically, let $\mathcal{V}_K(P) \subset \mathcal{V}$ denote the set of $K > 0$ tokens with the highest conditional probabilities under $p(\cdot \mid P)$. We define the *top-K normalized* conditional probability distribution over $\mathcal{V}_K(P)$ as

$$p_K(s \mid P) = \begin{cases} \dfrac{p(s \mid P)}{\sum_{t \in \mathcal{V}_K(P)} p(t \mid P)}, & s \in \mathcal{V}_K(P), \\ 0, & \text{otherwise.} \end{cases} \tag{3}$$

The *top-K token-level entropy* is then given by

$$H^K_{\text{tok}}(P) = -\sum_{s \in \mathcal{V}_K(P)} p_K(s \mid P) \log p_K(s \mid P). \tag{4}$$

When $K \ll |\mathcal{V}|$ while $\sum_{t \in \mathcal{V}_K(P)} p(t \mid P)$ is close to one, $H^K_{\text{tok}}(P)$ will not only closely approximate the full token-level entropy $H_{\text{tok}}(P)$, but also be much easier to compute.

A particularly tractable setting in which Eq. (2) can be implemented directly arises when the set of admissible outputs is explicitly constrained to be small. In certain applications, we may want to instruct the LLM to generate a response from a finite set of discrete labels, for example, choosing among options "A", "B", or "C," or providing a binary "Yes" or "No" answer. Consider the binary case, where the admissible token set is $\mathcal{S} = \{A, B\}$, the token-level entropy $H_{\text{tok}}(P)$ simplifies to

$$H_{\text{bin}}(P) = -p(A \mid P) \log p(A \mid P) - (1 - p(A \mid P)) \log(1 - p(A \mid P)), \tag{5}$$

which attains its maximum when $p(A \mid P) = p(B \mid P) = 0.5$.

Motivated by this observation, we define a simple inner confidence statistic for a given LLM output. With the probability of output s conditional on prompt P being $p(s \mid P)$, the probability that the output is not s is given by $1 - p(s \mid P)$. Given that the binary entropy is maximized when $p(s \mid P) = 0.5$, we measure the model's inner confidence on the output s based on the distance of the inner probability from maximum uncertainty:

$$C(s \mid P) = \big|p(s \mid P) - 0.5\big|. \tag{6}$$

Notice that this measure of inner confidence assumes that all outputs that are not s can be aggregated into a single complementary outcome. This assumption is valid when the prediction task is indeed binary, or when the non-s outcomes are equivalent from a decision perspective. For example, a trading strategy may only invest in a stock if the LLM predicts

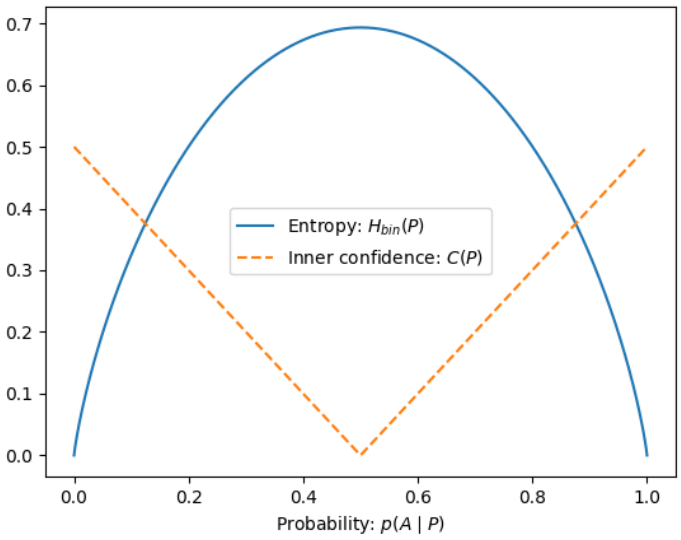


Figure 1: **Entropy and inner confidence in the binary case.** The solid blue curve plots the binary entropy $H_{\text{bin}}(P)$ as a function of $p(A \mid P) \in [0, 1]$. The dashed red curve plots the inner-confidence statistic $C(p)$.

a firm-specific news event is "positive," while all other outcomes ("negative," "neutral," or "uncertain") lead to no investment. As the following lemma shows, this inner confidence statistic is monotonically related to the binary entropy.

Lemma 1 (Monotone transform between inner-confidence and binary entropy)**.** *There exists a continuous, strictly decreasing function g such that*

$$C(s \mid P) = g\big(H_{\text{bin}}(s \mid P)\big), \tag{7}$$

where $H_{\text{bin}}(s \mid P)$ is defined for the conditional distribution over $\mathcal{S} = \{s, \neg s\}$. In particular, ranking predictions by $C(s \mid P)$ is equivalent (up to reversal) to ranking them by binary entropy.

Proof. See Appendix A. □

Lemma 1 shows that the inner-confidence statistic $C(s \mid P)$ is an order-preserving (more specifically, order-reversing) transformation of the entropy-based uncertainty measure. This result is also illustrated in Figure 1. Thus, in empirical applications where uncertainty measures are used ordinally, e.g., when ranking the uncertainties of different LLM outputs, sorting by the inner-confidence statistic $C(s \mid P)$ is equivalent to sorting by $H_{\text{bin}}(s \mid P)$.

2.2 Synonyms and Semantic Labels

The token-level entropy defined in Eq. (2) treats all tokens as distinct outcomes, even when multiple tokens are semantically equivalent (synonyms). When probability mass is distributed across such synonymous tokens, the entropy measure can become inflated by capturing lexical variation rather than focusing on economically meaningful uncertainty. For example, following the prompt "The stock market is expected to," an LLM might may assign similar probabilities to tokens like "rise" and "increase." Treating these tokens as distinct outcomes would inflate the uncertainty of output.

To better align uncertainty quantification with economic decision, we would ideally like to group tokens into equivalence classes based on their semantic interpretations in a given context, and then compute the entropy with respect to the conditional distribution over these semantic classes. Formally, denote a finite label space $\mathcal{Y}$ and a deterministic mapping $f : \mathcal{V} \to \mathcal{Y}$ that identifies tokens that are semantically equivalent (e.g., "rise" and "increase" will be assigned to the same label class). The induced conditional distribution in the label space is given by

$$p_Y(y \mid P) = \sum_{s \in f^{-1}(y)} p(s \mid P), \qquad y \in \mathcal{Y}. \tag{8}$$

We can then compute the corresponding semantic-label entropy:

$$H_{\text{sem}}(P) = -\sum_{y \in \mathcal{Y}} p_Y(y \mid P) \log p_Y(y \mid P). \tag{9}$$

It is easy to show that $H_{\text{sem}}(P) \leq H_{\text{tok}}(P)$. Intuitively, H_{sem} leaves out the dispersion across synonyms within each label class, while H_{tok} counts all variation. In practice, we can apply the mapping f to the top-K tokens with the highest conditional probabilities, and compute a top-K semantic-label entropy in an analogous manner.

Implementing the semantic-label entropy requires specifying the mapping f. There are several possibilities. First, we can use a curated synonym dictionary to create the mapping. Second, the mapping can be learned via embedding-based clustering. Specifically, we first map tokens into a high-dimensional semantic embedding space using pre-trained language model embeddings, and then group them into semantic classes using unsupervised clustering algorithms (e.g., using k-means clustering). This approach has the advantage of being flexible and adaptable to different contexts. For example, embeddings can be trained or adapted for domain-specific corpora, such as financial texts, while clustering can be performed locally at the prompt level to capture context-dependent semantic equivalences. Finally, similar to the binary entropy discussed in Section 2.1, we can avoid the synonym issue altogether

by constraining the LLM to generate outputs from a small controlled token set through prompting. Our empirical study in Section 3 will be built on this last approach.

2.3 Multi-token Uncertainty

The token-level entropy measures discussed above can be extended to multi-token outputs, e.g., a sentence. Let a multi-token sequence be denoted by $S = (s_1, \ldots, s_n)$. We can compute its conditional probability using the chain rule for the autoregressive model:

$$p(S \mid P) = \prod_{i=1}^{n} p(s_i \mid s_{<i}, P). \tag{10}$$

where $s_{<i} \equiv (s_1, \ldots, s_{i-1})$ denotes the sequence of previously-generated tokens up to but not including token i. Importantly, they become part of the augmented prompt for predicting token s_i.

Conceptually, we can measure the uncertainty of the entire sequence using the mean token entropy:

$$H_{\text{seq}}(P) = -\frac{1}{n} \sum_{S} p(S \mid P) \log p(S \mid P), \tag{11}$$

which is the sequence-level entropy normalized by sequence length n. The summation in Eq. (11) is over all possible sequences of length n, and the normalization ensures that the measure is comparable across sequences of different lengths. Without normalization, longer sequences would naturally have higher entropy.

However, $H_{\text{seq}}(P)$ is generally computationally intractable: the number of possible sequences, $|\mathcal{V}|^n$, grows exponentially with sequence length n. Even if we restrict the possible sequences based on top-K tokens at each step, we will only be able to compute the measure for short sequences.

For practical applications, we propose a computationally feasible implementation of Eq. (11) based on sampling. Specifically, we draw M independent sample sequences $S^{(1)}, \ldots, S^{(M)}$ based on the same prompt P and use stochastic decoding (with temperature ≥ 1.0) to ensure diversity. For each generated sequence, we compute its conditional probability according to Eq. (10), and then renormalize it:

$$p_M(S^{(j)} \mid P) = \frac{p(S^{(j)} \mid P)}{\sum_{\ell=1}^{M} p(S^{(\ell)} \mid P)}. \tag{12}$$

Finally, we can compute the mean token entropy as:

$$H^M_{\text{seq}}(P) = -\frac{1}{n}\sum_{j=1}^{M} p_M(S^{(j)} \mid P) \log p_M(S^{(j)} \mid P). \tag{13}$$

As in the case of single-token outputs, lexically different sentences could have similar semantic meanings. To deal with this issue, we can extend the semantic-label entropy from Eq. (9) to multi-token outputs. Similar to the procedure described in Section 2.2, we can first assign sequences into semantic clusters, then aggregate conditional probabilities for the M sequences at the cluster level, and finally compute the entropy over the cluster distribution. For sequence clustering, Kuhn, Gal, and Farquhar (2023) have proposed the bidirectional entailment clustering algorithm. Alternatively, we can also use embedding-based clustering as described in Section 2.2. In Appendix B we provide an example to illustrate how our method can be expanded to the multi-token setting.

The feedback loop. As mentioned earlier, the autoregressive decoding process creates a feedback loop: each selected token becomes part of the context for predicting subsequent tokens; therefore, the randomness with each token selection will influence the conditional distributions of subsequent tokens, especially when the selected token has significant and persistent influence on the distribution of subsequent tokens. This feedback mechanism has important implications for uncertainty quantification and prompt engineering of a multi-token sequence.

To formalize this feedback loop, consider the cases of one-token versus two-token generation. Let S_1 denote the first token generated by the model given a prompt P, and let S_2 be the subsequent token. While the conditional distribution of S_1 only depends on the prompt P, the marginal distribution of S_2 integrates over the randomness in S_1,

$$p(S_2 \mid P) = \sum_{s_1 \in \mathcal{V}} p(s_1 \mid P)\, p(S_2 \mid s_1, P). \tag{14}$$

The entropy for this marginal distribution of S_2 admits the decomposition

$$H(S_2 \mid P) = H(S_2 \mid S_1, P) + I(S_2; S_1 \mid P) \geq H(S_2 \mid S_1, P). \tag{15}$$

The term $H(S_2 \mid S_1, P)$ captures the average conditional uncertainty of the second token given a realized first token. The mutual information term, $I(S_2; S_1 \mid P)$, measures the conditional dependence between S_2 and S_1 given the prompt P. It quantifies the additional

uncertainty about S_2 induced by the randomness of the initial decoding step. Intuitively, if a particular realization of S_1 has significant influence on the conditional distribution of S_2, then $I(S_2; S_1 \mid P)$ will be large, resulting in larger marginal entropy for S_2. Finally, the inequality in Eq. (15) follows because $I(S_2; S_1 \mid P)$ is nonnegative; $I(S_2; S_1 \mid P) = 0$ if and only if S_2 is conditionally independent of S_1 given P.

