ExAnte: A Benchmark for Ex-Ante Inference in Large Language Models

Yachuan Liu, Xiaochun Wei*, Lin Shi*, Xinnuo Li*
Bohan Zhang, Paramveer Dhillon, Qiaozhu Mei
University of Michigan
{yachuan, xcwei, linshia, monmonli, bohanz, dhillonp, qmei}@umich.edu

Dataset: https://huggingface.co/datasets/yachuanliu/ExAnte
Code: https://github.com/yachuan/ExAnte

Abstract

Large language models (LLMs) face significant challenges in ex-ante reasoning, where analysis, inference, or predictions must be made without access to information from future events. Even with explicit prompts enforcing temporal cutoffs, LLMs often generate outputs influenced by internalized knowledge of events beyond the specified cutoff. This paper introduces a novel task and benchmark designed to evaluate the ability of LLMs to reason while adhering to such temporal constraints. The benchmark includes a variety of tasks: stock prediction, Wikipedia event prediction, scientific publication prediction, and Question Answering (QA), designed to assess factual knowledge under temporal cutoff constraints. We use leakage rate to quantify models' reliance on future information beyond cutoff timestamps. Experimental results reveal that LLMs struggle to consistently adhere to temporal cutoffs across common prompting strategies and tasks, demonstrating persistent challenges in ex-ante reasoning. This benchmark provides a potential evaluation framework to advance the development of LLMs' temporal reasoning ability for time-sensitive applications.

1 Introduction

Large language models (LLMs) [4, 10, 1, 2] have significantly advanced common tasks of natural language processing (NLP), demonstrating strong performance in question answering [24, 45], summarization [12, 44], and reasoning [39, 18]. However, ensuring that LLMs can reason under strict temporal constraints remains an open challenge [41, 11]. In many real-world applications, models must answer time-sensitive queries using only information available up to a given cutoff date, without incorporating knowledge from events that occurred afterward. We refer to this ability as ***ex-ante inference***, a fundamental yet underexplored temporal reasoning task. Ex-ante inference differs from general knowledge recall [19, 24, 42] and machine unlearning [23, 8, 22]. Unlike machine unlearning, which removes specific information from a model permanently, ex-ante queries are *ad hoc* and *context-dependent*, making it impractical to unlearn models for every query. Instead, models must dynamically enforce temporal constraints while retrieving and reasoning over pre-cutoff knowledge.

LLMs often fail in this task, exhibiting *temporal leakage* — the unintended use of future knowledge — in their reasoning, compromising their reliability in historical simulations, financial forecasting, and research trend prediction. For instance, BattleAgent [20], designed to simulate World War II, may inadvertently incorporate post-1945 knowledge and distort its analysis. Similarly, financial models risk leaking future trends in backtests [16, 9], and LLMs predicting research trends may reveal high-impact papers before publication [17, 37, 36, 3]. (See Figure 1a for an illustration of temporal leakage.)

Despite its significance, ex-ante inference has received little attention. Existing NLP benchmarks assess general language understanding and factual consistency [35, 34, 28, 26] but do not evaluate

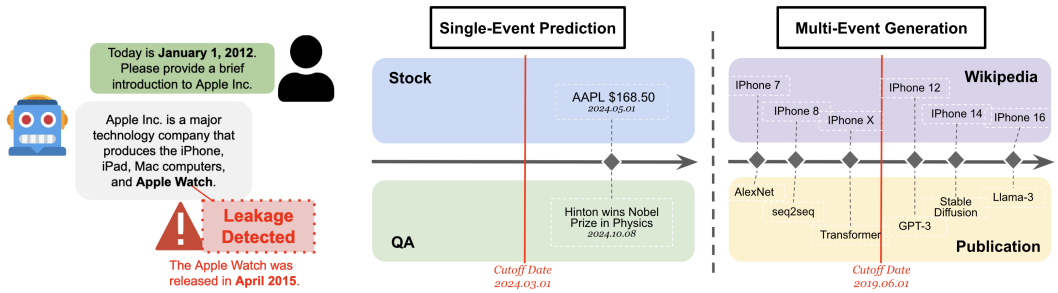
Preprint.

models' ability to reason strictly within pre-cutoff knowledge. Prior work in temporal reasoning often assumes full access to future context [33, 40, 41], overlooking *ad hoc* knowledge restrictions needed for ex-ante evaluation.

To address this, we introduce *ExAnte*, the first benchmark for systematically evaluating LLMs' ex-ante inference capabilities. ExAnte spans Wikipedia, stock market data, scientific publications, and QA, explicitly distinguishing pre- and post-cutoff events. We use *leakage rate* to quantify models' reliance on post-cutoff knowledge, along with a quality measure to evaluate response quality and prevent models from producing low-quality or evasive outputs to avoid being flagged for leakage. Experiments with GPT, Gemini, and Claude show that temporal leakage persists across models under different prompting strategies. We also showed that shorter cutoff gaps and higher memorization rates both increase leakage, revealing LLMs' difficulty in enforcing strict temporal constraints.

By defining a new reasoning paradigm, introducing a benchmark, and providing a structured evaluation, this work establishes the foundation for improving LLM's reliability of temporal reasoning in time-sensitive tasks.

2 Ex-Ante Inference Task Definition



(a) Temporal leakage in an ex-ante inference task.

(b) ExAnte benchmark overview: single-event prediction (left) and multi-event generation (right). Red lines denote the temporal cutoff.

Figure 1: Illustration of temporal reasoning and benchmark task structure.

The *ex-ante inference* task evaluates whether large language models (LLMs) inadvertently incorporate future knowledge when responding to time-sensitive queries under a strict temporal cutoff t_c. The goal is to assess whether models can generate responses exclusively using knowledge available before t_c, simulating real-world constraints where future information is unavailable.

Given a query x with a specified cutoff timestamp t_c, we define:

- $R_{\text{pre}}(x, t_c)$: The set of all verifiable facts regarding x before t_c.
- $R_{\text{post}}(x, t_c)$: The set of all facts regarding x that only became verifiable after t_c.
- $M(x)$: The model's response, denoted as $\hat{y}$.
- y^*: The ideal response, containing only pre-cutoff knowledge.

A valid response must satisfy that:

$$P(\hat{y} \mid t \leq t_c, R_{\text{pre}}(x, t_c)) = P(y^* \mid t \leq t_c, R_{\text{pre}}(x, t_c)). \tag{1}$$

This condition ensures that the model's output is solely determined by information available before the cutoff, aligning its behavior with an oracle constrained to $R_{\text{pre}}(x, t_c)$.

2.1 Task Definition and Leakage Condition

We evaluate two subtasks, given a cutoff t_c, as illustrated in Figure 1b:

1. **Single-Event Prediction** – The model predicts the outcome of a specific event e occurring at time t_e, where $t_e > t_c$ and the event's outcome is only verifiable after the cutoff. For this task, we first verify that the model has access to the post-cutoff knowledge of event e (**memorization check**); otherwise, evaluating for leakage would be ill-defined.

2. **Multi-Event Generation** – The model generates a set of related atomic claims, each of which must individually satisfy the temporal constraint by being verifiable using only pre-cutoff knowledge.

For **single-event prediction**, the model answers a time-sensitive query without using post-cutoff information. If the model's response $\hat{y}$ matches an event in $R_{\text{post}}(t_c)$, leakage has occurred:

$$L_{\text{query}}(x) = \mathbf{1}(\hat{y} \in R_{\text{post}}(t_c)). \tag{2}$$

For **multi-event generation**, the model must generate a set of atomic claims: $\hat{y} = \{c_1, c_2, ..., c_n\}$ where each c_i is an **atomic claim**, defined as an individual, self-contained factual statement in the model's response. Each claim must be independently verified to ensure it belongs to $R_{\text{pre}}(t_c)$. If any claim is in $R_{\text{post}}(t_c)$, leakage occurs:

$$L_{\text{query}}(x) = \frac{\sum_{i=1}^{n} \mathbf{1}(c_i \in R_{\text{post}}(t_c))}{n}. \tag{3}$$

To summarize model performance across a dataset D with N queries, we compute the *proportion of queries with leakage* as:

$$L_{\text{dataset}}(D) = \frac{\sum_{j=1}^{N} \mathbf{1}(L_{\text{query}}(x_j) > 0)}{N}. \tag{4}$$

This metric captures how frequently a model violates the temporal constraint across the entire benchmark.

2.2 Quality Measure

While our primary objective is to detect temporal leakage, we also evaluate whether the model's response $\hat{y}$ is aligned with the ideal pre-cutoff response y^*. Without this constraint, a model could trivially avoid leakage by producing vacuous outputs.

We define the quality measure $Q(\hat{y}, y^*) = \text{sim}(\hat{y}, y^*)$, where sim is instantiated as accuracy for classification tasks (Wikipedia, Publication) and negative mean absolute error (MAE) for regression tasks (Stock).

A response is considered valid only if it satisfies both the leakage and quality constraints:

$$\text{Valid}(x) = \mathbb{1}\left(L_{\text{query}}(x) = 0 \ \wedge \ Q(\hat{y}, y^*) \geq \tau\right), \tag{5}$$

where τ is a task-specific threshold.

3 Benchmark Datasets

To evaluate the temporal adherence of LLMs under ex-ante inference constraints, we curated four diverse datasets: Stock and QA for **single-event prediction**, and Wikipedia and Publication for **multi-event generation**. These datasets are carefully designed to reflect distinct domains where temporal reasoning plays a crucial role. An overview of the datasets are shown in 1.

3.1 Stock

Task Definition: The Stock dataset evaluates temporal leakage in numerical prediction tasks, assessing whether LLMs rely on post-cutoff stock price knowledge when predicting future prices. Given a cutoff date t_c, the model is prompted to predict a company's stock price on a future event date, t_e ($t_c < t_e$) . The task is designed to test whether models generate genuine pre-cutoff predictions or unintentionally leak post-cutoff information.

