Do Large Language Models (LLMs) Understand Chronology? (Student Abstract)

Pattaraphon Kenny Wongchamcharoen[1], **Paul Glasserman**[2]

[1] Department of Industrial Engineering & Operations Research, University of California, Berkeley
[2] Columbia Business School, Columbia University
pattaraphon.kenny@berkeley.edu,
pg20@gsb.columbia.edu

Abstract

Large language models have shown great potential as forecasting tools in finance and economics, but backtesting performance is subject to look-ahead bias if the period overlaps with an LLM's training window. Prompt-based attempts to avoid look-ahead bias require that LLMs understand chronology. We test LLMs' ability to understand and enforce chronological order in three types of tasks: sorting randomly shuffled historical events; conditional sorting of events defined by some conditions; and anachronism detection based on intersections of multiple timelines. Our experiments use events that we first confirm are known to the LLM; this ensures that we test chronological understanding on an LLM's pretrained internal knowledge. Across three LLM families— GPT-4.1 (standard), GPT-5 (hybrid-reasoning), and Claude 3.7 Sonnet (large-reasoning, with and without Extended Thinking), we find that performance degrades rapidly with problem complexity but improves greatly for reasoning models with test-time extended reasoning. These patterns are important for the real-time application of LLMs in finance.

Code — https://github.com/kennywong524/chronollm

Introduction

Large language models (LLMs) have shown potential as forecasting tools for finance and economics. Typical applications ask LLMs to predict the direction of stock prices based on text data, such as news reports or company earnings calls. Evaluating the performance of any forecasting method requires backtesting the method on historical data. Yet backtesting of LLM forecasts is subject to look-ahead bias (Glasserman and Lin (2024), Sarkar and Vafa (2024)) when the backtesting period overlaps with the LLM's training window: the LLM may be asked to predict an outcome it has already seen. Leakage of post-event information embedded in the LLM's pre-training corpus can inflate estimated forecast performance with poor out-of-sample results.

Various methods have been proposed to measure and mitigate this problem. The simplest approach is to limit testing to an LLM's post-training period (as in, e.g., Halawi et al. (2024), Lopez-Lira and Tang (2024)). This approach eliminates look-ahead bias, but it severely limits the testing window. Building models trained using only text available up to fixed dates in the past (Sarkar and Vafa (2024), (He et al. 2025)) provides a secure way to wall off future information, but it is computationally demanding and limited to training using time-stamped documents. For users, the most convenient solution would be to wall off future information by instructing an LLM to respond using only information available up to a fixed date. Sarkar and Vafa (2024) and Lopez-Lira, Tang, and Zhu (2025) find leakage in examples of this approach. More basically, a prompt-based instruction like "use only information from before 2016" presupposes that an LLM understands what "before 2016" means.

Prior approaches to NLP temporal testbeds (e.g., TimeQA (Chen, Wang, and Wang 2021), TRAM (Wang and Zhao 2024), TimeBench (Chu et al. 2024), and ChronoSense (Islakoglu, Yates, and de Rijke 2025)) seek to isolate and evaluate an LLM's performance on specific components of temporal reasoning, often using newly crafted, hypothetical scenarios and specially designed instructions to probe narrow inference types. We step back and ask a more fundamental question: *Do LLMs understand chronology?* Before asking an LLM to avoid leakage of future information in responding to new data, we want to evaluate how well the LLM understands chronological constraints in data on which it has been trained.

Methodology

We test performance on three task types: *(1) Chronological sorting* (putting shuffled events in chronological order); *(2) Conditional sorting* (selecting events that meet a specified condition and ordering those chronologically); *(3) Anachronism detection* (distinguishing possible vs. impossible events based on historical timing). For (2), we run paired trials on the same shuffle under two regimes: (i) *self-filtering & sorting* (the model filters events based on the condition, then orders them) and (ii) *given-names & sorting* (the model just chronologically orders the provided filtered events, presented in the same shuffled order as (i)). We evaluate results on both non-reasoning and reasoning models from multiple families (OpenAI's GPT-4.1, GPT-5 series with various reasoning effort and Anthropic's Claude Sonnet 3.7 with and without extended-thinking). We design our tests around basic historical facts, which are first confirmed to be "known" by the LLM. The three categories of tests above also allow us to vary problem complexity along distinct dimensions.