Extending Eq. (15) to a sequence with tokens $S_1, \ldots, S_L$, the marginal entropy of the final token admits the decomposition

$$H(S_L \mid P) = H(S_L \mid S_{<L}, P) + \sum_{i=1}^{L-1} I(S_L; S_i \mid S_{<i}, P), \tag{16}$$

where $S_{<i} \equiv (S_1, \ldots, S_{i-1})$. The term $H(S_L \mid S_{<L}, P)$ represents the average conditional uncertainty of the L-th token given a realized prefix, $S_{<i}$. Each mutual information term $I(S_L; S_i \mid S_{<i}, P)$ is nonnegative and captures the incremental uncertainty about the L-th token arising from the randomness of an earlier token S_i.

This result clearly shows that the realization of each decoding step alters the conditional distributions of the subsequent tokens, with the resulting dependence propagating forward through the sequence. Consequently, the marginal entropy of later tokens is weakly increasing in sequence length. Longer generated sequences therefore embed a larger uncertainty feedback loop, as randomness introduced at early stages persists and amplifies the unpredictability of subsequent outputs.

As a concrete example, consider the prompt:

```
Based on the recent 25 bps cut in the federal funds rate, provide a
concise forecast for the S&P 500 performance in the next month. End your
sentence with either the token ‘positive’ or ‘negative’.
```

If the model generates the final prediction directly, and the training data show that the market generally responds positively to rate cuts, then the model may assign a high probability to the token "positive," and the uncertainty will be low. However, in a multi-token generation, the first token S_1 drawn could be a conjunction like "while," which could then lead to a "narrative drift" that significantly alters the conditional distribution for the final token. An example could be:

```
While rate cuts often lead to short-term gains, market participants had
anticipated a more substantial reduction, suggesting that the overall
outlook for the index may be negative.
```

Implications for Prompt Engineering. The feedback loop provides guidance for how to design LLM prompts to minimize uncertainty amplification. Unconstrained or verbose responses mechanically increase uncertainty through the cumulative dependence on earlier decoding choices. To mitigate this effect, we recommend prompts be structured so that the payoff-relevant output is produced early and compactly, ideally as a single token or from a small, predefined set of labels. When multi-token responses are needed, ask for short and direct answers, with reasoning or explanatory text separated from the final output.

At first glance, this recommendation may appear to conflict with the widespread use of chain-of-thought (CoT) prompting to improve LLM performance on complex tasks (Wei and al., 2022). The apparent tension reflects a distinction between unconstrained token generation and structured intermediate reasoning aimed at extracting task-relevant information. Under CoT prompting, intermediate reasoning tokens are part of the conditioning context for the final output. If accurate and informative, they can effectively reduce the conditional entropy $H(S_L \mid S_{<L}, P)$ and improve the accuracy of the output, even though the additional generated tokens mechanically increase the marginal entropy $H(S_L \mid P)$ through the autoregressive feedback loop (as shown in Eq. (16)). This is why we recommend separating reasoning from final answers rather than avoiding CoT altogether. In practice, uncertainty amplification can be further limited by eliciting structured, task-specific intermediate steps instead of open-ended CoT, and by verifying intermediate outputs before producing the final prediction.

2.4 White-box vs. Black-box Uncertainty Quantification

We conclude this section by comparing the entropy-based measures developed in this section with alternative approaches to uncertainty quantification. The entropy-based measures rely on direct access to the LLM's conditional probabilities. We refer to them as white-box measures, in contrast to black-box approaches that infer uncertainty without observing the model's internal probabilities.

One type of black-box methods treat an LLM as a stochastic mapping from prompts to outputs and infer uncertainty by measuring the dispersion across model outputs generated

from repeated queries. Besides the obvious computational challenges, there are several issues associated with these sampling-based methods. First, these measures are built on a limited number of sampled realizations from a high-dimensional output space and therefore conflate intrinsic model uncertainty with randomness induced by the decoding strategy. As a result, these black-box measures are inherently noisy, sensitive to hyperparameter choices, and not readily comparable across models. Second, due to its lack of a clear probabilistic meaning, dispersion-based measures are difficult to interpret theoretically and compare across different prompts. Finally, sampling-based methods scale poorly when uncertainty is driven by rare but economically important outcomes, as low-probability events are systematically underrepresented in finite samples.

Another black-box approach for uncertainty quantification uses model-declared confidence (Lin, Hilton, and Evans, 2022). Under this approach, the LLM is prompted to generate a confidence score following its main output. For example, the model may be asked to rate its confidence on a scale from 1 to 10, which is then used as a proxy for uncertainty. While simple and intuitive, it is unclear how well declared confidence aligns with the model's internal uncertainty, especially in light of the feedback loop discussed in Section 2.3. We compare the empirical relationship between entropy-based measures and declared confidence and document a distinct bias associated with declared confidence in Section 5.

3 Economic Value of Uncertainty Quantification

In this section, we examine the economic value of quantifying the uncertainty of LLM outputs. We conduct two empirical experiments to address this question. In the first experiment, we examine whether lower uncertainty (higher inner confidence) is associated with higher accuracy of model predictions. Second, we assess the economic value of uncertainty quantification by examining how much it affects the profitability of trading strategies built on LLM predictions.

In addition, to study the sources of uncertainty, we investigate which topics in the news are most strongly associated with model confidence. Finally, we apply a Bayesian framework to assess the driver and stability of the inner confidence measure.

3.1 The Relationship Between Confidence and Accuracy

For our first test, we feed financial news articles during regular market hours to an LLM and ask it to classify the news as positive or negative for the stock price of a given firm. We

Table 1: **Summary Statistics**
The table below presents the summary statistics of the dataset of news articles that we use for the main analysis. For each article, we indicate the length of the headline and of the body of the article in number of characters. For each stock for which we have news, we indicate the market capitalisation in billions and the average return on the same day on which we record a news. For each item we show the mean value, the standard deviation, and the distribution.

		mean	std	min	20%	40%	60%	80%	max
Article	Headline Length	67.310	19.323	15	53	61	68	79	241
	Body Length	1226	1032	120	426	687	1051	1911	4992
Stock	Market Cap.	125.298	220.632	10.010	18.589	35.803	70.805	184.252	2686.711
	Return	0.002	0.036	-0.436	-0.015	-0.003	0.005	0.018	0.412

then use the same-day realized returns as ground truth to evaluate the model's performance. For this analysis, we use a sample of news articles from Reuters.com. We randomly select a subsample of articles that meet the following criteria:

1. The news must relate to a single company. This ensures that the firm is directly affected by the news rather than mentioned only indirectly. We apply the data-cleaning procedure described in Chen et al. (2022).
2. The news must have a timestamp between 10 a.m. and 2 p.m. (ET) to increase the likelihood that contemporaneous daily returns reflect the news impact.
3. The news must contain between 100 and 5,000 characters.
4. The news must concern firms with a market capitalization of at least 10 billion USD.
5. To remove look-ahead bias concerns, the news must have been published after the learning cutoff of the model we analyze.

We obtain a population of 100,000 financial news items. We supplement this subsample with daily returns and market capitalization data from the CRSP database. Table 1 presents the main summary statistics for the sample.

We classify news using Prompt 1, which elicits the LLM's assessment of the news' impact on the firm's stock price. As discussed in Section 2, issues related to synonymy/semantics and the effect of feedback loops on entropy can be mitigated by designing prompts that elicit concise responses in which, ideally, a single token encodes complex meaning. Accordingly, Prompt 1 is constructed to produce a single-letter output: "A" for positive news and "B" for negative news.

```
Based on the news below, please say if you think the return for the stock
 {target} will be A (positive) or B (negative).

Please write only a letter A or B. Add no other formatting or bolding.

{headline}
{body}
```

Prompt 1: **Classifying the News**
The prompt asks the LLM to predict the market reaction (positive or negative) to a news item related to a specific listed firm. The brackets $\{\cdots\}$ indicate dynamic inserts where we place the *target* (firm's ticker), *headline* (article's headline), and *body* (main text of the news article).

For a sufficiently capable transformer model, the two most probable next tokens are "A" and "B":

$$p(s_1 = A \mid P) \geq p(s_1 = X \mid P) \quad \text{and} \quad p(s_1 = B \mid P) \geq p(s_1 = X \mid P) \quad \forall X \neq A, B. \tag{17}$$

Furthermore, a model with perfect understanding of Prompt 1 would assign zero probability to any token other than "A" or "B":

$$p(s_1 = A \mid P) + p(s_1 = B \mid P) = 1. \tag{18}$$

Empirically, we find that while state-of-the-art models almost always satisfy Eq. (17), they do not consistently satisfy Eq. (18).

To eliminate residual probability mass assigned to irrelevant tokens (e.g., "C", *The*), we define the inner probability of "A" as the normalized probability over the acceptable token set:

$$\tilde{p}(s_1 = A \mid P) = \frac{p(s_1 = A \mid P)}{p(s_1 = A \mid P) + p(s_1 = B \mid P)}. \tag{19}$$

Using this normalized probability, the confidence metric defined in Eq. (6) becomes

$$C = \left|\tilde{p}(s_1 = A \mid P) - 1/2\right|. \tag{20}$$

We extract the inner conditional probability $\tilde{p}(s_1 = A \mid P)$ for each news article in

our sample.[7] We classify news as positive if the probability of generating "A" as the first token is at least 0.5 ($\tilde{p}(s_1 = A \mid P) \geq 0.5$). Conversely, we classify news as negative (B) if $\tilde{p}(s_1 = A \mid P) < 0.5$.

We use the realized return as *ground truth*, defining an event as positive if the daily open-to-close contemporaneous return is positive. We then measure accuracy as the share of news for which the return's sign matches the LLM's point estimate. Finally, we divide the 100,000 financial news items into quintiles based on confidence, ranging from 1 (lowest confidence) to 5 (highest confidence).

Figure 2 presents the results by quintile of confidence. Higher inner probabilities correspond to greater accuracy, suggesting that inner probabilities reflect the underlying economic reality rather than being an uninterpretable component of a black box model. Moreover, the accuracy gains are substantial. At low confidence levels, the LLM is correct about 50% of the time, no better than a coin toss, indicating its inability to classify with certainty. In contrast, at high confidence levels, accuracy reaches 64.5%, showing that the LLM's inner probabilities align with its actual confidence.

This result validates the use of inner probabilities as they correlate with model accuracy, rather than being a function of the black box nature of neural networks. Hence, the information contained in the inner probabilities is a reflection of the actual model's ability to successfully perform the task required.

3.2 Confidence-Based Portfolio Strategies

To further assess the economic significance of inner probabilities, we construct trading portfolios. We follow Chen et al. (2022) in collecting news from *Reuters* that mention a single U.S. firm through its ticker. In the spirit of Lopez-Lira and Tang (2023), we include only articles with at least 50 and less than 500 words and focus on overnight news, appearing after 4 p.m. on day $t-1$ and before 9 a.m. on day t. For each trading day t, we classify news items as either "positive" or "negative," following the procedure in Section 3.1. An important difference lies in the prompt. In this experiment, we use Prompt 2, which produces text structured as a sentiment label, followed by the delimiting character "|" and a numerical value representing the model-declared confidence (e.g., "A|0.7"). In addition to being compatible with the principle proposed in Section 2, this format enables a direct comparison, within a single prompt, between the model's *inner confidence* (computed from the distribution of the first token to obtain $\tilde{p}(s_1 = A \mid P)$) and its *declared confidence*, obtained

[7] We detail technical consideration of this extraction in Appendix C.