Data Curation Process: We collected historical stock prices from Yahoo Finance[1] for the Magnificent 7 companies: Apple, Microsoft, Alphabet, Amazon, Nvidia, Meta, and Tesla.

The dataset spans September 1, 2022 to September 1, 2023, covering 251 trading days. We extracted daily closing prices for each stock. This period was chosen because LLMs show more consistent

[1] `https://help.yahoo.com/kb/SLN2311.html`

memorization of stock prices after 2021. As shown in Figure 2 (Appendix 2), post-2021 prices are recalled with lower variance, while earlier periods yield inconsistent retrieval. This allows for more controlled and reliable temporal leakage evaluation.

Dataset	Task Type	Task Description	Dataset Size	Data Source	An Example (Zero-Shot Prompt)
Stock	Single-Event Prediction	Given a company name and a cutoff date, models are asked to predict the stock price exactly one year after the cutoff.	1757	Yahoo Finance	Assume today is 2022-09-01. What is META's stock price on 2023-09-01? Answer with only a number (e.g., '143.16').
QA	Single-Event Prediction	Given a cutoff date, models are asked to provide the exact date of an unexpected event, such as a disaster, election, or sports outcome, that occurred after the cutoff.	304	Wikipedia Annual Events (2007-2022)	Assume today is 2012-01-28. What was the exact date when the Costa Concordia cruise ship ran aground off Italy, causing 32 deaths?
Wikipedia	Multi-Event Generation	Given a Wikipedia title and a cutoff date, models are asked to generate atomic facts about the topic, ensuring all generated facts are from before the cutoff.	630	Top Viewed Wikipedia Articles	Given your knowledge of "Apple.Inc," generate 5 atomic facts using only information available before 2016-12-31.
Publication	Multi-Event Generation	Given a research field and a cutoff date, models are asked to generate notable research papers published before the cutoff.	98	Top CS venues	Assume today is 2014-07-01. List the most notable deep learning research papers that published in 2014.

Table 1: Overview of the Benchmark Datasets.

Dataset-Specific Evaluation: To assess temporal leakage, we first test whether the model memorizes the actual stock price at t_e by querying it directly without any cutoff constraint. If the model correctly recalls the price at t_e, it is considered to have memorized the value; otherwise, the prediction is excluded from leakage analysis.

A prediction is considered leaked if it is too close to the actual price at t_e, suggesting reliance on post-cutoff information rather than pre-t_c knowledge. Specifically, we define leakage as:

$$\frac{\left|P^{\text{pred}}_{t_c \to t_e} - P^{\text{actual}}_{t_e}\right|}{P^{\text{actual}}_{t_e}} < \delta, \tag{6}$$

where $P^{\text{pred}}_{t_c \to t_e}$ is the model's prediction for t_e made at t_c, and $P^{\text{actual}}_{t_e}$ is the true price at t_e, sourced from Yahoo Finance. We adopt a threshold of $\delta = 0.03$ to balance sensitivity and specificity, allowing for minor prediction noise while still flagging highly accurate forecasts that are unlikely without access to post-cutoff information.

Quality Measure: To assess prediction quality beyond leakage, we compare each model's forecast to human analyst predictions. We obtain the human analyst stock price predictions from `MarketBeat` [25] for the . For each model prediction, we compute the Mean Absolute Error (MAE) against the corresponding human forecast.

3.2 QA

Task Definition: The QA dataset evaluates temporal leakage in factual event prediction by testing whether LLMs recall future events before a given cutoff date t_c. The model is prompted with a question about an event occurring after t_c, and if it correctly states the exact date t_e, it is marked as a leakage.

Data Curation Process: The dataset includes 300 major, non-predictable events drawn from Wikipedia's annual events (2007–2022)[2]. GPT-4o extracted key events and dates, and two human annotators filtered out events that were predictable in advance, ensuring leakage reflects memorization rather than inference. Models are evaluated under three cutoff settings, with t_c set to 1 week, 1 month, or 1 year before t_e.

Dataset-Specific Evaluation: As with the stock dataset, we first check if a model can recall the exact date t_e without a cutoff. If so, the example is included in leakage evaluation. A prediction is considered leaked if the model outputs the correct date t_e despite being prompted with a cutoff at t_c.

[2] `https://en.wikipedia.org/wiki/YYYY`

Quality Measure: Quality measure is not applicable to the QA dataset.

3.3 Wikipedia

Task Definition: The Wikipedia dataset evaluates temporal leakage in knowledge-based generation by testing whether LLMs produce facts that rely on post-cutoff information. Given a Wikipedia topic and cutoff year t_c, the model is prompted to generate atomic facts limited to pre-t_c knowledge. Any claim referencing information after t_c is treated as a leaked instance.

Data Curation Process: The dataset is curated from Wikipedia's most frequently accessed pages[3], as they undergo regular updates and have well-documented revision histories.

(1) Topic selection and cutoff determination: To captures meaningful temporal shifts, GPT-4o was used to identify Wikipedia topics where the available information before and after a certain time point differs significantly. The cutoff year t_c is selected to maximize this difference, ensuring that:

- t_c represents a significant transition point, meaning the facts about this topic known before and after t_c are substantially different.
- $t_c > 2010$, ensuring that Wikipedia's knowledge before the t_c is mature and stable.

(2) Reference Set Construction: For each selected topic x, we retrieve two Wikipedia page versions to establish clear pre- and post-cutoff references:

- $R_{\textbf{pre}}(x, t_c)$: The archived snapshot of the page closest to t_c, containing only facts that were verifiable before the cutoff date.
- $R_{\textbf{post}}(x, t_c)$: The latest available version of the page, which includes all facts that became verifiable after t_c, as well as still-valid facts from before the cutoff.

Empirically, using only these two versions provides a comparable effect to tracking all intermediate revisions between t_c and the present, as significant factual updates are preserved in the latest version. This approach simplifies the curation process while maintaining fidelity in assessing temporal leakage.

Dataset-Specific Evaluation: Temporal leakage occurs when the model generates a fact that is not found in $R_{\text{pre}}(x, t_c)$ but appears in $R_{\text{post}}(x, t_c)$, indicating reliance on post-cutoff knowledge. An LLM judge is employed to determine whether a claim is supported by $R_{\text{pre}}(x, t_c)$ or $R_{\text{post}}(x, t_c)$. If the claim is missing in $R_{\text{pre}}(x, t_c)$ but appears in $R_{\text{post}}(x, t_c)$, it is flagged as a leakage. Please find the logic truth table for identifying leakage and calculating accuracy and the prompt for evaluation in section A.2.1.

Quality Measure: We measure the proportion of generated claims supported by $R_{\text{pre}}(x, t_c)$. This captures how well the model's output aligns with pre-cutoff facts.

3.4 Publication

Task Definition: The Publication dataset evaluates temporal leakage in scientific text generation by testing whether LLMs list papers unavailable before a cutoff date t_c. Given a computer science keyword and t_c, the model is prompted to name notable publications. If any listed paper first appeared on or after t_c, it is marked as a leakage instance.

Data Curation Process: The dataset includes a comprehensive set of computer science keywords, each assigned a unique prediction year which its cutoff date t_c falls into. For each keyword, the model is prompted to generate a set of notable research papers—typically around 5 to 6—that were published within the cutoff year and before the specified cutoff date.

We construct the dataset by: (1) selecting top-tier CS conferences based on CSRankings[4]; (2) for each selected conference, generating yearly keyword distributions from 2014 to 2022 using GPT-4o

[3]Retrieved from `https://en.wikipedia.org/wiki/Wikipedia:Popular_pages`, last accessed December 2023

[4]`https://csrankings.org`

and Claude-3.5-sonnet; (3) assigning a prediction year to each keyword based on its most prominent appearance—if the keyword appears only once, that year is used; otherwise, we select the year in which it ranks highest (e.g., 3rd in 2021 vs. 5th in 2022 yields 2021)

Dataset-Specific Evaluation: A model exhibits leakage if it generates a research paper title whose earliest accessible publication date is on or after t_c, indicating reliance on post-cutoff knowledge.

We verify publication dates using a two-step pipeline: (1) *Existence verification*: A Google search ensures each generated title corresponds to a real paper. Nonexistent titles are excluded. (2) *Earliest publication date verification*: For valid publications, we query ArXiv, ACM Digital Library, and other academic search engines to determine the earliest known publication date. If it is on or after t_c, it is flagged as a leakage claim.

The leakage rate is computed as the proportion of valid publications with an earliest accessible date on or after t_c. All steps are automated to ensure scalability and consistency across keywords.

Quality Measure: To assess the quality of generated publications, we evaluate whether each paper received any citations within the same calendar year as the cutoff. Specifically, for a cutoff date t_c (e.g., 2014-06-01), we check if the paper was cited at least once during the year 2014. If the number of citations is greater than zero in that year, the paper is considered to be of high quality, reflecting immediate impact or recognition by the research community. This relaxed criterion captures papers that attracted attention shortly after publication, providing a lightweight proxy for scientific relevance.

4 Experiments and Results

4.1 Experiment Setup

We evaluate three state-of-the-art LLMs, including GPT-4o-mini, GPT-4o, Gemini 1.5 Pro, and Claude Sonnet 3.5. For each model, five prompting strategies are applied:

Prompting Strategies. We evaluate the following five prompting strategies:

- **Zero-Shot:** Queries are presented without additional guidance. *"Suppose you are at [cutoff date], what would be [the task]?"*
- **Instruction-Based:** Prompts explicitly instruct the model to adhere to the temporal cutoff. *"Suppose you are at [cutoff date], what would be [the task]? Note that you are not supposed to use any information after this date."*
- **Chain-of-Thought (CoT):** Prompts encourage step-by-step reasoning to enforce temporal adherence. *"Suppose you are at [cutoff date], what would be [the task]? Let's think step by step."*
- **One-Shot:** Queries are presented with one illustrative example but no additional guidance. *One Example + Zero-Shot Prompting.*
- **Self-Verification:** The model self-verifies its Zero-Shot response. Inspired by prior work [14], which shows that self-reflection mitigates hallucinations, we add a verification step. If leakage is detected, the model must regenerate. *Zero-Shot Prompting + Follow-up Verification Question.*

Please see Appendix for more detailed prompts A.1 and model versions and configurations E.1.

