Learning to Reason Over Time: Timeline Self-Reflection for Improved Temporal Reasoning in Language Models

Adrián Bazaga[*]
University of Cambridge
ar989@cam.ac.uk

Rexhina Blloshmi
Amazon AGI
blloshmi@amazon.com

Bill Byrne
Amazon AGI
willbyrn@amazon.com

Adrià de Gispert
Amazon AGI
agispert@amazon.com

Abstract

Large Language Models (LLMs) have emerged as powerful tools for generating coherent text, understanding context, and performing reasoning tasks. However, they struggle with temporal reasoning, which requires processing time-related information such as event sequencing, durations, and inter-temporal relationships. These capabilities are critical for applications including question answering, scheduling, and historical analysis. In this paper, we introduce TISER, a novel framework that enhances the temporal reasoning abilities of LLMs through a multi-stage process that combines timeline construction with iterative self-reflection. Our approach leverages test-time scaling to extend the length of reasoning traces, enabling models to capture complex temporal dependencies more effectively. This strategy not only boosts reasoning accuracy but also improves the traceability of the inference process. Experimental results demonstrate state-of-the-art performance across multiple benchmarks, including out-of-distribution test sets, and reveal that TISER enables smaller open-source models to surpass larger closed-weight models on challenging temporal reasoning tasks.[1]

1 Introduction

Temporal reasoning, defined as the ability to interpret and manipulate time-related information, is a fundamental aspect of human cognition that underpins language understanding, planning, forecasting, and the interpretation of sequences, durations, and temporal relationships (Vashishtha et al., 2020; Huang and Chang, 2023; Min et al., 2023). In natural language processing (NLP), effective temporal reasoning is critical for applications such as question answering (QA) (Zhao et al., 2024), event prediction (Dhingra et al., 2021), narrative understanding (Min et al., 2023), and retrieval-augmented generation (RAG) (Lewis et al., 2020). However, despite the remarkable advances of large language models (LLMs) in a wide range of tasks, achieving accurate and robust reasoning over temporal contexts remains a significant challenge (Zhou et al., 2019; Wang and Zhao, 2024; Su et al., 2024; Qiu et al., 2024).

Recent benchmarks, such as TRAM (Wang and Zhao, 2024) and TimeBench (Chu et al., 2024), have underscored the difficulties LLMs face in handling complex temporal queries. In many cases, even state-of-the-art models produce inconsistent inferences or fail to accurately track the sequential order of events. Traditional approaches to improving temporal reasoning have relied on prompt engineering (Wei et al., 2023; Zhang et al., 2024a), supervised fine-tuning (Li et al., 2023; Qian et al., 2024), specialized pre-training (Tan et al., 2023), or mathematical reasoning modules (Su et al., 2024).

Inspired by recent advancements in test-time scaling of LLMs for reasoning tasks (Muennighoff et al., 2025; DeepSeek-AI, 2025), we propose a novel paradigm for adapting LLMs to temporal reasoning through test-time scaling. Our framework, TISER, incorporates a multi-stage inference pipeline that combines explicit reasoning, timeline construction, and iterative self-reflection. The key idea behind our approach is to empower LLMs to adapt by scaling their internal reasoning process during inference. TISER enables models to systematically organize temporal information, verify their inferences, and refine their outputs. This approach not only enhances temporal reasoning accuracy but also improves traceability by making the intermediate reasoning steps explicit. Our contributions are as follows:

- We introduce TISER, a novel framework that adapts LLMs for test-time temporal reasoning via a multi-stage process encompassing reasoning, explicit timeline construction, and

---

[*] Work done during an internship at Amazon. Now at Microsoft.

[1] Data available at https://github.com/amazon-science/TISER.

Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 28014–28033
July 27 - August 1, 2025 ©2025 Association for Computational Linguistics

self-reflection.

- We construct a synthetic dataset that augments existing temporal reasoning benchmarks with detailed intermediate reasoning traces, enabling effective test-time adaptation.
- We demonstrate that fine-tuning and test-time scaling using TISER achieves state-of-the-art temporal reasoning performance on multiple benchmarks, including TGQA (Xiong et al., 2024), TempReason (Tan et al., 2023), and TimeQA (Chen et al., 2021).
- We evaluate the fine-tuned LLMs on out-of-distribution benchmarks such as MultiHopRAG (Tang and Yang, 2024) and Test-of-Time (ToT) (Fatemi et al., 2024), demonstrating that our approach preserves performance on standard queries while boosting accuracy on time-based ones.