Key Results

Our main finding is that LLM performance on our chronology tasks degrades quickly with problem complexity. Table 1 summarizes GPT-4.1's results with shuffled lists of U.S. presidents. (We have obtained similar results on other collections of historical events.) GPT-4.1's performance on ordering drops drastically with list sizes. The exact match rate (EM) is nearly perfect on very short lists (e.g., $n = 2$), drops to roughly half on small lists (e.g., $n \approx 10$), and is effectively zero on longer lists ($n > 25$). The model often omits or adds extra presidents' names to the responded list, with increasing prevalence at larger list sizes. Interestingly, complexity is not a simple function of list length: models can do better listing *all* presidents in order than sorting a random subset of n=20; in Table 1, Spearman's ρ, Kendall's τ, and Cayley distance are U-shaped in n. Ordering a complete set of events, at least approximately, appears to be easier than ordering a random subset for the LLM. Performance improves substantially for models with extended reasoning, which achieve near flawless results, as seen in Figure 1 which shows that Claude 3.7 Sonnet with Extended Thinking (ET) and GPT-5 at *medium/high* reasoning effort achieve *perfect* chronological ordering across all list sizes (100% exact match), whereas GPT-5 at minimal/low reasoning effort, its non-reasoning variant (latest), and Claude 3.7 *without* ET behave similarly to GPT-4.1. Interestingly, poor performance at high complexity kicks in at problem scales much smaller than those seen in other tasks (math, coding, or QA benchmarks), suggesting that reasoning about chronology of real events may be inherently difficult. Table 2 illustrates the result of the conditional sorting task where non-reasoning GPT-4.1 failed entirely at filtering, producing no valid trials. In contrast, Claude 3.7 Sonnet (Extended Thinking) and GPT-5 (medium) filtered almost perfectly and achieved near-perfect ordering accuracy. Once models successfully performed the filtering step, their ordering performance improved—suggesting that the added difficulty of conditional sorting mainly comes from filtering, and that reasoning-driven self-selection can actually enhance chronological reasoning.

Our results include several more novel findings and contributions: (i) reasoning models consistently outperform base models across all three task families and all the evaluation metrics we study, and their performance generally increases with larger test-time reasoning budgets; (ii) errors often concentrate in the middle of lists, with more-salient starting and end points acting as anchors; and (iii) a basic anachronism detection task defined by two events proves easy for the LLMs, but performance declines on tests defined by the intersection of multiple events.

n	Exact match		Spearman's ρ		Kendall's τ		Cayley
	μ	SE	μ	SE	μ	SE	μ
2	0.96	0.04	0.998	0.00	0.997	0.00	0.00
5	0.87	0.08	0.973	0.02	0.960	0.03	0.20
10	0.36	0.11	0.961	0.02	0.932	0.02	1.72
15	0.20	0.09	0.960	0.01	0.927	0.02	3.27
20	0.10	0.07	0.971	0.01	0.929	0.01	6.20
30	0.00	0.00	0.963	0.01	0.948	0.02	6.60
40	0.00	0.00	0.993	0.00	0.989	0.01	3.10
43	0.00	0.00	1.000	0.00	1.000	0.00	0.05

Table 1: GPT-4.1 performance means and standard errors for ordering a random list of n U.S. presidents chronologically. EM collapses beyond small n despite high rank correlations.

