A Test of Lookahead Bias in LLM Forecasts

Zhenyu Gao Wenxi Jiang Yutong Yan

Department of Finance, CUHK Business School

ABFER 2026

Motivation: Are LLMs Forecasting, or Just Recalling?

LLMs outperform traditional methods in making financial forecasts (Lopez-Lira and Tang, 2023; Jha et al., 2024).

Forecast Prompt

"On 2020-07-28 , Kodak Triples on Loan to Make Covid-19 Drug Ingredients. Do you think this news is **good**, **bad**, or **neutral** for the stock price of Kodak ?"

LLM responds: {good}, confidence: 1.0 — *Kodak's stock surged* +318% *the next day.*

Two possibilities:

- **Genuine reasoning:** the model *reasons* the loan is good news for Kodak.
- **Memorization:** the model *recalls* the outcome it has seen — the surge was widely covered by media and likely in the LLM's training data.

This paper

A test to distinguish memorization from genuine reasoning in LLM forecasts.

Step 1: Measure memorization

Outcome-blind date-only recall query. Firm name + outcome date $(t{+}1)$, nothing else:

Recall Prompt

"On 2020-07-29, did the closing stock price of Kodak go up or down compared to the previous trading day? Respond with exactly one word: up, down, or unknown."

We look at the token probability of each response:

$$P_{\text{up}} \approx 0.9999, \qquad P_{\text{down}} < 0.0001, \qquad P_{\text{unknown}} < 0.0001.$$

Construct the **Lookahead Propensity** (LAP):

$$\text{LAP}^{P(\text{known } t+1)} \equiv P_{\text{up}} + P_{\text{down}}.$$

Validation regression:

$$Y_{i,t+1} \;=\; \alpha_i \;+\; \lambda_t \;+\; \boxed{\beta} \cdot (P_{\text{up}} - P_{\text{down}}) \;+\; \epsilon_{i,t+1}.$$

$\Rightarrow \hat{\beta} > 0$ is a diagnostic for **memorization**.

Step 2: Contamination test

True DGP: $Y_{t+1} = \mu(X_t) + \varepsilon_{t+1}$, where $\mu(X_t) = \mathbb{E}[Y_{t+1} \mid X_t]$.

Lookahead bias (Sarkar and Vafa, 2024): the LLM forecast $\hat{\mu}_t$ is a function of both X_t and Y_{t+1}, i.e., it already contains information about the future shock ε_{t+1}:

$$\mathrm{Cov}(\hat{\mu}_t, \varepsilon_{t+1}) \neq 0.$$

Contaminated forecast:

$$\hat{\mu}_t = \mu(X_t) + \boxed{\gamma} \cdot L_t \cdot \varepsilon_{t+1}.$$

- $L_t \in [0, 1]$: *memorization strength*; already confirmed in Step 1, proxied by LAP.
- $\gamma \in [0, 1)$: *contamination loading*, scalar, unobserved.

Interpretation:
L_t = whether or not the LLM has memorized the outcome;
γ = how much of that memorized outcome leaks into the LLM's forecast.

What's left

γ is unobserved. How can we still test $\gamma > 0$?

Step 2 — Forecast contamination test

Contamination regression:

$$Y_{t+1} = \beta_1 \hat{\mu}_t + \beta_2 \,\mathrm{LAP} + \boxed{\beta_3}\,(\mathrm{LAP} \times \hat{\mu}_t) + \epsilon_{t+1}.$$

- **High LAP, high LLM accuracy** ($\beta_3 > 0$):
  The model predicts well only when it knows the realized outcome.
  $\Rightarrow$ Suggests *recall*, not reasoning.
- **LAP orthogonal to LLM accuracy** ($\beta_3 = 0$):
  Accuracy is unrelated to the model's memorization.
  $\Rightarrow$ Suggests *reasoning*, not memorization.

Lookahead Bias

The sign of $\hat{\beta}_3$ is our diagnostic for lookahead bias.