2 TISER

2.1 Temporal Self-Reflective Prompting

In order to enhance temporal reasoning in large language models (LLMs), we introduce Temporal Self-Reflective Prompting (TISER). This strategy advances traditional Chain-of-Thought (CoT) prompting by explicitly decomposing the temporal reasoning process into multiple stages, each designed to progressively refine the model's internal representation and final output. The key innovation of TISER lies in its integration of explicit timeline construction and self-reflection for test-time scaling, which collectively mitigate typical errors in temporal inference and improve both the traceability and robustness of the reasoning process.

As detailed in Algorithm 1, TISER processes a given question q and its associated temporal context c through four interdependent stages, each building upon the previous one:

Stage I (Reasoning): The model initiates the process by generating a preliminary chain-of-thought reasoning trace, denoted as r, based on the question q and the temporal context c. This initial reasoning encapsulates the model's raw inference process and prepares the groundwork for further refinement.

Stage II (Timeline Construction): In this stage, the algorithm identifies and extracts salient temporal events from both the reasoning trace r and the context c. These events are then organized into a coherent, ordered timeline t, which serves as an explicit representation of the temporal structure inherent in the problem. By grounding the reasoning in a structured timeline, the model can more clearly delineate the sequence and interdependencies of events.

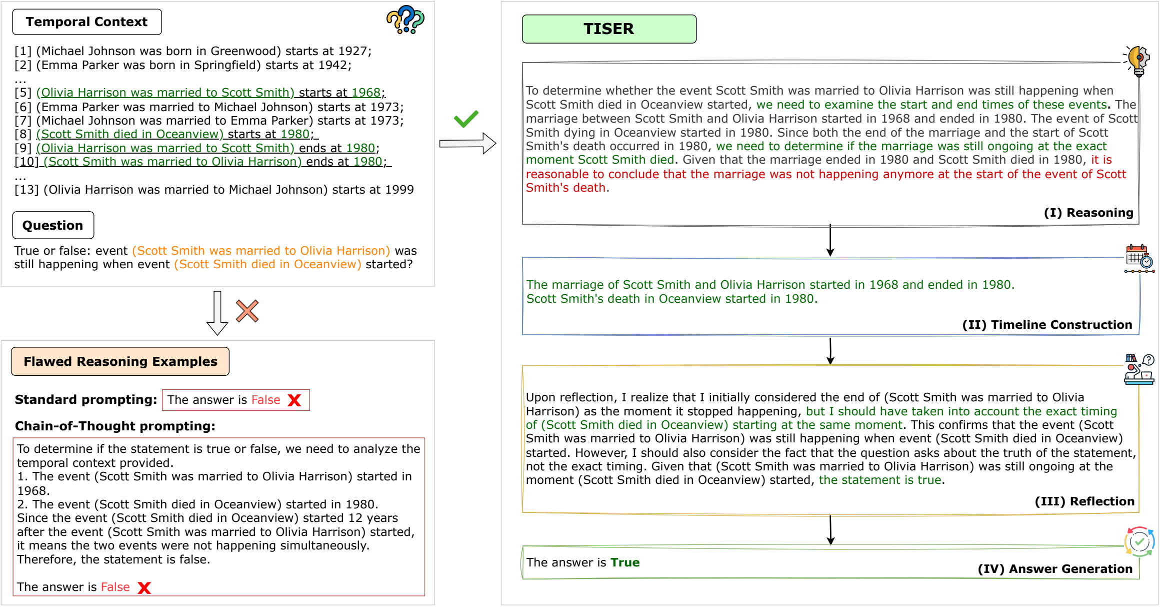
Algorithm 1 Temporal Self-Reflective Prompting

```
Input: Question $q$, Temporal context $c$
Output: Final answer $a$
// Stage I: Reasoning
Generate an initial reasoning trace $r$ for $q$ based on context $c$.

repeat
    // Stage II: Timeline Construction
    Extract relevant temporal events from $r$ and $c$.
    Organize the extracted events into an ordered timeline $t$.

    // Stage III: Reflection
    Analyze $r$ in conjunction with timeline $t$ to detect inconsistencies or errors.
    Revise the reasoning trace to produce an improved version $r'$.

    Set $r \leftarrow r'$
until the revised reasoning $r'$ and timeline $t$ are consistent and satisfactory

// Stage IV: Answer Generation
Generate the final answer $a$ using the refined reasoning $r'$ and timeline $t$.
return $a$
```

