Teaching Large Language Models When Not to Know: Learning Temporal Critique for Ex-Ante Reasoning

Chenlu Ding[1] **Jiancan Wu**[1,*] **Yanchen Luo**[2]
Zheyuan Liu[3] **Yancheng Yuan**[2,*] **Xiang Wang**[1]
[1]University of Science and Technology of China
[2]The Hong Kong Polytechnic University
[3]University of Notre Dame
dingchenlu200103@gmail.com

Abstract

Large language models (LLMs) often fail to reason under temporal cutoffs: when prompted to answer from the standpoint of an earlier time, they exploit knowledge that became available only later. We study this failure through the lens of ex-ante reasoning, where a model must rely exclusively on information knowable before a cutoff. Through a systematic analysis of prompt-level interventions, we find that temporal leakage is highly sensitive to cutoff formulation and instruction placement: explicit cutoff statements outperform implicit historical framings, and prefix constraints reduce leakage more effectively than suffix constraints. These findings indicate that prompting can steer models into a temporal frame, but does not endow them with the ability to verify whether a response is temporally admissible. We further argue that supervised fine-tuning is insufficient, since ex-ante correctness is not an intrinsic property of an answer, but a relation between the answer and the cutoff. To address this gap, we propose **TCFT**, a Temporal Critique Fine-Tuning framework that trains models to acquire cutoff-aware temporal verification. Given a query, a cutoff, and a candidate response, TCFT teaches the model to identify post-cutoff leakage, explain temporal boundary violations, and judge temporal admissibility. Experiments with Qwen2.5-7B-Instruct and Qwen2.5-14B-Instruct show that TCFT consistently outperforms prompting and SFT baselines, reducing average leakage by 41.89 and 37.79 percentage points[2], respectively.

1 Introduction

Suppose today is May 1, 2015. How would you describe Twitter? A human answering this question would naturally inhabit that moment — recalling a 140-character platform led by Dick Costolo — while instinctively setting aside what came later: Musk's 2022 acquisition, and the rebranding to X. This capability is referred to as **ex-ante reasoning** [1], the ability to reason under a strict temporal cutoff, drawing exclusively on information available up to that point. Far from a curiosity, it underpins any application whose validity depends on *what was knowable at the time*, from backtesting investment strategies [2, 3] and conducting historical counterfactual analysis [4] to tracing the genuine progression of scientific ideas [5]. In each, value hinges not on what a model can retrieve, but on **what it refuses to use**. Yet, today's most capable language models [6, 7] routinely fail at this refusal [1]: ask the question above with the cutoff stated unambiguously, they volunteer post-2015 facts about Musk and X, and this failure does not diminish with scale. Stronger models memorize post-cutoff knowledge more thoroughly, and thus leak it more fluently (*cf.* Figure 2b) [1, 8].

*Corresponding authors.
[2]Our code is available at https://github.com/oceanoceanna/TCFT_2026.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

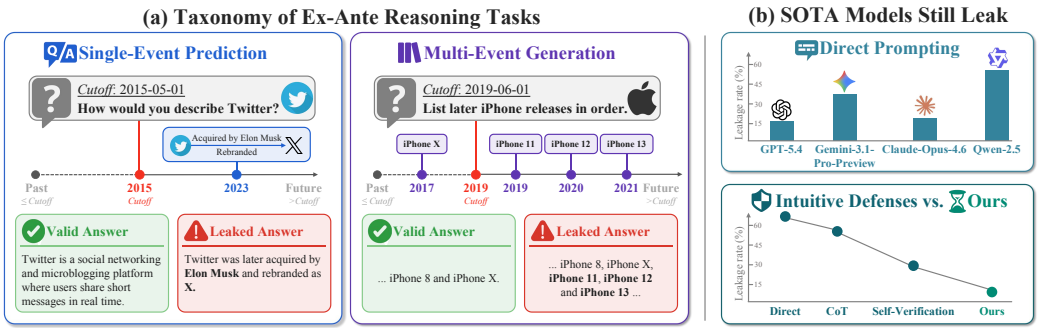


Figure 1: Overview of ex-ante reasoning tasks, temporal leakage in SOTA models, and the comparison between prompting-based defenses and our TCFT.

Confronted with such leakage, the natural first recourse is to push harder on the prompt: state the cutoff more forcefully, elicit chain-of-thought [9], or invoke self-verification [10]. We systematically investigate this family of interventions, treating their behavior as a diagnostic probe into how models handle temporal constraints. Two findings stand out, each difficult to reconcile with the hypothesis that models possess any stable notion of a cutoff:

- **Leakage tracks surface phrasing, not semantics.** Two logically equivalent cutoff specifications — an implicit historical framing (*"Assume today is May 1, 2009"*) and an explicit knowledge restriction (*"Use only information available before May 1, 2009"*) — yield markedly different leakage rates, with explicit phrasings consistently outperforming implicit ones.
- **Leakage tracks position, not proximity to the answer.** Placing the cutoff before the query (prefix) reduces leakage substantially more than placing it afterward (suffix), despite the latter's closer proximity to generation.

Taken together, these findings reveal an underlying weakness: prompt-based interventions modulate *how the model is steered into a temporal frame*, but they do not install a mechanism that asks, of each candidate fact, whether it is admissible under the given cutoff. **In essence, the model is being conditioned, not verified.** This diagnosis also explains why supervised fine-tuning offers no real escape: SFT is an imitation objective [11, 12], teaching the model to reproduce a target answer for a given prompt [13–16]. But ex-ante correctness is not a property of an answer in isolation. Instead, it is a relation between an answer and a cutoff. No volume of answer-imitation supervision can instill this contingency, because the supervision signal itself never asks the model to evaluate the relation.

What is missing, then, is supervision that operates on the relation rather than on the answer — training that teaches the model not what to say, but why a given response is or is not temporally admissible at a given time. We instantiate this principle as **TCFT** (Temporal Critique Fine-Tuning), a critique-based [11] training framework for ex-ante reasoning. Rather than fine-tuning the model to imitate temporally correct answers, TCFT trains it to acquire cutoff-aware temporal verification. Given a query, a cutoff, and a candidate response, the model learns to assess whether the response is temporally admissible, identify post-cutoff leakage, and explain boundary violations. In this way, TCFT directly supervises the missing relation between an answer and a cutoff, rather than treating ex-ante reasoning as a matter of prompt conditioning or answer imitation. We instantiate this supervision by using a teacher model to generate cutoff-conditioned critiques for candidate responses, where each critique highlights temporally invalid facts and provides an admissibility judgment [17, 18]. The student model is then fine-tuned on these critiques, acquiring the ability to verify whether a candidate response is valid under the specified cutoff. With this temporal verification capability, the model can reject temporally inadmissible facts before accepting an answer, thereby reducing the leakage of future knowledge. Experiments with Qwen2.5-7B-Instruct and Qwen2.5-14B-Instruct [19] demonstrate that TCFT consistently outperforms strong prompting and fine-tuning baselines, achieving better adherence to temporal constraints with leakage rate reductions of 41.89 and 37.79 percentage points, respectively.

2 Preliminary

To ground our empirical investigation and subsequent methodology, we first formalize the ex-ante reasoning problem, outline the task taxonomy, and establish rigorous evaluation metrics.

2.1 Task Definition

Given a query x and a specific cutoff timestamp t_c, ex-ante reasoning aims to produce a response $\hat{y}$ using only information verifiable before t_c. Let $\mathcal{A}(x)$ denote the universe of all factual knowledge associated with query x. We define a temporal mapping $\tau(f)$ that assigns each fact $f \in \mathcal{A}(x)$ the exact time it became verifiable in the real world. The temporal constraint t_c naturally partitions this knowledge space into two disjoint subsets:

$$R_{\text{prev}}(x, t_c) = \{f \in \mathcal{A}(x) \mid \tau(f) \leq t_c\} \tag{1}$$

$$R_{\text{post}}(x, t_c) = \{f \in \mathcal{A}(x) \mid \tau(f) > t_c\} \tag{2}$$

where $R_{\text{prev}}(x, t_c)$ represents the admissible historical context, and $R_{\text{post}}(x, t_c)$ denotes future facts that must remain inaccessible. For a model M parameterized by θ, the output is $\hat{y} = M(x, t_c)$. An oracle ideal response y^* is constructed entirely from $R_{\text{prev}}(x, t_c)$ and contains only the pre-cutoff information. Strict ex-ante validity therefore requires excluding all future knowledge: $F(\hat{y}) \cap R_{\text{post}}(x, t_c) = \varnothing$. Depending on the structural complexity of the required response, the ex-ante reasoning task can be categorized into the following two primary paradigms [1]:

- **Single-Event Prediction.** Single-event prediction asks the model to adopt the perspective of t_c and predict an outcome verifiable only afterward. This paradigm typically appears in time-sensitive QA or numerical forecasting. For an event whose true outcome a^* materializes at $t_e > t_c$, ex-ante reasoning requires a historically grounded prediction or abstention, rather than leaking the future ground truth.
- **Multi-Event Generation.** In open-ended generation, such as Wikipedia-style [20] summarization from a historical perspective, the response $\hat{y}$ is a complex text sequence rather than a single fact. For evaluation, we decompose it into independent atomic claims $F(\hat{y}) = \{f_1, f_2, \ldots, f_n\}$, each of which is verified against the cutoff t_c.

2.2 Evaluation Metrics

The core failure mode in ex-ante reasoning is *temporal leakage*: the unintended use of post-cutoff knowledge (R_{post}) under an explicit pre-cutoff restriction (R_{prev}). Leakage differs from *memorization*: the latter refers to the model's internalization of facts that became verifiable only after t_c, which is a prerequisite for leakage rather than a failure in itself. Without prior memorization of the relevant future facts, cutoff adherence is trivial; leakage evaluation is therefore meaningful only when such memorization is established. We formalize leakage by measuring whether the generated output intersects with the restricted future knowledge space.

