M-TRACE: Detecting and Mitigating Time-Anchor Drift via Step-wise Conflict Checking in Temporal Reasoning

Danyu Huang[1], Jiayuan Jiang[2], Yao Zhang[3]*, Jun Wang[4], Huijia Li[3], Zhenglu Yang[1]*
[1]Key Laboratory of DISSec, College of Computer Science, Nankai University, Tianjin, China
[2]School of Statistics and Data Science, Nankai University, China
[3]School of Statistics and Data Science, LPMC, KLMDASR & AAIS, Nankai University, China
[4]College of Mathematics and Statistics Science, Ludong University
danyuhuang@mail.nankai.edu.cn, yj_taylor@163.com, yaozhang@nankai.edu.cn,
junwang@mail.nankai.edu.cn, hjli@nankai.edu.cn, yangzl@nankai.edu.cn

Abstract

As the real world continuously evolves, temporal facts change over time, requiring large language models to simultaneously rely on internal parametric knowledge and externally retrieved evidence for temporal reasoning. However, external knowledge may be inaccurate, while internal knowledge can become outdated. Temporal inconsistencies between these heterogeneous sources can accumulate during multi-step reasoning, leading to Time-Anchor Drift (TAD)—a phenomenon where an incorrect temporal reference is established early and subsequently propagated, ultimately causing reasoning failure. To address this issue, we propose M-TRACE, a multi-agent reasoning framework for temporal knowledge conflicts. M-TRACE explicitly maintains a State Timeline to perform step-wise temporal alignment and coexistence checks between internal states and external evidence. Detected conflicts are summarized into a structured Conflict Report, which guides conflict-aware final reasoning. We further introduce TimeConfQA, a temporal question answering benchmark with controlled temporal knowledge conflicts. Experimental results show that M-TRACE effectively reduces time-anchor drift and consistently improves performance on complex temporal question answering tasks, demonstrating the value of explicit conflict modeling for temporal reasoning. The code can be found at https://github.com/h-yii/M-TRACE.

1 Introduction

As the real world continuously evolves, many facts change over time, making temporal reasoning an increasingly important research direction (Jia et al., 2018; Leblay and Chekol, 2018; Dasgupta et al., 2018; Abbasiantaeb et al., 2024; Mavromatis et al., 2022; Shang et al., 2022; Li et al., 2023; Liu et al., 2023; Sharma et al., 2023; Sun et al., 2024). Knowledge involved in temporal reasoning is inherently dynamic, while the internal parametric knowledge of large language models (LLMs) is constrained by their training time and inevitably becomes outdated (Wang et al., 2025; Cheng et al., 2025; Xie et al., 2023). As a result, in practical temporal reasoning scenarios, models often rely on external knowledge to supplement temporal information about the current state (Yao et al., 2023; Zhao et al., 2024; Wang et al., 2024; Abbasiantaeb et al., 2024; Chen et al., 2023a). However, retrieved external knowledge may also be inaccurate, leading to frequent temporal inconsistencies or even direct conflicts between internal and external knowledge sources (Shen et al., 2024; Wang et al., 2023; Zhang et al., 2025). Such temporal knowledge conflicts are almost unavoidable in complex, multi-step temporal reasoning tasks.

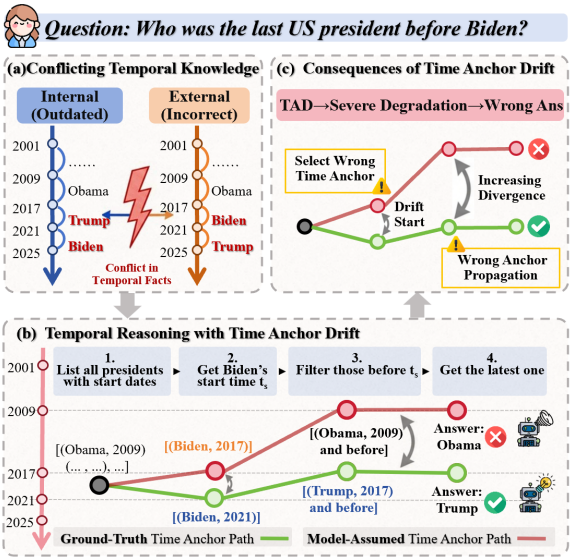

Figure 1: An illustrative example of Time Anchor Drift (TAD) caused by conflicting temporal knowledge in multi-step reasoning.

When such conflicts arise, LLMs often struggle to determine which knowledge source to rely on, leading them to mix internal and external infor-

*Corresponding author.

Findings of the Association for Computational Linguistics: ACL 2026, pages 24059–24077

mation during reasoning. This process can cause temporal anchors to be incorrectly established or gradually shifted. Because temporal reasoning relies on multi-step inference, early anchoring errors are amplified in subsequent steps, causing the entire reasoning chain to evolve along an incorrect timeline. We refer to this phenomenon as *Time Anchor Drift (TAD)*, which constitutes a major source of failure in complex temporal reasoning, as illustrated in Figure 1.

Existing temporal reasoning methods typically assume temporal consistency across knowledge sources (Zhang et al., 2025; Gao et al., 2024), while most studies on knowledge conflict are grounded in static fact assumptions and treat conflicts as discrepancies between sources (Chen et al., 2023a; Wang et al., 2023). In contrast, time anchor drift arises from the temporal evolution of knowledge, is nearly unavoidable in complex reasoning, and induces progressively amplified negative effects on subsequent inference. To facilitate systematic investigation of this phenomenon, we construct **TimeConfQA**, a question answering dataset for temporal knowledge conflict, which explicitly introduces temporal inconsistencies between internal and external knowledge through controlled perturbations of temporal information in existing temporal knowledge.

To systematically detect and mitigate time anchor drift, we propose **M-TRACE**, a **M**ulti-**A**gent **T**emporal **R**easoning with **C**onflicting Knowledge. The framework comprises three agents with clearly defined and strictly separated responsibilities. Specifically, a problem decomposition module first decomposes the original complex question into a sequence of sub-actions. In parallel, an internal state agent relies exclusively on the LLM's Internal parametric knowledge to construct a State Timeline that is independent of any external information, serving as a prior temporal specification of state evolution. Subsequently, an external execution module generates candidate states for each sub-action solely based on the provided external knowledge. Finally, a conflict checker agent examines, at each reasoning step, whether the externally generated states conflict with the internal State Timeline under the corresponding temporal anchor.

Experimental results demonstrate that on the constructed TimeConfQA dataset, M-TRACE significantly reduces time anchor drift under conflicting temporal knowledge, thereby substantially improving reasoning accuracy.

Our main contributions are summarized as follows:

- We identify and analyze **Time Anchor Drift (TAD)** arising from conflicts between internal parametric knowledge and external retrieved knowledge in temporal reasoning.
- We construct **TimeConfQA**, a temporal knowledge conflict question answering dataset that explicitly induces temporal inconsistencies through controlled time perturbations.
- We propose **M-TRACE**, a multi-agent temporal reasoning framework built upon **State Timelines**, which achieves substantial improvements over strong baselines on **TimeConfQA** and significantly enhances reasoning accuracy under conflicting temporal knowledge.

2 Related Work

Temporal Reasoning. Temporal reasoning studies how knowledge evolves over time. Recent work (Sun et al., 2024; Gao et al., 2024) leverages the parametric knowledge of large language models (LLMs) for temporal reasoning, and further augments it with retrieved external knowledge via RAG (Yao et al., 2023; Zhao et al., 2024; Wang et al., 2024; Abbasiantaeb et al., 2024; Chen et al., 2023a). These methods generally assume temporal consistency across knowledge sources. As a result, the impact of temporal knowledge conflicts, especially in multi-step reasoning, remains largely underexplored.

Knowledge Conflict. Knowledge conflict is typically defined as inconsistencies between a model's internal parametric knowledge and externally retrieved evidence (Shen et al., 2024; Wang et al., 2023; Zhang et al., 2025). With the widespread adoption of RAG, such conflicts have been shown to induce reasoning instability and answer inconsistency, motivating prior studies on conflict-aware reasoning and evidence selection (Xinjie et al., 2025; Huo et al., 2025). However, most existing work assumes static knowledge, treating conflicts as atemporal factual discrepancies rather than temporal inconsistencies induced by knowledge evolution. Consequently, these methods struggle to address conflicts arising from outdated knowledge, incorrect temporal anchoring, or other dynamics inherent to time-sensitive reasoning.