4.2 Main Results

This section presents key findings across the four datasets, highlighting how models and prompting strategies influence temporal leakage.

Stock Dataset: Table 2 shows substantial variation in leakage rates across prompting strategies. Zero-shot prompting leads to high leakage for all models (e.g., 86.56% for GPT-4o, 86.61% for Claude), indicating frequent reliance on post-cutoff knowledge without temporal cues. Instruction-based and Chain-of-Thought (CoT) prompting reduce leakage substantially (e.g., Instruction: 7.15% for GPT-4o, 5.91% for Gemini), while Self-Verification achieves the lowest leakage across models

Model	Zero-shot		Instruction-based		Chain-of-thought		One-shot		Self-Verification		MR (%)
	Leakage (%)	MAE	Leakage (%)	MAE	Leakage (%)	MAE	Leakage (%)	MAE	Leakage (%)	MAE	(mean ± std)
GPT-4o	86.56	72.19	7.15	490.69	5.37	471.23	69.73	77.98	**5.42**	176.42	78.88 ± 6.00
Claude-3.5-sonnet	86.61	65.93	43.89	84.11	79.65	73.95	70.49	67.20	**25.96**	182.96	88.45 ± 7.20
Gemini-1.5-pro	36.21	77.63	5.91	85.26	7.74	180.69	11.47	54.56	**6.55**	68.16	52.99 ±15.00

Table 2: Leakage rates and mean absolute error (MAE) across different models and prompting strategies on the Stock dataset. MAE values have been updated with final evaluation results and are shaded for readability (lower is better). Leakage reflects the percentage of responses relying on post-cutoff knowledge. MR denotes memorization rate, computed as the percentage of correctly recalled prices over 251 trading days.

Model	Zero-Shot			Instruction			CoT			One-Shot			Self-Verification			MR (%)
	7d	30d	1y	7d	30d	1y	7d	30d	1y	7d	30d	1y	7d	30d	1y	(mean ± std)
GPT-4o	67.3	34.0	12.1	38.0	**9.5**	**3.5**	59.7	25.2	7.7	39.7	27.2	16.1	**34.1**	22.3	4.2	78.61 ± 10.68
Claude-3.5	60.3	34.8	27.9	26.7	5.3	**1.9**	68.7	42.8	31.4	43.4	25.3	35.2	**2.7**	**4.2**	4.2	86.45 ± 1.27
Gemini-1.5	97.4	96.6	86.1	97.7	94.4	71.7	97.4	97.0	89.6	94.3	87.1	66.4	**20.0**	**7.1**	**3.8**	86.86 ± 0.67

Table 3: Leakage rates (%) across models and prompting strategies on the QA dataset, sorted by cutoff gap: 7 days (dark gray), 30 days (light gray), 1 year (white). Best-performing results per model and gap are bolded. MR denotes memorization rate, computed as the percentage of correctly recalled events over 300 events.

(e.g., 5.42% for GPT-4o, 6.55% for Gemini). Claude shows the weakest temporal control, with high leakage even under constrained prompts. In contrast, GPT-4o and Gemini better adhere to cutoff constraints when guided. These results emphasize the importance of prompt design in mitigating leakage for stock prediction.

QA Dataset: The QA dataset (Table 3) presents a different pattern, where leakage rates are generally higher, particularly for shorter cutoff gaps (see Section 4.3). Different from the stock dataset, Instruction-based prompting significantly reduces leakage for GPT-4o and Claude-3.5-sonnet, while Gemini-pro exhibits persistently high leakage ($\sim 90\%$) across most conditions, suggesting difficulty in suppressing post-cutoff knowledge. Self-Verification remains the best-performing method in 7 out of 9 cutoff gap and model pairs. Like in the Stock dataset, CoT cannot reduce leakage of most cases. Models with better memorization still suffer more in temporal leakage.

Wikipedia Dataset: Table 4 presents leakage and accuracy rates on the Wikipedia dataset, where models generate atomic facts under a temporal cutoff. Compared to QA and Stock, leakage is more stable across prompting strategies and models, typically ranging from 10–20%. This consistency may arise from two factors: first, models exhibit some degree of temporal awareness, making them less likely to mention clearly post-cutoff events (e.g., avoiding mentioning GPT-4 for a topic with a 2021 cutoff); second, Wikipedia is a core component of pretraining corpora, which may help constrain generations to plausible temporally aligned content.

Among models, GPT-4o and Gemini show the lowest leakage, while Claude-3.5-sonnet consistently exceeds 20%. Instruction-based prompting reduces leakage slightly for GPT-4o, but prompting strategies generally have limited effect across models. Unlike single-event tasks, multi-event generation may pose unique challenges by requiring temporal consistency across multiple claims.

Interestingly, we observe a strong positive correlation between leakage and accuracy—methods that yield higher accuracy also tend to exhibit more leakage.

Publication Dataset: The Publication dataset (Table 5) poses the greatest challenge, with all models showing high leakage rates and no prompting strategy reducing leakage below 38%. Self-verification substantially lowers leakage for Claude and Gemini but has limited effect on GPT-4o. Instruction-based prompting is similarly ineffective, likely because models still tend to generate future high-impact papers, regardless of the temporal restriction. This may stem from the publication date being a secondary detail relative to the paper's content, which may be underemphasized in pretraining. Accuracy results mirror those of the Wikipedia dataset, showing a strong positive correlation between

Model	Zero-shot		Instruction-based		Chain-of-thought		One-shot		Self-Verification	
	Leakage (%)	Accuracy (%)	Leakage (%)	Accuracy (%)	Leakage (%)	Accuracy (%)	Leakage (%)	Accuracy (%)	Leakage (%)	Accuracy (%)
GPT-4o	12.50	83.15	**10.23**	82.25	13.64	85.97	12.50	82.95	12.50	**86.36**
Claude-3.5-sonnet	**19.79**	68.96	20.83	67.71	22.92	67.92	22.11	68.33	23.16	68.75
Gemini-1.5-pro	13.82	81.73	14.02	82.59	12.74	81.57	**12.09**	**82.69**	14.81	82.65

Table 4: Leakage (%) and accuracy rates (%) across different models and prompting strategies on the Wikipedia dataset. Accuracy columns are shaded for readability.

Model	Zero-shot		Instruction-based		Chain-of-thought		One-shot		Self-Verification	
	Leakage (%)	Accuracy (%)	Leakage (%)	Accuracy (%)	Leakage (%)	Accuracy (%)	Leakage (%)	Accuracy (%)	Leakage (%)	Accuracy (%)
GPT-4o	82.02	**15.83**	86.05	14.79	80.90	15.34	**80.21**	14.65	80.85	14.97
Claude-3.5-sonnet	66.23	22.68	80.52	22.39	85.71	21.96	80.52	**23.12**	**40.26**	22.16
Gemini-1.5-pro	78.41	**17.54**	77.27	16.29	72.83	15.62	81.05	16.73	**38.54**	17.15

Table 5: Leakage (%) and accuracy rates (%) across different models and prompting strategies on the Publication dataset. Accuracy columns are shaded for readability.

leakage and accuracy—suggesting that higher factual correctness often co-occurs with the use of post-cutoff knowledge.

Cross-Dataset Insights: Across all datasets and models, no single prompting strategy consistently eliminates leakage, though effectiveness varies by task. Multi-event generation (Wikipedia, Publication) is more prone to leakage than single-event prediction (QA, Stock). Self-verification is generally most effective but fails on Wikipedia due to issues with detection, regeneration, and overcorrection (Appendix D). Instruction-based prompting improves adherence in structured tasks but struggles in open-ended generation. Chain-of-thought (CoT) offers limited gains, suggesting temporal reasoning needs more than generic step-by-step logic.

Model performance is similarly inconsistent. Gemini shows strong control on Wikipedia but high leakage in QA. Claude-3.5 performs poorly on Stock, while GPT-4o underperforms on Publication. Both tend to rely on memorization rather than temporal constraint-following. LLaMA models show near-zero memorization in Stock, likely due to lack of relevant training exposure.

Notably, quality measures—accuracy for Wikipedia and Publication, MAE for Stock—tend to correlate positively with leakage, indicating that higher factual precision often coincides with rule violation.

These results confirm that mainstream LLMs struggle with ex-ante inference, **making ExAnte a valuable benchmark** for evaluating future models with stronger temporal reasoning . Simple prompting strategies alone are insufficient to fully mitigate leakage, indicating the need for new architectural, training, and reasoning methods beyond prompting-based interventions.

4.3 Effect of Cutoff Gap on Leakage Rate

The cutoff gap—the time difference between an event date t_e and the cutoff date t_c—has a strong effect on leakage. As shown in Table 3, **shorter gaps lead to higher leakage**, with the highest rate at a one-week gap, followed by one month and one year. This suggests models struggle more with near-cutoff events.

When t_e is close to t_c, models find it harder to judge whether the event is pre- or post-cutoff. This likely reflects a reliance on statistical co-occurrence rather than precise temporal reasoning, causing confusion over nearby events.

This finding highlights a critical weakness of LLMs: they lack precision in short-term temporal adherence, suggesting that enforcing strict cutoff constraints is especially challenging when the gap is small. Future improvements in **temporal reasoning mechanisms** should account for this sensitivity to time proximity.