Step 2 — Forecast contamination test (Cont'd)

Contamination regression:

$$Y_{t+1} = \beta_1 \hat{\mu}_t + \beta_2 \, \mathrm{LAP} + \boxed{\beta_3} \, (\mathrm{LAP} \times \hat{\mu}_t) + \epsilon_{t+1}.$$

Proposition 1

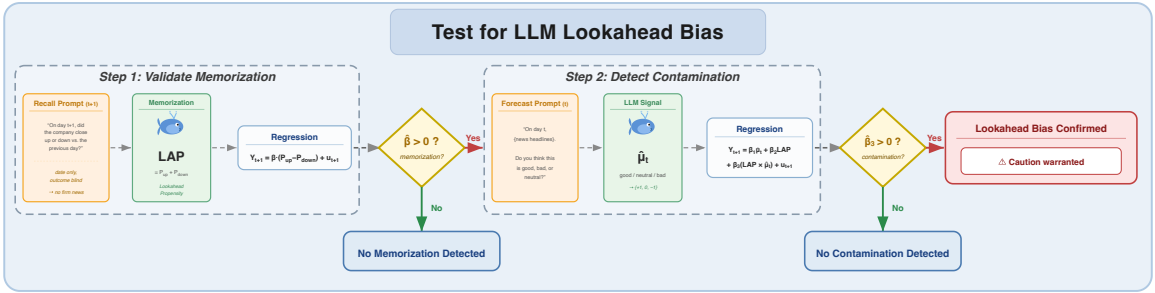
Under standard assumptions, β_3 satisfies:

(a) Under the null $\gamma = 0$: $\beta_3 = 0$.

(b) Under the alternative $\gamma > 0$: $\beta_3 > 0$.

$\Rightarrow \hat{\beta}_3 > 0$ is a diagnostic for contamination loading $\gamma > 0$.

Method overview



Data

Two applications:

- **News Headlines:** predict next-day stock returns from headlines, following Lopez-Lira and Tang (2023) (out-of-sample with ChatGPT-4).
  - Bloomberg news headlines and CRSP stock returns.
  - IS: $N = 91{,}357$ headlines, 1,587 firms (Jan 2012 – Dec 2023).
  - OOS: $N = 7{,}568$ (year 2024).
- **Earnings Calls:** predict capital expenditure 2 quarters ahead from transcripts, following Jha et al. (2024) (in-sample; we use their sample period).
  - Thomson Reuters StreetEvents transcripts and Compustat capex.
  - IS: $N = 106{,}994$ firm-quarters, 3,920 firms (2006Q1 – 2020Q4).
  - OOS: $N = 6{,}744$ (2023Q3 – 2024Q1).

Model: Llama-3.3-70B (open-source, December 2023 knowledge cutoff). The cutoff defines the IS / OOS split.

Step 1 prompts

Two design choices: firm name + outcome-period date (day $t+1$ / quarter $q+2$).

News Headlines (date: day $t+1$):

Recall Prompt

"On 2020-07-29, did the closing stock price of Kodak go up or down compared to the previous trading day? Respond with exactly one word: up, down, or unknown."

Output: $P_{\text{up}}, P_{\text{down}}, P_{\text{unknown}}$. LAP $\equiv P_{\text{up}} + P_{\text{down}}$.

Earnings calls (date: quarter $q+2$):

Recall Prompt

"In Q3 2020, did the capital expenditure of Amazon increase or decrease compared to the previous quarter? Respond with exactly one word: up, down, or unknown."

Output: $P_{\text{up}}, P_{\text{down}}, P_{\text{unknown}}$. LAP $\equiv P_{\text{up}} + P_{\text{down}}$.

Step 2 prompts

News Headlines (Lopez-Lira and Tang, 2023)**:**

Forecast Prompt

"Here is a piece of news: {headline}. Do you think this news is **good** , **bad** , or **neutral** for the stock price of {company} in the short term?"

Mapping: `good`$\rightarrow +1$, `neutral`$\rightarrow 0$, `bad`$\rightarrow -1$. Output: $\text{LLM}_{i,t} \in \{-1, 0, +1\}$.