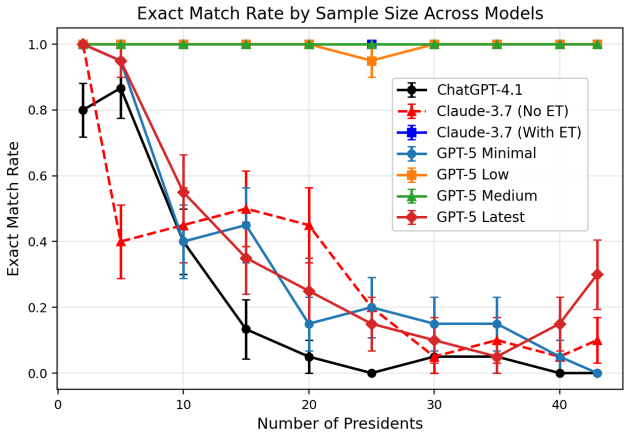

Figure 1: Exact Match rate by list size across various reasoning model families. Error bars are ±2 s.e. around mean.

Conclusion

Models show a workable but brittle grasp of chronology unless test-time reasoning is enabled, weakening prompt-based mitigation against look-ahead bias. Our findings identify key complexity drivers but are limited by our event sets, partial model coverage, and the need to better understand GPT-5's automatic reasoning modes.

Filtering accuracy (OHIOORVIRGINIA)				
Model	Acc.	n_{correct}	n_{total}	ET?
Claude 3.7 Sonnet	0.02	2	100	No
Claude 3.7 Sonnet	0.98	98	100	Yes
GPT-4.1	0.00	0	100	N/A
GPT-5 (medium)	1.00	100	100	N/A
Ordering metrics (Claude 3.7 Sonnet + ET)				
Metric	Value	n_{correct}	n_{total}	Cond.
Spearman's ρ	0.997	99	100	Given
Kendall's τ	0.995	99	100	Given
Exact match	0.82	82	100	Given
Spearman's ρ	1.000	100	100	Self
Kendall's τ	1.000	100	100	Self
Exact match	0.97	97	100	Self

Table 2: Conditional sorting on OHIOORVIRGINIA: filtering accuracy across models and ordering performance for Claude 3.7 Sonnet with Extended Thinking (ET).

Acknowledgements

This work was conducted as part of Pattaraphon Kenny Wongchamcharoen's summer research internship at Columbia Business School. We gratefully acknowledge the AI in Business Initiative for their generous support, including computational resources and access to the OpenAI API, which made this research possible.

References

Chen, W.; Wang, X.; and Wang, W. Y. 2021. TimeQA: A Dataset for Answering Time-Sensitive Questions. In *Proceedings of NeurIPS 2021*.

Chu, Z.; Chen, J.; Chen, Q.; Yu, W.; Wang, H.; Liu, M.; and Qin, B. 2024. TimeBench: A Comprehensive Evaluation of Temporal Reasoning Abilities in LLMs. In *Proceedings of ACL 2024*.

Glasserman, P.; and Lin, C. 2024. Assessing Look-Ahead Bias in Stock Return Predictions Generated by GPT Sentiment Analysis. *Journal of Financial Data Science*, 6(1).

Halawi, D.; Zhang, F.; Chen, Y.; and Steinhardt, J. 2024. Approaching Human-Level Forecasting with Language Models. arXiv:2402.18563.

He, S.; Lv, L.; Manela, A.; and Wu, J. 2025. Chronologically Consistent Large Language Models. arXiv:2502.21206.

Islakoglu, D. S.; Yates, A.; and de Rijke, M. 2025. ChronoSense: Exploring Temporal Understanding in LLMs. In *Proceedings of ACL 2025: Short Papers*.

Lopez-Lira, A.; and Tang, Y. 2024. Can ChatGPT Forecast Stock Price Movements? Return Predictability and Large Language Models. arXiv:2304.07619.

Lopez-Lira, A.; Tang, Y.; and Zhu, M. 2025. The Memorization Problem: Can We Trust LLMs' Economic Forecasts? arXiv:2504.14765.

Sarkar, S. K.; and Vafa, K. 2024. Lookahead Bias in Pretrained Language Models. SSRN 4754678.

Wang, Y.; and Zhao, Y. 2024. TRAM: Benchmarking Temporal Reasoning for Large Language Models. arXiv:2310.00835.