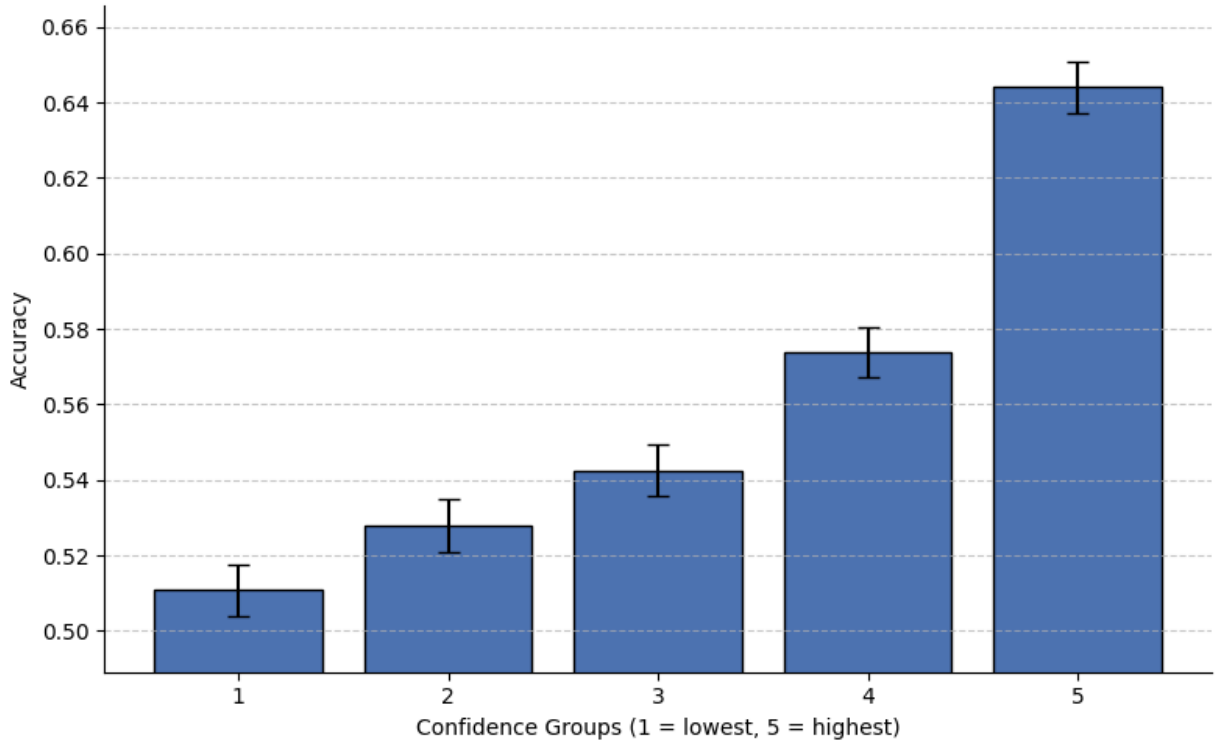


Figure 2: **Baseline Accuracy by Confidence Groups.** The figure shows the results of using prompt 1 to estimate sentiment and the associated inner probability for 100,000 randomly selected news articles. Sentiment is classified as positive when the probability of generating the first token as "A" exceeds 0.5 ($\tilde{p}(s_1 = A \mid P) \geq 0.5$). Confidence is defined as $c = |\tilde{p}(s_1 = A \mid P) - 0.5|$, and its distribution is divided into 5 quintiles from 1 (low confidence) to 5 (high confidence). For each quintile group, we report the average accuracy and 95% confidence intervals.

by parsing the number after the "|" symbol. The latter, used in prior work (see, e.g., Bybee, 2023; Fedyk et al., 2024), offers an intuitive means of eliciting model confidence without accessing token-level probabilities, making it a natural benchmark for our methodology. We further explore *declared confidence* and potential biases introduced by the feedback loop in Section 5.

We then construct five intraday strategies on open-to-close returns:[8]

- **Baseline Portfolio:** Equally weights all stocks with a net positive signal in a long leg and all stocks with a net negative signal in a short leg. The signal for firm i on day t is determined by the majority of its overnight news items.
- **Confidence Portfolios:** For both the *inner* and *declared* confidence we build two portfolios:

[8]Appendix D presents a more comprehensive description of the portfolio construction methodology.

```
Forget all your previous instructions.

You are a financial expert with stock recommendation experience.
Is this financial news good or bad for the stock price of {target} in the
 short term?
Answer A if good news, B if bad news. After you answer with A or B,
please write the symbol | and then a number between 1 (very confident)
and 0 (not confident) to indicate how confident you are in your answer.
Please only write the symbol "|" and the number, nothing else.

ARTICLE:
{headline}
{body}
```

Prompt 2: **Joint Declared and Inner Confidence.**
The prompt asks the LLM to predict the market reaction (positive or negative) to a news item related to a specific listed firm and to assess its own confidence in the prediction. The brackets $\{\cdots\}$ indicate dynamic inserts where we place the *target* (firm's ticker), *headline* (article's headline), and *body* (main text of the news article).

- **High-Confidence Portfolio:** Includes only news items whose classification confidence is above a specific threshold. The confidence threshold level $c(\nu_t)$ determines the percentage of news items classified as low confidence. For each portfolios (inner and declared confidence), we choose the threshold monthly by an expanding-window procedure to maximize the strategy's past Sharpe ratio. We build the portfolio on the resulting subsample following the same procedure as for the benchmark one.
- **Low-Confidence Portfolio:** Uses the remaining news items with confidence at or below the threshold $c(\nu_t)$. Its construction mirrors the high-confidence approach but with low-confidence news.

We start our out-of-sample analysis in October 2023 to avoid look-ahead bias related to GPT-4o's training cutoff date. The threshold $c(\nu_t)$ is recalibrated monthly, using only past data.[9] All portfolios are equally weighted and rebalanced daily. Table 2 summarizes the results. The *High-Confidence* portfolio achieves a Sharpe ratio substantially larger than that of the *Baseline* portfolio, indicating that filtering out low-confidence signals improves per-

[9]The average $c(\nu_t)$ is 0.18 for the inner confidence-based portfolios and 0.05 for the declared confidence ones. For robustness, we repeat the analysis using a set of fixed thresholds instead of the dynamic threshold. The results remain qualitatively similar and are presented in Appendix E.

Table 2: **Performance of Confidence-Based Portfolios**
This table reports out-of-sample performance of three long-short investment strategies based on the model confidence in its classification. Inner Confidence refers to the classification obtained following the method presented in Section 1, while Declared Confidence refers to the self reported confidence obtained with Prompt 2. The *Baseline* portfolio includes all news, the *High-Confidence* portfolio includes only news for which the model generated a confident classification, and the *Low-Confidence* portfolio includes non-confident classifications only. Annualized return, volatility, and Sharpe ratio are reported in the first three columns. The analysis starts in October 2023, following the training cutoff of GPT-4o. All portfolios are equally weighted and rebalanced daily.

	Inner Confidence		**Declared Confidence**		**Baseline**
	High Confidence	Low Confidence	High Confidence	Low Confidence	
Mean	0.50	-0.10	0.41	-0.03	0.40
Std	0.10	0.25	0.11	0.31	0.09
Sharpe ratio	5.15	-0.39	3.71	-0.10	4.24
Avg # assets	481.50	111.82	452.42	130.82	583.24

formance. By contrast, the *Low-Confidence* portfolio exhibits a negative Sharpe ratio, confirming that low confidence is associated with poor model performance. Using the approach suggested by Jobson and Korkie (1981) after making the correction pointed out in Memmel (2003), we test the significance of these results. We find that the *High-Confidence* portfolio's Sharpe ratio is significantly higher than the *Baseline* portfolio's (Test statistic = 2.84). The Sharpe ratio of the *Low-Confidence* portfolio does not significantly differ from zero (Test statistic = -0.41). In contrast to inner confidence, declared confidence leads to weaker performance. The Sharpe ratio is lower than that of the *Baseline* portfolio, although the difference is not statistically significant (Test statistic = -1.28).[10]

To better understand the source of the high Sharpe ratio of the inner confidence-based high-confidence portfolio, we sort the high-confidence observations based on several characteristics and split the portfolio in high and low based on the sorting criteria. For each sub-sample we repeat the analysis from Table 2. Table 3 presents the results.

We find that the Sharpe ratio follows intuitive and predictable patterns. Performance is concentrated among small firms, which are covered by fewer analysts and are more illiquid. Most of the returns come from the short leg, with a marginal role of the long leg. These results suggest that the model's performance does not stem from a deeper understanding of financial markets, but rather exploits information and trading frictions that make the

[10]In 2% of our sample, the LLMs failed to follow instructions, and the generated text did not contain a parsable declared confidence. In such cases, we removed the observation from the declared confidence portfolios. By contrast, the inner confidence always produces at least one probability for both A and B. This additional stability represents a further advantage of our methodology over declared confidences.

Table 3: **Sub-Sample Sharpe Ratios of High-Confidence Portfolios**
This table provides a breakdown of the "High-Confidence" portfolio built including only news items classified with high confidence by the LLM and presented in Table 2. We partition the daily sample based on analyst coverage, illiquidity, and market capitalization. "Higher" and "Lower" refer to the split based on the indicated criterion. For each criterion and split, we report the Sharpe ratios of portfolios constructed using the top and bottom halves of the respective split. The first two columns present results for the Long-Short strategy, the next two for the short leg, and the final two for the long leg. The analysis uses data beyond the training cutoff of GPT-4o in October 2023. All portfolios are equally weighted and rebalanced daily.

Portfolio	Long_Short		Short Leg		Long Leg	
Criterion Split	Higher	Lower	Higher	Lower	Higher	Lower
Analyst coverage	1.916	5.261	-4.293	-2.365	-3.973	0.869
Bid-Ask spread	4.368	3.183	-3.059	-0.459	0.087	1.619
Illiquidity (Amihud)	5.542	1.961	-3.202	-0.832	1.025	0.472
Market Cap	2.350	5.019	0.122	-3.247	1.563	0.335
# News articles	1.202	3.759	-4.631	-1.601	-2.181	1.245
Turnover	2.728	5.563	-2.140	-2.375	-0.227	1.568

investment strategy difficult to implement in practice.

3.3 Confidence Determinants

Next, to explore the determinants of LLM confidence we estimate the following regression with a Logit model:

$$\begin{aligned} T_{i,t} = \beta_1 \,\text{realized\_vol}_{i,t} + \beta_2 \,\text{realized\_return}_{i,t} \\ + \beta_3 \,\text{abn\_volume}_{i,t} + \beta_4 \,\text{nb\_news\_firm}_{i,t} \\ + \beta_5 \,\text{nb\_news\_tot}_t + \beta_6 \,\text{vix}_t + FE_i + \epsilon_{i,t} \end{aligned} \tag{21}$$

Here, $T_{i,t}$ is a dummy variable equal to 1 if the LLM assigns high inner confidence to its classification of a news item. High confidence means that the model confidence exceeds a percentile cut-off of the confidence distribution, specifically the 10th, or 30th percentile. These thresholds correspond to flagging approximately the top 90%, 80%, or 70% of the distribution, respectively.

Regarding the independent variables, $realized\_vol_{i,t}$ is the thirty-day forward realized volatility of firm i, computed as the square root of the mean squared daily returns over the next 30 trading days. We consider this variable as a proxy for news complexity. Indeed,

a news followed by a period of volatility was likely harder to interpret and should be associated with lower levels of confidence in the LLM classification. $realized\_return_{i,t}$ is the arithmetic mean of those forward daily returns and captures the firm's subsequent performance. $abn\_volume_{i,t}$ measures abnormal trading volume on day t as the deviation of raw volume from its 60-day trailing mean, normalized by the corresponding standard deviation, since days of high volume might be associated with hard-to-interpret news. $nb\_news\_firm_{i,t}$ counts the number of Reuters headlines mentioning firm i on day t and serves as a proxy for the magnitude of firm-specific information flow. $nb\_news\_tot_t$ is the total number of Reuters headlines across all firms on that day and reflects market-wide news pressure. Finally, vix_t is the closing value of the CBOE Volatility Index, summarizing overall equity-market uncertainty on day t. We divide all dependent variables by their standard deviation to make the coefficient easier to interpret. Additionally, we include firm fixed-effect, which in our setting captures all of the information that the LLM has about a firm, both before and after the specific news, capturing any potential bias in terms or optimism or confidence. Standard errors are clustered at the date level. Table 4 presents the results.

The LLM assigns high confidence to observations followed by positive returns and lower volatility. Abnormal trading volume and the market-wide headline count apply downward pressure, indicating that background noise reduces the model's decisiveness. The daily count of firm-specific headlines becomes positively associated with confidence only when the threshold is set at the 30th percentile of the confidence distribution. The VIX is negatively associated with high confidence, but the relationship is not significant. These patterns suggest that the LLM issues its sharpest probability splits when the subsequent price path is directional and trading volumes are low, consistent with standard forecasting intuition.

We proceed by examining the relationship between model confidence and the topical content of news articles. To assign topics to each news item, we follow Bybee, Kelly, Manela, and Xiu (2020), who employ a Latent Dirichlet Allocation (LDA) model trained on the *Wall Street Journal* to identify 130 distinct topics. Their publication of "word weights per topic" enables efficient topic assignment to new texts. While these weights are insufficient for precise probabilistic assignment across all topics, they are adequate for identifying the most dominant topic in a given article.

Accordingly, each news item i is associated with a sparse vector $X_i \in \mathbb{R}^{130}$, where the j^{th} component equals 1 if topic j is the most strongly associated with that article, and 0 otherwise.