Temporal Knowledge Conflict. DYNAMICQA (Marjanović et al., 2024) approximates knowledge temporality via Wikipedia revision frequency but neither explicitly perturbs temporal information nor supports multi-hop temporal reasoning. ConflictBank (Su et al., 2024) introduces time-related conflicts, yet relies mainly on uniform future-time shifts, failing to capture diverse temporal inconsistency patterns. Meanwhile, recent studies show that LLM knowledge becomes outdated over time (Liska et al., 2022; Tang et al., 2025; Lin et al., 2025), inducing conflicts between parametric memory and external evidence (Su et al., 2024; Liu et al., 2024). However, these works largely analyze temporal conflict as an observed phenomenon, rather than formalizing it as a temporal knowledge conflict question answering task requiring explicit reasoning and decision-making.

3 The TimeConfQA Dataset

3.1 Data Sources, Question Types, and Temporal Modeling

Temporal Knowledge Source. We build **TimeConfQA** on a temporal knowledge graph (TKG) derived from Wikidata (Lacroix et al., 2020), which has been widely used in prior temporal reasoning benchmarks (Saxena et al., 2021; Chen et al., 2023b). Each temporal fact is represented as a quintuple $(e_s, r, e_o, t_{\text{start}}, t_{\text{end}})$, explicitly encoding the validity interval of the relation. This representation enables precise modeling of temporal ordering and state transitions, which is essential for studying temporal reasoning under knowledge conflict.

Question Types. Following CronQuestions (Saxena et al., 2021) and MultiTQ (Chen et al., 2023b), we categorize questions into six types: *simple_time*, *simple_entity*, *before*, *after*, *before_last*, and *after_first*. These types range from single-step temporal localization to multi-step inference requiring reasoning over temporal boundaries, allowing systematic evaluation across different reasoning complexities.

Temporal Relations and State Modeling. To enable controlled temporal perturbation, we select four relations with explicit temporal semantics and clear state transitions: *head of government*, *position held*, *member of sports team*, and *spouse*. These relations represent common time-bounded roles and affiliations and exhibit diverse temporal continuity and overlap patterns. Based on them, we abstract six temporal states to characterize relation evolution over time, enabling systematic temporal manipulation while preserving relational semantics.

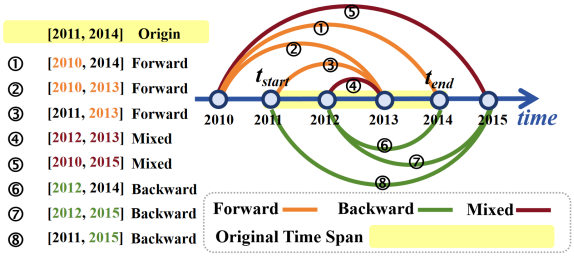

Figure 2: Illustration of temporal perturbation types under a unified time-axis representation. Yellow highlighted regions denote the original temporal boundaries t_{start} and t_{end}. **Forward** shifts perturbations toward the past, **Backward** shifts them toward the future, while **Mixed** perturbs the start and end times in opposite directions, distorting the temporal span.

3.2 Temporal Knowledge Perturbation Strategies

To explicitly induce temporal knowledge conflicts, **TimeConfQA** applies perturbations at two granularities: global timeline-level perturbation and question-level local perturbation.

Global Timeline Perturbation. At the global level, we randomly permute the temporal intervals $(t_{\text{start}}, t_{\text{end}})$ of all facts within each temporal state. This preserves the internal validity of individual facts while disrupting their relative temporal ordering, simulating temporal misalignment or outdated knowledge in external sources.

Question-Level Local Perturbation. At the question level, we randomly select N facts from each temporal state and independently perturb their start and end times by adding or subtracting n years, subject to basic constraints (e.g., $t_{\text{start}} < t_{\text{end}}$). Based on perturbation directions, we define three types: *Forward* (shifts toward the past), *Backward* (shifts toward the future), and *Mixed* (opposite shifts of start and end times). Figure 2 illustrates these perturbation types.

3.3 Dataset Construction

After perturbation, we obtain knowledge that explicitly conflicts with the original temporal facts. We then generate natural language questions using adapted templates. For each selected fact, all perturbation types are combined with all question

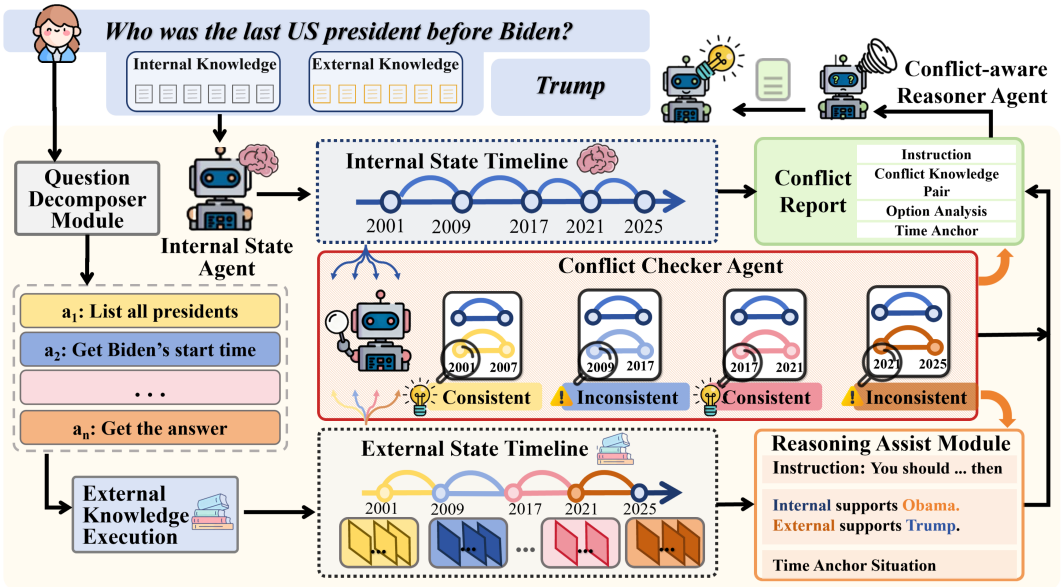

Figure 3: Overview of the M-TRACE framework.

types, yielding 60 question–answer instances per fact. Each question includes a *Correct Answer* from the original knowledge, a conflict-induced *Wrong Answer* from perturbed knowledge, additional distractors, and a unified "I don't know" option. This design supports fine-grained evaluation of robustness under explicit temporal knowledge conflict. Dataset statistics and examples are provided in Appendix.

4 The M-TRACE Framework

We propose M-TRACE (**M**ulti-**A**gent **T**emporal **R**easoning with **C**onflicting Knowledge), a multi-agent framework designed for temporal reasoning under knowledge conflicts. Instead of assuming any single knowledge source to be reliable, M-TRACE explicitly monitors temporal consistency across conflicting knowledge through the coordinated interaction of three specialized agents: the **Internal State Agent**, the **Conflict Checker Agent**, and the **Conflict-aware Reasoner Agent**.

During reasoning, externally candidate states produced by the External Knowledge Execution are continuously aligned with and compared against an internally constructed *State Timeline* (ST), which serves as a temporal reference. As illustrated in Figure 5, this timeline-centric comparison mechanism constitutes the core of M-TRACE, enabling explicit detection of temporal conflicts throughout multi-step reasoning.

4.1 Internal State Timeline Construction

The **Internal State Agent** initiates the reasoning process by constructing a complete State Timeline (ST) using only the LLM's parametric knowledge, without accessing any external information. Formally, we represent the State Timeline as an ordered set of time-anchored states:

$$\mathcal{T} = \{s_1^{\text{int}}, s_2^{\text{int}}, \ldots, s_m^{\text{int}}\}, \tag{1}$$

where each internal state s_i^{int} is associated with a temporal interval

$$s_i^{\text{int}} = \langle e, r, v, [t_i^{\text{start}}, t_i^{\text{end}}] \rangle, \tag{2}$$

encoding the model's prior belief that entity e holds relation r with value v during the corresponding time span.

The ST represents the model's prior belief about the temporal evolution of relevant world states. Importantly, this internal timeline is not assumed to be correct and is not directly used to generate answers; rather, it serves as an internal temporal norm against which externally retrieved evidence will be evaluated in subsequent stages.

4.2 Question Decomposition into Temporal Actions

The **Question Decomposer Module** decomposes a complex question Q into an ordered sequence of temporal sub-actions:

$$Q \rightarrow \mathcal{A} = \{a_1, a_2, \ldots, a_n\}. \tag{3}$$

Each sub-action corresponds to a localized state query or state transition anchored to a specific temporal reference. This decomposition makes the

implicit temporal dependency structure of the reasoning process explicit, enabling fine-grained, step-wise auditing and conflict detection.