For **single-event prediction**, leakage is a binary indicator of whether the model directly reveals the future outcome:

$$L_{\text{single}}(x) = \mathbf{1}[\hat{y} = a^* \wedge t_e > t_c]. \tag{3}$$

This captures direct disclosure of a fact that should be unknown at cutoff t_c. In the more complex setting of **multi-event generation**, we calculate a claim-level leakage rate, defined as the proportion of generated atomic facts that fall into the restricted future knowledge set:

$$L_{\text{multi}}(x) = \frac{1}{n} \sum_{i=1}^{n} \mathbf{1}[f_i \in R_{\text{post}}(x, t_c)]. \tag{4}$$

A query is flagged as leaked if any generated claim depends on post-cutoff information. A high leakage rate indicates a systemic failure of temporal reasoning: the model cannot reliably separate pre-cutoff knowledge from post-cutoff facts. Unlike forecasting [21, 22], ex-ante reasoning concerns the *admissibility* of information. Thus, even a factually "correct" future prediction is a failure if it relies on post-cutoff information.

Evaluating temporal adherence requires capturing both the violation of constraints and the utility of the generated text. Relying solely on constraint satisfaction is insufficient, as models might exploit degenerate strategies to bypass the rules. This joint criterion prevents temporal gains from coming at the cost of task utility. The validity evaluation metric is detailed in Appendix B.

3 Investigating the Brittleness of Prompt-based Control

The most intuitive approach to enforcing temporal boundaries in LLMs is through prompt engineering. However, prior evaluations suggest that simple instructions often fail to consistently eliminate tempo-

ral leakage [1]. To better understand the limitations of prompt-based interventions, we systematically investigate how instruction paradigms affect model adherence to temporal cutoffs.

3.1 Formalizing Temporal Instruction Paradigms

To systematically analyze the impact of prompting, we represent a temporal template T as a tuple $(t_c, e_T, \phi_T, \ell_T)$, capturing key dimensions of prompt-based temporal control. Here, t_c denotes the target cutoff date and $e_T \in \{\text{implicit}, \text{explicit}\}$ specifies boundary explicitness. The predicate $\phi_T(f_i; t_c) \in \{0, 1\}$ evaluates whether a fact f_i is admissible under template T and cutoff t_c. Finally, $\ell_T \in \{\text{prefix}, \text{suffix}\}$ indicates whether the temporal constraint is placed before or after the query.

Given this formalization, we investigate how variations in e_T and ℓ_T affect ex-ante reasoning performance, focusing on two dimensions: the explicitness of the temporal constraint and its placement within the prompt. These factors are hypothesized to influence how effectively a model can internalize and adhere to temporal boundaries, thereby affecting leakage rates and overall validity.

3.2 Insight 1: Leakage Tracks Surface Phrasing

The first critical finding concerns how the phrasing of temporal boundaries influences model behavior. In ex-ante reasoning, models are typically prompted into a temporal frame either implicitly through historical framing (*e.g.,* "Assume today is May 1, 2009") or explicitly through direct knowledge restriction (*e.g.,* "Use only information available before May 1, 2009").

Formally, given a candidate fact f_i, a template induces an admissibility predicate $\phi_T(f_i; t_c) \in \{0, 1\}$. A strong template explicitly verbalizes this predicate, ideally requiring $\forall f_i \in \mathcal{F}(y), \phi_T(f_i; t_c) = 1$, whereas a weak template only implicitly points to the same boundary. Specifically:

- **Explicit templates** specify the knowledge boundary, often through prohibitive language (*e.g., "Use only information available before t_c"* or *"Do not rely on facts after t_c"*). We find that these templates consistently reduce leakage across both single-event and multi-event tasks. By verbalizing $\phi_T(f_i; t_c) = \mathbf{1}[\tau(f_i) \leq t_c]$, explicit prompts give the model a clear surface rule to follow.
- **Implicit templates** instead frame the task through a historical persona or perspective (*e.g., "Assume today is t_c"* or *"Imagine you are answering on t_c"*). Although the temporal boundary is identical, models perform substantially worse under implicit constraints. This sensitivity suggests that current LLMs rely on surface linguistic cues rather than a stable, cutoff-aware reasoning mechanism.

As shown in Figure 2a, explicit templates consistently yield lower future-information leakage than implicit templates. This formulation sensitivity suggests that current models lack a stable understanding of temporal boundaries. If models had mastered cutoff-aware reasoning, surface-level phrasing of the same boundary should not substantially change performance [1]. Instead, models appear to be linguistically conditioned rather than temporally verified.

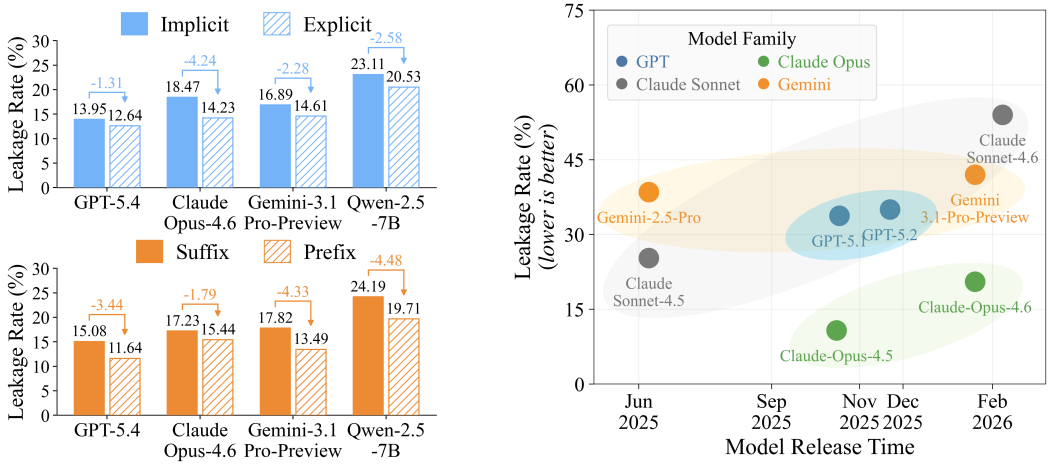
3.3 Insight 2: Leakage Tracks Prompt Position

The second finding involves the positional variation of constraints. Beyond formulation, the position ℓ_T of the temporal constraint also affects adherence: it can appear either as a prefix before the query or as a suffix after it. For instance, consider the following two templates for a query about Twitter:

- **Prefix**: "Use only information available before May 1, 2009. How would you describe Twitter?"
- **Suffix**: "How would you describe Twitter? Use only information available before May 1, 2009."

This positional variation is orthogonal to formulation explicitness, allowing us to independently assess how placement affects model behavior.

Empirically, we observe a clear **prefix advantage**: constraints placed at the beginning of the prompt yield markedly lower leakage than those appended as a suffix, as shown in Figure 2a. This finding is counterintuitive, since suffix constraints are closer to generation and might be expected to exert stronger control. Instead, the result aligns with known *positional biases* in in-context learning and long-context processing [23, 21], where information near the start of a prompt is often more stable. For ex-ante reasoning, the prefix advantage suggests that temporal boundaries function as global conditioning signals that are more easily ignored when treated as trailing instructions. Consequently, prompt engineering induces position-sensitive conditioning rather than robust temporal verification.



(a) Temporal template sensitivity.

(b) QA leakage of proprietary LLMs over release time.

Figure 2: Investigating the brittleness of prompt-based control and temporal leakage in proprietary LLMs on the QA task. Effect of temporal constraint formulation and placement on QA leakage (2a). Model capability does not guarantee cutoff awareness (2b).

4 Method

We address the core challenge identified in Section 3: ex-ante reasoning is fundamentally a temporal verification problem rather than a generation problem. Prompt-based control and conventional SFT fall short because they separate answer production from valid-at-cutoff judgment. Instead of imitating target answers, our approach trains models to critique candidate responses under temporal constraints. The model learns to assess whether a response is admissible at cutoff t_c, identify temporal violations, and explain its judgment. By making temporal validity a direct training target, this framework enables more robust cutoff-aware reasoning across diverse cutoff specifications.

4.1 Rethinking the Learning Objective: From Imitation to Verification

Standard supervised fine-tuning optimizes $\min_\theta \mathbb{E}_{(x,y^*)}\ell(M_\theta(x), y^*)$, where y^* is a target answer. This treats correctness as an answer-level property independent of context. In ex-ante reasoning, however, validity is cutoff-dependent: the validity of y depends on the temporal boundary t_c:

$$\text{Valid}(y \mid x, t_c) \neq \text{Valid}(y \mid x). \tag{5}$$

Thus, answer-matching supervision alone cannot teach a model when not to use memorized post-cutoff facts. Section 3 reveals that this limitation runs deeper than prompt phrasing.

The formulation and positional sensitivity of temporal constraints suggest that models lack an internalized procedure for judging temporal admissibility. Prompt engineering only steers surface behavior, without installing a verification mechanism. We therefore reframe training to directly supervise cutoff-conditioned temporal judgments, shifting the objective from matching target answers to determining whether a response is valid under the given cutoff.

4.2 Temporal Critique Task Definition

In our TCFT framework, we extend the Critique Fine-Tuning (CFT) [11] paradigm to the temporal setting. Classical CFT mainly targets reasoning verification: given a question and a potentially flawed solution, the model learns to critique the reasoning process and identify errors. In our setting, however, critique is intrinsically time-dependent, since the validity of a claim depends on the temporal boundary.

Given a query x, cutoff timestamp t_c, and candidate response y, the teacher model $\mathcal{C}_\Theta$ (*e.g.,* GPT-5.4 [24]) outputs a structured tuple:

$$\mathcal{C}_\Theta(x, t_c, y) = (r, v), \tag{6}$$

where v is a binary verdict indicating whether y is temporally admissible at cutoff t_c and r provides the supporting rationale.

This decomposition captures two coupled capabilities: **leakage detection**, which identifies claims in y that are not verifiable at t_c; and **temporal reasoning**, which explains why those claims violate the cutoff based on their verifiability timelines.