We then perform a Lasso regression, using model confidence (y_i) as the dependent variable, measured on a decile-based scale from 1 (lowest confidence) to 10 (highest confidence).

Table 4: **Determinants of High Inner-Confidence Classification**
The table reports logit regression coefficients from Eq. (21), where the dependent variable is an indicator equal to one if the LLM's inner confidence for a news item exceeds the 10$^{\text{th}}$, 20$^{\text{th}}$, or 30$^{\text{th}}$ percentile of the confidence distribution (columns 1–3, respectively). Explanatory variables include 30-day forward realized volatility and return, abnormal trading volume, the daily count of firm-specific Reuters headlines, the aggregate market-wide headline count, and the VIX. Explanatory variables are all normalized by their standard deviation. All regressions include firm fixed effects, and standard errors are clustered by date. Asterisks indicate statistical significance at the 10% (*), 5% (**), and 1% (***) levels.

High-Confidence Threshold (percentile cut-off)	**10th**	**20th**	**30th**
realized_vol	-0.034***	-0.051***	-0.075***
	(0.012)	(0.013)	(0.013)
realized_return	0.023**	0.034***	0.046***
	(0.010)	(0.007)	(0.008)
abn_volume	-0.039***	-0.068***	-0.119***
	(0.011)	(0.017)	(0.038)
nb_news_firm	0.010	0.019	0.047***
	(0.012)	(0.012)	(0.013)
nb_news_tot	-0.032***	-0.076***	-0.078***
	(0.010)	(0.011)	(0.010)
vix	-0.011	-0.024*	-0.013
	(0.012)	(0.013)	(0.013)
Firm FE	Yes	Yes	Yes
Observations	327,625	340,792	344,266
Squared Correlation	0.0586	0.1376	0.1630
Pseudo R^2	0.0795	0.1191	0.1277

The independent variables are the topic indicators X_i. The regression solves:

$$\hat{\beta} \;=\; \arg\min_{\beta\in\mathbb{R}^p} \left\{ \frac{1}{N}\sum_{i=1}^{N}\left(y_i - X_i^\top \beta - \alpha\right)^2 \;+\; \lambda \sum_{j=1}^{p} |\beta_j| \right\}, \tag{22}$$

where α is an intercept term and λ is the regularization parameter. For each configuration, we increment λ until the model retains only 10 non-zero coefficients $\hat{\beta}_j$, corresponding to the ten topics most predictive of confidence levels.

This analysis is repeated across the full panel of news articles and separately for each of the ten largest firms. Figure 3 summarizes the results. Red hues denote topics associated with low confidence (negative $\hat{\beta}_j$), while blue hues indicate topics associated with high confidence (positive $\hat{\beta}_j$).

We find that news topics related to lawsuits and product announcements are consistently

Figure 3: **Lasso-Selected Topic Coefficients vs. Confidence Decile.** This figure presents the topics most strongly associated with the LLM's inner confidence. We estimate the penalized least squares model in Eq. (22), predicting the confidence decile y_i using topic dummies X_i, and tune the penalty parameter λ to ten selected topics. The top panel displays the full-sample coefficient estimates $\hat{\beta}$, while the lower panels show firm-specific estimates for the ten firms with the largest number of news articles. Blue tones indicate larger positive coefficients (associated with increased model confidence), while red tones represent larger negative coefficients (associated with decreased model confidence).

associated with higher levels of model confidence. This pattern aligns with economic intuition, as such events are typically straightforward to interpret as negative or positive news, respectively. In contrast, topics concerning corporate governance and foreign markets tend to correlate with lower confidence, likely reflecting their greater complexity and contextual ambiguity. These findings are consistent with fundamental economic intuition regarding which types of information yield clearer signals and, thus, more confident model assessments.

3.4 Confidence and Rationality

The experiment presented in this section should be interpreted through a Bayesian lens. LLMs, through their training, are implicitly endowed with prior beliefs about each firm. These priors may be conceptualized as a mean and variance, representing the model's estimate of the probability of event A and its associated confidence in the absence of new

information. News, in this context, functions as a signal characterized by an unobservable mean and variance. What is observable are the LLM's outputs $\tilde{p}(\text{“}A''|P)$ and c_i, which we interpret as the posterior mean and dispersion, respectively.

This Bayesian framework motivates a natural experiment: Can prompting alter either the model's prior belief about a firm or the variance of the signal, and does the resulting posterior behave in a manner consistent with rational updating?

To test this, we designed three pairs of prompts intended to positively or negatively perturb the model. The first pair aims to modify the model's prior variance by asserting that analysts either uniformly agree or disagree on the company's prospects. The second pair addresses the variance of the signal, informing the model that analysts either agree or disagree on the interpretation of recent news. Lastly, the third pair serves as a placebo. In this experiment, we prepend Prompt 1 with words associated either with high confidence–"Confidence, clarity, conviction, precision, certainty"–or with low confidence–"Doubt, anxiety, mistrust, uncertainty, hesitation"–to assess whether semantically vacuous keywords can influence the model's internal representation of confidence.

The initial prompt used in this experiment is presented below. Additional prompts are detailed in Appendix F.

```
Yesterday, most analysts agreed on the prospects of the firm {target}.
Today, new news has emerged. Based on this news, please assess the
expected return for the stock. Answer A if the news is good, B if the
news is bad. Is this financial news good or bad for the stock price of {
target} in the short term? Please start by writing only the letter A or B
. Add no other formatting. ARTICLE: {headline} {body}
```

Prompt 3: **Prior Consensus**
The prompt asks the LLM to predict the market reaction (positive or negative) to a news item related to a specific listed firm while introducing an initial conditioning on the model's prior. The brackets $\{\cdots\}$ indicate dynamic inserts where we place the *target* (firm's ticker), *headline* (article's headline), and *body* (main text of the news article).

We employ these three prompt pairs to replicate the classification exercise presented in Section 3.1. Table 5 summarizes the results. For each prompt we present the mean confidence, its variance, and the confidence level per quartile. The confidence distributions are clustered around 0.5 but as we have shown in Section 3, the ordinal ranking carries strong economic significance. Hence, to better visualize the results, we report the mean confidence both as absolute value and as the corresponding percentile in the baseline exercise, meaning

Table 5: **Aggregated Inner Confidence Across Eight Prompting Conditions**
This table reports the mean and variance of the LLM's inner confidence for three semantic contexts (Analyst Prior, Analyst Signal, and Noise), each evaluated under a Consensus and a Disagreement variant. For each experiment, we report the LLM's inner confidence variance, mean and its "corresponding percentile," which indicates the percentile rank of the mean confidence within the unconditional baseline distribution. The final row lists the t-stats from two-sided tests that compare the Consensus and Disagreement means within each context.

	Analyst Prior		**Analyst Signal**		**Noise**	
Conditioning	consensus	disagreement	consensus	disagreement	consensus	disagreement
Mean	0.480	0.476	0.481	0.478	0.485	0.484
Corresponding Percentile	0.6531	0.5857	0.6460	0.5588	0.6096	0.5725
Variance	0.005	0.005	0.005	0.005	0.003	0.003
T-stat difference	7.250		7.860		1.698	

what percentile of confidence does the mean level correspond to in the original baseline distribution without conditioning.

We find that providing the model with information regarding analyst consensus–whether pertaining to prior beliefs or posterior interpretation–induces a significant shift in the model's reported confidence. When informed that analysts are in agreement, without reference to the direction of the consensus, the model exhibits greater confidence in its classification. Conversely, when informed of analyst disagreement, the model's confidence diminishes. The contrast between these two conditions is highly statistically significant.

This behavior is consistent with rational updating: the model integrates the additional information and modulates its internal belief structure, treating consensus as an informative signal about the underlying classification difficulty. By contrast, introducing affectively charged but semantically irrelevant terms–serving as either positive or negative noise–has no measurable effect on the model's confidence. While such cues might influence human judgment, their irrelevance to the classification task renders any reaction to them irrational. In this respect, the model's behavior aligns with rational decision-making principles.

4 Confidence and Look-Ahead Bias

Look-ahead bias is a crucial methodological concern in economic and financial analyses involving LLMs (see, e.g., Sarkar and Vafa, 2024; Lopez-Lira et al., 2025). Commercial LLMs trained on extensive datasets may inadvertently "recall" information unavailable at the prediction time. Such recall can bias results and produce false positives, inflating performance in classification and prediction tasks.

In this section, we examine the connection between model memory and internal probabilities, focusing on months near the training cutoff.[11]

To identify this relationship clearly, we first conduct a test in a controlled setting. In the spirit of Didisheim et al. (2025); Lopez-Lira et al. (2025) and query the LLM with the following prompt:

```
What was the daily return of index {index name} on {date}?
Answer with only a number.
```

Prompt 4: **Pure Look-Ahead Bias**
The prompt above is designed to extract the LLM's pure look-ahead bias, defined as ability to recall the specific data point without any context. The brackets $\{\cdots\}$ indicate dynamic inserts where we place the index's name, ticker, and date.

The prompt evaluates the LLM's recall capability in a parsimonious way. In the absence of contextual cues for predicting returns, any accurate recall must arise solely from look-ahead bias. Following the methodology of Didisheim et al. (2025), we query the LLM about daily returns for eleven major indices between 2018 and 2023. Treating the "recalled" return as a forecast and the true return as the ground truth, we use the absolute error as a proxy for pure look-ahead bias. Since the LLM receives no additional information, a low absolute error implies that it successfully retrieved the true return from its internal "memory." Such memorization capacity could inadvertently introduce look-ahead bias in downstream applications.

For each "recollection," we further compute the average internal probabilities of the k tokens used to generate the response, given by $\sum_k \frac{1}{k}\tilde{p}(s_k)$. A higher value of this metric indicates greater model confidence in the generated return.

Figure 4 illustrates the relationship between absolute recall error and average internal probability. Specifically, we sort the observations by absolute error and group them into buckets representing 4% of the sample for visualization. The X-axis reports the average absolute error per group, while the Y-axis shows the corresponding average internal probability within each group.

The figure shows a clear difference between high recall (absolute error below 0.5 basis points), which are associated with average confidence exceeding 95%, and an average confidence of around 75% for more inaccurate recalls. The sharp angle in the plot, and the fairly

[11] All other tests presented in the rest of the paper are conducted out of sample, using only observations that occur after the model's training cutoff date.

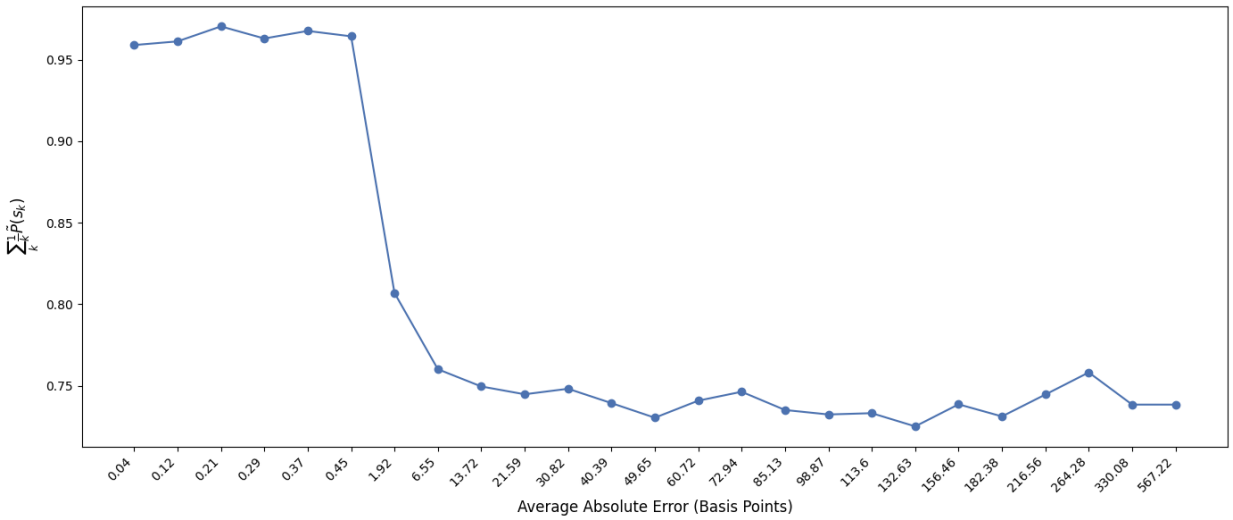


Figure 4: **Recall Confidence by Accuracy.** The horizontal axis shows buckets sorted by recall accuracy (left = most accurate, right = least accurate), while the vertical axis indicates the average inner confidence. Accuracy measures the LLM's ability to correctly retrieve a specific daily index return occurring before its training cutoff, without receiving any related information. Accurate recalls indicate the presence of look-ahead bias (Didisheim et al., 2025).

stable distribution for the high and low absolute errors, suggests a clear association between look-ahead bias and inner confidence.