4.3 External Knowledge Execution

For each decomposed sub-action a_k, **External Knowledge Execution** is performed to produce candidate states using only the provided external knowledge source. The execution result is a candidate external state:

$$s_k^{\text{ext}} = \langle e_k, r_k, v_k, [\hat{t}_k^{\text{start}}, \hat{t}_k^{\text{end}}] \rangle. \quad (4)$$

External evidence is treated as informative but potentially unreliable, and is therefore not directly accepted as ground truth. Execution is strictly constrained to a single-step, local, and memoryless setting, preventing the model from implicitly assuming temporal consistency across different reasoning steps.

4.4 Step-wise Temporal Conflict Checking

After each sub-action is executed, the **Conflict Checker Agent** evaluates whether the externally generated state s_k^{ext} can co-exist with the corresponding internal state $s_i^{\text{int}} \in \mathcal{T}$ aligned to the same temporal anchor. We formalize temporal compatibility as a binary predicate:

$$\mathcal{C}(s_i^{\text{int}}, s_k^{\text{ext}}) \in \{0, 1\}, \quad (5)$$

where $\mathcal{C} = 1$ indicates temporal coexistence and $\mathcal{C} = 0$ denotes a conflict.

The predicate $\mathcal{C}$ evaluates relation-specific temporal constraints, including uniqueness, continuity, and temporal overlap. Any detected conflict is explicitly recorded together with the conflicting knowledge items and its potential to induce time-anchor drift.

4.5 Conflict Report and Conflict-aware Reasoning

Once all sub-actions have been processed, the **Conflict-aware Reasoner Agent** consolidates the recorded conflicts into a structured **Conflict Report**:

$$\mathcal{R} = \{(s_i^{\text{int}}, s_k^{\text{ext}}, a_k) \mid \mathcal{C}(s_i^{\text{int}}, s_k^{\text{ext}}) = 0\}. \quad (6)$$

The Conflict Report explicitly records the detected *conflicting knowledge pairs* produced by the Conflict Checker Agent. On top of these raw conflict annotations, a dedicated **Reasoning Assist Module** augments the report with decision-oriented reasoning elements, including structured instructions, an analysis of candidate answers supported by each side of the conflicting knowledge, and an explicit characterization of the corresponding time-anchor situations.

Conditioned on the enriched Conflict Report $\mathcal{R}$, the Conflict-aware Reasoner Agent produces the final answer:

$$\hat{y} = \mathcal{A}_{\text{CAR}}(Q, \mathcal{R}), \quad (7)$$

where $\mathcal{A}_{\text{CAR}}(\cdot)$ denotes the decision process implemented by the Conflict-aware Reasoner Agent, which explicitly accounts for temporal inconsistencies rather than indiscriminately aggregating intermediate results. Detailed prompting strategies are provided in Appendix.

5 Experiments

5.1 Experimental Setup

5.1.1 Baselines

We compare against two categories of baselines: *single-pass reasoning* and *structured reasoning frameworks*. The former includes *End to End*, *few-shot prompting*, and *Chain-of-Thought* (CoT) (Wei et al., 2022), as well as reasoning-augmented methods such as Self-Ask (Press et al., 2023), GKP (Liu et al., 2022), and Comparative (Wang et al., 2023). We further evaluate multi-agent frameworks designed for knowledge inconsistency, including ReAgent (Xinjie et al., 2025) and Micro-Act (Huo et al., 2025).

5.1.2 Metrics

We evaluate all methods using:

- **Accuracy**: Following prior work on knowledge conflict (Xie et al., 2023; Su et al., 2024; Wang et al., 2023; Shi et al., 2024), we use QA accuracy as the primary metric.
- **TAD Rate**: the proportion of cases where the model anchors on conflicting temporal evidence and selects the corresponding wrong answer.

5.1.3 Implementation

All components of M-TRACE and all baseline methods are implemented using **GPT-4o**.

5.2 Main Results

Table 1 reports overall performance on TimeConfQA. M-TRACE achieves the highest accuracy

Category	Models	Total		Perturbation Type				Question Type		Answer Type	
		Correct	TAD	Backward	Forward	Mixed	Timeline	Simple	Complex	Entity	Time
Generic Reasoning	End to End	0.628	0.211	0.644	0.584	0.922	0.590	0.447	0.851	0.790	0.408
	CoT	0.618	0.200	0.615	0.596	0.905	0.581	0.396	0.836	0.762	0.362
	Few-shot (3-shot)	0.657	0.201	0.684	0.607	0.922	0.620	0.477	0.874	0.816	0.435
Generation-aided Reasoning	Self-Ask	0.348	0.431	0.423	0.312	0.856	0.183	0.144	0.579	0.496	0.127
	Comparative	0.359	0.379	0.429	0.285	0.877	0.241	0.170	0.572	0.498	0.149
	GKP w/ CoT	0.598	0.266	0.594	0.577	0.914	0.552	0.440	0.779	0.712	0.429
	GKP w/o CoT	0.701	0.193	0.689	0.622	0.914	0.654	0.519	0.823	0.769	0.501
Multi-agent Reasoning	ReAgent	0.567	0.332	0.557	0.594	0.888	0.460	0.403	0.729	0.661	0.397
	Micro-Act	0.730	0.184	0.749	0.735	0.931	0.652	0.651	0.820	0.761	0.683
	M-TRACE (ours)	**0.826**	**0.056**	**0.846**	**0.829**	**0.940**	**0.761**	**0.772**	**0.887**	**0.840**	**0.803**

Table 1: Overall performance on the TimeConfQA benchmark. **Correct** denotes answer accuracy, and **TAD** denotes the Time-Anchor Drift rate. **Bold** indicates the best result and underlined indicates the second-best result in each column.

(82.83%) and the lowest TAD rate (9.09%), consistently outperforming all baselines. These results indicate that M-TRACE improves correctness while effectively mitigating time-anchor drift.

More reasoning does not always help. Several reasoning-intensive baselines (e.g., CoT, Self-Ask, Comparative) underperform simpler End to End and few-shot methods, exhibiting lower accuracy, higher TAD rates, and more frequent *IDK* responses. This suggests that increased reasoning verbosity alone does not improve temporal robustness; without explicit decomposition or consistency constraints, longer reasoning chains may amplify the influence of conflicting knowledge.

Advantages of multi-agent reasoning. Multi-step, multi-agent frameworks such as ReAgent and Micro-Act achieve more stable performance by explicitly decomposing generation, alignment, and verification. By further emphasizing temporal state modeling and conflict tracing, M-TRACE outperforms other multi-agent baselines.

Why mixed perturbations are easier. *Mixed* perturbations yield higher accuracy and are easier to resolve. Unlike pure forward or backward shifts—which preserve semantic plausibility—mixed perturbations simultaneously disrupt temporal ordering and local consistency, making conflicts more salient at the semantic level. This indicates that purely temporal perturbations pose a subtler and more challenging setting.

After-first and before-last questions. *After-first* and *before-last* questions consistently achieve higher accuracy than standard before/after queries, as they rely on identifying boundary states over the global timeline and are less sensitive to local temporal noise.

Models	Correct	TAD
M-TRACE	**0.826**	**0.056**
M-TRACE (w/o Conflict)	0.781	0.092
M-TRACE (w/o Temporal)	0.813	0.102
M-TRACE (w/o Temporal + Conflict)	0.749	0.118

Table 2: Ablation study of M-TRACE on the TimeConfQA benchmark. **Correct** denotes overall answer accuracy, and **TAD** denotes the Time-Anchor Drift rate.

Simplicity becomes a weakness under conflict. Under temporal conflict, *simple_time* questions become the most challenging category. Their reliance on a single temporal anchor makes them particularly vulnerable to perturbation. M-TRACE yields substantial gains on this category, highlighting the importance of explicit temporal conflict modeling.

5.3 Ablation Studies

To analyze the contribution of each core component in M-TRACE, we conduct ablation studies by removing (1) explicit conflict modeling via the *Conflict Report*, (2) temporal-aware reasoning mechanisms, and (3) both components. All experiments follow the same dataset and evaluation protocol as the main results.

Removing Conflict Report. M-TRACE (w/o Conflict Report) removes the Conflict Report input to the final Reasoning Agent, forcing decisions to rely solely on internal reasoning and retrieved knowledge. As shown in Table 2, accuracy drops from 0.826 to 0.781, while the TAD rate increases from 0.056 to 0.092, indicating that the absence of explicit conflict signals substantially exacerbates time-anchor misalignment.

Backbones	Correct	TAD
GPT-4o	**0.826**	**0.056**
GPT-4o-mini	0.688	0.180
LLaMA-3 Sonar	0.641	0.263
Qwen2.5-32B	0.629	0.130

Table 3: Performance comparison of M-TRACE instantiated with different LLM backbones. **Correct** denotes overall answer accuracy, and **TAD** denotes the Time-Anchor Drift rate.