4.3 Temporal Critique Dataset Construction

To construct temporal critique supervision, we pair each query with multiple cutoffs, candidate responses, and corresponding temporal verdicts. The resulting dataset is structured as:

$$\mathcal{D}_{\text{tc}} = \{(x_i, t_{c,i}, y_i, c_i)\}_{i=1}^{|\mathcal{D}|}, \tag{7}$$

where each instance $(x_i, t_{c,i})$ is paired with a critique-conclusion target $c_i = (r_i, v_i)$. We augment the data with diverse prompt formulations to improve cutoff-aware judgment in a task-specific manner:

QA Augmentation. For each QA pair from the unaugmented ex-ante reasoning dataset [1], we generate variants along three dimensions: (1) **Cutoff variation**, where the cutoff date is set to 1, 14, and 30 days before the event, exposing the model to the same factual context under different temporal boundaries; (2) **Formulation variation**, where each cutoff is paired with explicit knowledge restriction ("Use only information available before t_c") and implicit historical framing ("Assume today is t_c"); and (3) **Position variation**, where each template is placed either before the query as a prefix or after it as a suffix. This results in $3 \times 2 \times 2 = 12$ distinct training instances per original pair.

Wikipedia Augmentation. For generation tasks, we focus on formulation and position variation: (1) cutoff dates are kept fixed, since Wikipedia-style prompts often involve historical entities or long-span events, making fine-grained cutoff shifts less well-defined; and (2) each example is expanded into $2 \times 2 = 4$ variants by crossing explicit/implicit formulation with prefix/suffix placement.

All splits (train, validation, and test) undergo the same augmentation, ensuring balanced exposure to strong and weak prompt formulations and enabling fair comparison across conditions. This construction supervises temporal validity relations, *i.e.,* query-cutoff pairs with valid/invalid labels, rather than a single target answer. By annotating cross-cutoff validity changes, the model learns that correctness is fundamentally contingent on t_c.

4.4 Learning via Temporal Critique Supervision

In practice, TCFT is implemented as a supervised fine-tuning process over the temporal critique tuples. Given a training instance (x, t_c, y, c) with $c = (r, v)$, we optimize the following negative log-likelihood objective:

$$\mathcal{L} = -\log p(c \mid x, t_c, y; \theta). \tag{8}$$

The joint probability can be factorized as:

$$p_\theta(c \mid x, t_c, y) = p_\theta(r \mid x, t_c, y) \cdot p_\theta(v \mid x, t_c, y, r), \tag{9}$$

where the model first generates a rationale r that examines the candidate response under the cutoff, and then summarizes this analysis into a verdict v. This order encourages the verdict to be grounded in an explicit temporal critique rather than produced as an isolated label. In implementation, the rationale and verdict are concatenated into a single critique-conclusion sequence and optimized via teacher forcing, yielding the full sequence likelihood.

Notably, this objective does not directly supervise final-answer generation (*e.g.,* predicting y from x and t_c). Instead, the model learns to critique candidate answers and summarize the temporal judgment. This inversion is deliberate: by decoupling generation from verification, the model can reuse its base knowledge while learning a conditional gate that flags leaked content.

4.5 Inference

At inference time, the model directly generates a response $\hat{y}$ from the query and cutoff (x, t_c). Although TCFT is trained on temporal critiques rather than final answers, it equips the model with cutoff-aware verification. By learning from temporally violated candidates and their critiques, the model internalizes which facts are admissible under a cutoff and which constitute leakage. Thus, inference requires neither explicit critique generation nor answer refinement; the learned verification capability implicitly guides $\hat{y}$ away from post-cutoff information.

Table 1: Evaluation results on QA and Wiki. For QA, 1d, 14d, and 30d denote cutoff gaps of 1, 14, and 30 days before the target event date, and Avg. denotes the average QA leakage rate. For Wiki, we report leakage rate and claim accuracy. Lower leakage indicates stronger cutoff adherence, and higher accuracy indicates better factual utility.

Models / Methods	QA				Wiki	
	1d ↓	14d ↓	30d ↓	Avg. ↓	Leakage Rate ↓	Accuracy ↑
GPT-5.4	49.60	13.11	7.19	22.25	13.28	81.37
Claude-Opus-4.6	48.00	11.48	5.23	20.50	16.86	81.30
Gemini-3.1-Pro-Preview	44.00	41.80	40.52	42.00	15.75	79.82
Qwen-2.5-7B-Instruct						
Zero-shot	80.80	76.23	73.86	76.75	21.82	61.67
Few-shot	20.80	26.23	18.95	21.75	26.76	59.96
Chain-of-thought	59.20	58.20	51.63	56.00	22.77	62.35
Self-verification	30.40	33.61	25.49	29.50	22.01	62.81
SFT	35.20	41.80	30.72	35.50	20.79	62.18
Ours	**1.60**	**0.00**	**0.65**	**0.75**	**14.04**	**66.24**
Qwen-2.5-14B-Instruct						
Zero-shot	81.44	65.66	62.28	69.80	23.34	69.49
Few-shot	23.35	20.48	13.77	19.20	30.93	66.68
Chain-of-thought	74.25	56.02	56.29	62.20	23.15	67.86
Self-verification	24.55	11.45	9.58	15.20	20.49	70.73
SFT	20.80	25.41	17.00	20.75	16.32	71.01
Ours	**5.99**	**6.02**	**5.39**	**5.80**	**11.76**	**73.81**

Formally, training optimizes the critique objective $p(c \mid x, t_c, y)$, while inference samples $\hat{y}$ from $p(\hat{y} \mid x, t_c)$. Although the two stages differ in form, both rely on the same learned capability: judging temporal admissibility under the cutoff. In this sense, TCFT does not teach the model to imitate a fixed target answer; instead, it teaches the model a temporal validity criterion that guides generation.

Accordingly, TCFT does not require an explicit answer-repair objective, *i.e.,* optimizing $p(y^* \mid x, t_c, y, c)$. Rather than learning to rewrite a flawed answer into a target response, the model learns to recognize temporal admissibility itself, in line with prior critique-based learning [11]. Introducing a separate refinement objective would add a competing learning signal and dilute the focus on temporal verification. Instead, the verdict v and rationale r provide direct supervision for temporal judgment, which can support downstream filtering or post-hoc correction when needed.

5 Experiment

This section provides a comprehensive experimental evaluation of TCFT, with the analysis structured around the following key research questions: **RQ1:** How does TCFT perform compared with state-of-the-art closed-source models and strong baseline methods? **RQ2:** How well does TCFT generalize across boundary explicitness e_T and constraint placement ℓ_T? **RQ3:** How do the temporal critique rationale r and verdict v contribute to learning cutoff-aware verification?

5.1 Experimental Setups

We use Qwen2.5-7B-Instruct and Qwen2.5-14B-Instruct as the LLM backbones, excluding smaller models whose limited memorization makes leakage evaluation ill-defined. We focus on standard instruction models to isolate whether TCFT itself can induce cutoff-aware temporal verification, without relying on reasoning-specialized post-training. We build on the original ex-ante reasoning benchmark, augmenting its single-event QA and Wikipedia-style generation tasks. For evaluation, we use leakage rate as the primary metric: for QA, we report leakage rates under 1-day, 14-day, and 30-day cutoffs, together with the average leakage rate; for Wikipedia generation, we report leakage rate and claim accuracy. We compare TCFT with closed-source models and open-source baselines, including Zero-shot, Few-shot, Chain-of-thought, Self-verification, and SFT. Comprehensive details on the benchmark, baselines, evaluation metrics, and TCFT implementation are provided in Appendix D.5.

5.2 Results Analysis of Ex-Ante Reasoning (RQ1)

We conduct experiments on two representative ex-ante inference tasks: QA and Wiki. Table 1 reports results for proprietary LLMs and open-weight Qwen models under standard prompting, prompt-level interventions, SFT, and our method. Lower leakage indicates stronger cutoff adherence, and higher Wiki accuracy indicates better factual utility. We draw the following observations:

- **Proprietary LLMs generally achieve stronger temporal adherence than open-weight baselines.** Among them, GPT-5.4 achieves the best overall performance, with 22.25% QA leakage, 13.28% Wiki leakage, and 81.37% Wiki accuracy. This indicates that stronger general-purpose models exhibit better implicit temporal alignment, but still cannot reliably enforce strict cutoff constraints.
- **Prompt-level interventions do not consistently improve temporal adherence.** For the QA task, prompt-level interventions generally reduce leakage, with few-shot and self-verification reducing leakage by 52.80 and 50.93 percentage points, respectively, while CoT yields a smaller reduction of 14.18 percentage points. In contrast, prompt-level methods fail to consistently reduce leakage on the Wiki task. This suggests that prompt-level interventions mainly impose surface-level behavioral constraints, rather than enabling models to acquire genuine cutoff awareness.
- **Shorter cutoff gaps lead to higher QA leakage.** QA leakage is highest under the 1-day cutoff gap and decreases as the gap extends to 14 and 30 days, with average leakage rates of 40.00%, 32.50%, and 27.90%, respectively. This observation is consistent with ExAnte [1], suggesting that LLMs struggle most with near-cutoff events due to weak fine-grained temporal boundary awareness.
- **TCFT achieves the strongest temporal leakage control.** Compared with zero-shot, SFT reduces average QA and Wiki leakage by 45.15 and 4.03 percentage points, while TCFT achieves larger reductions of 70.00 and 9.68 percentage points, respectively. Meanwhile, TCFT improves Wiki accuracy from 65.58% to 70.03%. This suggests that the key to ex-ante reasoning is learning cutoff-aware verification, rather than simply imitating target answers.