The previous exercise show a relationship between pure-look ahead and inner-probabilities. We next analyze the relationship between inner confidence derived from processing news and the look-ahead bias. Specifically, we compute a daily average confidence score obtained by processing all news items in our sample using Prompt 1, following the procedure described in Section 3. Figure 5 displays this average daily confidence over the one-year period before and after the GPT-4o learning cutoff in October 2023.[12]

If a strong look-ahead bias were present, implying substantial reliance on memorization of news or associated returns, we would expect higher average confidence in the pre–learning-cutoff period relative to the post-cutoff period. Instead, we observe a small but statistically significant *increase* in inner confidence after the learning cutoff. This finding aligns with Didisheim et al. (2025), who report that the look-ahead bias is relatively minor for daily-frequency recollection of individual firm returns, and with Glasserman and Lin (2023), who document that the *distraction effect*—where general knowledge of the companies mentioned interferes with sentiment measurement—can be stronger than the look-ahead bias.

[12]The learning cutoff is defined as the date corresponding to the last observation included in the model's training dataset. This information is typically disclosed by the provider.

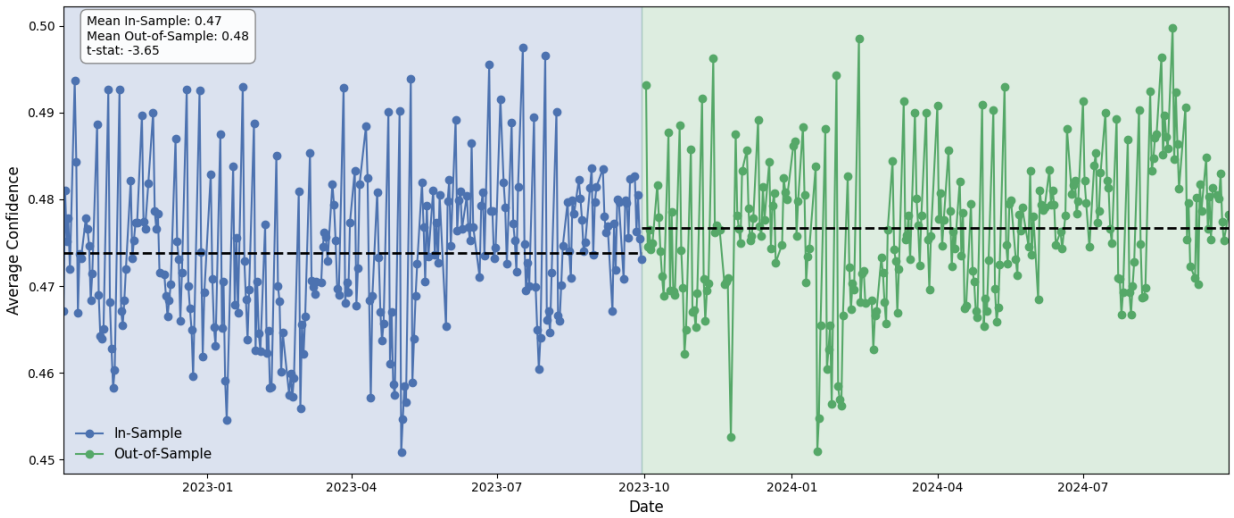


Figure 5: **Daily Average Inner Confidence In- and Out-of-Sample.** The figure shows the average daily inner confidence one year before and after GPT-4o training sample cutoff in October 2023. Shaded areas indicate in-sample (blue) and out-of-sample (green) periods. Dashed lines show mean confidence per period, with top-left inset reporting means and t-statistic.

A natural question is whether this pattern differs for firms more likely to appear in the LLMs' training data. To examine this, in Appendix G we replicate the analysis from Figure 5 for the top 10% of firms ranked by annual news coverage and market capitalization. Intuitively, these subsets are more likely to be represented in the LLMs' training corpus and therefore more exposed to potential look-ahead contamination (Didisheim et al., 2025). However, the resulting patterns closely resemble those observed for the full sample shown in Figure 5. This finding is consistent with Lopez-Lira et al. (2025), who report no systematic relationship between look-ahead bias and firm characteristics or news coverage.

Finally, we remove daily averaging and compare the distribution of individual inner-confidence values in the year before and after the learning cutoff. Figure 6 presents the distribution of news-level inner confidence for the in-sample period (left) and the out-of-sample period (right). Once again, the in- and out-of-sample distributions do not exhibit patterns consistent with a strong influence of look-ahead bias. These plots also highlight that inner probabilities tend to cluster near 0% and 100%, leading inner confidence to concentrate around 0.5. This clustering complicates the cardinal interpretation of inner probabilities. Nevertheless, these probabilities remain economically meaningful when interpreted ordinarily. For example, although the difference between 0.490 and 0.499 may appear negligible, the relative ranking carries important economic information, as discussed in Section 3.

Overall, the results in this section indicate that the internal probability can serve as a

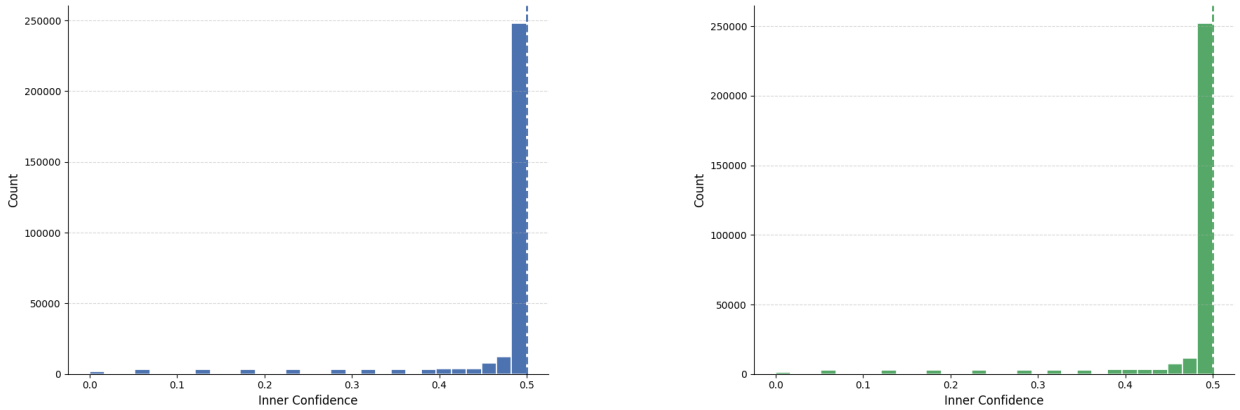


Figure 6: **Inner Confidence Distributions (Non-Ranked).** The figure shows the distribution of inner confidence in absolute terms. Left: in-sample distribution (blue) with median dashed line. Right: out-of-sample distribution (green) with median dashed line.

proxy for the risk of look-ahead bias contamination. Moreover, the findings suggest that this risk remains relatively low for current LLMs when applied to the processing daily information from financial news about individual stocks.

5 Declared Confidence

In Section 3, we used declared confidence as a robustness check when evaluating the performance of the inner confidence-based method. In this Section, we further explore the effectiveness of declared confidence as an uncertainty measure. Directly asking the LLM about its confidence is a simple and intuitive alternative to our method and several studies have adopted variations of this approach (Bybee, 2023; Fedyk et al., 2024).

While declared confidence and inner probabilities may appear similar at first glance, the extraction method introduces two key sources of discrepancy. First, the LLM cannot directly observe its own inner confidence, as it simply predicts the next token based on its training data. When asked about its confidence, it can not directly extract it but it must estimate it. Second, obtaining a declared confidence metric requires the model to generate an answer first. As detailed in Section 2, Eq. (15) indicates that the second token's probability depends on $H(S_2 \mid S_1, P)$. Because this value can exceed zero, it may introduce unforeseen biases.

We empirically test wether $H(S_2 \mid S_1, P)$ is larger than zero with an experiment that isolates the impact of the additional conditioning. Using Prompt 2, we construct a dataset of matched inner probabilities, answers, and declared confidence levels, and proceed as follows:

- We select 500 news articles where the inner confidence is $C \leq 0.2$, meaning the inner

probability falls within $0.4 \leq \tilde{p}(s_1 = A \mid P) \leq 0.6$. These are cases where the LLM has low inner confidence.

- We rerun the prompt 100 times for each of the 500 news articles, generating a total of 50,000 observations.
- We estimate the following regression:

$$D_{i,j} = \gamma_j + \beta_0, A_{i,j} + \epsilon_{i,j}, \tag{23}$$

where $D_{i,j}$ represents the declared confidence, i.e., the generated token representing the model's confidence. $A_{i,j}$ is a binary variable equal to 1 if the first generated token is "A". As explained in Section 1 this point estimate does not necessarily correspond to the most likely token based on the inner probability but it is the outcome of the decoding strategy. The index i denotes the individual observation, while j refers to the specific prompt. The key coefficient, β_0, captures whether the LLM systematically declares higher confidence when the first generated token is "A", given the same inner probabilities. As a robustness check, we perform sample splits based on realized returns and inner probabilities. In all regressions, we cluster standard errors at the prompt level.

Table 6 presents the results. The LLM consistently reports significantly higher confidence when the point estimate is "A", even in cases where the inner probability is actually higher for B (column 3) or when the market reaction is negative (column 5). These results demonstrate a distinct bias within the declared confidence metric, introduced by the feedback loop. Unlike inner probabilities, which are conditional only on the prompt, declared confidence is also influenced by the generated response. In this case, the bias may stem from the LLM arbitrarily favoring the letter "A" over B or from a general tendency to express higher confidence when predicting positive outcomes.

Next, we examine the distribution of declared confidence, shown in Figure 7 (left).

We can derive two main observations from the data presented in this figure. First, the LLM strongly favors confidence values with a single digit after the comma, even though we do not specify the number of digits in the prompt. Nearly all declared confidence estimates have one decimal digit, with few exceptions at 0.75 and 0.85. Second, the model declares a confidence of exactly 0.8 in over 35% of cases.

These distributional features likely arise from patterns in the training data and the model's difficulty distinguishing between numerical and alphabetical tokens. LLMs are trained to predict the most likely next token from a large corpus of text. In this process, tokens for words and numbers are treated identically. The training procedure provides no mechanism for the model to acquire a mathematical understanding of numbers. For in-

Table 6: **Identifying the Effect of Conditioning**
This table presents the estimation results for Eq. (23). The sample comprises matched observations of inner probabilities, point estimates, and declared probabilities. The dependent variable is the declared confidence, while the independent variable is $S_1 = A$, a binary indicator equal to 1 if the point estimate is A, as defined in Eq. (6). The first column reports the estimated coefficient for the full sample. The next two columns present results for subsamples in which the LLM assigns a probability to A above or below 0.5, respectively. The final two columns show results for the subset of news articles corresponding to firms whose return on the day of the news was above or below zero, respectively. For each estimation we report the point estimate of the parameters and its standard error (SE), and the sample size. *, **, and *** denote statistical significance at the 10%, 5%, and 1% level, respectively.

	(1)	(2)	(3)	(4)	(5)
$A_{i,j}$	0.0337***	0.0432***	0.0216***	0.0367***	0.0307***
	(0.0038)	(0.0047)	(0.0048)	(0.0055)	(0.0052)
Article FE	YES	YES	YES	YES	YES
Sample	Full	$\tilde{p}(A) > 0.5$	$\tilde{p}(A) \leq 0.5$	$r_{i,t} > 0$	$r_{i,t} \leq 0$
Observations	49,999	22,944	27,055	25,000	24,999
R^2	0.5684	0.5154	0.5993	0.5695	0.5673

stance, in a simple random number generation task, where one might anticipate an LLM would simulate a random distribution, the actual output probabilities reveal a distinct bias. Rather than sampling uniformly, GPT-4o shows a marked tendency to favor the number 7 (Figure 8).

These findings caution against attributing inherent mathematical understanding to LLMs or treating their numerical outputs as statistically meaningful. For instance, a declared confidence score of 2 should not be interpreted as twice the confidence of 1. More broadly, researchers should avoid treating LLM-generated tokens as ordinal measures.