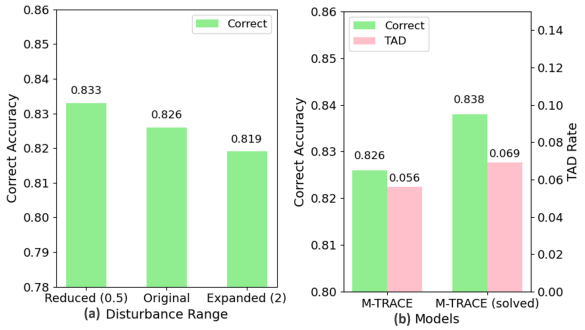

Figure 4: (a) Effect of perturbation strength on M-TRACE (**Reduced/Original/Expanded** = 0.5×/1×/2× perturbation). (b) Effect of conflict resolution. **Correct**: accuracy; **TAD**: Time-Anchor Drift rate.

Removing Temporal Modeling. M-TRACE (w/o Temporal) removes all temporal-specific mechanisms, including State Timeline construction and drift detection. This results in an accuracy of 0.813 and a higher TAD rate of 0.102, with notable degradation on time-sensitive question types, suggesting that implicit temporal reasoning alone is insufficient under conflicting knowledge.

Removing Both Components. M-TRACE (w/o Conflict Report & Temporal) removes both conflict modeling and temporal constraints, reducing the framework to a generic reasoning process. Accuracy further drops to 0.749 and the TAD rate rises to 0.118, demonstrating that explicit conflict awareness and temporal anchoring are jointly critical for stable temporal reasoning.

Different Backbone LLMs. We evaluate the robustness of M-TRACE across different backbone LLMs by replacing GPT-4o with GPT-4o-mini, LLaMA-3-Sonar, and Qwen2.5-32B, while keeping all other settings unchanged. Results show that GPT-4o achieves the best performance (Acc. 0.826, TAD 0.056), whereas GPT-4o-mini and the two open-source models suffer notable accuracy drops (down to 0.629) and substantially higher TAD rates (up to 0.263), particularly on *time*-type answers. This indicates that although M-TRACE exhibits consistent performance trends across models, its effectiveness is constrained by the backbone LLM's temporal understanding and multi-step reasoning capability.

Model	Overall Accuracy	API Calls
GKP (w cot)	0.598	2.00
Micro-act	0.730	9.83
M-TRACE	0.826	11.46

Table 4: Efficiency comparison in terms of overall accuracy and average number of API calls per query.

5.4 Effect of Perturbation Magnitude

During dataset construction, we introduce moderate temporal perturbations adaptively based on the temporal span of the State Timeline (ST). To assess robustness under different perturbation strengths, we further scale the perturbation magnitude to 0.5× and 2× while preserving the perturbation structure.

As shown in Figure 4-(a), smaller perturbations (0.5×) slightly outperform the original setting, as they tend to preserve global temporal order. In contrast, larger perturbations (2×) severely disrupt local timeline structure, substantially increasing temporal reasoning difficulty.

5.5 Can Conflicts Be Resolved?

To investigate whether identified conflicts can be resolved, we use GPT-5 to adjudicate conflicts in the Conflict Report by replacing conflicting entries with temporally correct facts for final reasoning. As shown in Figure 4-(b), this procedure slightly improves accuracy from 0.826 to 0.838, but also increases the TAD rate from 0.056 to 0.069.

Further analysis shows that the proportion of *IDK* responses decreases (from 0.072 to 0.064), indicating reduced conservatism. However, this shift does not lead to more stable temporal reasoning; instead, it increases susceptibility to time-anchor drift. These results suggest that explicitly resolving conflicts is less effective than preserving conflicts and reasoning under explicit temporal constraints.

5.6 Efficiency Analysis

Beyond accuracy improvements, inference efficiency is a critical factor in assessing the practicality of complex reasoning frameworks. We therefore evaluate efficiency from two aspects: overall accuracy and the average number of API calls, as summarized in Table 4.

As shown in the table, increasing the number of API calls—corresponding to more sophisticated

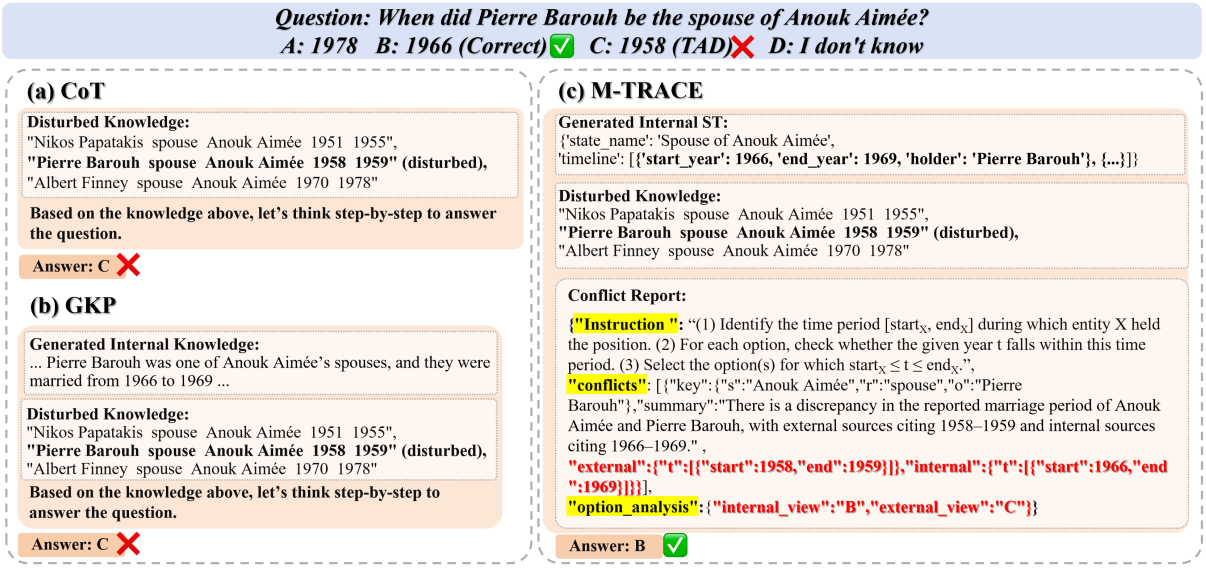

Figure 5: Case study on temporal anchor drift (TAD) under conflict knowledge. Both CoT and GKP are misled by incorrect temporal facts and select a wrong answer, while M-TRACE identifies temporal conflicts via a Conflict Report, and successfully resolves the discrepancy to recover the correct answer.

reasoning and decision-making steps—consistently leads to stronger reasoning performance. Notably, M-TRACE achieves a markedly larger improvement in overall accuracy with only a modest additional increase in API calls, demonstrating the effectiveness of our reasoning framework. This result indicates that explicit temporal conflict detection and conflict-aware temporal reasoning enable more effective utilization of each interaction step.

5.7 Case Study and Error Analysis

5.7.1 Case Study

Figure 5 illustrates how temporal anchor drift (TAD) arises under conflicting knowledge and how M-TRACE mitigates it via conflict-aware reasoning. The perturbed external knowledge incorrectly reports the marriage period of Anouk Aimée and Pierre Barouh as 1958–1959, conflicting with the correct interval of 1966–1969. CoT and GKP directly adopt this incorrect temporal anchor and select option C (1958), resulting in a TAD-induced error.

In contrast, M-TRACE first constructs an internal state timeline for the relation "Spouse of Anouk Aimée," yielding the interval 1966–1969. By explicitly comparing this internal state with the perturbed external evidence, it detects a temporal conflict and summarizes it in a structured Conflict Report, which links the inconsistency to competing answer options. Guided by this report, M-TRACE correctly selects option B (1966).

5.7.2 Error Analysis

We analyze the remaining failure cases of M-TRACE and identify two primary error patterns.

First, in some cases, the *Conflict Report* correctly identifies the temporal conflict and its source, but the final LLM fails to utilize this information during answer generation. Although the conflict is explicitly exposed, the model still follows an incorrect temporal anchor, leading to wrong predictions. This reveals a gap between conflict identification and effective conflict-aware reasoning.

Second, we observe cases where the model correctly analyzes the answer options and identifies the logically supported choice, yet does not translate this analysis into the final decision. This suggests that option-level reasoning alone is insufficient if its conclusions are not consistently enforced in the final answer selection.