5.3 Generalization Across Temporal Templates (RQ2)

To investigate whether TCFT learns general temporal verification rather than template-specific patterns, we evaluate cross-template transfer on QA and Wiki. We vary templates along two dimensions: implicit vs. explicit temporal constraints and prefix vs. suffix cutoff placement. TCFT is trained on one template and tested on both the seen template and its counterpart, measuring template-specific performance and cross-template generalization. As shown in Figure 3, we can find that:

- **TCFT remains effective across different temporal templates.** Across implicit/explicit formulations and prefix/suffix cutoff placements, TCFT consistently achieves low leakage. Under the same template setting, QA leakage remains within 0.00–1.00%, and Wiki leakage ranges from 14.19% to 16.13%. These results suggest that TCFT captures the underlying cutoff-verification behavior across different surface formulations, rather than merely memorizing template-specific patterns.
- **TCFT remains effective under cross-template transfer.** When evaluated on unseen template variants, leakage increases only slightly and remains well controlled. Cross-template QA leakage stays low at 1.00–4.48%, while Wiki leakage remains within 16.13–19.01%. This suggests that TCFT acquires a transferable temporal verification capability rather than relying on prompt-specific surface patterns, enabling robust generalization across cutoff formulations.

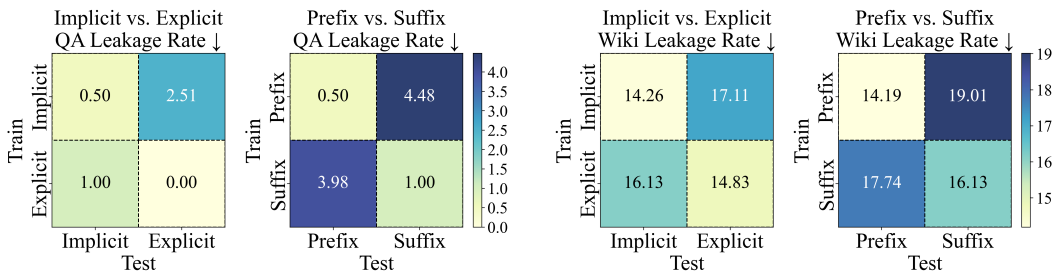
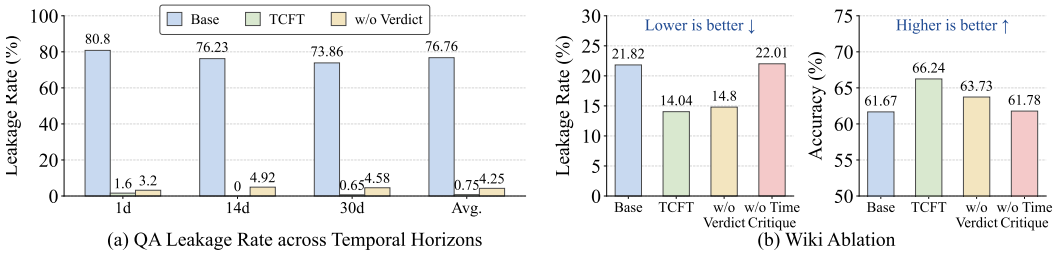


Figure 3: Cross-template generalization of TCFT on QA and Wiki with Qwen2.5-7B-Instruct. Rows denote the temporal template used for training, and columns denote the template used for testing.



(a) QA Leakage Rate across Temporal Horizons (b) Wiki Ablation

Figure 4: Ablation results of TCFT on QA and Wiki with Qwen2.5-7B-Instruct.

5.4 Ablation Study of TCFT (RQ3)

We conduct ablation experiments to analyze the effects of the verdict signal v and the rationale signal r in TCFT. For QA, we remove the verdict signal, yielding the *w/o Verdict* variant. For Wiki, we ablate both the verdict and rationale components. Since Wiki requires multi-fact generation, we replace the time-aware rationale with a generic hallucination critique, which checks factuality but does not verify whether each fact is supported before the cutoff. This *w/o Time Critique* variant isolates the contribution of cutoff-aware reasoning. Figure 4 presents the ablation results. We can observe that:

- **Temporal-validity rationales are essential for reliable ex-ante generation.** Replacing the time-aware critique with a generic factuality critique increases Wiki leakage from 14.04% to 22.01% and reduces accuracy from 66.24% to 61.78%, nearly reverting to the base model. This shows that factual correctness alone is insufficient for ex-ante generation, since true facts may still be invalid under the cutoff constraint. Cutoff-aware critique helps the model distinguish factual validity from temporal validity, enabling it to retain pre-cutoff facts while filtering out post-cutoff knowledge.
- **The verdict signal provides useful supervision for identifying cutoff violations.** Removing the verdict signal increases average QA leakage from 0.75% to 4.25%, with consistent degradation across all cutoff gaps. On Wiki, leakage also rises from 14.04% to 14.80%, while accuracy drops from 66.24% to 63.73%. These results show that the verdict signal helps models identify cutoff violations. However, the relatively mild degradation suggests that binary judgment alone provides limited supervision; robust ex-ante generation requires learning the underlying temporal verification rationale, *i.e.,* the ability to verify whether a fact is temporally valid under the cutoff constraint.

6 Conclusion

In this work, we study ex-ante reasoning as a temporal verification problem, where models must judge whether a response is admissible under a given cutoff. We show that prompt-level interventions provide brittle control, with leakage remaining sensitive to formulation and placement. To address this, we propose TCFT, a Temporal Critique Fine-Tuning framework that trains models to identify post-cutoff leakage, explain boundary violations, and judge temporal admissibility. Experiments on QA and Wiki show that TCFT reduces leakage while preserving factual utility. Ablation and cross-template studies further show that TCFT learns transferable cutoff-aware verification beyond prompt-specific patterns. Overall, reliable ex-ante reasoning requires verification rather than stronger prompting or answer imitation.

7 Limitations

While TCFT provides an effective framework for cutoff-aware verification, limitations remain. Our evaluation focuses on QA and Wiki-style generation, leaving broader time-sensitive scenarios such as finance, law, and scientific trend analysis unexplored. We use Qwen2.5-7B-Instruct and Qwen2.5-14B-Instruct, as many smaller open-weight models weakly memorize post-cutoff facts, making leakage evaluation and fine-tuning less informative. Extending TCFT to stronger backbones and reasoning-specialized models remains future work. TCFT also relies on teacher-generated temporal critiques, whose quality may vary across teacher models, critique styles, and evidence sources; developing more robust critique-generation strategies is another important direction.

Acknowledgements

This research is supported by the New Generation Artificial Intelligence-National Science and Technology Major Project (2025ZD0123305). This research is supported by the National Natural Science Foundation of China (62625608, 62572449, 62302321) and the Fundamental Research Funds for the Central Universities (WK2100250065). The work of Yancheng Yuan was supported by the NSFC Young Scientists Fund (Project No. 12501440). This research is also supported by the advanced computing resources provided by the Supercomputing Center of the USTC.

References

[1] Yachuan Liu, Xiaochun Wei, Lin Shi, Xinnuo Li, Bohan Zhang, Paramveer Dhillon, and Qiaozhu Mei. ExAnte: A benchmark for ex-ante inference in large language models. In *EACL (Volume 1: Long Papers)*, pages 1551–1571. Association for Computational Linguistics, 2026.

[2] Jean Lee, Nicholas Stevens, and Soyeon Caren Han. Large language models in finance (finllms). *Neural Comput. Appl.*, 37(30):24853–24867, 2025.

[3] Ties de Kok. Chatgpt for textual analysis? how to use generative llms in accounting research. *Manag. Sci.*, 71(9):7888–7906, 2025.

[4] Shuhang Lin, Wenyue Hua, Lingyao Li, Che-Jui Chang, Lizhou Fan, Jianchao Ji, Hang Hua, Mingyu Jin, Jiebo Luo, and Yongfeng Zhang. Battleagent: Multi-modal dynamic emulation on historical battles to complement historical analysis. In *EMNLP (System Demonstrations)*, pages 172–181. Association for Computational Linguistics, 2024.

[5] Jinheon Baek, Sujay Kumar Jauhar, Silviu Cucerzan, and Sung Ju Hwang. Researchagent: Iterative research idea generation over scientific literature with large language models. In *NAACL (Long Papers)*, pages 6709–6738. Association for Computational Linguistics, 2025.

[6] OpenAI. GPT-4 technical report. *CoRR*, abs/2303.08774, 2023.

[7] Gemini Team. Gemini: A family of highly capable multimodal models. *CoRR*, abs/2312.11805, 2023.

[8] Jeffrey Cheng, Marc Marone, Orion Weller, Dawn J. Lawrie, Daniel Khashabi, and Benjamin Van Durme. Dated data: Tracing knowledge cutoffs in large language models. *CoRR*, abs/2403.12958, 2024.

[9] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models. In *NeurIPS*, 2022.

[10] Ziwei Ji, Tiezheng Yu, Yan Xu, Nayeon Lee, Etsuko Ishii, and Pascale Fung. Towards mitigating LLM hallucination via self reflection. In *EMNLP (Findings)*, Findings of ACL, pages 1827–1843. Association for Computational Linguistics, 2023.

[11] Yubo Wang, Xiang Yue, and Wenhu Chen. Critique Fine-Tuning: Learning to critique is more effective than learning to imitate. *CoRR*, abs/2501.17703, 2025.

[12] Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. In *ICLR*. OpenReview.net, 2022.

[13] Victor Sanh, Albert Webson, Colin Raffel, Stephen H. Bach, Lintang Sutawika, Zaid Alyafeai, Antoine Chaffin, Arnaud Stiegler, Arun Raja, Manan Dey, M Saiful Bari, Canwen Xu, Urmish Thakker, Shanya Sharma Sharma, Eliza Szczechla, Taewoon Kim, Gunjan Chhablani, Nihal V. Nayak, Debajyoti Datta, Jonathan Chang, Mike Tian-Jian Jiang, Han Wang, Matteo Manica, Sheng Shen, Zheng Xin Yong, Harshit Pandey, Rachel Bawden, Thomas Wang, Trishala Neeraj, Jos Rozen, Abheesht Sharma, Andrea Santilli, Thibault Févry, Jason Alan Fries, Ryan Teehan, Teven Le Scao, Stella Biderman, Leo Gao, Thomas Wolf, and Alexander M. Rush. Multitask prompted training enables zero-shot task generalization. In *ICLR*. OpenReview.net, 2022.