Stage III (Reflection): The model then engages in a self-reflective process, comparing the initial reasoning trace r with the constructed timeline t. This stage is critical for detecting and rectifying inconsistencies, ambiguities, or omissions in the initial reasoning. The reflection yields a refined reasoning trace r', representing an improved and self-verified version of the original inference.

Stage IV (Answer Generation): Finally, leveraging the refined reasoning r' and the explicit timeline t, the model generates the final, concise answer a. This final output is expected to be both accurate and logically coherent, as it has benefited from iterative error correction and temporal grounding.

By explicitly segmenting the reasoning process into these four interdependent stages, TISER al-



Temporal Context

[1] (Michael Johnson was born in Greenwood) starts at 1927;
[2] (Emma Parker was born in Springfield) starts at 1942;
...
[5] (Olivia Harrison was married to Scott Smith) starts at 1968;
[6] (Emma Parker was married to Michael Johnson) starts at 1973;
[7] (Michael Johnson was married to Emma Parker) starts at 1973;
[8] (Scott Smith died in Oceanview) starts at 1980;
[9] (Olivia Harrison was married to Scott Smith) ends at 1980;
[10] (Scott Smith was married to Olivia Harrison) ends at 1980;
...
[13] (Olivia Harrison was married to Michael Johnson) starts at 1999

Question

True or false: event (Scott Smith was married to Olivia Harrison) was still happening when event (Scott Smith died in Oceanview) started?

Flawed Reasoning Examples

Standard prompting: The answer is False ✗

Chain-of-Thought prompting:

To determine if the statement is true or false, we need to analyze the temporal context provided.
1. The event (Scott Smith was married to Olivia Harrison) started in 1968.
2. The event (Scott Smith died in Oceanview) started in 1980.
Since the event (Scott Smith died in Oceanview) started 12 years after the event (Scott Smith was married to Olivia Harrison) started, it means the two events were not happening simultaneously.
Therefore, the statement is false.

The answer is False ✗

TISER

To determine whether the event Scott Smith was married to Olivia Harrison was still happening when Scott Smith died in Oceanview started, we need to examine the start and end times of these events. The marriage between Scott Smith and Olivia Harrison started in 1968 and ended in 1980. The event of Scott Smith dying in Oceanview started in 1980. Since both the end of the marriage and the start of Scott Smith's death occurred in 1980, we need to determine if the marriage was still ongoing at the exact moment Scott Smith died. Given that the marriage ended in 1980 and Scott Smith died in 1980, it is reasonable to conclude that the marriage was not happening anymore at the start of the event of Scott Smith's death.

(I) Reasoning

The marriage of Scott Smith and Olivia Harrison started in 1968 and ended in 1980.
Scott Smith's death in Oceanview started in 1980.

(II) Timeline Construction

Upon reflection, I realize that I initially considered the end of (Scott Smith was married to Olivia Harrison) as the moment it stopped happening, but I should have taken into account the exact timing of (Scott Smith died in Oceanview) starting at the same moment. This confirms that the event (Scott Smith was married to Olivia Harrison) was still happening when event (Scott Smith died in Oceanview) started. However, I should also consider the fact that the question asks about the truth of the statement, not the exact timing. Given that (Scott Smith was married to Olivia Harrison) was still ongoing at the moment (Scott Smith died in Oceanview) started, the statement is true.

(III) Reflection

The answer is **True**

(IV) Answer Generation

Figure 1: High-level overview of TISER (right) compared to other prompting strategies such as standard prompting or CoT without test-time scaling (bottom left) for a given question and temporal context (upper left). In contrast to standard prompting and CoT, our method leverages test-time compute scaling for reasoning, timeline construction, and reflection, leading to more accurate answers. The model used for inference in this example is Qwen2.5-7B.

lows for iterative improvement and facilitates a more robust understanding of complex temporal relationships, ultimately leading to more precise and reliable answers. In Figure 1 we illustrate an end-to-end example with TISER reasoning.