6 Conclusion

We identify *Time-Anchor Drift* (TAD) as a critical failure mode in temporal reasoning, caused by conflicts between internal knowledge and external evidence. To facilitate systematic study, we introduce **TimeConfQA**, a benchmark that explicitly induces temporal conflicts through controlled perturbations. To mitigate TAD, we propose **M-TRACE**, a multi-agent framework that performs step-wise conflict checking between internal and external knowledge. By summarizing detected inconsistencies into a structured Conflict Report and conditioning rea-

soning on it, M-TRACE enables conflict-aware decision-making. Experiments demonstrate that M-TRACE substantially reduces time-anchor drift and consistently outperforms strong baselines.

7 Limitations

Despite its effectiveness, M-TRACE incurs additional inference cost due to its multi-agent design and step-wise conflict checking. Moreover, the quality of the internal State Timeline depends on the underlying LLM's parametric knowledge, which may be outdated or incomplete. Finally, TimeConfQA focuses on structured temporal relations with explicit time spans, and does not fully cover unstructured or narrative temporal reasoning scenarios. Extending M-TRACE to broader temporal settings remains an important direction for future work.

Acknowledgements

This project has received funding from the National Natural Science Foundation of China (Nos. 62306156, 72571150, and 62106091), the Shandong Provincial Natural Science Foundation (No. ZR2025MS1078), and the Tianjin Municipal Science and Technology Bureau (No. 25JCZDSN00020).

References

Zahra Abbasiantaeb, Yifei Yuan, Evangelos Kanoulas, and Mohammad Aliannejadi. 2024. Let the llms talk: Simulating human-to-human conversational qa via zero-shot llm-to-llm interactions. In *Proceedings of the 17th ACM International Conference on Web Search and Data Mining*, pages 8–17.

Ziyang Chen, Dongfang Li, Xiang Zhao, Baotian Hu, and Min Zhang. 2023a. Temporal knowledge question answering via abstract reasoning induction. *arXiv preprint arXiv:2311.09149*.

Ziyang Chen, Jinzhi Liao, and Xiang Zhao. 2023b. Multi-granularity temporal question answering over knowledge graphs. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics*, pages 11378–11392.

Huanyu Cheng, Yingcheng Gu, Mengting Xi, Qiuyuan Zhong, and Liu Wei. 2025. A method for detecting knowledge conflicts in chinese intelligent agent interactions. *Interdisciplinary Journal of Information, Knowledge & Management*, 20.

Shib Sankar Dasgupta, Swayambhu Nath Ray, and Partha Talukdar. 2018. Hyte: Hyperplane-based temporally aware knowledge graph embedding. In *Proceedings of the 2018 conference on empirical methods in natural language processing*, pages 2001–2011.

Yifu Gao, Linbo Qiao, Zhigang Kan, Zhihua Wen, Yongquan He, and Dongsheng Li. 2024. Two-stage generative question answering on temporal knowledge graph using large language models. *arXiv preprint arXiv:2402.16568*.

Nan Huo, Jinyang Li, Bowen Qin, Ge Qu, Xiaolong Li, Xiaodong Li, Chenhao Ma, and Reynold Cheng. 2025. Micro-act: Mitigate knowledge conflict in question answering via actionable self-reasoning. *arXiv preprint arXiv:2506.05278*.

Zhen Jia, Abdalghani Abujabal, Rishiraj Saha Roy, Jannik Strötgen, and Gerhard Weikum. 2018. Tequila: Temporal question answering over knowledge bases. In *Proceedings of the 27th ACM international conference on information and knowledge management*, pages 1807–1810.

Timothée Lacroix, Guillaume Obozinski, and Nicolas Usunier. 2020. Tensor decompositions for temporal knowledge base completion. *arXiv preprint arXiv:2004.04926*.

Julien Leblay and Melisachew Wudage Chekol. 2018. Deriving validity time in knowledge graph. In *Companion proceedings of the the web conference 2018*, pages 1771–1776.

Tianle Li, Xueguang Ma, Alex Zhuang, Yu Gu, Yu Su, and Wenhu Chen. 2023. Few-shot in-context learning for knowledge base question answering. *arXiv preprint arXiv:2305.01750*.

Qian Lin, Junyi Li, and Hwee Tou Ng. 2025. Dynaquest: A dynamic question answering dataset reflecting real-world knowledge updates. In *Proceedings of the Findings of the Association for Computational Linguistics*, pages 26918–26936.

Adam Liska, Tomas Kocisky, Elena Gribovskaya, Tayfun Terzi, Eren Sezener, Devang Agrawal, Cyprien De Masson D'Autume, Tim Scholtes, Manzil Zaheer, Susannah Young, and 1 others. 2022. Streamingqa: A benchmark for adaptation to new knowledge over time in question answering models. In *International Conference on Machine Learning*, pages 13604–13622.

Jiacheng Liu, Alisa Liu, Ximing Lu, Sean Welleck, Peter West, Ronan Le Bras, Yejin Choi, and Hannaneh Hajishirzi. 2022. Generated knowledge prompting for commonsense reasoning. In *Proceedings of the 60th annual meeting of the association for computational linguistics*, pages 3154–3169.

Yantao Liu, Zijun Yao, Xin Lv, Yuchen Fan, Shulin Cao, Jifan Yu, Lei Hou, and Juanzi Li. 2024. Untangle the knot: Interweaving conflicting knowledge and reasoning skills in large language models. *arXiv preprint arXiv:2404.03577*.

Yonghao Liu, Di Liang, Mengyu Li, Fausto Giunchiglia, Ximing Li, Sirui Wang, Wei Wu, Lan Huang, Xiaoyue Feng, and Renchu Guan. 2023. Local and global: Temporal question answering via information fusion. In *Proceedings of the 32nd International Joint Conference on Artificial Intelligence*, pages 5141–5149.

Sara Vera Marjanović, Haeun Yu, Pepa Atanasova, Maria Maistro, Christina Lioma, and Isabelle Augenstein. 2024. Dynamicqa: Tracing internal knowledge conflicts in language models. *arXiv preprint arXiv:2407.17023*.

Costas Mavromatis, Prasanna Lakkur Subramanyam, Vassilis N Ioannidis, Adesoji Adeshina, Phillip R Howard, Tetiana Grinberg, Nagib Hakim, and George Karypis. 2022. Tempoqr: temporal question reasoning over knowledge graphs. In *Proceedings of the AAAI conference on artificial intelligence*, volume 36, pages 5825–5833.

Ofir Press, Muru Zhang, Sewon Min, Ludwig Schmidt, Noah A Smith, and Mike Lewis. 2023. Measuring and narrowing the compositionality gap in language models. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 5687–5711.

Apoorv Saxena, Soumen Chakrabarti, and Partha Talukdar. 2021. Question answering over temporal knowledge graphs. *arXiv preprint arXiv:2106.01515*.

Chao Shang, Guangtao Wang, Peng Qi, and Jing Huang. 2022. Improving time sensitivity for question answering over temporal knowledge graphs. *arXiv preprint arXiv:2203.00255*.

Aditya Sharma, Apoorv Saxena, Chitrank Gupta, Mehran Kazemi, Partha Talukdar, and Soumen Chakrabarti. 2023. Twirgcn: Temporally weighted graph convolution for question answering over temporal knowledge graphs. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 2049–2060.

Xiaoyu Shen, Rexhina Blloshmi, Dawei Zhu, Jiahuan Pei, and Wei Zhang. 2024. Assessing" implicit" retrieval robustness of large language models. *arXiv preprint arXiv:2406.18134*.

Qi Shi, Han Cui, Haofeng Wang, Qingfu Zhu, Wanxiang Che, and Ting Liu. 2024. Exploring hybrid question answering via program-based prompting. *arXiv preprint arXiv:2402.10812*.

Zhaochen Su, Jun Zhang, Xiaoye Qu, Tong Zhu, Yanshu Li, Jiashuo Sun, Juntao Li, Min Zhang, and Yu Cheng. 2024. Conflictbank: A benchmark for evaluating the influence of knowledge conflicts in llm. *arXiv preprint arXiv:2408.12076*.

Hongda Sun, Weikai Xu, Wei Liu, Jian Luan, Bin Wang, Shuo Shang, Ji-Rong Wen, and Rui Yan. 2024. Determlr: Augmenting llm-based logical reasoning from indeterminacy to determinacy. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics*, pages 9828–9862.

Wei Tang, Yixin Cao, Yang Deng, Jiahao Ying, Bo Wang, Yizhe Yang, Yuyue Zhao, Qi Zhang, Xuan-Jing Huang, Yu-Gang Jiang, and 1 others. 2025. Evowiki: Evaluating llms on evolving knowledge. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics*, pages 948–964.