In addition to the aforementioned bias, the distribution of declared confidence is also less granular and with many observations at identical values, Figure 7 (right) indicates a positive relationship between accuracy and declared confidence. Nevertheless, this distribution is not as economically significant as the inner confidence one as shown in the portfolio exercise presented in Section 3 in Table 2.

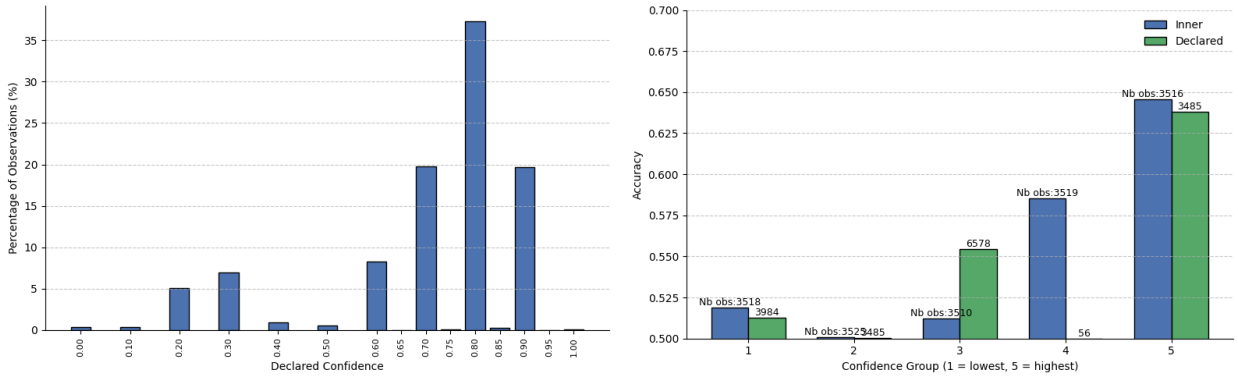


Figure 7: **Declared Confidence Distribution and Accuracy.** *Left:* Distribution of declared confidence values across observations, expressed as a percentage of the total sample. *Right:* Accuracy for each confidence quintile (1 = lowest, 5 = highest) comparing inner and declared confidences, with sample counts annotated on each bar.

6 Conclusion

This paper argues that leveraging the inner probabilities produced by LLMs substantially enhances performance, interpretability, and usefulness for economic research compared to relying solely on generated tokens. Our methodology establishes an entropy-based framework for uncertainty quantification that is designed to be semantics-aware and feasible for large-scale applications. We show that by restricting output to a fixed set of labels and extracting inner probabilities directly from the model's conditional distribution, we can construct an interpretable confidence metric that is monotonically related to Shannon entropy. This approach effectively eliminates the biases and unpredictability introduced by opaque decoding strategies and the LLM feedback loop.

Our empirical analyses confirm that these inner probabilities carry significant economic value. Predictions with high inner confidence are systematically more accurate in financial tasks, achieving an accuracy rate of 64.5% in the highest decile compared to close to 50% in the lowest quintiles in a sentiment classification exercise. Filtering by confidence substantially improves performance: a news-based high-confidence portfolio achieved a Sharpe ratio of 5.15 while low-confidence predictions yielded returns not statistically different from zero.

We identify critical determinants of model confidence. Confidence rises during clear market trends and diminishes in conditions of increased volatility or uncertainty. Unambiguous news events, such as lawsuits or product launches, typically induce higher confidence, whereas complex or ambiguous news reduces confidence. Through prompt manipulation experiments designed to simulate shifts in prior beliefs or uncertainty, we find the model updates its beliefs consistent with Bayesian reasoning. We also find that confidence is strictly related to

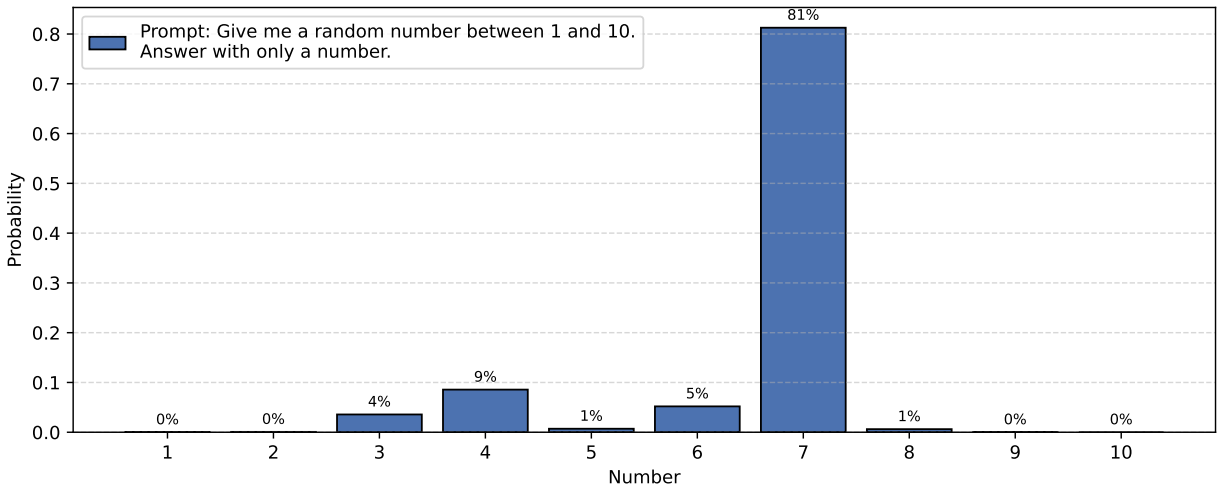


Figure 8: **Random Number Generation.** The figure shows the inner probabilities generated by the model when prompted to select a random number between 1 and 10.

the look-ahead bias, with the model being more confident when it is able to recall an answer as opposed to guessing it.

Our paper also critically evaluates methods based on self-reported confidence and reveals their weaknesses, such as vulnerability to decoding strategies and sensitivity to prompt variations. These limitations reinforce the advantage of relying on inner probabilities to obtain credible confidence measures.

Collectively, these findings suggest that inner probabilities offer an informative, scalable, and theoretically grounded metric for evaluating LLM outputs in finance. While the generated tokens are mere point estimates, the underlying distributions capture essential nuances that distinguish high-conviction insights from unreliable hallucinations. By utilizing this measure, researchers and practitioners can reduce model opacity and integrate LLM predictions more systematically into economic applications.

References

Acikalin, Utku, Tolga Caskurlu, Gerard Hoberg, and Gordon M Phillips, 2025, Intellectual property protection lost and competition: An examination using large language models, *FEB-RN Research Paper* .

Allena, Rohit, 2025, Confident risk premiums and investments using machine learning uncertainties, *The Review of Financial Studies* hhaf087.

Ashokkumar, Ashwini, Luke Hewitt, Isaias Ghezae, and Robb Willer, 2024, Predicting results of social science experiments using large language models, arXiv preprint arXiv:2504.01167.

Babina, Tania, Anastassia Fedyk, Alex He, and James Hodson, 2024, Artificial intelligence, firm growth, and product innovation, *Journal of Financial Economics* 151, 103745.

Brown, Tom B, et al., 2020, Language models are few-shot learners, *Advances in Neural Information Processing Systems (NeurIPS)* 33, 1877–1901.

Bybee, Leland, 2023, Surveying generative ai's economic expectations, arXiv preprint arXiv:2305.02823.

Bybee, Leland, Bryan Kelly, Asaf Manela, and Dacheng Xiu, 2024, Business news and business cycles, *The Journal of Finance* 79, 3105–3147.

Bybee, Leland, Bryan T Kelly, Asaf Manela, and Dacheng Xiu, 2020, The structure of economic news, Available at SSRN 3522305.

Caragea, Doina, Mark Chen, Theodor Cojoianu, Mihai Dobri, Kyle Glandt, and George Mihaila, 2020, Identifying fintech innovations using bert, Proceedings of the 2020 IEEE International Conference on Big Data (Big Data).

Chang, Anne, Xi Dong, Xiumin Martin, and Changyun Zhou, 2023, AI democratization, return predictability, and trading inequality, Available at SSRN 4543999.

Chen, Yifei, Bryan T Kelly, and Dacheng Xiu, 2022, Expected returns and large language models, Available at SSRN 4416687.

Cheng, Qiang, Pengkai Lin, and Yue Zhao, 2024, Does generative ai facilitate investor trading? evidence from chatgpt outages, Available at SSRN 4872189.

Didisheim, Antoine, Martina Fraschini, and Luciano Somoza, 2025, AI's predictable memory and its implications for finance, *Economics Letters* 256.

Dubey, Abhimanyu, et al., 2024, The llama 3 herd of models, arXiv preprint arXiv:2407.21783.

Eisfeldt, Andrea L, and Gregor Schubert, 2024, AI and finance, Available at SSRN 4988553.

Engelberg, Joseph, Asaf Manela, William Mullins, and Luka Vulicevic, 2025, Entity neutering, Available at SSRN 5182756.

Fan, Angela, Mike Lewis, and Yann Dauphin, 2018, Hierarchical neural story generation, Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers).

Fedyk, Anastassia, Ali Kakhbod, Peiyao Li, and Ulrike Malmendier, 2024, Chatgpt and perception biases in investments: An experimental study, Available at SSRN 4787249.

Glasserman, Paul, and Caden Lin, 2023, Assessing look-ahead bias in stock return predictions generated by gpt sentiment analysis, arXiv preprint arXiv:2309.17322.

He, Songrun, Linying Lv, Asaf Manela, and Jimmy Wu, 2025, Chronologically consistent large language models, *arXiv preprint arXiv:2502.21206* .

Holtzman, Ari, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi, 2019, The curious case of neural text degeneration, arXiv preprint arXiv:1904.09751.

Jobson, J Dave, and Bob M Korkie, 1981, Performance hypothesis testing with the sharpe and treynor measures, *Journal of Finance* 889–908.

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

Krockenberger, Vanessa S, Anthony Saunders, Sascha Steffen, and Paulina M Verhoff, 2024, CovenantAI-new insights into covenant violations, Available at SSRN 4640653.

Kuhn, Lorenz, Yarin Gal, and Sebastian Farquhar, 2023, Semantic uncertainty: Linguistic invariances for uncertainty estimation in natural language generation, in *International Conference on Learning Representations (ICLR)*.

Levy, Bradford, 2024, Caution ahead: Numerical reasoning and look-ahead bias in ai models, Available at SSRN 5082861.

Li, Shuting, Zhen Shi, Yusen Xia, and Baozhong Yang, 2025, The value of stock analysis in news, *Available at SSRN 5754262* .

Lin, Stephanie, Jacob Hilton, and Owain Evans, 2022, Teaching models to express their uncertainty in words, *Transactions on Machine Learning Research* .

Lopez-Lira, Alejandro, and Yuehua Tang, 2023, Can chatgpt forecast stock price movements? return predictability and large language models, arXiv preprint arXiv:2304.07619.

Lopez-Lira, Alejandro, Yuehua Tang, and Mingyin Zhu, 2025, The memorization problem: Can we trust llms' economic forecasts?, *arXiv preprint arXiv:2504.14765* .

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

Ludwig, Jens, Sendhil Mullainathan, and Ashesh Rambachan, 2025, Large language models: An applied econometric framework, Technical report, National Bureau of Economic Research.

Memmel, Christoph, 2003, Performance hypothesis testing with the sharpe ratio, Available at SSRN 412588.

Radford, Alec, et al., 2019, Language models are unsupervised multitask learners, OpenAI blog post.

Sarkar, Suproteem K., and Keyon Vafa, 2024, Lookahead Bias in Pretrained Language Models, Available at SSRN 4754678.

Sutskever, Ilya, Oriol Vinyals, and Quoc V Le, 2014, Sequence to sequence learning with neural networks, *Advances in Neural Information Processing Systems* 27, 3104–3112.

Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin, 2017, Attention is all you need, *Advances in neural information processing systems* 30.

Wei, Jason, and al., 2022, Chain-of-thought prompting elicits reasoning in large language models, in *Advances in Neural Information Processing Systems*, volume 35, 24824–24837.

Appendix

A Proof of Lemma 1

Proof. Both $H_{\text{bin}}(s \mid P)$ and $C(s \mid P)$ can be expressed as a function of $p \equiv p(s \mid P)$: $H_{\text{bin}}(s \mid P) = H(p)$ and $C(s \mid P) = C(p)$. It is easy to see that both $H(p)$ and $C(p)$ are symmetric about $p = 0.5$, i.e. $H(p) = H(1-p)$ and $C(p) = C(1-p)$, so it suffices to study $p \in [1/2, 1]$. In this region, Eq. (6) simplifies to $C(p) = p - 0.5$.