Earnings calls (Jha et al., 2024)**:**

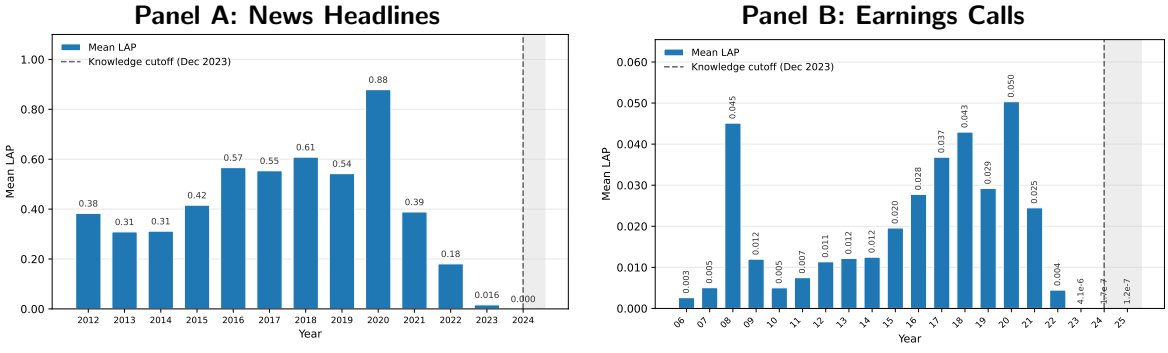
Forecast Prompt

"Here is the earnings-call transcript of {company}: {transcript}. Do you think the company plans to **significantly increase** , **slightly increase** , **not change** , **slightly decrease** , or **significantly decrease** its capital expenditures over the next year?"

Mapping: 5-way capex direction $\rightarrow \{-1, -0.5, 0, +0.5, +1\}$. Output: $\text{LLM}_{i,q}$.

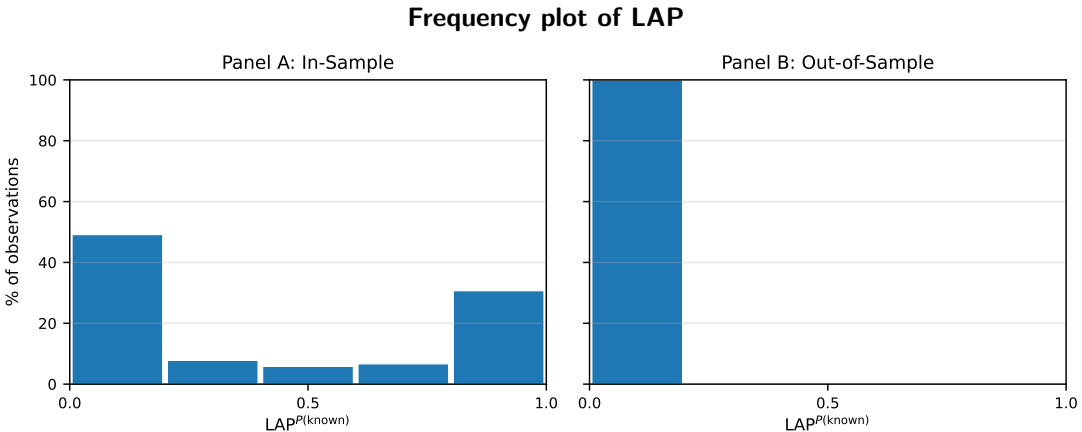
LAP collapses at the training cutoff

Figure I. Mean LAP$^{P(\text{known})}$ by year.



$\Rightarrow$ Both collapse at the Dec 2023 cutoff.

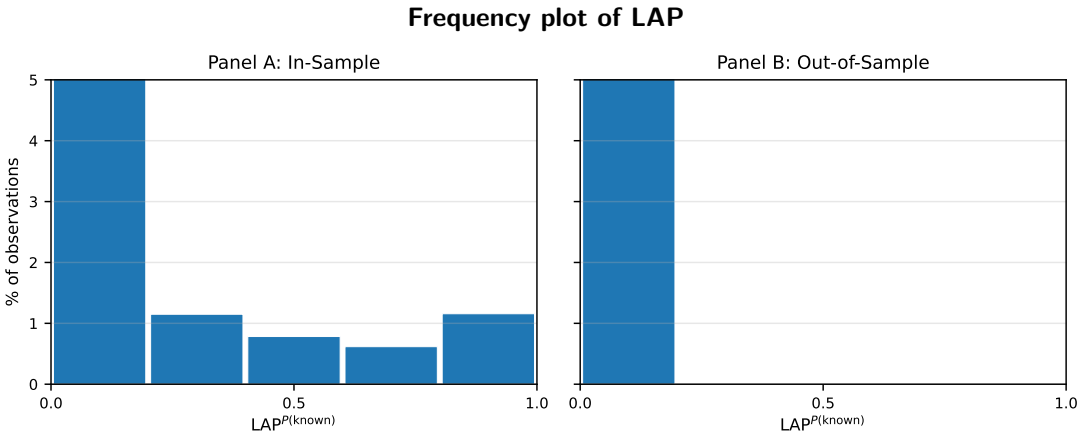
LAP distribution: News Headlines (IS vs OOS)