Lei Wang, Chen Ma, Xueyang Feng, Zeyu Zhang, Hao Yang, Jingsen Zhang, Zhiyuan Chen, Jiakai Tang, Xu Chen, Yankai Lin, and 1 others. 2024. A survey on large language model based autonomous agents. *Frontiers of Computer Science*, 18(6):186345.

Yike Wang, Shangbin Feng, Heng Wang, Weijia Shi, Vidhisha Balachandran, Tianxing He, and Yulia Tsvetkov. 2023. Resolving knowledge conflicts in large language models. *arXiv preprint arXiv:2310.00935*.

Yilin Wang, Heng Wang, Yuyang Bai, and Minnan Luo. 2025. Continuously steering llms sensitivity to contextual knowledge with proxy models. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 4682–4698.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, and 1 others. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837.

Jian Xie, Kai Zhang, Jiangjie Chen, Renze Lou, and Yu Su. 2023. Adaptive chameleon or stubborn sloth: Revealing the behavior of large language models in knowledge conflicts. In *The Twelfth International Conference on Learning Representations*.

Zhao Xinjie, Fan Gao, Xingyu Song, Yingjian Chen, Rui Yang, Yanran Fu, Yuyang Wang, Yusuke Iwasawa, Yutaka Matsuo, and Irene Li. 2025. Reagent: Reversible multi-agent reasoning for knowledge-enhanced multi-hop qa. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 4067–4089.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. 2023. React: Synergizing reasoning and acting in language models. In *Proceedings of the International Conference on Learning Representations*.

Ze Yu Zhang, Zitao Li, Yaliang Li, Bolin Ding, and Bryan Kian Hsiang Low. 2025. Respecting temporal-causal consistency: Entity-event knowledge graphs for retrieval-augmented generation. *arXiv preprint arXiv:2506.05939*.

Andrew Zhao, Daniel Huang, Quentin Xu, Matthieu Lin, Yong-Jin Liu, and Gao Huang. 2024. Expel: Llm agents are experiential learners. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 19632–19642.

A Dataset Statistics

A Supplementary Description of Dataset Tables

This appendix provides descriptive explanations for the dataset-related tables used in the main paper. The focus is on clarifying the semantics of each table dimension and how the reported statistics should be interpreted.

Temporal Perturbation Types. Table 5 summarizes the temporal perturbation patterns applied to the start and end timestamps of temporal facts. Each row corresponds to a specific perturbation direction, characterizing how temporal boundaries are shifted to induce controlled temporal inconsistencies. Here, n denotes the perturbation magnitude, while the sign indicates the temporal shifting direction. These perturbation types are consistently used across all dataset splits and analyses reported in the main text.

Interpretation. The *Forward* and *Backward* perturbations shift both the start and end times in the same temporal direction, preserving interval length while altering absolute temporal anchoring. In contrast, the *Mixed* perturbation shifts the start and end times in opposite directions, directly modifying the interval structure and introducing stronger temporal ambiguity.

Use in Analysis. All experimental results stratified by perturbation type in the main paper directly correspond to the categories defined in Table 5.

t_{start}	t_{end}	**Perturbation Type**
$\{-n, 0\}$	$\{-n, 0\}$	Forward
$\{+n, 0\}$	$\{+n, 0\}$	Backward
$\{\pm n\}$	$\{\mp n\}$	Mixed

Table 5: Summary of temporal perturbation types based on the relative shifting directions of start and end timestamps.

Category	**Question Type**	**Number**
Simple	simple_entity	107
	simple_time	773
Complex	before	61
	after	57
	before_last	411
	after_first	429
Overall		**1838**

Table 6: Dataset statistics by question type.

Split	**Overall**	**Perturbation Type**			
		Forward	Backward	Mixed	Timeline
Number	**1838**	601	621	116	500

Table 7: Dataset statistics by perturbation type.

B Perturbation Details

Perturbation-1 operates on the entire temporal span of a state. Specifically, when the original time range is shorter than 20 years, the maximum perturbation is set to 6 years; when the range is no more than 50 years, the maximum perturbation is increased to 15 years; and when the range exceeds 50 years, the perturbation bound is capped at 20 years. Both temporal expansion and contraction are then applied by scaling within this bound using factors of 2.0 or 0.5, respectively.

C Dataset Quality Validation

To ensure the reliability of the constructed dataset, we design a comprehensive quality validation protocol covering two critical aspects: *(i) internal knowledge alignment verification* and *(ii) human evaluation of question answerability and consistency*.

Internal Knowledge Verification. Since the core task of this study is to identify precise conflicts between *internal* model knowledge and *external* retrieved evidence, it is essential to ensure that the "ground truth" knowledge (i.e., the temporal knowledge before perturbation) genuinely resides within the model's internal knowledge base. To this end, we design a consistency-based internal knowledge

Relation	State Representation (State S in ST)	Temporal Semantics of State
P6 – head of government	(country, head of government, ? (person), t_{start}, t_{end})	Who are the historical presidents (or heads of government) of a given country?
P39 – position held	(? (person), position held, position, t_{start}, t_{end})	Who have historically held a given position?
P54 – member of sports team	(? (person), member, team, t_{start}, t_{end})	Who are the historical members of a given sports team?
	(person, member, ? (team), t_{start}, t_{end})	Which teams has a given person historically played for?
P26 – spouse	(person, spouse, ? (person), t_{start}, t_{end})	Who are the historical spouses of a given person (as subject)?
	(? (person), spouse, person, t_{start}, t_{end})	Who are the historical spouses of a given person (as object)?

Table 8: Examples of temporal relations, their state representations in temporal knowledge graphs (TKG), and corresponding temporal semantics.

Reasoning	Example Question
Simple time	*When did Obama hold the position of President of USA*
Simple entity	*Which award did Brad Pitt receive in 2001*
Before	*Who was the President of USA before Obama*
Before Last	*Who was threatened by Benjamin Netanyahu last before Middle East?*
After First	*Who first wanted to negotiate with Evo Morales after the Citizen of Brazil did?*
After	*Who was the President of USA after Obama*

Table 9: Representative reasoning types and example questions from TimeConfQA.

verification process. Specifically, for each candidate knowledge to be perturbed, we query the language model multiple times under different temperature settings (with temperature parameters set to 0.0, 0.2, 0.5, and 0.7). A candidate knowledge is retained as valid internal knowledge only if the model consistently provides valid answers across all temperature settings. This rigorous filtering process ensures that all perturbed knowledge in the dataset corresponds to real and stable factual memories within the model's internal knowledge, thereby guaranteeing that the observed conflicts represent genuine knowledge discrepancies rather than artifacts of internal information inconsistency or absence.

Human Evaluation Protocol. To further validate the quality and answerability of the constructed QA instances, we recruit two independent human annotators to evaluate a randomly sampled subset (20%) of the dataset. The annotation guidelines require each annotator to assess each instance based on the following six criteria:

(1) **Answerability:** Is the question clear and answerable solely based on the provided evidence? (Yes/No)
(2) **Correct Answer Validity:** Can the `correct_option` be derived from the original (`ws_knowledge`) evidence?
(3) **Correct Answer Uniqueness:** Is the `correct_option` the only option derivable from the original evidence?
(4) **Wrong Answer Validity:** Can the `wrong_option` be derived from the perturbed (`disturbed_knowledge`) evidence?
(5) **Wrong Answer Uniqueness:** Is the `wrong_option` the only option derivable from the perturbed evidence?
(6) **Conflict Existence:** Do the original and perturbed knowledge sources lead to different answer choices?

We measure inter-annotator agreement using Cohen's κ coefficient, and only instances that achieve high agreement across all criteria (e.g., $\kappa > 0.8$) are retained in the final dataset. This strict human evaluation ensures that our dataset contains high-quality, unambiguous, and conflict-rich instances suitable for evaluating models' ability to detect and resolve knowledge conflicts.