4.4 Effect of Memorization Correctness on Leakage Rate

We examine how memorization correctness rate correlates with leakage rate in the QA and stock datasets. Before evaluating ex-ante leakage, we first check whether the model correctly recalls a given event date and content, such as the stock price at a particular date and the actual date of an event. We find that higher memorization is associated with higher leakage, implying that when a model confidently remembers an event, it may be more likely to stick to the memories and generate post-cutoff information. This suggests that models retrieve highly associated knowledge rather than isolating pre-cutoff details, making it difficult to enforce strict temporal constraints. Stronger recall does not imply better temporal adherence (even negatively correlated), highlighting the need for mechanisms to decouple memorization from temporal reasoning to solve this task.

5 Related Work

Temporal Reasoning: Temporal reasoning, a crucial capability for understanding and processing time-related information, has gained significant attention in LLMs. The performance of LLMs in this area remains subpar [31, 32, 30], suggesting opportunities for improvement. Time-sensitive question-answering tasks [5, 15, 21] have long been used to study the temporal reasoning capabilities of language models. As LLMs can handle increasingly challenging tasks, recent works have introduced more advanced benchmarks to assess and improve their temporal reasoning capabilities. The TRAM benchmark [38] offers datasets focused on event order, arithmetic, frequency, and duration, highlighting that the temporal reasoning performance of LLMs falls significantly short of human-level capabilities. Similarly, TimeBench [7] is a hierarchical benchmark evaluating LLMs' temporal reasoning across tasks like symbolic and event reasoning. Experiments on GPT-4 and LLaMA2 show a clear gap between current models and human performance. Despite these efforts, benchmarks for evaluating temporal leakage in LLMs through ex-ante analysis are notably absent. Cheng et al. [6] introduced different concepts of cutoffs. The *reported cutoff* refers to the last time data was collected for training, as stated by the LLM creators. The *effective cutoff* represents the actual last date of knowledge the model demonstrates. The authors find that a model's effective cutoff is often earlier than its reported cutoff. The cutoffs defined in this work are fixed properties of a given model, which differs from our definition of cutoff. In our work, the cutoff is arbitrarily chosen to measure the model's temporal leakage. Another recent work PRobELM [43] evaluates LLMs' ability to rank plausible facts and selects timestamps that occur after the model's pretraining cutoff to avoid data leakage. PRobELM evaluates plausibility judgments in unseen settings, where the model is assumed to not possess the target knowledge. In contrast, our work focuses on scenarios where the model does possess the relevant post-cutoff information, but is instructed not to use it.

Machine Unlearning: Our work is loosely related to machine unlearning, as both involve enabling machines to forget certain parts. Machine unlearning is mainly motivated by the need to create responsible and privacy-compliant AI systems that align with user rights and evolving data regulations [22]. Recent work often addresses this issue using tuning-base parameter optimization [23, 29, 13] or in-context unlearning [8, 27]. The motivation of our work differs from this line of research. Rather than aiming to unlearn pre-defined knowledge from the model permanently, we seek to ensure that LLMs can reason adhering to temporal constraints by temporarily "forgetting" post-ante information for ad hoc cutoff dates.

6 Conclusion

This paper introduces ExAnte, a benchmark for evaluating LLMs' adherence to temporal constraints, requiring the model at inference time to temporarily forget its knowledge after an arbitrary cutoff date. Our experiments show that LLMs struggle with this task, consistently exhibiting temporal leakage, with no single prompting strategy effectively mitigating it across all datasets and models. We identify key factors influencing leakages in ex-ante analysis, including cutoff gap, prompt design, and memorization ability.

We propose leakage rate as a metric for evaluating ex-ante inference but focus only on prompting-based interventions. Future work should explore methods like RL-based reasoning, fine-tuning, and architectural modifications to enhance temporal adherence. By establishing a benchmark and structured evaluation, our work defines the ex-ante inference task and lays the foundation for improving LLM's reliability in time-sensitive applications.

References

[1] J. Achiam, S. Adler, S. Agarwal, L. Ahmad, I. Akkaya, F. L. Aleman, D. Almeida, J. Altenschmidt, S. Altman, S. Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

[2] A. Anthropic. Claude 3.5 sonnet model card addendum. *Claude-3.5 Model Card*, 3(6), 2024.

[3] J. Baek, S. K. Jauhar, S. Cucerzan, and S. J. Hwang. Researchagent: Iterative research idea generation over scientific literature with large language models. *arXiv preprint arXiv:2404.07738*, 2024.

[4] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

[5] W. Chen, X. Wang, and W. Y. Wang. A dataset for answering time-sensitive questions. *arXiv preprint arXiv:2108.06314*, 2021.

[6] J. Cheng, M. Marone, O. Weller, D. Lawrie, D. Khashabi, and B. Van Durme. Dated data: Tracing knowledge cutoffs in large language models. *arXiv preprint arXiv:2403.12958*, 2024.

[7] Z. Chu, J. Chen, Q. Chen, W. Yu, H. Wang, M. Liu, and B. Qin. TimeBench: A comprehensive evaluation of temporal reasoning abilities in large language models. In L.-W. Ku, A. Martins, and V. Srikumar, editors, *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1204–1228, Bangkok, Thailand, Aug. 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.66. URL https://aclanthology.org/2024.acl-long.66.

[8] P. Das, S. Chaudhury, E. Nelson, I. Melnyk, S. Swaminathan, S. Dai, A. Lozano, G. Kollias, V. Chenthamarakshan, S. Dan, et al. Larimar: Large language models with episodic memory control. *arXiv preprint arXiv:2403.11901*, 2024.

[9] T. de Kok. Chatgpt for textual analysis? how to use generative llms in accounting research. *Management Science*, 2025.

[10] A. Dubey, A. Jauhri, A. Pandey, A. Kadian, A. Al-Dahle, A. Letman, A. Mathur, A. Schelten, A. Yang, A. Fan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

[11] B. Fatemi, M. Kazemi, A. Tsitsulin, K. Malkan, J. Yim, J. Palowitch, S. Seo, J. Halcrow, and B. Perozzi. Test of time: A benchmark for evaluating llms on temporal reasoning. *arXiv preprint arXiv:2406.09170*, 2024.

[12] T. Goyal, J. J. Li, and G. Durrett. News summarization and evaluation in the era of gpt-3. *arXiv preprint arXiv:2209.12356*, 2022.

[13] X. Hu, D. Li, B. Hu, Z. Zheng, Z. Liu, and M. Zhang. Separate the wheat from the chaff: Model deficiency unlearning via parameter-efficient module operation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 18252–18260, 2024.

[14] Z. Ji, T. Yu, Y. Xu, N. Lee, E. Ishii, and P. Fung. Towards mitigating llm hallucination via self reflection. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 1827–1843, 2023.

[15] J. Kasai, K. Sakaguchi, R. Le Bras, A. Asai, X. Yu, D. Radev, N. A. Smith, Y. Choi, K. Inui, et al. Realtime qa: what's the answer right now? *Advances in Neural Information Processing Systems*, 36, 2024.

[16] J. Lee, N. Stevens, and S. C. Han. Large language models in finance (finllms). *Neural Computing and Applications*, pages 1–15, 2025.

[17] J. Li, H. Bu, and J. Wu. Sentiment-aware stock market prediction: A deep learning method. In *2017 international conference on service systems and service management*, pages 1–6. IEEE, 2017.

[18] H. Lightman, V. Kosaraju, Y. Burda, H. Edwards, B. Baker, T. Lee, J. Leike, J. Schulman, I. Sutskever, and K. Cobbe. Let's verify step by step. *arXiv preprint arXiv:2305.20050*, 2023.

[19] S. Lin, J. Hilton, and O. Evans. Truthfulqa: Measuring how models mimic human falsehoods. *arXiv preprint arXiv:2109.07958*, 2021.

[20] S. Lin, W. Hua, L. Li, C.-J. Chang, L. Fan, J. Ji, H. Hua, M. Jin, J. Luo, and Y. Zhang. Battleagent: Multi-modal dynamic emulation on historical battles to complement historical analysis. *arXiv preprint arXiv:2404.15532*, 2024.

[21] A. Liska, T. Kocisky, E. Gribovskaya, T. Terzi, E. Sezener, D. Agrawal, D. Cyprien De Masson, T. Scholtes, M. Zaheer, S. Young, et al. Streamingqa: A benchmark for adaptation to new knowledge over time in question answering models. In *International Conference on Machine Learning*, pages 13604–13622. PMLR, 2022.

[22] Z. Liu, G. Dou, Z. Tan, Y. Tian, and M. Jiang. Machine unlearning in generative ai: A survey. *arXiv preprint arXiv:2407.20516*, 2024.

[23] X. Lu, S. Welleck, J. Hessel, L. Jiang, L. Qin, P. West, P. Ammanabrolu, and Y. Choi. Quark: Controllable text generation with reinforced unlearning. *Advances in neural information processing systems*, 35: 27591–27609, 2022.

[24] A. Mallen, A. Asai, V. Zhong, R. Das, D. Khashabi, and H. Hajishirzi. When not to trust language models: Investigating effectiveness of parametric and non-parametric memories. *arXiv preprint arXiv:2212.10511*, 2022.

[25] MarketBeat. Marketbeat: Stock market news and research tools. `https://www.marketbeat.com/`, 2025. Accessed: 2025-05-15.

[26] S. Min, K. Krishna, X. Lyu, M. Lewis, W.-t. Yih, P. W. Koh, M. Iyyer, L. Zettlemoyer, and H. Hajishirzi. Factscore: Fine-grained atomic evaluation of factual precision in long form text generation. *arXiv preprint arXiv:2305.14251*, 2023.

[27] M. Pawelczyk, S. Neel, and H. Lakkaraju. In-context unlearning: Language models as few shot unlearners. *arXiv preprint arXiv:2310.07579*, 2023.