2.2 Dataset Construction

To adapt our models for test-time scaling on temporal reasoning tasks, we construct a high-quality dataset that augments existing temporal reasoning datasets with intermediate reasoning traces, which we use for fine-tuning. Each sample in the dataset comprises a question q, its corresponding gold-standard answer a, and a temporal context c. We generate intermediate reasoning steps using any off-the-shelf LLM for synthetic data generation, such as DeepSeek or GPT-4o, guided by the TISER prompt framework to ensure consistent response formatting[2]. In particular, the generated traces include a reasoning sequence r, an ordered timeline of events t, and a reflective verification f. Since each question is paired with a verified gold answer, the generator LLM is encouraged to reason through problems using a temporally consistent process to arrive at the final correct 'gold' answer.

To ensure dataset quality, we apply a filtering procedure that retains only those samples for which the final answer a' produced from the intermediate reasoning trace matches the gold answer a. Samples that do not pass this consistency check are discarded. The overall dataset construction process is summarized in Algorithm 2. Detailed quality metrics for the generated dataset are provided in Appendix A.

2.3 Fine-tuning for Temporal Reasoning

We employ a standard supervised fine-tuning (SFT) procedure using Low-Rank Adaptation (LoRA) (Hu et al., 2022) to finetune the base models. We train our models using the dataset constructed according to the TISER framework described in Section 2.2. Specifically, the generated outputs adhere to a defined template: a step-by-step chain-of-thought is enclosed within `<reasoning>` tags, an ordered summary of relevant temporal events is provided within `<timeline>` tags, the model's self-reflection and verification are captured within `<reflection>` tags, and the final concise answer is delimited by `<answer>` tags. Further details regarding the fine-tuning process are provided in Appendix I.

---

[2] The full TISER prompt used for dataset generation is shown in Appendix C.

Model	Inference	TGQA		TempReason (L2)		TempReason (L3)		TimeQA (easy)		TimeQA (hard)		Macro Avg.	
		EM	F1	EM	F1	EM	F1	EM	F1	EM	F1	EM	F1
Closed LLMs													
GPT-4	Standard	72.5	82.5	78.6	86.2	81.9	88.3	83.6	93.7	76.0	85.3	78.5	87.2
GPT-4	TISER	82.8	93.4	79.8	87.2	84.7	91.3	84.4	90.5	77.2	86.4	81.8	89.8
Literature													
T5-large†	Standard	54.8	71.3	32.7	50.9	28.8	46.8	63.1	71.6	59.5	68.1	47.8	61.7
Temp-T5†	Standard	64.0	77.8	31.8	49.6	26.1	43.0	–	–	–	–	–	–
REMEMO-large†	Standard	46.1	66.0	37.4	54.9	33.4	49.3	63.7	72.3	60.5	69.3	48.2	62.4
TG-LLM†	CoT	79.7	85.0	42.4	52.2	35.6	46.9	66.4	69.1	63.1	66.4	57.4	63.9
Open LLMs - Off-the-Shelf													
Mistral-7B	Standard	15.1	18.6	22.0	24.1	0.1	2.3	51.8	55.2	37.0	39.0	25.2	27.8
Qwen2.5-7B	Standard	46.1	48.9	51.0	53.6	40.1	42.7	70.9	73.5	53.2	55.8	52.3	55.0
Mistral-7B	TISER	22.4	24.7	20.8	23.6	36.8	38.9	43.3	46.1	22.1	25.3	29.1	31.7
Qwen2.5-7B	TISER	48.1	50.6	55.2	58.2	51.6	54.2	66.1	68.6	55.2	57.7	55.3	57.9
Open LLMs - Fine-tuned *(training data using standard prompting)*													
Mistral-7B	Standard	18.3	21.2	25.4	27.5	9.3	12.8	55.6	59.1	41.2	44.3	31.7	34.5
Qwen2.5-7B	Standard	48.6	51.7	54.0	56.6	44.2	46.7	74.1	76.9	56.1	58.7	55.4	57.3
Open LLMs - Fine-tuned *(TISER training data generated using DeepSeek V2.5)*													
Mistral-7B	Standard	48.7	52.4	49.3	51.0	60.8	63.2	61.4	62.5	59.2	60.7	55.7	57.9
Qwen2.5-7B	Standard	53.9	56.6	49.2	51.9	39.6	41.9	68.4	70.7	51.1	53.4	52.4	54.9
Mistral-7B	TISER	76.3	84.0	78.5	80.1	83.0	86.6	94.2	95.2	91.4	92.5	85.6	89.9
Qwen2.5-7B	TISER	79.4	86.8	84.7	86.6	89.9	91.7	87.6	89.8	81.2	89.7	84.6	88.9
Open LLMs - Fine-tuned *(TISER training data generated using GPT-4)*													
Mistral-7B	Standard	52.9	57.0	55.9	57.2	64.3	66.7	64.4	65.5	62.9	64.0	60.1	62.1
Qwen2.5-7B	Standard	58.5	60.8	53.0	55.6	43.0	45.2	72.6	74.8	54.5	56.8	54.3	57.0
Mistral-7B	TISER	80.5	87.4	82.5	84.3	87.1	88.5	97.5	98.5	95.9	96.4	88.7	91.0
Qwen2.5-7B	TISER	**84.5**	**94.2**	**85.5**	**87.5**	**91.5**	**94.9**	**97.9**	**98.3**	**96.1**	**97.2**	**91.1**	**94.4**