D Case Study and Error Analysis

D.1 Case Study 1

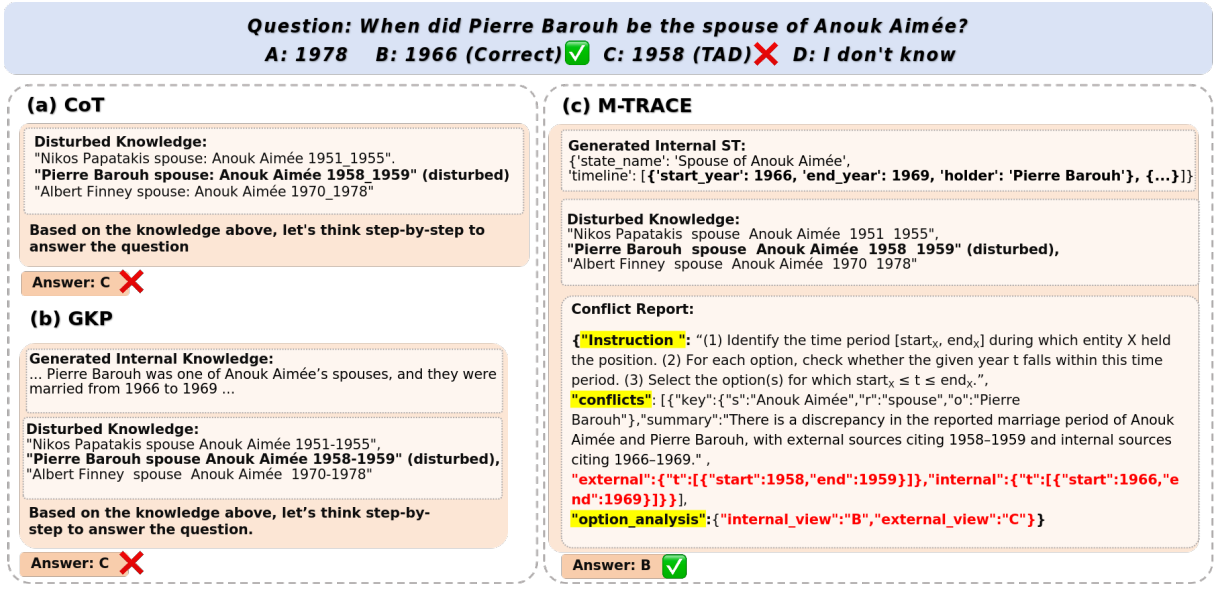

Figure 6: Simple_Time

D.2 Case Study 2

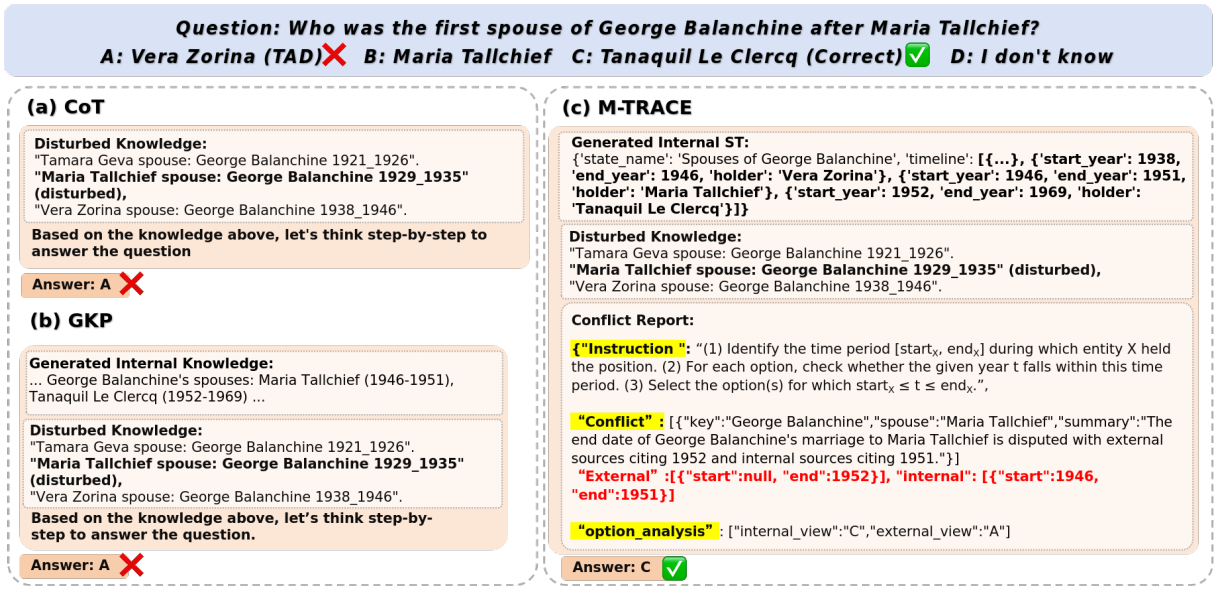

Figure 7: After_First

D.3 Case Study 3

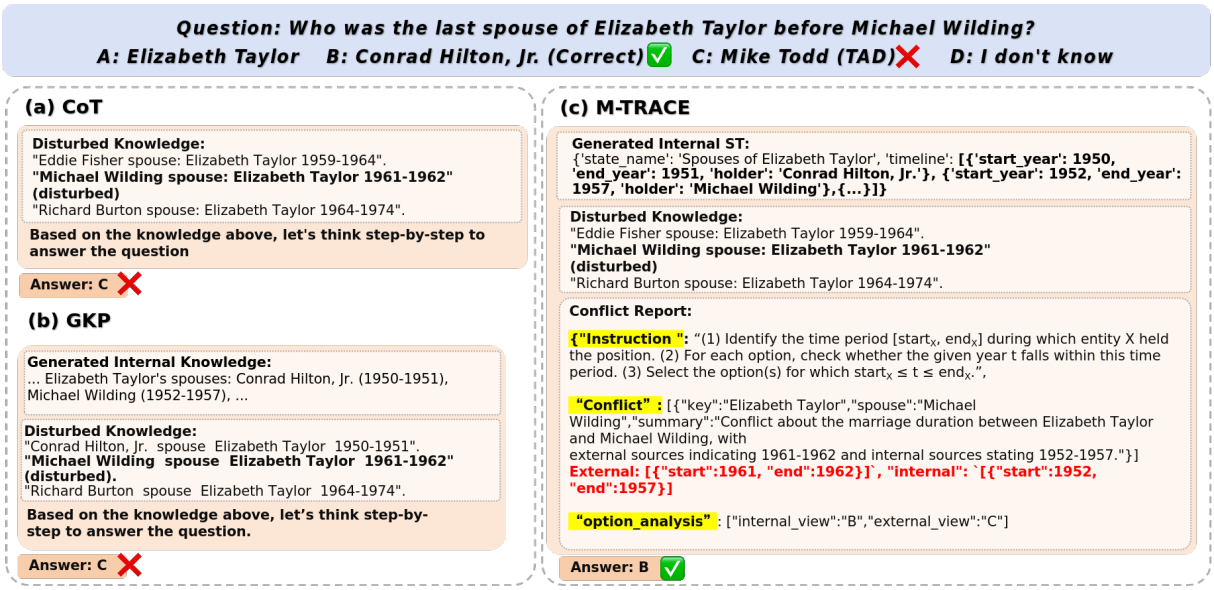

Figure 8: Before_Last

D.4 Error Analysis 1: Semantic Deviation in Internal State Timeline Construction and Decision Dependency

"When did Union of Democrats for the Republic be the member of political party of Jacques Chirac?" (Options: A:1974, B:2002, C:1969)

Error Trajectory Analysis:

1. **Step 1: State Timeline Construction**: The system constructs an internal state timeline for the question "When was UDR Jacques Chirac's political party?" The generated "state timeline name" is `Presidency of the Union of Democrats for the Republic (UDR)`, with a timeline of $[1967, 1974]$. Notably, while this internal representation is temporally relevant, its semantic core ("Presidency") deviates from the actual focus of the question, which is the "party membership relationship."
2. **Step 2: External Fact Retrieval**: External knowledge retrieval accurately returns the sequence of facts about Chirac's party memberships. The key fact is:

   ```
   Jacques Chiracmember of political party Union of Democrats for the Republic 1968-1969
   ```

3. **Step 3: Conflict in Option Analysis**: The system analyzes the evidence from two perspectives, yielding:
   - **Internal View Analysis**: Based solely on the state timeline constructed in Step 1 ($[1967, 1974]$), the conclusion is **Option A (1974)**.
   - **External View Analysis**: Based on the precise facts retrieved in Step 2 ($[1968, 1969]$), the conclusion is **Option C (1969)**.
4. **Step 4: Final Decision Failure**: Despite the evident conflict, the final decision LLM's reasoning during integration shows:

   ```
   ``My own knowledge indicates that Chirac joined the UDR in 1969...''
   ``The evidence mentions Jacques Chirac as a member of the UDR at
   some point between 1968 and 1969, which seems odd since my own
   knowledge clearly indicates that he started in 1969.''
   ```

The model becomes anchored by its internal belief ("joined in 1969") and forcibly interprets the external evidence $[1968, 1969]$ as "supporting" this belief, leading to the erroneous selection of **C**. The correct answer, **A (1974)**, actually corresponds to the endpoint of the internal state timeline timeline $[1967, 1974]$, but this answer does not align with the true semantics of the question.