[14] Jason Wei, Maarten Bosma, Vincent Y. Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M. Dai, and Quoc V. Le. Finetuned language models are zero-shot learners. In *ICLR*. OpenReview.net, 2022.

[15] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul F. Christiano, Jan Leike, and Ryan Lowe. Training language models to follow instructions with human feedback. In *NeurIPS*, 2022.

[16] Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A. Smith, Daniel Khashabi, and Hannaneh Hajishirzi. Self-instruct: Aligning language models with self-generated instructions. In *ACL (1)*, pages 13484–13508. Association for Computational Linguistics, 2023.

[17] Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. In *ICLR*. OpenReview.net, 2024.

[18] Jonathan Uesato, Nate Kushman, Ramana Kumar, H. Francis Song, Noah Y. Siegel, Lisa Wang, Antonia Creswell, Geoffrey Irving, and Irina Higgins. Solving math word problems with process- and outcome-based feedback. *CoRR*, abs/2211.14275, 2022.

[19] An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. Qwen2.5 technical report. *CoRR*, abs/2412.15115, 2024.

[20] Wikimedia Foundation. Wikipedia. https://www.wikipedia.org/, 2026. Accessed: 2026-05-07.

[21] Nate Gruver, Marc Finzi, Shikai Qiu, and Andrew G Wilson. Large language models are zero-shot time series forecasters. *Advances in neural information processing systems*, 36:19622–19635, 2023.

[22] Ming Jin, Shiyu Wang, Lintao Ma, Zhixuan Chu, James Y Zhang, Xiaoming Shi, Pin-Yu Chen, Yuxuan Liang, Yuan-Fang Li, Shirui Pan, et al. Time-llm: Time series forecasting by reprogramming large language models. *arXiv preprint arXiv:2310.01728*, 2023.

[23] Jie Zeng, Qianyu He, Qingyu Ren, Jiaqing Liang, Weikang Zhou, Zeye Sun, Fei Yu, and Yanghua Xiao. Order matters: Investigate the position bias in multi-constraint instruction following. In *ACL (Findings)*, volume ACL 2025 of *Findings of ACL*, pages 12479–12492. Association for Computational Linguistics, 2025.

[24] OpenAI. Introducing gpt-5.4. https://openai.com/index/introducing-gpt-5-4/, 2026. Accessed: 2026-05-07.

[25] Zhaochen Su, Juntao Li, Jun Zhang, Tong Zhu, Xiaoye Qu, Pan Zhou, Yan Bowen, Yu Cheng, and Min Zhang. Living in the moment: Can large language models grasp co-temporal reasoning? In *ACL (1)*, pages 13014–13033. Association for Computational Linguistics, 2024.

[26] Qingyu Tan, Hwee Tou Ng, and Lidong Bing. Towards benchmarking and improving the temporal reasoning capability of large language models. In *ACL (1)*, pages 14820–14835. Association for Computational Linguistics, 2023.

[27] Yuqing Wang and Yun Zhao. TRAM: benchmarking temporal reasoning for large language models. In *ACL (Findings)*, Findings of ACL, pages 6389–6415. Association for Computational Linguistics, 2024.

[28] Zheng Chu, Jingchang Chen, Qianglong Chen, Weijiang Yu, Haotian Wang, Ming Liu, and Bing Qin. TimeBench: A comprehensive evaluation of temporal reasoning abilities in large language models. In *ACL (1)*, pages 1204–1228. Association for Computational Linguistics, 2024.

[29] Bahare Fatemi, Mehran Kazemi, Anton Tsitsulin, Karishma Malkan, Jinyeong Yim, John Palowitch, Sungyong Seo, Jonathan Halcrow, and Bryan Perozzi. Test of Time: A benchmark for evaluating llms on temporal reasoning. In *ICLR*. OpenReview.net, 2025.

[30] Adam Liska, Tomás Kociský, Elena Gribovskaya, Tayfun Terzi, Eren Sezener, Devang Agrawal, Cyprien de Masson d'Autume, Tim Scholtes, Manzil Zaheer, Susannah Young, Ellen Gilsenan-McMahon, Sophia Austin, Phil Blunsom, and Angeliki Lazaridou. StreamingQA: A benchmark for adaptation to new knowledge over time in question answering models. In *ICML*, Proceedings of Machine Learning Research, pages 13604–13622. PMLR, 2022.

[31] Shahriar Golchin and Mihai Surdeanu. Time travel in llms: Tracing data contamination in large language models. In *ICLR*. OpenReview.net, 2024.

[32] Manley Roberts, Himanshu Thakur, Christine Herlihy, Colin White, and Samuel Dooley. To the cutoff... and beyond? A longitudinal perspective on LLM data contamination. In *ICLR*. OpenReview.net, 2024.

[33] Martin Pawelczyk, Seth Neel, and Himabindu Lakkaraju. In-context unlearning: Language models as few-shot unlearners. In *ICML*. OpenReview.net, 2024.

[34] Zheyuan Liu, Guangyao Dou, Zhaoxuan Tan, Yijun Tian, and Meng Jiang. Machine unlearning in generative AI: A survey. *CoRR*, abs/2407.20516, 2024.

[35] Chenlu Ding, Jiancan Wu, Yancheng Yuan, Jinda Lu, Kai Zhang, Alex Su, Xiang Wang, and Xiangnan He. Unified parameter-efficient unlearning for llms. In *ICLR*. OpenReview.net, 2025.

[36] Zheyuan Liu, Guangyao Dou, Zhaoxuan Tan, Yijun Tian, and Meng Jiang. Towards safer large language models through machine unlearning. In *ACL (Findings)*, Findings of ACL, pages 1817–1829. Association for Computational Linguistics, 2024.

[37] Anvith Thudi, Gabriel Deza, Varun Chandrasekaran, and Nicolas Papernot. Unrolling sgd: Understanding factors influencing machine unlearning. In *2022 IEEE 7th European Symposium on Security and Privacy (EuroS&P)*, pages 303–319. IEEE, 2022.

[38] Tianyu Zhang, Chenlu Ding, Junfeng Fang, Jiancan Wu, and Xiang Wang. Task-aligned unlearning for multimodal large language models. In *WWW (Companion Volume)*, pages 1386–1390. ACM, 2026.

[39] Ruiqi Zhang, Licong Lin, Yu Bai, and Song Mei. Negative preference optimization: From catastrophic collapse to effective unlearning. *CoRR*, abs/2404.05868, 2024.

[40] Chenlu Ding, Jiancan Wu, Leheng Sheng, Fan Zhang, Yancheng Yuan, Xiang Wang, and Xiangnan He. Mllmeraser: Achieving test-time unlearning in multimodal large language models through activation steering. *CoRR*, abs/2510.04217, 2025.

[41] Chenlu Ding, Jiancan Wu, Yanchen Luo, Tianyu Zhang, Yancheng Yuan, Xiang Wang, and Xiangnan He. Cuse: Fine-grained multimodal unlearning with counterfactual guidance and on-policy self-distillation. *arXiv preprint arXiv:2610.14763*, 2026.

[42] Suproteem K Sarkar and Keyon Vafa. Lookahead bias in pretrained language models. In *ICML 2025 Workshop on Reliable and Responsible Foundation Models*, 2024.

[43] Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, Shashank Gupta, Bodhisattwa Prasad Majumder, Katherine Hermann, Sean Welleck, Amir Yazdanbakhsh, and Peter Clark. Self-refine: Iterative refinement with self-feedback. In *NeurIPS*, 2023.

[44] Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: language agents with verbal reinforcement learning. In *NeurIPS*, 2023.

[45] Zhibin Gou, Zhihong Shao, Yeyun Gong, Yelong Shen, Yujiu Yang, Nan Duan, and Weizhu Chen. Critic: Large language models can self-correct with tool-interactive critiquing. *arXiv preprint arXiv:2305.11738*, 2023.

[46] Xin Xu, Tianhao Chen, Fan Zhang, Wanlong Liu, Pengxiang Li, Ajay Kumar Jaiswal, Yuchen Yan, Jishan Hu, Yang Wang, Hao Chen, et al. Double-checker: Enhancing reasoning of slow-thinking llms via self-critical fine-tuning. *arXiv preprint arXiv:2506.21285*, 2025.

[47] Karthik Valmeekam, Matthew Marquez, and Subbarao Kambhampati. Can large language models really improve by self-critiquing their own plans? *arXiv preprint arXiv:2310.08118*, 2023.

[48] Peiyi Wang, Lei Li, Zhihong Shao, Runxin Xu, Damai Dai, Yifei Li, Deli Chen, Yu Wu, and Zhifang Sui. Math-shepherd: Verify and reinforce llms step-by-step without human annotations. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 9426–9439, 2024.

[49] Lifan Yuan, Wendi Li, Huayu Chen, Ganqu Cui, Ning Ding, Kaiyan Zhang, Bowen Zhou, Zhiyuan Liu, and Hao Peng. Free process rewards without process labels. *arXiv preprint arXiv:2412.01981*, 2024.

[50] Zachary Ankner, Mansheej Paul, Brandon Cui, Jonathan D Chang, and Prithviraj Ammanabrolu. Critique-out-loud reward models. *arXiv preprint arXiv:2408.11791*, 2024.

[51] Junkang Wu, Xue Wang, Zhengyi Yang, Jiancan Wu, Jinyang Gao, Bolin Ding, Xiang Wang, and Xiangnan He. Alphadpo: Adaptive reward margin for direct preference optimization. In *ICML*, volume 267 of *Proceedings of Machine Learning Research*. PMLR / OpenReview.net, 2025.

[52] Junkang Wu, Yuexiang Xie, Zhengyi Yang, Jiancan Wu, Jiawei Chen, Jinyang Gao, Bolin Ding, Xiang Wang, and Xiangnan He. Towards robust alignment of language models: Distributionally robustifying direct preference optimization. In *ICLR*. OpenReview.net, 2025.