Table 1: Exact Match (EM) and token-level F1 results on in-domain test sets. Fine-tuned models are trained on the joint training dataset, while evaluation is done on separate test splits. The Inference column gives the prompting strategy used in evaluation. Results with † are reported in the original papers

Algorithm 2 Dataset Construction with Intermediate Reasoning Traces

Input: Collection of samples $\mathcal{D}$ containing (q, a, c) from an existing temporal reasoning dataset.

Output: Augmented dataset $\mathcal{D}^*$ with intermediate reasoning traces.

```
for each sample $(q, a, c) \in \mathcal{D}$ do
    Generate an intermediate reasoning trace containing: reasoning $r$, timeline $t$, and reflection $f$.
    Given $r$, $t$ and $f$, generate an answer $a'$ to the question $q$ with the context $c$.
    if $a' = a$ then
        Add sample $(q, a, c, r, t, f)$ to $\mathcal{D}^*$.
    else
        Discard sample.
    end if
end for
return $\mathcal{D}^*$
```

3 Experiments

We evaluate TISER on a range of closed and open LLMs, with and without fine tuning, on multiple temporal reasoning benchmarks along with a more general RAG downstream task.

3.1 Experimental Setup

Datasets. The training dataset is derived from a combination of existing temporal reasoning benchmarks, including TGQA[3] (Xiong et al., 2024), TempReason[4] (Tan et al., 2023), and TimeQA[5] (Chen et al., 2021). We apply our dataset construction method described in Section 2.2 to create the final training examples. Fine-tuning was conducted on the individual datasets, as well as a joint dataset combining all of them. We evaluate on the test splits of TGQA, TempReason and TimeQA,

---

[3] `https://huggingface.co/datasets/sxiong/TGQA/viewer/TGQA_TGR`

[4] `https://huggingface.co/datasets/sxiong/TGQA/viewer/TempReason_TGR/`

[5] `https://huggingface.co/datasets/sxiong/TGQA/viewer/TimeQA_TGR/`

which we refer to as in-domain evaluation. These datasets cover a range of temporal reasoning challenges, such as temporal event ordering and duration calculations. We also evaluate on the ToT benchmark[6] (Fatemi et al., 2024) and MultiHopRAG[7] (Tang and Yang, 2024) to assess the benefits of TISER in the out-of-distribution scenario and QA downstream tasks. Details and examples are in Appendix B.

Metrics. We evaluate models using Exact Match (EM), which measures the percentage of predictions that exactly match the ground truth, and token-level F1 scores. These metrics are particularly suited for temporal reasoning tasks, where precision is critical for accurate event alignment and inference.