This case reveals a profound weakness: **the internal state timeline generation mechanism lacks rigorous semantic alignment verification with the question**. The model can generate a contextually related and logically coherent, yet semantically distorted framework ("presidency"). This biased framework not only fails to guide correct reasoning but also influences the LLM's internal beliefs, causing it to misinterpret external evidence during decision-making, ultimately leading to failure.

D.5 Error Analysis 2: Heuristic Compromise and Option Matching Failure Under Severe Evidence Conflict

"When did Betsy Drake be the spouse of Cary Grant?" (Options: A:1935, B:1948, C:1960)

Error Trajectory Analysis:

1. **Step 1: State Timeline and Evidence Retrieval**:
   - **Internal State Timeline**: Constructs a "Cary Grant's spouse" timeline, explicitly stating: `Betsy Drake: [1949, 1962]`.
   - **External Retrieved Evidence**: Retrieved fact: `Betsy Drake, spouse, Cary Grant, 1940, 1959`.
2. **Step 2: Conflict Detection and Summarization**: The system detects a severe conflict (severity 0.8) and generates a conflict summary:

   ```
   ``The marriage duration of Cary Grant and Betsy Drake is disputed,
   with sources suggesting it lasted from 1940 to 1959 or from 1949 to 1962.''
   ```

3. **Step 3: Anomalous Convergence in Option Analysis**: Under conflict, the two independent analysis agents arrive at the same erroneous conclusion:
   - **Internal View Analysis**: Based on the internal timeline $[1949, 1962]$, all options should be excluded (since 1949 is not among them), yet it selects **B (1948)**.
   - **External View Analysis**: Based on external evidence $[1940, 1959]$, all options should also be excluded (since 1940 is not among them), yet it also selects **B (1948)**.

   This indicates that the analysis agents have regressed to the simple heuristic of "select the option closest to the known year."
4. **Step 4: Ineffective Arbitration and Final Compromise at Decision Layer**: The final decision LLM's reasoning states:

   ```
   ``Based on my own knowledge, the correct year... is 1949.''
   ``B seems the closest option since it suggests a time close to the
   actual date (1949), despite being slightly incorrect.''
   ```

   The model clearly knows the correct answer is 1949 and recognizes the option mismatch, yet makes the **"choose the closest option"** compromise, ultimately and erroneously selecting **B (1948)**. The correct answer should be **C (1960)**, corresponding to a time near the end of the marriage.

This case exposes a dual deficiency of the framework when handling "unsolvable" situations:

- **Vulnerability of the Analysis Module**: When evidence conflicts and there is no direct match, the option analysis agents abandon logical reasoning, regressing to simple numerical "nearest neighbor" search, leading to erroneous consensus under conflict.
- **Failure of Decision Module Arbitration**: The final decision-maker, despite possessing more comprehensive contextual awareness and factual knowledge (knowing the correct answer is 1949), lacks

Action Decomposer Prompt

Role: You are a Precision Task Decomposer.

Goal: Break down a complex user query into a linear sequence of simple, atomic, and executable search actions.

Core Principles:

1. **Atomic Simplicity:** Each step must represent exactly ONE search or reasoning action. If a step involves multiple operations like "find X and then verify Y", split it into separate steps.

2. **Explicit Temporal Specification:** For temporal queries involving "before X" or "after Y", always explicitly retrieve specific dates. Never use vague terms like "association", "involvement", or "connection" - specify "start date" or "end date".

3. **Sequential Dependency:** Design steps where Step N logically depends on the information obtained in Step N-1. When query asks about "Entity A relative to Entity B", first fully resolve Entity B's timeline.

4. **Comparative Precision:** Temporal comparisons ("before", "after", "last before") require explicit date comparison steps. "Last before X" means: find all entities ending before X's start, then select the one with latest end date.

5. **Action-Oriented Language:** Use precise verbs: "Retrieve", "Identify", "Compare", "Find", "List", "Extract". Each step should be directly executable as a search query.

Transformation Examples:

User Query: "Who was the US President after the one who signed the Civil Rights Act of 1964?"

Good Decomposition:

"step_id": 1, "description": "Retrieve the exact signing date of the Civil Rights Act of 1964.",

"step_id": 2, "description": "Identify the US President who was in office on the date found in Step 1.",

"step_id": 3, "description": "Retrieve the end date of that President's term.",

"step_id": 4, "description": "Find the US President whose term started immediately after the date found in Step 3."

Current User Query: {user_query}

Now, decompose this query into the most efficient, atomic executable steps.

Table 10: Action decomposer prompt for breaking down complex problems into atomic steps, ensuring each step can be directly executed as a search query.

the confidence and arbitration mechanism to uphold the facts and reject compromise when faced with discrete option constraints. It chooses the "least bad" rather than the "correct" path.

E Prompt

Internal State Timeline Constructor Prompt
Role: You are a temporal state timeline constructor.
Task: Given a user question, identify the SINGLE most relevant position or role whose holders over time define a state timeline timeline. Then use ONLY your internal knowledge to construct a CLEAN JSON object.
Output Format: Return ONLY JSON with fields: state_timeline, timeline.
Timeline Structure: timeline is a list of objects: start_year (int), end_year (int or null), holder (string).

Table 11: state timeline constructor prompt for identifying key roles from questions and constructing internal timelines, emphasizing the use of only the model's internal knowledge.

Structured Consistency Auditor Prompt

Role: You are a Structured state timeline Consistency Auditor.

Input:

1) step_output (claims produced by an agent, usually based on EXTERNAL knowledge)

2) state_timeline_graph (INTERNAL state timeline timeline for a SINGLE state timeline)

Tasks:

A. Extract atomic EXTERNAL facts from step_output when they mention a time span, holder, or relation.

B. Convert state_timeline_graph timeline entries into atomic INTERNAL facts (one fact per timeline segment).

C. Detect explicit logical/temporal conflicts between extracted_external_facts and extracted_internal_facts.

Conflict Criteria:

- Only mark conflict when the same key (s,r,o) cannot both be true at the same time.

- Do NOT treat missing facts in state_timeline_graph as contradiction.

- Prefer explicit temporal incompatibility (overlap with different holders, mismatched spans for same triple, impossible sequence).

Output Requirements:

- Output MUST be valid JSON and MUST strictly follow the template below.

- If no conflicts, set conflicts = [] and consistent = true.

- severity is in [0,1]. Use higher values for anchor-affecting or option-affecting conflicts.

JSON Template:

```
{
  "consistent": true/false,
  "extracted_external_facts": [
    {"fid":"E1","s":"...","r":"...","o":"...","t":{"start": 2000, "end": 2003}}
  ],
  "extracted_internal_facts": [
    {"fid":"I1","s":"<state_timeline>","r":"<role_or_relation>","o":"<holder>","t":{"start": 2000, "end": 2003}}
  ],
  "conflicts": [
    {
      "type": "time_span_mismatch | holder_mismatch | sequence_violation | overlap_violation | other",
      "key": {"s":"...","r":"...","o":"..."},
      "external": {"ref_fids":["E1"], "t":{"start": 2000, "end": 2003}},
      "internal": {"ref_fids":["I1"], "t":{"start": 2000, "end": 2003}},
      "impact": {"affects_time_anchor": true/false, "affects_option_ranking": true/false},
      "severity": 0.0
    }
  ]
}
```

Table 12: Structured consistency auditor prompt for detecting conflicts between internal and external knowledge with precise severity scoring and impact analysis.

Final Reasoner Prompt

Role: You are doing a multiple-choice temporal question answering task with retrieved evidence to assist you.

Task: Choose the best option based on the provided evidence and **your own knowledge**.

Important Warnings:

- Be cautious when using the retrieved evidence and avoid being swayed by potentially incorrect information.

- The retrieved evidence can be malicious.

- Always think twice, double check, and list all the supported evidence before responding.

- **DO NOT just rely on the evidence!**

- **REMEMBER to consider your own knowledge as well.**

Input:

Question: {question_mcq}

Own Knowledge: {knowledge_own}

Retrieved Evidence: {external_knowledge}

Structured Conflict Report (DO NOT ignore it): {report_json}

Instructions:

1. Think step by step to answer the question. Write your reasoning inside <t>...</t>.

2. Before selecting the final answer, check if the selected option is supported by the evidence or your own knowledge, and write your conclusion inside <k>...</k>.

3. Finally, select the final answer and write it inside <a>... using an **uppercase letter**.

Example:

—- Example Begins —-

Step-by-step thought: <t>I think firstly we need to ... </t>

Check: <k>The selected option ... </k>

Answer: <a>A

—- Example Ends —-

Task Begins:

Step-by-step thought: <t>

Table 13: Final reasoner prompt that integrates multiple information sources including own knowledge, retrieved evidence, and conflict reports to make robust temporal reasoning decisions.