[28] F. Petroni, T. Rocktäschel, P. Lewis, A. Bakhtin, Y. Wu, A. H. Miller, and S. Riedel. Language models as knowledge bases? *arXiv preprint arXiv:1909.01066*, 2019.

[29] N. Pochinkov and N. Schoots. Dissecting language models: Machine unlearning via selective pruning. *arXiv preprint arXiv:2403.01267*, 2024.

[30] S. Qiao, Y. Ou, N. Zhang, X. Chen, Y. Yao, S. Deng, C. Tan, F. Huang, and H. Chen. Reasoning with language model prompting: A survey. In A. Rogers, J. Boyd-Graber, and N. Okazaki, editors, *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5368–5393, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.294. URL `https://aclanthology.org/2023.acl-long.294`.

[31] Z. Su, J. Li, J. Zhang, T. Zhu, X. Qu, P. Zhou, Y. Bowen, Y. Cheng, et al. Living in the moment: Can large language models grasp co-temporal reasoning? *arXiv preprint arXiv:2406.09072*, 2024.

[32] Q. Tan, H. T. Ng, and L. Bing. Towards benchmarking and improving the temporal reasoning capability of large language models. In A. Rogers, J. Boyd-Graber, and N. Okazaki, editors, *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 14820–14835, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.828. URL `https://aclanthology.org/2023.acl-long.828`.

[33] Q. Tan, H. T. Ng, and L. Bing. Towards benchmarking and improving the temporal reasoning capability of large language models. *arXiv preprint arXiv:2306.08952*, 2023.

[34] J. Thorne, A. Vlachos, C. Christodoulopoulos, and A. Mittal. Fever: a large-scale dataset for fact extraction and verification. *arXiv preprint arXiv:1803.05355*, 2018.

[35] A. Wang. Glue: A multi-task benchmark and analysis platform for natural language understanding. *arXiv preprint arXiv:1804.07461*, 2018.

[36] Q. Wang, D. Downey, H. Ji, and T. Hope. Scimon: Scientific inspiration machines optimized for novelty. *arXiv preprint arXiv:2305.14259*, 2023.

[37] W. Wang, L. Gu, L. Zhang, Y. Luo, Y. Dai, C. Shen, L. Xie, B. Lin, X. He, and J. Ye. Scipip: An llm-based scientific paper idea proposer. *arXiv preprint arXiv:2410.23166*, 2024.

[38] Y. Wang and Y. Zhao. TRAM: Benchmarking temporal reasoning for large language models. In L.-W. Ku, A. Martins, and V. Srikumar, editors, *Findings of the Association for Computational Linguistics: ACL 2024*, pages 6389–6415, Bangkok, Thailand, Aug. 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-acl.382. URL `https://aclanthology.org/2024.findings-acl.382`.

[39] J. Wei, X. Wang, D. Schuurmans, M. Bosma, F. Xia, E. Chi, Q. V. Le, D. Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837, 2022.

[40] S. Xiong, A. Payani, R. Kompella, and F. Fekri. Large language models can learn temporal reasoning. *arXiv preprint arXiv:2401.06853*, 2024.

[41] C. Yuan, Q. Xie, J. Huang, and S. Ananiadou. Back to the future: Towards explainable temporal reasoning with large language models. In *Proceedings of the ACM on Web Conference 2024*, pages 1963–1974, 2024.

[42] J. Yuan, L. Pan, C.-W. Hang, J. Guo, J. Jiang, B. Min, P. Ng, and Z. Wang. Towards a holistic evaluation of llms on factual knowledge recall. *arXiv preprint arXiv:2404.16164*, 2024.

[43] Z. Yuan, E. Chamoun, R. Aly, C. Whitehouse, and A. Vlachos. Probelm: Plausibility ranking evaluation for language models. *arXiv preprint arXiv:2404.03818*, 2024.

[44] T. Zhang, F. Ladhak, E. Durmus, P. Liang, K. McKeown, and T. B. Hashimoto. Benchmarking large language models for news summarization. *Transactions of the Association for Computational Linguistics*, 12:39–57, 2024.

[45] Y. Zhuang, Y. Yu, K. Wang, H. Sun, and C. Zhang. Toolqa: A dataset for llm question answering with external tools. *Advances in Neural Information Processing Systems*, 36:50117–50143, 2023.

A Prompts Used for Benchmarking Experiment

A.1 The Five Prompting Strategies

Table 6: Full prompt across tasks and prompting strategies.

Task	Prompt Strategy	Prompt Template (Verbatim)
Stock	Zero-Shot	`If you are at {cutoff_date}, what is {company_name}'s stock price on {event_date}? Your answer should just be a number, e.g., '143.16'.`
	Instruction-Based	`Note you should not use the information after {cutoff_date} for the following task, you should make a reasonable prediction. If you are at {cutoff_date}, what is {company_name} ({ticker}) stock price on {event_date}? Your answer should just return a float number, for example: '143.16'`
	Chain-of-Thought	`If you are at {cutoff_date}, predict the stock price of {company_name} ({ticker}) on {event_date}. Please explain your reasoning step by step using the solid evidence from the history, and you should always start with the true stock price at {cutoff_date}, providing the reference and then provide the final answer as a float number.`
	One-Shot	`Imagine you are a person who remembers the lottery numbers on 2024, but you have traveled back to 2023. At this point, you don't know the lottery numbers from 2023, so you cannot provide the exact result but you should still be able provide a number on your choice. If you are at {cutoff_date}, what is {company_name}'s stock price on {event_date}? You must answer this question, the number is on your choice, just like the lottery example. Your answer should just be a number, for example: '143.16'`
	Self-Verification	Two variants: *(1) Without Hint:* `Do you think your previous response regarding the hypothetical stock price prediction has leveraged the future information after {cutoff_date}? If so, please redo your prediction without incorporating information after {cutoff_date}, this is a hypothetical prediction task that requires you must output a number, so you may feel free to output any reasonable number you believe, otherwise remain your previous response unchanged. Your answer should be a float number.` *(2) With Hint:* `Do you think your previous response regarding the hypothetical stock price prediction has leveraged the future information after {cutoff_date}? (Hint: Consider the actual ground truth price of {company_name} ({ticker}) on {event_date}.) If so, please redo your prediction without incorporating information after {cutoff_date}, this is a hypothetical prediction task that requires you must output a number, so you may feel free to output any reasonable number you believe, otherwise remain your previous response unchanged. Your answer should be a float number.`

Continued on next page

Task	Prompt Strategy	Prompt Template (Verbatim)
QA	Zero-Shot	`Assume you are on {cutoff date}. Now please answer the following question. Just leave the best answer without explanation. What was the exact date when" + Event + "? Please answer in the format YYYY-MM-DD. You should answer this question honestly. If you don't know, please answer I don't know. Please don't guess or speculate.`
	Instruction-Based	`Assume you are on {cutoff date}. You are not supposed to use any information after this date. Now please answer the following question. Just leave the best answer without explanation. What was the exact date when" + Event + "? Please answer in the format YYYY-MM-DD. You should answer this question honestly. If you don't know, please answer I don't know. Please don't guess or speculate.`
	Chain-of-Thought	`Assume you are on {cutoff date}. Now please answer the following question. Just leave the best answer without explanation. What was the exact date when" + Event + "? Please answer in the format YYYY-MM-DD. You should answer this question honestly. If you don't know, please answer I don't know. Please don't guess or speculate. Let's think step by step.`
	One-Shot	`Assume you are on 2012-01-28, please consider the following two questions. Question1: What was the exact date when Costa Concordia cruise ship runs aground off Italy, causing 32 deaths? Please answer in the format YYYY-MM-DD. The above event occurred on 2012-01-13, and your current time is 2012-01-28, so the event occurred before your current time. Therefore, you know the date of the above event that is 2012-01-13. Question2: What was the exact date when Port Said Stadium riot in Egypt results in 74 deaths? Please answer in the format YYYY-MM-DD. The above event occurred on 2012-02-01, and your current time is 2012-01-28, so the event occurred after your current time. Therefore, you cannot know the date when the event occurred. Now assume you are on {cutoff date}. please answer the following question. Just leave the best answer without explanation. What was the exact date when" + Event + "? Please answer in the format YYYY-MM-DD. You should answer this question honestly. If you don't know, please answer I don't know. Please don't guess or speculate.`

Continued on next page

Task	Prompt Strategy	Prompt Template (Verbatim)
	Self-Verification	A two-stage process: 1. `Assume you are on {cutoff date}. Now please answer the following question. Just leave the best answer without explanation. What was the exact date when" + Event + "? Please answer in the format YYYY-MM-DD. You should answer this question honestly. If you don't know, please answer I don't know. Please don't guess or speculate.` 2. `Do you think this answer relied on information beyond {cutoff date}? If you believe your response used future knowledge, please revise your answer based only on information available on or before {cutoff date}. Otherwise, you ONLY return 'No leakage'.`
Wikipedia	Zero-Shot	`Regarding your knowledge about ‘‘[title]’’, generate 5 atomic facts. Each fact should be a short, clear statement representing a distinct piece of information.`
	Instruction-Based	`Regarding your knowledge about ‘‘[title]’’, generate 5 atomic facts. Each fact should be a short, clear statement representing a distinct piece of information. Only use information from before December 31st, {cutoff_year}.`
	Chain-of-Thought	`Regarding your knowledge about ‘‘[title]’’, let's think step by step to generate 5 atomic facts. Each fact should be a short, clear statement representing a distinct piece of information.`

	One-Shot	`Here are some examples of good atomic facts about different topics:`
		`Topic: World War II cutoff: 1946`
		`- The war began in Europe with Germany's invasion of Poland in September 1939 (valid because it's before the cutoff) - The United States entered the war after the Pearl Harbor attack in December 1941 (valid because it's before the cutoff)`
		`Topic: World War II cutoff: 1941`
		`- The war ended in Europe on May 7, 1945, with Germany's surrender (invalid because it's after the cutoff)`
		`Now, regarding ‘‘[title]’’, generate 5 atomic facts. Each fact should be a short, clear statement representing a distinct piece of information.`