In-sample (2012–2023): clearly bimodal.

Out-of-sample (2024): the right tail vanishes.

LAP distribution: Earnings Calls (IS vs OOS)



In-sample (2006Q1–2020Q4): sharply right skewed.

Out-of-sample (2023Q3–2024Q1): the small right tail disappears.

Step 1: Validation

Test. Does the date-only recall signal $P_{\text{up}} - P_{\text{down}}$ predict realized Y_{t+1}?

	$r_{i,t+1}$ (Headlines)			CapEx$_{i,q+2}$ (Earnings calls)		
	(1) Pooled	(2) High LAP *(Top 50%)*	(3) Low LAP *(Bottom 50%)*	(4) Pooled	(5) High firm-LAP *(Top 50%)*	(6) Low firm-LAP *(Bottom 50%)*
$P_{\text{up}} - P_{\text{down}}$	**0.264*** (3.53)**	0.306*** (3.59)	−0.159 (−0.33)	**0.368** (2.13)**	0.539*** (2.90)	27.05 (0.71)
Firm FE	Yes	Yes	Yes	Yes	Yes	Yes
Time FE	Yes	Yes	Yes	Yes	Yes	Yes
N	91,357	45,432	45,443	106,994	53,444	53,550

$\Rightarrow$ Both applications: memorization exists $(L_t > 0)$.

Step 2 — Forecast contamination: News Headlines

Testing forecast contamination. Does the contaminated forecast $\hat{\mu}_t$ use memorized content L_t when predicting Y_{t+1}?

	$r_{i,t+1}$ (1) LLM only	$r_{i,t+1}$ (2) LLM + LAP
$\text{LLM}_{i,t}$	0.209*** (12.18)	0.141*** (7.55)
LAP		0.024 (0.22)
$\text{LLM}_{i,t} \times \text{LAP}$		**+0.162*** (3.64)**
Firm FE	Yes	Yes
Time FE	Yes	Yes
R^2 / N	0.179 / 91,357	0.180 / 91,357

$\Rightarrow \hat{\beta}_3 > 0$: contamination detected (contamination loading $\gamma > 0$). 1-sd $\uparrow$ in LAP raises marginal LLM effect by $\approx 32\%$.

Step 2 — Forecast contamination: Earnings Calls

Testing forecast contamination. Does the contaminated forecast $\hat{\mu}_t$ use memorized content L_t when predicting Y_{t+1}?

	$\text{CapEx}_{i,q+2}$ (1) LLM only	$\text{CapEx}_{i,q+2}$ (2) LLM + LAP
$\text{LLM}_{i,q}$	0.547*** (16.69)	0.534*** (16.31)
LAP		0.051 (0.26)
$\text{LLM}_{i,q} \times \text{LAP}$		**+0.512** (2.01)**
Firm FE	Yes	Yes
Time FE	Yes	Yes
R^2	0.628	0.628
N	106,994	106,994

$\Rightarrow$ Same pattern: contamination detected in earnings calls. 1-sd $\uparrow$ in LAP raises marginal LLM effect by $\approx 12\%$.

Inner Confidence

Step 2: Forecast Prompt

"Here is a piece of news: {headline}. Do you think this news is **good** , **bad** , or **neutral** for the stock price of {company} in the short term?"

Inner Confidence (IC) (Chen et al., 2024): the LLM picks the most likely label, and IC is the probability of that chosen label :

$$\mathrm{IC}_{i,t} = \begin{cases} P_{\texttt{good}} & \text{if LLM responds } \texttt{good}, \\ P_{\texttt{neutral}} & \text{if LLM responds } \texttt{neutral}, \\ P_{\texttt{bad}} & \text{if LLM responds } \texttt{bad}. \end{cases}$$

IC vs LAP.