[53] Sewon Min, Kalpesh Krishna, Xinxi Lyu, Mike Lewis, Wen-tau Yih, Pang Koh, Mohit Iyyer, Luke Zettlemoyer, and Hannaneh Hajishirzi. Factscore: Fine-grained atomic evaluation of factual precision in long form text generation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 12076–12100, 2023.

[54] Anthropic. Introducing claude opus 4.6. https://www.anthropic.com/news/claude-opus-4-6, 2026. Accessed: 2026-05-07.

[55] Google. Gemini 3.1 pro preview. https://ai.google.dev/gemini-api/docs/models/gemini-3.1-pro-preview, 2026. Accessed: 2026-05-07.

[56] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

A Related Work

A.1 Temporal Reasoning

Evaluating LLMs on time-related dimensions has historically centered on temporal reasoning and time-sensitive question answering [25, 26]. Benchmarks such as TRAM [27], TimeBench [28], and Test of Time (ToT) [29] rigorously assess models on event ordering, durations, and temporal arithmetic [1, 8]. Similarly, time-sensitive QA datasets evaluate how models adapt to evolving knowledge [30]. However, these existing benchmarks generally assume full access to future context or focus on static facts, largely overlooking the specific requirement of cutoff-aware temporal restriction [31, 32].

To bridge this gap, the ExAnte benchmark [1] was introduced to systematically evaluate LLMs' ability to reason while strictly adhering to temporal cutoffs. Experimental results across multiple tasks

reveal that existing LLMs frequently violate these temporal constraints, with no single prompting strategy—including Zero-shot, Chain-of-Thought (CoT) [9], and Self-Verification [10]—consistently eliminating temporal leakage [1]. Beyond general failure, empirical observations indicate specific vulnerabilities: (1) **Cutoff Explicitness Effect**: Models exhibit a significant sensitivity to whether a cutoff is phrased through implicit or explicit templates, suggesting a reliance on surface-level linguistic cues rather than robust time-conditioned validity reasoning. (2) **Constraint Placement Sensitivity**: A distinct "prefix advantage" is observed, where temporal constraints at the beginning of a prompt yield more effective control than those at the end—a phenomenon that contradicts the intuition that proximity to the output should enhance constraint adherence. These insights suggest that failures in ex-ante reasoning are fundamentally temporal verification failures, indicating a lack of genuine temporal verification capability in current models. Since surface-level prompting proves insufficient to resolve these internal reasoning gaps, research has begun to explore more intrusive forms of knowledge control [33–35].

A.2 LLM Unlearning

As prompting strategies fail to provide a robust solution for temporal adherence, machine unlearning has emerged as a potential paradigm to suppress or remove specific knowledge within model weights [34, 36, 35]. Subsequent research has proposed various frameworks, such as gradient-ascent fine-tuning [37, 38], preference-based unlearning [39], and in-context unlearning [33], to selectively erase the influence of specific data points [40, 41].

However, ex-ante reasoning presents a challenge fundamentally distinct from traditional unlearning. While unlearning focuses on the permanent removal of data for privacy or safety, ex-ante inference requires a dynamic, context-dependent suppression of knowledge based on an arbitrary temporal cutoff [1]. A critical obstacle is the "memorization-leakage" paradox: evidence suggests that stronger recall and memorization of post-cutoff events are often positively correlated with higher leakage rates, as models struggle to decouple internalized memories from temporal constraints [1, 42]. Consequently, ex-ante reasoning demands temporary knowledge suppression and active admissibility judgment—capabilities that traditional unlearning methods, which target permanent erasure, are not designed to achieve [1].

A.3 Critique Learning

To overcome the limitations of standard prompting and Supervised Fine-Tuning (SFT) [13, 16, 15, 14], recent research has turned toward self-correction and critique learning [43–45]. While self-correction aims to refine outputs, studies show that LLMs typically struggle to identify complex flaws without specialized training [46]. Critique Fine-Tuning (CFT) was proposed to shift the training paradigm from simple imitation to active verification, training models to critique noisy responses and identify errors [11, 47].

In the context of ex-ante reasoning, standard SFT is inadequate because its objective is static imitation; yet, in ex-ante scenarios, the validity of a response is strictly conditioned on the specified temporal cutoff. This necessitates an extension of the critique paradigm to the temporal domain. By training models to evaluate the temporal admissibility of responses and identify specific leakage violations, it is possible to move beyond superficial prompt adherence. Such an approach aims to instill an intrinsic cutoff-aware verification capability, enabling models to judge whether a response remains valid under shifting temporal constraints [17, 18, 48–50].

Beyond critique-based supervision, preference optimization provides another direction for improving the alignment of language models. Recent studies investigate adaptive preference margins and distributionally robust optimization to improve the robustness of preference learning [51, 52]. However, these approaches primarily focus on preference alignment, whereas our work explicitly supervises cutoff-conditioned temporal validity.

B Validity Evaluation Metric

Another core dimension of evaluating ex-ante reasoning is guarding against degenerate low-leakage strategies [1]. For instance, a naive model could trivially achieve zero leakage by refusing to answer

(*e.g., "I cannot provide the related information"*) or by generating vacuous, low-quality text, which while technically adhering to the cutoff, offers no functional utility.

To distinguish genuine reasoning from mere abstention, a quality guardrail $Q(\hat{y}, y^*)$ is introduced, where Q is a task-specific similarity or factual precision function evaluated against the ideal pre-cutoff response [1]. The instantiation of Q is task-dependent to reflect diverse output structures [1]. For numerical single-event tasks, the quality function can be instantiated with task-specific regression metrics such as negative mean absolute error [1]. For QA, we do not evaluate validity separately, since the task asks for the exact date of an unexpected future event, and providing the correct date is itself temporal leakage. Thus, there is no meaningful pre-cutoff target answer for quality evaluation; rewarding exact correctness would reward leakage, while penalizing uncertainty or abstention would be inappropriate. We therefore evaluate QA using leakage rate only.

In the context of multi-event generation, Q is operationalized as factual precision, representing the proportion of generated atomic claims that are verifiable within the R_{prev} knowledge space [1, 53]. Formally, a response is *valid* only if it satisfies both the temporal boundary and a minimum utility threshold γ [1]:

$$\text{Valid}(x) = \mathbf{1}[L(x) = 0] \wedge \mathbf{1}[Q(\hat{y}, y^*) \geq \gamma]. \tag{10}$$

C Discussion

C.1 Relation to Supervised Fine-Tuning

Formally, standard SFT [16, 15, 14, 13] optimizes the following objective:

$$\min_{\theta} \mathbb{E}_{(x,t_c,y^*)} - \log p(y^* \mid x, t_c; \theta),$$

where y^* denotes the oracle response under cutoff t_c. This objective treats (x, t_c) as the input context and y^* as the target output, thereby learning a direct mapping from prompt to answer. However, because y^* is constructed to be temporally admissible by design, SFT only exposes the model to positive answers and provides no supervision on what future leakage looks like or why it violates the cutoff. As a result, it remains an answer-generation objective rather than a temporal verification objective.

TCFT directly addresses this limitation. Instead of training only on admissible target answers, TCFT pairs diverse candidate responses, including both admissible and leaked ones, with explicit temporal judgments. The model therefore learns to evaluate a response against the cutoff, distinguishing temporally valid content from post-cutoff leakage, rather than merely memorizing a mapping from context to answer.

C.2 Relation to Conventional Critique Fine-Tuning

Template CFT [11] has shown promise in teaching models to identify errors in problem-solving chains and self-correct. Its key contribution is to shift learning from output imitation to error identification. TCFT extends this inversion to the temporal setting, where the object of critique is not only whether a response is factually correct, but whether it is admissible under a given cutoff.

In standard CFT, the critique of a reasoning error is largely context-independent: whether a solution step is flawed depends on the problem and the step itself. In temporal critique, however, admissibility is cutoff-dependent. For example, the statement "Elon Musk acquired Twitter" is factually correct, but temporally inadmissible if the cutoff precedes 2022. Likewise, the same response y may be valid when $t_c = 2023$ but invalid when $t_c = 2020$. Thus, the critique function must be conditioned on the cutoff:

$$c^{(t_c)} = \mathcal{C}_\theta(x, t_c, y) \neq \mathcal{C}_\theta(x, y). \tag{11}$$

This makes the learned critique capability inherently temporally parameterized. Without conditioning on t_c, a CFT-style critique would assign the same judgment across different cutoffs, failing to capture the temporal validity relation central to ex-ante reasoning.

Table 2: Claim-level evaluation for Wiki generation.

Pre-cutoff support	Post-cutoff support	Temporal leakage	Claim accuracy
$f_i \in R_{\text{pre}}$	$f_i \notin R_{\text{post}}$	No	Yes
$f_i \notin R_{\text{pre}}$	$f_i \in R_{\text{post}}$	Yes	No
$f_i \notin R_{\text{pre}}$	$f_i \notin R_{\text{post}}$	No	No
$f_i \in R_{\text{prev}}$	$f_i \in R_{\text{post}}$	No	Yes

D Experimental Setups

D.1 Datasets

We build our experiments on the ExAnte benchmark [1], focusing on two representative ex-ante inference tasks: single-event QA and Wikipedia-style generation. These two tasks cover different levels of response complexity: QA requires the model to decide whether it can answer a future-event question under a given cutoff, while Wiki requires the model to generate multiple factual claims that must all remain valid before the cutoff. To improve generalization, we augment both datasets along several dimensions. Specifically, we vary whether constraints are expressed implicitly or explicitly, diversify constraint positions in the prompt, and introduce different cutoff-date settings. We further mix multiple prompt templates during data construction, so the model is exposed to semantically similar tasks with varied surface forms rather than overfitting to a single prompt pattern. The QA dataset contains 9,072 examples in total, split into 8:1:1 for train/validation/test. The Wiki dataset contains 5,268 examples in total, split into 8:1:1 for train/validation/test.