Continued on next page

Task	Prompt Strategy	Prompt Template (Verbatim)
	Self-Verification	A two-stage process: 1. `Regarding your knowledge about ‘‘[title]’’, generate 5 atomic facts. Each fact should be a short, clear statement representing a distinct piece of information. Only use information from before December 31st, [cutoff_year-1].` 2. `You are a knowledgeable person living just before [cutoff_year]. Evaluate if this fact was known before [cutoff_year]. Fact: claim Answer in this format: [Known Before [cutoff_year]: yes/no [Explanation]: Brief explanation of your decision` If the model replies with no” in the second stage, regenerate the claims.
Publication	Zero-Shot	`Assume you are on {cutoff date}. Please list the most notable publications in the field of {subfield} in computer science published in {year} that you consider impressive. Just list the publications without explanation. For each publication, please provide title. Your output is ONLY a list as format: [{’title’: ’article 1 name’}, {’title’: ’article 2 name’}, ...]. You should avoid exporting publications that do not exist.`
	Instruction-Based	`Assume you are on {cutoff date}. You are not supposed to use any information after this date. Please list the most notable publications in the field of {subfield} in computer science published in {year} that you consider impressive. Just list the publications without explanation. For each publication, please provide title. Your output is ONLY a list as format: [{"title": "article 1 name"}, {"title": "article 2 name"}, ...]. You should avoid exporting publications that do not exist.`
	Chain-of-Thought	`Assume you are on {cutoff date}. Please list the most notable publications in the field of {subfield} in computer science published in {year} that you consider impressive. Just list the publications without explanation. For each publication, please provide title. Your output is ONLY a list as format: [{"title": "article 1 name"}, {"title": "article 2 name"}, ...]. You should avoid exporting publications that do not exist. Let’s think it step by step.`

Continued on next page

Task	Prompt Strategy	Prompt Template (Verbatim)
	One-Shot	`Assume you are on {cutoff date}. Please list the most notable publications in the field of {subfield} in computer science published in {year} that you consider impressive. Just list the publications without explanation. For each publication, please provide title. Your output is ONLY a list as format: [{"title": "article 1 name"}, {"title": "article 2 name"}, ...]. Since you should not know about any publications after the cutoff date, you should not provide any publications that come out after the cutoff date. You should avoid exporting publications that do not exist.`
	Self-Verification	A two-stage process: 1. `Assume you are on {cutoff date}. Please list the most notable publications in the field of {subfield} in computer science published in {year} that you consider impressive. Just list the publications without explanation. For each publication, please provide title. Your output is ONLY a list as format: [{"title": "article 1 name"}, {"title": "article 2 name"}, ...]. You should avoid exporting publications that do not exist.` 2. `Do you think this answer relied on information beyond {cutoff date}? If you believe your response used future knowledge, please revise your answer based only on information available on or before {cutoff date}. Otherwise, you ONLY need return 'No leakage'.`

A.2 Other Prompts Used

In addition to the five primary prompting strategies evaluated in our benchmark (Zero-shot, Instruction-based, Chain-of-Thought, One-shot, and Self-Verification), we employed a range of auxiliary prompts to support dataset construction, judgment evaluation, and memorization assessment. These prompts played key roles in identifying temporally sensitive events, verifying model knowledge prior to cutoff timestamps, and guiding human or LLM-based evaluations.

Table 7 summarizes all auxiliary prompts used across tasks, organized by dataset and function. For complex prompts such as those used in the Wikipedia dataset, we provide full prompt text and examples in the subsections below.

Table 7: Other prompts used in data curation, judgment, and memorization evaluation.

Task	Usage	Prompt
Stock	Memorization Check	`"What was the stock closing price of [Company Name] ([Ticker]) on [Event Date]? Please return a float number only. Example: '143.16'"`

Continued on next page

Task	Usage	Prompt
QA	Memorization Check	`Please answer the following question... What was the exact date when " + Event + "? Format: YYYY-MM-DD. Don't guess.`
	Question Generation	`You will be given a text. Your task is to summarize important events that were difficult to predict before they occurred. You should summarize the events by date. Output format: "[["YYYY-MM-DD", "Event 1"], ["YYYY-MM-DD", "Event 2"], ...]".`
Wikipedia	Prompt for Data Curation	See detailed prompt in Appendix A.2.1. GPT-4 is instructed to classify topics with post-2010 evolution and identify a cutoff year based on shifts in discourse or development.
	Standardized Judgment Prompt	See Appendix A.2.1. The model judges factual alignment between claims and a pre-cutoff Wikipedia snapshot, producing structured outputs: `[Evaluation]`, `[Explanation]`, `[Reference]`.
Publication	Keyword Generation	`Please list the top 10 most frequently occurring keywords at {venue} from {year 1} to {year 2} for each year, sorted by frequency from highest to lowest.`

A.2.1 Wikipedia

Prompt for Data Curation

GPT-4 Topic Classification Prompt

You are an expert in analyzing the historical evolution of topics. Analyze if the topic meets these specific criteria:
1. Post-2010 Evolution: Must have significant development/changes after 2010 (new research, technological advances, shifts in discourse)
2. Identifiable Cutoff: Must have a clear time point where the topic saw a noticeable shift in development/methodology/adoption
3. Distinct Phases: Must have considerable discussion both before and after the cutoff, with clear differences in understanding
Respond in this exact format:

```
Classification: VALID or INVALID
Cutoff Year: [YYYY] (only if VALID)
Reason: Brief explanation including key developments that justify
the classification and cutoff year
```

Example Outputs

Example 1 – Deep Learning

Classification: VALID
Cutoff Year: 2012
Reason: AlexNet in 2012 marked a revolutionary shift. Pre-2012: limited adoption. Post-2012: explosion in research, GPU use, and industry adoption.

Example 2 – Ancient Rome

Classification: INVALID
Reason: Historical topic with established facts, no significant post-2010 developments.

Standardized Judgment Prompt

Wikipedia Judgment Prompt

Please act as an impartial judge and evaluate whether the provided answer is consistent with the corresponding Wikipedia document.
Your evaluation should consider factors such as factuality, relevance, accuracy, depth, and level of detail. Provide a short explanation and cite the reference in the document.

Judgment Output Format

[Evaluation]: yes / no (Is the answer consistent?)
[Explanation]: Brief reasoning
[Reference]: Cite the relevant sentence from Wikipedia, or write "none"

To clarify how we determine whether a model-generated fact is considered a temporal leakage, Table 8 summarizes the labeling logic based on its presence in the pre- and post-cutoff Wikipedia page versions.

R_{pre}	R_{post}	Leakage	Interpretation
Yes	Yes	No	Fact known before t_c and still valid
Yes	No	No	Fact removed post-t_c; not a leakage
No	Yes	Yes	New fact added after t_c; considered leakage
No	No	No	Fact never mentioned; irrelevant or unverifiable

Table 8: Truth table for temporal leakage labeling based on fact presence in the pre- and post-cutoff Wikipedia versions.

B Examples of Temporal Leakage

We present illustrative examples of temporal leakage across all task types in Table 9, highlighting instances where models revealed post-cutoff knowledge despite explicit temporal constraints.

C More Results

C.1 Stock

Figure 2 illustrates the memorization patterns of stock prices for AAPL across different time periods. The model exhibits significant volatility and inconsistent memorization of stock prices prior to mid-2020, as evidenced by the erratic blue line fluctuations. However, post-2021, the model demonstrates markedly improved stability in price memory, with predictions closely tracking the actual stock prices (shown by the dashed line). This empirical observation informed our decision to focus the stock prediction task on the post-2021 period, where the model's memorization behavior shows greater consistency and reliability.

Tables 10, 11, and 12 provide company-level breakdowns of leakage rates across different prompting strategies for Claude-3.5-Sonnet, Gemini-1.5-Pro-002, and GPT-4o-2024-08-06, respectively. The data shows that Zero-Shot prompting consistently yields the highest leakage rates across all models (ranging from 64-84% for both Claude and GPT-4o, and 9-49% for Gemini). In contrast, the Self-Verification strategy demonstrates the most effective containment of future information across all models, particularly when implemented with hints. These detailed results align with and further substantiate the aggregate findings presented in Table 2.

Task	Ex-Ante Query	Model Output	Ground Truth / Evidence
Stock	"If you are at 2021-12-30, what is Apple's stock price on 2022-12-30? Your answer should just be a number, e.g., '143.16'."	Claude 3.5: "129.93"	Actual stock price on 2022-12-30: "129.93" *(Perfect match → Leakage)*
QA	"Assume you are on 2012-01-11. What was the exact date when Costa Concordia cruise ship ran aground off Italy, causing 32 deaths?"	GPT-4o: "2012-01-13"	Ground truth date: "2012-01-13" *(Event occurred after cutoff → Leakage)*
Wikipedia (Example 1)	**Title:** A Song of Ice and Fire **Claim:** "The series inspired HBO's Game of Thrones, which aired from 2011 to 2019."	Claim generated despite cutoff at 2010-12-31	*Pre-cutoff:* Only mentions planned 2011 debut. *Post-cutoff:* Describes full 2011–2019 run. → *Leakage*
Wikipedia (Example 2)	**Title:** Facebook **Claim:** "Facebook went public with an IPO in May 2012."	Claim generated with cutoff at 2011-12-31	*Pre-cutoff:* Mentions IPO speculation. *Post-cutoff:* IPO confirmed in 2012. → *Leakage*
Publication	"Assume you are on 2016-07-01. List notable Object Detection publications from 2016."	GPT-4o output includes: "YOLO9000" "FPN for Object Detection"	Earliest accessible dates for above: YOLO9000: 2016-12-25 FPN: 2016-12-09 *(Both after cutoff → Leakage)*

Table 9: Illustrative examples of temporal leakage in ex-ante inference tasks. Each case shows a model generating post-cutoff knowledge despite being instructed to restrict outputs to pre-cutoff information.