From Eq. (5),

$$H'(p) = \log \frac{1-p}{p}.$$

For $p \in (1/2, 1)$, we have $H'(p) < 0$, i.e., H is strictly decreasing on $(1/2, 1)$, while the opposite is true for $C(p)$. This already implies that ranking by $H(p)$ is equivalent to ranking by $C(p)$ for $p \in [1/2, 1]$.

Next, the function g can be constructed through the inverse function of H subject to a change of variable. Given $C = p - 0.5$, define $\Phi(C)$ for $C \in [0, 1/2]$ as:

$$\Phi(C) \equiv H(C + 0.5).$$

Differentiating Φ gives

$$\Phi'(C) = \log \frac{\frac{1}{2} - C}{\frac{1}{2} + C},$$

and $\Phi'(C) < 0$ for $C \in (0, 1/2)$. Therefore, Φ is continuous on $[0, 1/2]$ and strictly decreasing on $(0, 1/2)$, and thus a bijection between $[0, 1/2]$ and $[H(1), H(1/2)] = [0, \log 2]$. The inverse function theorem (applied to a continuous strictly monotone map on a compact interval) guarantees that Φ admits a continuous inverse $\Phi^{-1} : [0, \log 2] \to [0, 1/2]$. Setting $g = \Phi^{-1}$ yields $C = g(H(p))$ for all $p \in [1/2, 1]$. By symmetry the same results hold for $p \in [0, 1/2]$. □

B Multi-token Uncertainty

To illustrate the multi-token approach discussed in Section 2.3, we use Prompt 5 to query GPT-4o 1,000 times with a temperature of 2, thereby generating 1,000 multi-token sequences $S^{(1)}, S^{(2)}, \cdots, S^{(1000)}$. Setting a high temperature ensures substantial diversity across the generated sequences $S^{(j)}$. At the same time, the OpenAI API provides the default *inner*

probabilities prior to the temperature transformation.

```
In five words or fewer, what is the most likely cause of a U.S. recession
 in 2030?
```

Prompt 5: **Multi-token illustration:**
Simple prompt designed to generate short answer on economic topics to illustrate multi-token uncertainty measure.

We use these probabilities to compute the conditional probability of each sequence, $p(S \mid P)$, via the chain rule (Equation (10)). We then focus on the 20 most likely sequences according to this conditional probability and compute the normalized conditional probability $p_M(S^{(j)} \mid P)$, as defined in Equation (12).

Table 6 reports the results. While this sequence-level approach offers flexibility and generality, it introduces semantic ambiguity: identical meanings expressed using different phrasings are treated as distinct sequences. In most applications, addressing this issue would require the construction of semantic clusters (see, e.g. Kuhn et al., 2023), which in turn introduces additional modeling opacity and computational overhead.

Consequently, as argued in Section 2.3, in applications where predefined categories or semantic clarity are paramount, token-level responses are often preferable. Moreover, careful prompt design can bridge the gap between open-ended generation and structured classification. For example, rather than eliciting free-form responses, one may prompt the model to enumerate candidate answers in a structured list, each associated with a token-level identifier such as a letter. The model can then be instructed to select the most plausible option via a single-token response (e.g., "—b"), thereby preserving interpretability while retaining the generative richness of the initial prompt. This hybrid strategy combines the semantic granularity of generation with the analytical clarity of classification.

C Technical Considerations

Extracting Inner Probabilities A brute-force estimation of inner probabilities–sampling numerous generated texts from the same prompt–would be computationally expensive and yield only an approximate estimate. Fortunately, such an approach is unnecessary. Most LLM providers, including all major open-source implementations, offer functionality to return both the generated text *and* the associated token-level probabilities. In the case of OpenAI's GPT models, these probabilities can be retrieved with minimal modification to

Table 7: **Multi-Token Illustration**
This table displays the top 20 completions sampled from GPT-4o in response to the prompt *"In five words or fewer, what is the most likely cause of a U.S. recession in 2030?"*, ranked by normalized likelihood ($p_M(S^{(j)} \mid P)$).

message	$p_M(S^{(j)} \mid P)$
Economic policy missteps	0.2811
Technological disruption and job displacement	0.2213
Economic policy mismanagement	0.0631
Technological disruption or geopolitical tensions	0.0590
Technological disruption and job loss	0.0506
Debt crisis or technological disruption	0.0460
Monetary policy or geopolitical tensions	0.0447
Global economic slowdown	0.0327
Rising interest rates and inflation	0.0306
Tight monetary policy	0.0290
Global economic downturn	0.0272
Interest rate hikes	0.0178
Monetary policy or economic imbalance	0.0177
Global economic instability	0.0167
Rising debt and inflation pressures	0.0140
Debt crisis or geopolitical tensions	0.0132
High interest rates or inflation	0.0102
Technological disruption or economic imbalance	0.0101
Technological disruption or policy missteps	0.0087
Monetary tightening or geopolitical tensions	0.0064

the API call–typically by adding two lines of code to the prompt processing request (see Figure 9).

Without Logprobs

```
client.chat.completions.create(
    messages=[{
        "role": "user",
        "content": prompt,
    }],
    model="chatgpt-4o-latest"
)
```

With Logprobs

```
client.chat.completions.create(
    messages=[{
        "role": "user",
        "content": prompt,
    }],
    model="chatgpt-4o-latest",
    logprobs=True,
    top_logprobs=10
)
```

Figure 9: The figure shows the necessary code for extracting inner probabilities via the OpenAI API. The 'logprobs=True" parameter enables access to token-level probabilities for the next predicted tokens, returned as log-probabilities. This is essential for obtaining $\tilde{p}(s_1 \mid P)$ from model outputs.

Parsing and Failed Prompts Throughout the paper, we present various tests using different prompts. To facilitate token parsing, we sometimes request that results be generated within special symbols, such as $[\cdot]$ or $< \cdot >$. This structure helps automate the processing of large volumes of generated text.

However, some prompts occasionally fail to produce parsable text. This can occur due to a rare token draw or a poor conditional distribution of the model. When this happens, we simply discard the observation. As a result, the total number of observations may not always sum precisely to the declared sample size. These discrepancies are minimal and unlikely to affect the significance of our results.

Reproducibility and Hardware Non-determinism For all our tests, we use the API provided by OpenAI. Our default model is *gpt-4o*, which we compare to GPT-3.5 using *gpt-3.5-turbo-0125*. As mentioned above, we obtain the inner probabilities, $\tilde{p}(s_1 \mid P)$, directly from the API. However, obtaining the exact same probability values across repeated runs is often impossible, even with identical settings (e.g., temperature set to 0). This is caused by the non-associative nature of floating-point arithmetic on GPUs (**?**). Large-scale inference relies on massive parallelism, where the order of operations in summation (such as parallel reductions) varies based on microscopic timing differences between GPU threads. Consequently, $(a + b) + c$ may not equal $a + (b + c)$ at the level of machine precision. This hardware-level noise introduces slight variations in the model's output logits. However, this variation does not affect any of the experiments or arguments in this paper; it merely adds noise, which we mitigate by conducting sufficiently large tests.

Quantization LLMs require immense computational resources, particularly for inference, due to the vast number of parameters and operations involved. To reduce computational cost and memory usage, many providers implement *quantization*–a technique that reduces the precision of model weights and activations.

In essence, quantization approximates high-precision floating-point numbers (typically 32-bit floats) using lower-precision representations, such as 16-bit or even 8-bit integers. For example, a model weight originally represented as a float32 value of 0.1234567 may be rounded to an int8-equivalent level such as 0.12 or 0.125. This process significantly decreases storage requirements and accelerates inference by enabling the use of faster, low-precision arithmetic.

However, quantization introduces rounding error and numerical imprecision. While these effects may be negligible for some applications, they can materially affect analyses–such as ours–that rely on subtle differences in token-level probabilities. As we show later in this paper, small variations in the model's inner probabilities can translate into economically meaningful differences. If such distinctions are flattened or distorted by quantization, the resulting analyses may be biased or less informative. Therefore, whenever precision matters (as is the case when evaluating $\tilde{p}(s_1 = A \mid P)$) quantization must be treated not merely as a technical footnote but as a first-order concern.

D Portfolio Construction Methodology

Benchmark:

The benchmark portfolio is an equally weighted strategy that buys companies with good news based on the LLM's predictions and shorts those with bad news. On any given day, a company may have more than one news item; we aggregate the signals at the firm level. Specifically, we define the signal as positive if a company receives more positive signals than negative ones.

Formally, let each news item for firm i on day t be represented by a discrete generated token $S_{i,k,t} \in \{A, B\}$, where A represents good news and B represents bad news.

We define the daily signal for firm i on day t by a single letter $Z_{i,t}$ taking values 1 for a "good" signal and -1 for a "bad" signal. The signal is good if the number of good new for

firm i on day t is larger than the number of bad news for the same firm on the same day:

$$Z_{i,t} \;=\; \begin{cases} 1, & \text{if } \sum_k \mathbf{1}\{S_{i,k,t} = A\} > \sum_k \mathbf{1}\{S_{i,k,t} = B\}, \\ -1, & \text{otherwise.} \end{cases} \tag{24}$$

We construct our benchmark portfolio by placing \$1 in the long ("good") leg and \$1 in the short ("bad") leg. Let $N_t^+ = \sum_{i=1}^N \mathbf{1}\{Z_{i,t} = 1\}$ and $N_t^- = \sum_{i=1}^N \mathbf{1}\{Z_{i,t} = -1\}$ denote the number of long and short positions on day t, respectively. If $r_{i,t+1}$ is the next-day return of firm i, the benchmark portfolio return on day t is:

$$r_t^{\text{bench}} \;=\; \frac{1}{N_t^+} \sum_{\{i:Z_{i,t}=1\}} r_{i,t+1} \;-\; \frac{1}{N_t^-} \sum_{\{i:Z_{i,t}=-1\}} r_{i,t+1}. \tag{25}$$

Confidence Portfolios:

We now exploit the full distributional information from $\tilde{p}\big(S_{i,k,t} = A \mid P_{i,k,t}\big)$ rather than relying on the discrete token $S_{i,k,t}$. Define the confidence for each news item k of firm i on day t as

$$C_{i,k,t} \;=\; \Big|\tilde{p}\big(S_{i,k,t} = A \mid P_{i,k,t}\big) \;-\; 1/2\Big|. \tag{26}$$

High Confidence Portfolio. We select only those items whose confidence exceeds a threshold $c(\nu_t)$ where ν_t is a hyperparameter defined in the next subsection:

$$\overline{\mathcal{N}}_{i,t} \;=\; \Big\{k : \Big|\tilde{p}\big(S_{i,k,t} = A \mid P_{i,k,t}\big) - 1/2\Big| > c(\nu_t)\Big\}. \tag{27}$$

For those selected items, we average the probabilities:

$$\overline{p}_{i,t} \;=\; \frac{1}{\big|\overline{\mathcal{N}}_{i,t}\big|} \sum_{k \in \overline{\mathcal{N}}_{i,t}} \tilde{p}\big(S_{i,k,t} = A \mid P_{i,k,t}\big). \tag{28}$$

We then define the high-confidence signal $\overline{Z}_{i,t}$:

$$\overline{Z}_{i,t} \;=\; \begin{cases} +1, & \text{if } \overline{p}_{i,t} > 0.5, \\ -1, & \text{otherwise.} \end{cases} \tag{29}$$

As with the benchmark, we form two equally weighted long and short legs to build a portfolio,

based on $\overline{Z}_{i,t}$. Define

$$N_t^{+,h} \;=\; \sum_{i=1}^{N} \mathbf{1}\{\overline{Z}_{i,t} = +1\}, \quad N_t^{-,h} \;=\; \sum_{i=1}^{N} \mathbf{1}\{\overline{Z}_{i,t} = -1\}. \tag{30}$$

The daily return of this high-confidence portfolio is

$$r_t^{\text{high-conf}} \;=\; \frac{1}{N_t^{+,h}} \sum_{\{i:\overline{Z}_{i,t}=+1\}} r_{i,t+1} \;-\; \frac{1}{N_t^{-,h}} \sum_{\{i:\overline{Z}_{i,t}=-1\}} r_{i,t+1}. \tag{31}$$