QA. The QA task evaluates single-event temporal leakage. Each example contains an event query, the true event date, and a cutoff date preceding the event. Following our main experiments, we evaluate three cutoff gaps: 1 day, 14 days, and 30 days before the target event date. A response is counted as leaked if the model reveals the correct post-cutoff event information under the specified cutoff. For TCFT training, each QA example is augmented along three dimensions: cutoff gap, temporal formulation, and constraint placement. Specifically, we use three cutoff gaps, two formulations (implicit historical framing and explicit knowledge restriction), and two placements (prefix and suffix), yielding $3 \times 2 \times 2 = 12$ template variants per original QA example.

Wiki. The Wiki task evaluates multi-event generation under temporal cutoffs. Given a topic and a cutoff year, the model is asked to generate factual claims using only information available before the cutoff. Since Wiki-style prompts often involve long-span entities or events, we keep the cutoff fixed for each example and augment the prompt along two dimensions: temporal formulation and constraint placement. This yields $2 \times 2 = 4$ template variants per original Wiki example. During evaluation, generated responses are decomposed into atomic claims, and each claim is checked for whether it is supported before the cutoff or leaks post-cutoff information.

D.2 Evaluation Metrics

We use leakage rate as the primary metric for both QA and Wiki. Lower leakage indicates stronger cutoff adherence.

QA leakage. For QA, leakage is measured under 1-day, 14-day, and 30-day cutoff gaps. A response is considered leaked if it provides the correct post-cutoff event information despite being instructed to answer under the cutoff. We report leakage rates for each cutoff gap and their average.

Wiki leakage and accuracy. For Wiki, we report leakage rate and claim accuracy. Leakage rate measures the proportion of generated claims that rely on post-cutoff information. Claim accuracy measures the proportion of generated claims that are factually supported by pre-cutoff information. This metric is included to prevent degenerate low-leakage strategies, such as refusing to answer or producing overly vague content. Detailed definitions of leakage rate and claim accuracy are provided in Table 2.

D.3 Baselines

We compare TCFT with proprietary LLMs, prompt-level interventions, and a parameter-level fine-tuning baseline.

Proprietary LLMs. We evaluate GPT-5.4 [24], Claude-Opus-4.6 [54], and Gemini-3.1-Pro-Preview [55] under the same evaluation protocol. These models serve as strong proprietary references for assessing whether general model capability alone leads to reliable cutoff adherence.

Zero-shot. The model is directly prompted with the task query and the cutoff constraint, without additional demonstrations or verification steps.

Few-shot [56]. The model is provided with demonstration examples that illustrate how to answer under temporal cutoffs. The demonstrations are designed to encourage the model to avoid post-cutoff information while preserving useful pre-cutoff content.

Chain-of-thought [9]. The model is prompted to reason step by step before answering. This baseline tests whether eliciting intermediate reasoning improves temporal cutoff adherence.

Self-verification [10] . The model first generates an initial response and then performs an additional verification step to check whether the response contains post-cutoff information. If leakage is detected, the model is instructed to revise the response.

SFT. We train a supervised fine-tuning baseline on temporally admissible target answers generated by GPT-5.4 [24] for each query under the corresponding cutoff. This baseline uses the same backbone models and augmented prompt formats as TCFT, but optimizes final-answer imitation rather than temporal critique [15, 14, 13]. It is included to test whether answer-level supervision alone is sufficient for learning ex-ante reasoning.

D.4 Prompt Templates

We use two types of temporal formulations and two constraint placements. We present several representative template examples here, with the complete set shown in Table 3. In practice, training incorporates diverse temporal templates to enhance generalization across different cutoff formulations.

Implicit temporal formulation. The cutoff is expressed through a historical framing:

> Assume today is `<cutoff date>`. `<query>`

Explicit temporal formulation. The cutoff is expressed as a direct knowledge restriction:

> Use only information available before `<cutoff date>`. `<query>`

Prefix placement. The temporal constraint is placed before the query:

> Use only information available before `<cutoff date>`. `<query>`

Suffix placement. The temporal constraint is placed after the query:

> `<query>` Use only information available before `<cutoff date>`.

D.5 Implementation Details

We use Qwen2.5-7B-Instruct and Qwen2.5-14B-Instruct [19] as the open-weight backbones. We exclude smaller models because their limited memorization of post-cutoff facts makes leakage evaluation less informative. All models are evaluated on the same QA and Wiki test sets.

Table 3: Representative explicit and implicit temporal templates for the QA task, where {Q} denotes the input query.

ID	Explicit template	Implicit template
01	**Constrain your response to pre-t_c information only. Any information beyond t_c must be excluded.** {Q} You should provide answer follow the format of YYYY-MM-DD. If you don't know, please answer I don't know.	**Assume today is t_c.** {Q} You should provide answer follow the format of YYYY-MM-DD. If you don't know, please answer I don't know.
02	**Base your response entirely on information available by t_c, without using any later information.** {Q} You should provide answer follow the format of YYYY-MM-DD. If you don't know, please answer I don't know.	**Suppose you are answering this question on t_c.** {Q} You should provide answer follow the format of YYYY-MM-DD. If you don't know, please answer I don't know.
03	**Your response must rely only on information available on or before t_c. Any later information must not be used.** {Q} You should provide answer follow the format of YYYY-MM-DD. If you don't know, please answer I don't know.	**Write your answer as it would be given on t_c.** {Q} You should provide answer follow the format of YYYY-MM-DD. If you don't know, please answer I don't know.
04	**Restrict your answer to information available on or before t_c only. Do not include any facts that were not available by t_c.** {Q} You should provide answer follow the format of YYYY-MM-DD. If you don't know, please answer I don't know.	**Provide your answer as of t_c.** {Q} You should provide answer follow the format of YYYY-MM-DD. If you don't know, please answer I don't know.
05	**Your answer must be based exclusively on information available by t_c. Do not include any facts from after t_c.** {Q} You should provide answer follow the format of YYYY-MM-DD. If you don't know, please answer I don't know.	**Frame your answer as of t_c.** {Q} You should provide answer follow the format of YYYY-MM-DD. If you don't know, please answer I don't know.

Temporal critique generation. For TCFT, we construct critique supervision using a teacher model (*i.e.,* GPT-5.4 [24]). Given a query, cutoff, and candidate response, the teacher generates a temporal-validity rationale and a binary verdict. The rationale explains which parts of the response are valid or invalid under the cutoff, while the verdict summarizes whether the response contains post-cutoff leakage.

Training. TCFT is trained using standard supervised fine-tuning over critique-conclusion sequences. The input consists of the query, cutoff, and candidate response, while the target output consists of the temporal rationale and verdict. We optimize the model with the standard next-token prediction objective. For fair comparison, SFT uses the same backbone and prompt augmentation but is trained to imitate final target answers rather than critiques.

Inference. At inference time, TCFT directly generates the final response from the query and cutoff. It does not require explicit critique generation or answer repair during evaluation. The learned temporal verification capability is expected to guide generation away from post-cutoff information.

Hyperparameters. Unless otherwise stated, we use the same training configuration for SFT and TCFT. The learning rate is set to `1e-5`, warmup ratio to `0.03`. The effective batch size is 64,

implemented with a per-device batch size of 16, gradient accumulation steps of 2, and 2 GPUs. We train for 3 epochs with a maximum sequence length of 1024. Training uses `bfloat16` precision and gradient checkpointing. All experiments use a fixed random seed of 42. For LoRA, we use rank 16, LoRA alpha 32, and LoRA dropout 0.05, and apply adapters to `q_proj`, `k_proj`, `v_proj`, `o_proj`, `gate_proj`, `up_proj`, and `down_proj`. Experiments are run on a server with NVIDIA A100 GPUs (80 GB memory per GPU).

E Prompt Templates

E.1 Temporal Critique Prompt for QA

System Prompt

You are an expert critic for ex-ante temporal reasoning.
You will receive an event, its true date, a cutoff date, the original prompt, and a model answer. Your job is to critique the answer under the cutoff constraint.
Rules:

- If the true event date is later than the cutoff date, then the event had not happened yet at the cutoff. Any answer that gives a specific future date or otherwise relies on post-cutoff knowledge is wrong.
- If the true event date is on or before the cutoff date, then the date was already knowable at the cutoff. In this case, the answer is right only if it gives the correct date in the required format. If it is wrong, explain the factual mistake. In this case, it is allowed to mention the correct date in the critique.
- Focus first on temporal validity, then on factual correctness, then on format.
- Keep the critique concise, natural, and useful for training.
- End with exactly one of: "Conclusion: right" or "Conclusion: wrong".

User Prompt

Event: {event}
True event date: {true_date}
Cutoff date: {ask_date}
Original prompt: {leakage_prompt}
Model answer: {leakage_answer_raw}
Write a concise critique of the model answer under the cutoff date.

E.2 Temporal Critique Prompt for Wiki

System Prompt

You are an expert critic for ex-ante temporal reasoning on Wikipedia-style factual claims.
You will receive a topic, a cutoff date, the original prompt, and a model answer containing several atomic claims. Your job is to critique whether the claims are valid under the cutoff constraint.
Rules:

- Evaluate each claim according to what could reasonably be known on or before the cutoff date.
- A claim is wrong if it relies on information from after the cutoff date.
- For future events, distinguish scheduled or planned information from completed-event information:
  - It is acceptable to say that an event was scheduled, planned, announced, selected, or expected before the cutoff if that was knowable then.
  - It is not acceptable to state a future event as already having happened, completed, concluded, won, opened, released, died, etc., if that happened after the cutoff.
- If any claim is temporally invalid, the overall answer is wrong.
- If all claims are temporally valid and factually plausible under the cutoff, the overall answer is right.
- Keep the critique concise, natural, and useful for training.
- End with exactly one of: "Conclusion: right" or "Conclusion: wrong".

User Prompt

Topic: {title}
Cutoff date: {cutoff_date}
Original prompt: {prompt}
Model answer: {raw_response}
Claims: {claims}
Write a concise critique of the model answer under the cutoff date.