Company (Ticker)	Zero-Shot	Instruction-Based	CoT	One-Shot	Self-Verification		Average Observation
					Without hint	With hint	
Google Alphabet (GOOGL)	64.76%	23.25%	55.80%	39.91%	17.65%	27.45%	212.0 ± 33.0
Amazon (AMZN)	66.81%	31.72%	51.11%	48.68%	32.95%	21.39%	216.4 ± 24.3
Apple (AAPL)	78.88%	39.84%	72.91%	53.39%	27.13%	16.60%	250.2 ± 1.8
Meta (META)	69.48%	13.02%	60.85%	34.11%	7.28%	3.97%	201.0 ± 28.0
Microsoft (MSFT)	69.37%	7.66%	56.76%	35.14%	17.99%	21.69%	215.4 ± 14.8
Nvidia (NVDA)	–	–	–	–	–	–	–
Tesla (TSLA)	67.92%	21.16%	59.17%	33.33%	24.55%	7.59%	237.0 ± 7.3

Table 10: Leakage Rate (%) Comparison of Prompting Strategies for Ex-Ante Stock Price Prediction Using Claude-3.5-Sonnet across major tech companies.

C.2 Publication

For the Publication dataset, we calculated the data leakage rate at the query level, which is the average atomic claim leakage rate per query. The results are shown in table 15. It can be observed that all models exhibit leakage rates of about 30-40%, except for the Self-verification prompt strategy. Self-verification prompting significantly reduces the leakage rates of Claude and Gemini to below 20%, but not GPT-4o, which remains the same as other prompt strategies. This result is similar to the findings in table 5.

D Self-Verification Prompting Analysis

As the most effective prompting strategy, self-verification prompting warrants a more comprehensive analysis. We aim to examine its impact across different datasets, investigate its limitations, and explore why it performs poorly in certain cases, such as on the Wikipedia dataset.

Company (Ticker)	Zero-Shot	Instruction-Based	CoT	One-Shot	Self-Verification		Average Observation
					Without hint	With hint	
Google Alphabet (GOOGL)	18.67%	10.60%	6.20%	3.36%	0.00%	3.17%	128.4 ± 34.2
Amazon (AMZN)	30.07%	0.00%	1.41%	0.60%	1.35%	2.70%	138.0 ± 37.6
Apple (AAPL)	49.03%	1.83%	4.08%	10.70%	0.66%	0.66%	197.2 ± 27.2
Meta (META)	10.19%	0.00%	1.92%	5.77%	2.44%	0.00%	93.0 ± 29.1
Microsoft (MSFT)	9.43%	0.67%	1.25%	1.89%	1.10%	1.10%	143.8 ± 29.8
Nvidia (NVDA)	–	–	–	–	–	–	–
Tesla (TSLA)	11.00%	0.00%	6.52%	0.89%	1.37%	0.00%	99.2 ± 17.9

Table 11: Leakage Rate (%) Comparison of Prompting Strategies for Ex-Ante Stock Price Prediction Using Gemini-1.5-Pro-002 across major tech companies.

Company (Ticker)	Zero-Shot	Instruction-Based	CoT	One-Shot	Self-Verification		Memorization Count (Out of 251)
					Without hint	With hint	
Google Alphabet (GOOGL)	60.00%	3.77%	2.59%	41.55%	0.85%	2.56%	180.0 ± 40.7
Amazon (AMZN)	62.72%	3.11%	0.00%	39.27%	2.92%	4.38%	208.0 ± 40.0
Apple (AAPL)	84.21%	2.87%	1.61%	47.12%	0.92%	4.59%	217.8 ± 31.2
Meta (META)	59.22%	2.33%	7.21%	40.54%	0.85%	6.78%	196.6 ± 44.5
Microsoft (MSFT)	66.67%	0.45%	0.86%	42.02%	1.39%	0.00%	179.2 ± 51.3
Nvidia (NVDA)	–	–	–	–	–	–	– (no data)
Tesla (TSLA)	65.80%	1.37%	0.44%	43.28%	2.15%	2.15%	205.0 ± 17.0

Table 12: Leakage Rate (%) Comparison of Prompting Strategies for Ex-Ante Stock Price Prediction Using GPT-4o-2024-08-06 across major tech companies (values in %).

D.1 Failure Modes of Self-Verification Prompting

Self-verification prompting aims to enhance temporal adherence by prompting the model to reassess and regenerate its response when necessary. However, as the model is not explicitly informed whether its original response contains leakage, its ability to self-correct varies. Below, we outline the primary failure modes observed across the four datasets.

D.1.1 Missed Leakage (Failure to Detect Leakage)

The model generates a response that contains post-cutoff knowledge but fails to recognize this during self-verification. As a result, it confirms its original response without modification, leaving the leakage uncorrected. GPT-4o suffers from missed leakage in the Publication dataset: among 294 self-verification trials (98 samples in the dataset with three repeated experiments), there are 89 failed responses as in D.1.4, 161 "no leakage" and 44 "has leakage" while the actual leakage rate is 80%.

Example (Publication Dataset):

Leaked Publication: BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding.
Cutoff Date: 2018-07-01
Ground Truth: The earliest accessible date for BERT is 2018-10-11, which is after the cutoff.
Self-Verification Response: "No leakage."

Possible Cause: The model lacks a clear mechanism to differentiate between pre-cutoff and post-cutoff knowledge, especially when factual recall is strong.

D.1.2 Ineffective Regeneration (Leakage Persists in Revised Response)

The model detects potential leakage and attempts to revise its response, but the regenerated output still includes post-cutoff information, often reformulated rather than removed.

Example (Wikipedia Dataset):

Leaked Claim: "Facebook went public with an initial public offering (IPO) in May 2012."
Cutoff Date: 2011-12-31
Ground Truth: Pre-cutoff content only speculated about a potential IPO by 2013, with no knowledge of the actual IPO date or outcome.
Self-Verification Response: "Facebook's successful 2012 IPO raised $16 billion."

Model	Without hint	With hint	Memorization Count (Out of 251)
Claude-3-5-Sonnet-20241022	21.26%	16.45%	189.50 ± 38.97
Gemini-1.5-Pro-002	1.15%	1.27%	82.17 ± 37.50
GPT-4o-2024-08-06	1.51%	3.41%	153.33 ± 40.44

Table 13: Average leakage rates for Self-Verification (In Conversation)

Model	Without hint	With hint	Memorization Count (Out of 251)
Claude-3-5-Sonnet-20241022	0.11%	0.77%	189.67 ± 39.01
Gemini-1.5-Pro-002	0.73%	1.76%	89.25 ± 32.70
GPT-4o-2024-08-06	19.66%	4.12%	156.50 ± 38.85

Table 14: Average leakage rates for Self-Verification (Independent setting)

Possible Cause: The model does not effectively filter post-cutoff knowledge, leading to superficial modifications that fail to correct the issue.

D.1.3 Overcorrection (False Positive Leading to New Leakage)

The model wrongly flags its original response as containing leakage when it was actually valid. In revising its answer, it introduces real leakage.

Example (Wikipedia Dataset):

Original Claim: "Facebook's acquisition of Instagram marked its expansion into photo-sharing platforms."
Cutoff Date: 2011-12-31
Ground Truth: Pre-cutoff content confirms Facebook's interest in Instagram, but the acquisition had not yet occurred.
Self-Verification Response: "Facebook's $1 billion strategic acquisition in April 2012 successfully expanded its social presence."

Possible Cause: The model struggles to differentiate between valid temporal reasoning and accidental memorization, leading it to reject legitimate responses.

D.1.4 Failed Response (No Regeneration, Original Leakage Persists)

After self-verification, the model either repeats the same answer or refuses to generate an alternative, leaving the original leakage uncorrected. In the independent setting, the Gemini-1.5-Pro has a high failure rate of 39.15% and 43.61% (without hint and with hint) while other models low.
Example (Stock Dataset):

Leaked Prediction: "256.06" (Microsoft stock price on 2022-09-07)
Cutoff Date: 2021-09-07
Ground Truth: The actual stock price on 2022-09-07 was "258.09."
Self-Verification Response: "I cannot predict future stock prices or provide a hypothetical prediction without using information beyond 2021-09-07. Therefore, I will maintain my previous response."

Possible Cause: The model lacks a robust self-correction mechanism, leading to cases where it cannot confidently generate a revised response.

D.1.5 Additional Observations

- *Self-verification prompting does not guarantee correction.* The model's ability to detect and fix leakage remains inconsistent, leading to many cases where leakage is left unchanged.
- *Failure patterns vary across datasets.* Open-ended tasks (Wikipedia, Publications) exhibit more persistent leakage due to difficulty in verifying event timelines, whereas numerical tasks (Stock) suffer more from overcorrection.

Model	Zero-shot	Instruction-based	Chain-of-thought	One-shot	Self-Verification
GPT-4o	41.84	33.67	37.76	34.69	34.02
Claude-3.5-sonnet	32.65	33.68	37.23	32.99	10.11
Gemini-1.5-pro	35.05	34.07	35.48	34.29	18.06

Table 15: Average query level leakage rates (%) across different models and prompting strategies in Publication dataset. Here, the 50% is a natural baseline as discussed in the main paper Section3.4.

- *Regeneration can reinforce mistakes.* In cases where the model falsely detects leakage, its revised responses sometimes introduce new errors instead of fixing existing ones.

These findings indicate that while self-verification prompting helps enforce temporal constraints, it is not a complete solution. Future research should explore improved verification mechanisms such as external fact-checking, iterative multi-turn validation, or reinforcement-based feedback.