Low Confidence Portfolio. We also construct a low-confidence portfolio by selecting the news items whose confidence is at or below the threshold $c(\nu_t)$:

$$\underline{\mathcal{N}}_{i,t} \;=\; \Big\{k : \Big|\tilde{p}\big(S_{i,k,t} = A \mid P_{i,k,t}\big) - 0.5\Big| \le c(\nu_t)\Big\}. \tag{32}$$

Define

$$\underline{p}_{i,t} \;=\; \frac{1}{\big|\underline{\mathcal{N}}_{i,t}\big|} \sum_{k \in \underline{\mathcal{N}}_{i,t}} \tilde{p}\big(S_{i,k,t} = A \mid P_{i,k,t}\big). \tag{33}$$

Then the low-confidence signal $\underline{Z}_{i,t}$ is

$$\underline{Z}_{i,t} \;=\; \begin{cases} +1, & \underline{p}_{i,t} > 0.5, \\ -1, & \underline{p}_{i,t} \le 0.5. \end{cases} \tag{34}$$

We form the low-confidence portfolio analogously:

$$N_t^{+,\ell} \;=\; \sum_{i=1}^{N} \mathbf{1}\{\underline{Z}_{i,t} = +1\}, \quad N_t^{-,\ell} \;=\; \sum_{i=1}^{N} \mathbf{1}\{\underline{Z}_{i,t} = -1\}, \tag{35}$$

$$r_t^{\text{low-conf}} \;=\; \frac{1}{N_t^{+,\ell}} \sum_{\{i:\underline{Z}_{i,t}=+1\}} r_{i,t+1} \;-\; \frac{1}{N_t^{-,\ell}} \sum_{\{i:\underline{Z}_{i,t}=-1\}} r_{i,t+1}. \tag{36}$$

Threshold Selection for the High-Confidence Portfolio

We determine the threshold parameter ν_t for $c(\nu_t)$ using an expanding-window procedure that starts after a minimum of one year of data. This approach ensures that ν_t is chosen exclusively using historical data (up to each recalibration date), thus avoiding look-ahead bias and preserving out-of-sample integrity. Concretely, we implement the following steps:

1. **Define Grid:** For each recalibration date, we consider a grid of values for ν_t ranging

from 0 to 0.5 in increments of 0.01.

2. **Quantile Threshold:** Given a chosen ν_t, the quantity $c(\nu_t)$ is defined as the ν_t-quantile of the confidence on the full expanding window. For instance, if $\nu_t = 0.25$, then $c(\nu_t)$ is the first quartile in our *training* sample.
3. **Compute High-Conf Returns:** For each ν_t on the grid, we construct the High-Confidence portfolio (using data up to day t) and compute its Sharpe ratio over the expanding window.
4. **Select ν_t:** We choose the ν_t that maximizes the Sharpe ratio of the High-Confidence portfolio over the expanding window. This selection is based solely on past data, preventing any form of "cheating" for out-of-sample analysis.
5. **Monthly Recalibration:** We repeat this procedure at the start of each month. The selected ν_t is then held fixed for that month and recalibrated again at the next month's start using the updated expanding window.

E Fixed-Threshold Portfolios

Table 8: **Performance of Confidence-Based Portfolios with Fixed Threshold**
This table reports out-of-sample performance of two long-short investment strategies based on the model confidence in its classification: the *High-Confidence* portfolio including only news for which the model generated a confident classification, and the *Low-Confidence* portfolio including non-confident classifications only. Annualized return, volatility, and Sharpe ratio are reported in the first three columns. The first column indicates the percentage of the total sample that is allocated to the *Low-confidence* portfolio. The analysis starts in October 2023, following the training cutoff of GPT-4o. All portfolios are equally weighted and rebalanced daily.

Quantile	Confidence Level	Position	Annualized Mean Return	Annualized Volatility	Sharpe Ratio	Number of Assets
10%	High	Long	0.0937	0.1416	0.6613	256.58
		Short	-0.3709	0.1649	-2.2491	252.96
		Long–Short	0.4645	0.0954	4.8706	254.77
	Low	Long	-0.4427	0.2807	-1.5771	34.41
		Short	-0.2603	0.2433	-1.0701	39.77
		Long–Short	-0.1885	0.3294	-0.5721	36.82
20%	High	Long	0.0884	0.1418	0.6232	248.91
		Short	-0.4075	0.1678	-2.4283	230.69
		Long–Short	0.4959	0.0987	5.0254	239.80
	Low	Long	-0.3501	0.2279	-1.5364	50.96
		Short	-0.2088	0.1893	-1.1032	71.41
		Long–Short	-0.1308	0.2180	-0.6001	61.10
30%	High	Long	0.0961	0.1406	0.6835	233.19
		Short	-0.3987	0.1706	-2.3379	220.62
		Long–Short	0.4948	0.1043	4.7432	226.90
	Low	Long	-0.1429	0.2162	-0.6611	73.41
		Short	-0.3203	0.1786	-1.7936	83.60
		Long–Short	0.1867	0.1951	0.9569	78.25

F Confidence Conditioning Prompts

```
Yesterday, most analysts disagreed on the prospects of the firm {target}.
 Today, new news has emerged. Based on this news, please assess the
expected return for the stock. Answer A if the news is good, B if the
news is bad. Is this financial news good or bad for the stock price of {
target} in the short term? Please start by writing only the letter A or B
. Add no other formatting. ARTICLE: {headline} {body}
```

Prompt 6: **Prior Disagreement**
The prompt asks the LLM to predict the market reaction (positive or negative) to a news item related to a specific listed firm with an initial conditioning on its prior. The brackets $\{\cdots\}$ indicate dynamic inserts where we place the *target* (firm's ticker), *headline* (article's headline), and *body* (main text of the news article).

```
Most analysts agree about the consequence of the news below for stock
with ticker {target}. Based on this news, please say if you think the
return for the stock with ticker {target} will be positive or negative in
 the short term. Answer A if good news, B if bad news. Is this financial
news good or bad for the stock price of {target} in the short term?
Please start by writing only a letter A or B. Add no other formatting.
ARTICLE: {headline} {body}
```

Prompt 7: **Analyst Consensus**
The prompt asks the LLM to predict the market reaction (positive or negative) to a news item related to a specific listed firm with an initial conditioning on the market interpretation. The brackets $\{\cdots\}$ indicate dynamic inserts where we place the *target* (firm's ticker), *headline* (article's headline), and *body* (main text of the news article).

```
Most analysts disagree about the consequence of the news below for stock
with ticker {target}. Based on this news, please say if you think the
return for the stock with ticker {target} will be positive or negative in
 the short term. Answer A if good news, B if bad news. Is this financial
news good or bad for the stock price of {target} in the short term?
Please start by writing only a letter A or B. Add no other formatting.
ARTICLE: {headline} {body}
```

Prompt 8: **Analyst Disagreement**
The prompt asks the LLM to predict the market reaction (positive or negative) to a news item related to a specific listed firm with an initial conditioning on the market interpretation. The brackets $\{\cdots\}$ indicate dynamic inserts where we place the *target* (firm's ticker), *headline* (article's headline), and *body* (main text of the news article).

```
Confidence, clarity, conviction, precision, certainty. Based on the news
below, please say if you think the return for the stock with ticker {
target} will be positive or negative in the short term. Answer A if good
news, B if bad news. Is this financial news good or bad for the stock
price of {target} in the short term? Please start by writing only a
letter A or B. Add no other formatting. ARTICLE: {headline} {body}
```

Prompt 9: **Positive Noise**
The prompt asks the LLM to predict the market reaction (positive or negative) to a news item related to a specific listed firm with an an initial conditioning with positive noise. The brackets $\{\cdots\}$ indicate dynamic inserts where we place the *target* (firm's ticker), *headline* (article's headline), and *body* (main text of the news article).

```
Doubt, anxiety, mistrust, uncertainty, hesitation. Based on the news
below, please say if you think the return for the stock with ticker {
target} will be positive or negative in the short term. Answer A if good
news, B if bad news. Is this financial news good or bad for the stock
price of {target} in the short term? Please start by writing only a
letter A or B. Add no other formatting. ARTICLE: {headline} {body}
```

Prompt 10: **Negative Noise**
The prompt asks the LLM to predict the market reaction (positive or negative) to a news item related to a specific listed firm with an an initial conditioning with negative noise. The brackets $\{\cdots\}$ indicate dynamic inserts where we place the *target* (firm's ticker), *headline* (article's headline), and *body* (main text of the news article).

G Look-ahead Bias Robustness

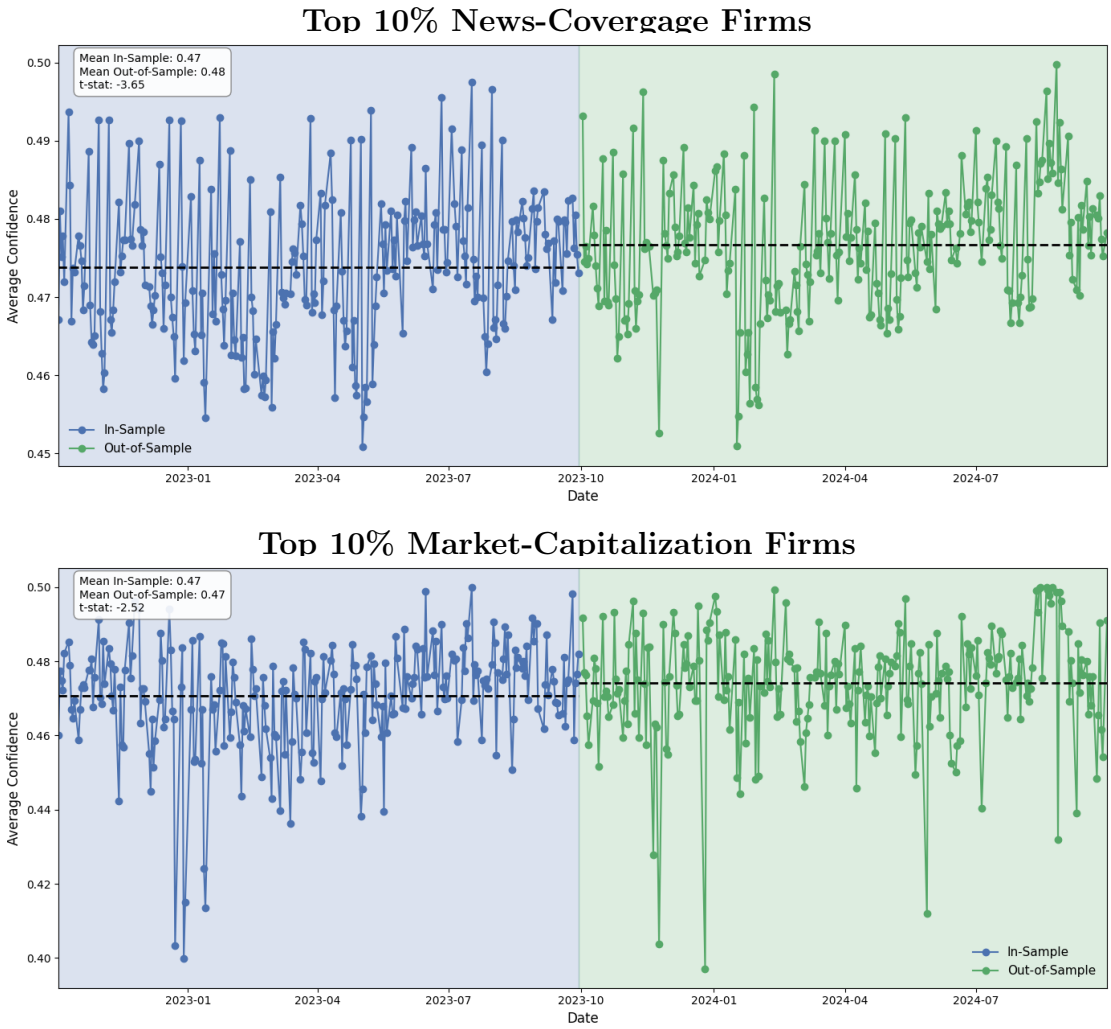


Figure 10: **Daily Average Inner Confidence — Robustness Checks.** This figure replicates Figure 5 for two sub-samples: firms in the top 10% of (a) news coverage and (b) market capitalization, computed annually so that thresholds are defined within each year. The patterns are consistent with the baseline, showing similar shifts in inner confidence around the GPT-4o training sample cutoff in October 2023.