E.3 Example of Wiki Temporal Critique Data

Input

Original prompt:
Regarding your knowledge about "2020_Summer_Olympics", generate 5 atomic facts. Each fact should be a short, clear statement representing a distinct piece of information. Do not guess or fill in missing details. Format each fact as a numbered list (1–5). Frame your answer as of January 1st, 2020.

Model answer:
1. The 2020 Summer Olympics were scheduled to take place in Tokyo, Japan.
2. The opening ceremony of the 2020 Summer Olympics was planned for July 24, 2020.
3. The 2020 Summer Olympics were originally set to include 33 sports across 339 events.
4. The Olympic Village for the 2020 Summer Olympics was located in Musashino Forest Park.
5. The torch relay for the 2020 Summer Olympics began on March 26, 2020.

Critique

1–3 are fine as of Jan. 1, 2020: Tokyo was the host city, the opening ceremony was scheduled for July 24, 2020, and the program was planned for 33 sports and 339 events.

4 is wrong factually. Musashino Forest Park was a venue, not the Olympic Village. The Olympic Village was in the Harumi waterfront area of Tokyo.

5 is not valid under the prompt's time framing. As of Jan. 1, 2020, one could say the torch relay was scheduled/planned to begin on March 26, 2020, but saying it "began" states a future event as already occurred.

Conclusion: wrong

E.4 Example of QA Temporal Critique Data

Input

Original prompt:
Your response must rely only on information available on or before 2018-06-10. Any later information must not be used. Now please answer the following question. Just leave the best answer without explanation. Question: On what date Twelve boys and their football coach are rescued from the flooded Tham Luang Nang Non cave in Thailand? You should provide answer follow the format of YYYY-MM-DD. If you don't know, please answer I don't know.

Model answer:
2018-07-10

Critique

The answer uses post-cutoff information: the rescue had not happened yet by 2018-06-10, so the exact rescue date was not knowable at the cutoff. It should have abstained and answered "I don't know" instead of giving a future date.

Conclusion: wrong

E.5 Wiki Critique Prompt w/o temporal critique

System Prompt

You are an expert critic for Wikipedia-style factual claims.
You will receive a title, an original prompt, and a model answer containing several atomic claims. Your job is to critique whether the model answer is factually correct.
Rules:

- Evaluate each claim according to general factual correctness.
- A claim is wrong if it is factually incorrect, unsupported, or overly specific in an unsupported way.
- Prefer concise, natural critiques useful for training.
- End with exactly one of: "Conclusion: right" or "Conclusion: wrong".

User Prompt

Title: {title}
Cutoff year: {cutoff_year}
Original prompt:
{prompt}
Model answer:
{model_answer}
Write a concise factual critique of the model answer.

E.6 Wiki Critique Prompt w/o verdict

System Prompt

You are an expert critic for ex-ante temporal reasoning on Wikipedia-style factual claims.
You will receive a topic, a cutoff date, the original prompt, and a model answer containing several atomic claims. Your job is to critique whether the claims are valid under the cutoff constraint.
Rules:

- Evaluate each claim according to what could reasonably be known on or before the cutoff date.
- A claim is wrong if it relies on information from after the cutoff date.
- For future events, distinguish scheduled or planned information from completed-event information:
  - It is acceptable to say that an event was scheduled, planned, announced, selected, or expected before the cutoff if that was knowable then.
  - It is not acceptable to state a future event as already having happened, completed, concluded, won, opened, released, died, etc., if that happened after the cutoff.
- If any claim is temporally invalid, the overall answer is wrong.
- If all claims are temporally valid and factually plausible under the cutoff, the overall answer is right.
- Keep the critique concise, natural, and useful for training.

F Case Study

User Prompt

Topic: {title}
Cutoff date: {cutoff_date}
Original prompt: {prompt}
Model answer: {raw_response}
Claims: {claims}
Write a concise critique of the model answer under the cutoff date.

Prompt

Assume today is January 1st, 2011. Regarding your knowledge about "Emma_Watson", generate 5 atomic facts. Each fact should be a short, clear statement representing a distinct piece of information. Do not guess or fill in missing details. Format each fact as a numbered list (1–5).

Zero-shot

1. Emma Watson was born on April 15, 1990.
2. Emma Watson played the character Hermione Granger in the Harry Potter film series.
3. Emma Watson is an English actress and UN Women Goodwill Ambassador *(appointed in Jul. 2014)*.
4. Emma Watson graduated from Brown University in 2014 *(after the cutoff)*.
5. Emma Watson has blonde hair and blue eyes.

Few-shot

1. Emma Watson was born on April 15, 1990.
2. She played the role of Hermione Granger in the Harry Potter film series.
3. Emma Watson was appointed as a UN Women Goodwill Ambassador in 2014 *(after the cutoff)*.
4. She graduated from Brown University in 2017 with a degree in English literature and French studies *(post-cutoff; actual graduation was in 2014)*.
5. As of January 1st, 2011, she was known for her roles in the Harry Potter films and had not yet become a UN Women Goodwill Ambassador.

Chain-of-thought

1. Emma Watson was born on April 15, 1990.
2. Emma Watson played the character Hermione Granger in the Harry Potter film series.
3. Emma Watson is an English actress and UN Women Goodwill Ambassador *(appointed in Jul. 2014)*.
4. Emma Watson graduated from Brown University in 2014 *(after the cutoff)*.
5. Emma Watson has blonde hair and blue eyes.

Self-verification

1. Emma Watson was born on April 15, 1990.
2. On January 1, 2011, Emma Watson was starring in the film "The Perks of Being a Wallflower."
3. As of January 1, 2011, Emma Watson had completed her role as Hermione Granger in the Harry Potter film series.
4. Emma Watson was on an academic leave from Brown University *(announced after the cutoff)* as of January 1, 2011.
5. By January 1, 2011, Emma Watson had been involved in various UN Women campaigns promoting gender equality *(UN Women involvement began after the cutoff)*.

Ours (TCFT)

1. Emma Watson was born on April 15, 1990.
2. She was known for her role as Hermione Granger in the Harry Potter film series.
3. On January 1, 2011, she was 20 years old.
4. She had appeared in the film "Harry Potter and the Deathly Hallows – Part 1" before this date.
5. She was a British actress.

NeurIPS Paper Checklist

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 abstract and introduction state the main problem of temporal leakage in ex-ante reasoning, analyze the brittleness of prompt-level control, and introduce TCFT as a temporal critique fine-tuning framework. The claimed improvements are supported by experiments on QA and Wiki tasks.

   Guidelines:

   - The answer [N/A] 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 [N/A] 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: The paper includes a limitations discussion, noting that the evaluation focuses on QA and Wiki-style generation, that many smaller open-weight models have limited memorization of post-cutoff facts, and that TCFT depends on teacher-generated temporal critiques.

   Guidelines:

   - The answer [N/A] 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: [N/A].

   Justification: The paper does not present formal theoretical results or theorems requiring proofs. The mathematical formulations are used to define ex-ante reasoning, leakage, and the TCFT objective.

   Guidelines:

   - The answer [N/A] 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: The paper describes the evaluated tasks, model backbones, baselines, metrics, data augmentation strategy, and training objective. Additional implementation details are provided in the appendix, and the code is released in an anonymized repository.

   Guidelines:

   - The answer [N/A] 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: The paper uses the public ExAnte benchmark and provides the implementation in an anonymized repository. The repository includes instructions for data preprocessing, training, and evaluation.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (https://neurips.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://neurips.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) necessary to understand the results?

   Answer: [Yes].

   Justification: The paper specifies the evaluated tasks, model backbones, cutoff settings, baselines, evaluation metrics, and data augmentation protocol. Additional training and implementation details are provided in the appendix.

   Guidelines:

   - The answer [N/A] 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: The main experiments report leakage rates and accuracy scores, but do not include error bars, confidence intervals, or statistical significance tests. The evaluation is based on fixed benchmark splits and deterministic aggregate metrics.

Guidelines:

- The answer [N/A] 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: The appendix reports the GPU type, training time, and total compute required for the main experiments.

   Guidelines:

   - The answer [N/A] 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: The work studies temporal leakage and cutoff-aware verification in LLMs using public benchmark data and standard model evaluations. It does not involve deception, human subjects, private data collection, or harmful deployment.

   Guidelines:

   - The answer [N/A] 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: The work can improve the reliability of LLMs in time-sensitive settings such as historical analysis, financial backtesting, and scientific trend analysis. A potential negative use is that models may be trained to conceal the use of future knowledge; the paper frames TCFT as a verification method for reducing leakage rather than enabling misuse.

Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] 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., pre-trained language models, image generators, or scraped datasets)?

Answer: [N/A].

Justification: The paper does not release a high-risk pretrained model, image generator, or unsafe scraped dataset. The released assets are code and temporal critique data for evaluating and improving cutoff adherence.

Guidelines:

- The answer [N/A] 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 paper cites the existing ExAnte benchmark, Qwen model backbones, and proprietary LLM APIs used in evaluation or critique generation. The corresponding licenses and terms of use are documented in the appendix.

Guidelines:

- The answer [N/A] 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 paper introduces TCFT training data derived from temporal critique generation and releases the code for constructing and evaluating this data. Documentation for the data format, training pipeline, and evaluation scripts is provided in the repository.

Guidelines:

- The answer [N/A] 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: [N/A].

Justification: The paper does not involve crowdsourcing or research with human subjects. Temporal critiques are generated by teacher models and evaluated on existing benchmark data.

Guidelines:

- The answer [N/A] 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: [N/A].

    Justification: The paper does not involve human subjects, user studies, or crowdsourced annotation, so IRB approval is not applicable.

    Guidelines:

    - The answer [N/A] 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 rigor, or originality of the research, declaration is not required.

    Answer: [Yes].

    Justification: The paper uses proprietary LLMs for evaluation and teacher-generated temporal critiques, and fine-tunes Qwen2.5-7B-Instruct and Qwen2.5-14B-Instruct as student backbones. LLMs were also used for language polishing.

    Guidelines:

    - The answer [N/A] 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 in the NeurIPS handbook for what should or should not be described.