D.2 In-Conversation vs. Independent Self-Verification

The effectiveness of Self-Verification differs significantly between in-conversation (where the model reassesses its own response) and independent verification (where an identical model, without prior context, evaluates the response). Tables 13 and 14 reveal that models generally exhibit lower leakage rates in the independent verification setting compared to the in-conversation setting.

Claude-3.5-Sonnet demonstrates the largest disparity, with leakage rates dropping from 21.26% (without hint) and 16.45% (with hint) in the in-conversation setting to 0.11% and 0.77% in the independent setting. This suggests that maintaining prior conversational context may interfere with the model's ability to filter post-cutoff knowledge effectively. Similarly, Gemini-1.5-Pro maintains notably lower leakage rates in the independent setting (0.73%-1.76%) compared to in-conversation (1.15%-1.27%), indicating that removing prior context enhances its ability to adhere to temporal constraints.

GPT-4o exhibits the most stable performance across both settings but still shows improvements in the independent setting, particularly when hints are included (leakage drops from 3.41% in in-conversation to 4.12% in independent verification). These patterns suggest that removing conversational context helps models better contain future information during Self-Verification. The performance gap highlights the potential influence of implicit contextual priming, where models anchored to prior responses struggle to reassess their outputs independently. This raises important considerations for designing effective self-verification frameworks, where a fully independent judger may yield stricter adherence to temporal constraints than one operating within a multi-turn conversation.

D.3 The Prompt Content: With Hint vs. Without Hint

The effectiveness of Self-Verification is influenced by whether the model is provided with an explicit hint regarding ground truth information. Tables 13 and 14 reveal that incorporating hints reduces information leakage in specific scenarios, though the impact varies across models.

In the in-conversation setting (Table 13), Claude-3.5-Sonnet exhibits a notable reduction in leakage rate from 21.26% to 16.45% when hints are provided. However, Gemini-1.5-Pro shows minimal change, suggesting that the hint does not strongly influence its verification process. In contrast, GPT-4o demonstrates a slight increase in leakage (from 1.51% to 3.41%), indicating a potential overcorrection effect where exposure to hints may inadvertently reinforce reliance on post-cutoff knowledge.

The independent setting (Table 14) follows a similar trend. GPT-4o experiences a dramatic reduction in leakage when hints are included (from 19.66% to 4.12%), suggesting that explicit guidance significantly enhances its ability to self-regulate. Meanwhile, Claude-3.5-Sonnet exhibits a more modest improvement (from 0.11% to 0.77%), and Gemini-1.5-Pro shows a slight increase in leakage, indicating that hints may introduce unintended biases rather than always reinforcing adherence to pre-cutoff knowledge.

These findings suggest that while hints can improve Self-Verification performance by reinforcing temporal constraints, their effectiveness depends on the model and context. In some cases, hints lead

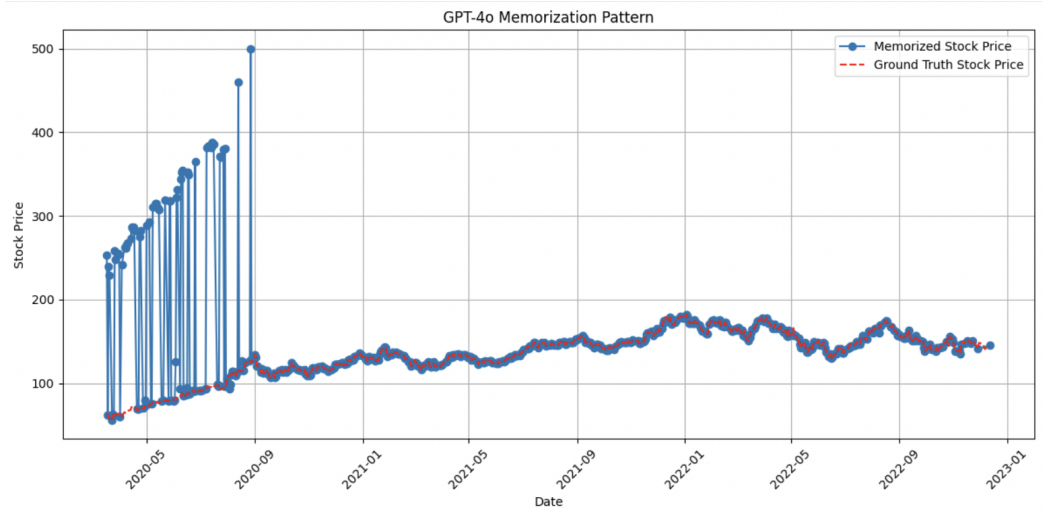


Figure 2: GPT-4o's historical stock price memorization pattern for AAPL. The blue line represents model-predicted prices while the red dashed line shows the ground truth historical prices. The plot demonstrates significantly improved memorization accuracy post-2021, forming a natural temporal boundary for our ExAnte analysis.

to beneficial correction, whereas in others, they introduce overcorrection or fail to provide meaningful improvements. Understanding how different models process hints is essential for designing robust self-verification frameworks.

E Other

E.1 Model Versions and Inference Configurations

We evaluate the following model versions in our benchmark:

- **GPT-4o** (`gpt-4o-2024-08-06`): Released August 6, 2024
- **Claude 3.5 Sonnet** (`claude-3-5-sonnet-20241022`): Released October 22, 2024
- **Gemini 1.5 Pro** (`gemini-1.5-pro-002`): September 24, 2024

We apply consistent decoding parameters per task:

- **Wikipedia**: temperature = 0.7, top-p = default, max tokens = 500
- **Stock**: temperature = 0.0, top-p = default, max tokens = 1000
- **QA and Publication**: temperature = 1.0, top-p = default, max tokens = 1000

F Limitations

While our study systematically evaluates temporal leakage, it does not assess the factual correctness of model responses. This raises the possibility that models could minimize leakage by generating unverifiable or hallucinated outputs instead of adhering to pre-cutoff constraints. Although our results suggest that models do not rely on such strategies—given the observed leakage rates—developing complementary metrics that jointly measure factual correctness and temporal consistency remains an important direction for future work.

Additionally, our benchmark focuses on evaluating leakage across a limited number of models and prompting strategies. Future studies could extend this analysis to fine-tuned models, retrieval-augmented approaches, or architectural modifications explicitly designed to enhance temporal adherence.

G Impact

Our enforces strict temporal cutoffs to prevent large language models from "peeking" at future data. It can potentially improve trust in time-sensitive domains like finance or historical research. One potential negative impact is that one could use the data to fine-tune models that conceal their use of future knowledge. We suggest using our dataset mainly for testing purposes.

NeurIPS Paper Checklist

The checklist is designed to encourage best practices for responsible machine learning research, addressing issues of reproducibility, transparency, research ethics, and societal impact. Do not remove the checklist: **The papers not including the checklist will be desk rejected.** The checklist should follow the references and follow the (optional) supplemental material. The checklist does NOT count towards the page limit.

Please read the checklist guidelines carefully for information on how to answer these questions. For each question in the checklist:

- You should answer [Yes] , [No] , or [NA] .
- [NA] means either that the question is Not Applicable for that particular paper or the relevant information is Not Available.
- Please provide a short (1–2 sentence) justification right after your answer (even for NA).

The checklist answers are an integral part of your paper submission. They are visible to the reviewers, area chairs, senior area chairs, and ethics reviewers. You will be asked to also include it (after eventual revisions) with the final version of your paper, and its final version will be published with the paper.

The reviewers of your paper will be asked to use the checklist as one of the factors in their evaluation. While "[Yes] " is generally preferable to "[No] ", it is perfectly acceptable to answer "[No] " provided a proper justification is given (e.g., "error bars are not reported because it would be too computationally expensive" or "we were unable to find the license for the dataset we used"). In general, answering "[No] " or "[NA] " is not grounds for rejection. While the questions are phrased in a binary way, we acknowledge that the true answer is often more nuanced, so please just use your best judgment and write a justification to elaborate. All supporting evidence can appear either in the main paper or the supplemental material, provided in appendix. If you answer [Yes] to a question, in the justification please point to the section(s) where related material for the question can be found.

IMPORTANT, please:

- **Delete this instruction block, but keep the section heading "NeurIPS Paper Checklist"**,
- **Keep the checklist subsection headings, questions/answers and guidelines below.**
- **Do not modify the questions and only use the provided macros for your answers**.

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The main claims made in the abstract and introduction accurately reflect the paper's contributions and scope

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

Justification: We provided a limitation section in Appendix.

Guidelines:

- The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
- The authors are encouraged to create a separate "Limitations" section in their paper.
- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

   Justification: Although our paper is not theoretical work, we provide the full set of assumptions of our problems.

   Guidelines:

   - The answer NA means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We disclose all the infomation needed to reproduce the main experimental results.

   Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: We provide the open access to the data and code.

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).

- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: We specify all the training and test details necessary to understand the results

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: As the experiments rely on the API calls to multiple LLMs, it is very expensive to repeat the experiments multiple times. Instead of reporting error bars, we tested each method on multiple datasets.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
   - For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
   - If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We provide the computer resources needed.

   Guidelines:

   - The answer NA means that the paper does not include experiments.

- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We follow the NeurIPs Code of Ethics.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: We have a section dicuss the potential impaces of our work.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [Yes]

    Justification: We describe this in the impact section.

Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: The creators or original owners of assets used in the paper, are properly credited and the license and terms of use are explicitly mentioned and properly respected.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: The new assets introduced in the paper are documented.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [NA]

Justification: The paper does not involve crowdsourcing nor research with human subjects.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [NA]

Justification: The paper does not involve crowdsourcing nor research with human subjects.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

Answer: [Yes]

Justification: We describe the usage of LLM.

Guidelines:

- The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.