- IC uses token probabilities from the forecast prompt , not the recall prompt .
- IC uses the probability of the response (e.g., $P_{\texttt{good}}$); LAP $\equiv P_{\text{up}} + P_{\text{down}}$, probability of *known*.

Why it matters here. We need to check that LAP is not just another confidence measure.

Horse race: LAP vs. IC

Test. Is LAP just another measure of LLM confidence (IC)?

	News Headlines	Earnings Calls
LLM × LAP	**+0.163*** (3.64)**	**+0.506** (1.99)**
LLM × IC	0.605*** (2.81)	0.314** (2.24)
Firm FE	Yes	Yes
Time FE	Yes	Yes
N	91,357	106,994

⇒ Both interactions load *independently*. The memorization channel is distinct from general LLM confidence.

Related work

Three strands relevant to this paper:

1. **Documenting lookahead bias in LLMs** (Glasserman and Lin, 2023; Sarkar and Vafa, 2024): a simple date cutoff prompt (e.g., "use only data before 2019") cannot eliminate it.
2. **Prompting-based mitigation:** iterative information masking (Engelberg et al., 2025); but it causes information loss (Wu et al., 2025) and LLMs still recall historical macro values and big-tech stock prices (Lopez-Lira et al., 2025).
3. **Retraining small-scale ($\leq$ 4B) lookahead-bias-free LLMs:** StoriesLM (Sarkar, 2024), ChronoGPT (1.5B) (He et al., 2025), DatedGPT (1.3B) (Yan et al., 2026), Point-in-Time LM (4B) (Kelly et al., 2026). Great for simple tasks, but limited compared to the frontier LLMs used in practice.

This paper

We provide a portable *test* for LLM lookahead bias. It works on any LLM.

Our contribution

Lookahead bias is task-specific.

Our method has three advantages:

- **No researcher discretion:** no choices about what to mask.
- **Cost-effective and easy to implement:** only requires first-token output probabilities; no retraining.
- **Generalizable:** the LAP test applies to any LLM forecasting application in empirical economics.

Thank you!

References

Chen, H., Didisheim, A., and Somoza, L. (2024). Out of the black box: Uncertainty quantification for llms via conditional probabilities. *Available at SSRN*.

Engelberg, J., Manela, A., Mullins, W., and Vulicevic, L. (2025). Entity neutering. *Available at SSRN*.

Glasserman, P. and Lin, C. (2023). Assessing look-ahead bias in stock return predictions generated by gpt sentiment analysis. *arXiv preprint arXiv:2309.17322*.

He, S., Lv, L., Manela, A., and Wu, J. (2025). Chronologically consistent large language models. *arXiv preprint arXiv:2502.21206*.

Jha, M., Qian, J., Weber, M., and Yang, B. (2024). Chatgpt and corporate policies. *arXiv preprint arXiv:2409.17933*.

Kelly, B. T., Malamud, S., Schwab, J., and Xu, T. A. (2026). Scaling point-in-time language models. *Swiss Finance Institute Research Paper*, (26-37).

Lopez-Lira, A. and Tang, Y. (2023). Can chatgpt forecast stock price movements? return predictability and large language models. *arXiv preprint arXiv:2304.07619*.

Lopez-Lira, A., Tang, Y., and Zhu, M. (2025). The memorization problem: Can we trust llms' economic forecasts? *arXiv preprint arXiv:2504.14765*.

Sarkar, S. K. (2024). StoriesLM: A family of language models with time-indexed training data. *Available at SSRN 4881024*.

Sarkar, S. K. and Vafa, K. (2024). Lookahead bias in pretrained language models. *Available at SSRN 4754678*.

Wu, K., Yang, B., Ying, Z., and Zhou, D. (2025). Anonymization and information loss. *arXiv preprint arXiv:2511.15364*.

Yan, Y., Tang, R., Gao, Z., Jiang, W., and Lu, Y. (2026). DatedGPT: Preventing lookahead bias in large language models with time-aware pretraining. *arXiv preprint arXiv:2603.11838*.