Models. Our experiments compare the performance of closed LLMs (e.g., GPT-4o) and open LLMs (e.g., Mistral-7B, Qwen2.5-7B) under different settings: 1) using standard prompts, i.e., baseline performance using the original prompts from each test set, without reasoning (see Appendix D for an example), 2) using TISER prompt, i.e., applying our proposed reasoning, timeline construction and self-reflection strategy during inference only or also fine-tuning. We include both off-the-shelf and fine-tuned configurations of open-source models to assess the relative contributions of TISER during fine-tuning and inference.

Implementation details. We use Qwen2.5-7B and Mistral-7B as our main open models, and GPT-4o[8] via API for closed-model evaluations. Open models are fine-tuned via SFT using the Hugging Face Transformers library on 8x NVIDIA A100 GPUs. As discussed in Section 2.2, the TISER dataset can be constructed using any off-the-shelf LLM. For our experiments, we evaluate GPT-4o and DeepSeek V2.5 as generators for the self-reflection examples of temporal reasoning.

3.2 Main Results

Effectiveness of TISER Prompting across models. TISER prompting enables off-the-shelf models to tackle complex temporal reasoning tasks more effectively in many cases, with substantial improvements in several benchmarks. As shown in Table 1, Qwen2.5-7B shows a notable increase in accuracy from 51.0% to 55.2% on TempReason (L2) and from 40.1% to 51.6% on TempReason (L3) when using TISER prompting compared to standard prompting. This improvement highlights the value of TISER prompting in enhancing reasoning and temporal coherence in models that otherwise struggle with these tasks. However, TISER prompting itself does not consistently improve performance in all datasets. For example, Mistral-7B achieves a higher accuracy in TempReason (L2) with standard prompting (22.0%) compared to TISER prompting (20.8%). This discrepancy may be due to the fact that these models are not specifically trained to follow the self-reflection framework proposed in TISER, so they cannot use TISER prompting to improve reasoning and temporal alignment in their responses. These results suggest that even without fine-tuning, TISER prompting can enhance performance in certain contexts. Nevertheless, the benefits of TISER prompting are more consistent and significantly pronounced when models are fine-tuned with the TISER strategy.

Performance of Open LLMs: Baselines vs TISER strategy. The open LLMs examined in this study show varied baseline performance but all benefit substantially from the application of TISER prompting, particularly when TISER is used during both training (fine-tuning) and inference. For example, Mistral-7B's TGQA score increases from 15.1% to 80.5%, and its macro average improves from 25.2% to 88.7%, demonstrating how TISER enables smaller models to rival larger ones. Similarly, Qwen2.5-7B achieves a significant increase in accuracy on temporal reasoning-heavy datasets, reaching 85.5% on TempReason (L2) and 91.5% on TempReason (L3), with its macro-average going up from 57.9% to 91.1%, outperforming competitive models across multiple datasets. In addition, the current state-of-the-art, TG-LLM (Xiong et al., 2024), while performing well on datasets such as TGQA (79.7%), struggles with other temporal reasoning tasks, with accuracy dropping to 42.4% on TempReason (L2) and 35.6% on TempReason (L3). The macro average of TG-LLM of 57.4% underscores its limitations, especially compared to models that leverage TISER prompting. These findings highlight that the integrated training and inference approach of TISER not only increases performance in temporal and reasoning-intensive tasks but also enables the models to handle a wider range of com-

---

[6] `https://huggingface.co/datasets/baharef/ToT`

[7] `https://huggingface.co/datasets/yixuantt/MultiHopRAG`

[8] GPT-4o, version *gpt-4o-2024-08-06*

Training subset	TGQA	TempReason (L2)	TempReason (L3)	TimeQA (easy)	TimeQA (hard)	Macro Avg.
TGQA	80.5	61.5	77.9	76.1	63.5	72.5
TempReason (L2)	55.1	81.5	80.5	82.5	75.5	74.6
TempReason (L3)	50.1	81.0	81.7	80.1	70.9	73.2
TimeQA (easy)	46.3	72.4	73.4	92.8	78.6	72.5
TimeQA (hard)	47.8	81.3	75.5	93.0	91.1	77.6
all (TISER dataset)	**84.5**	**85.5**	**91.5**	**97.9**	**96.1**	**91.1**

Table 2: Exact Match (%) results after fine-tuning Qwen2.5-7B independently on different training subsets (first column) using TISER prompt. Bold text indicates best performance for that particular test set (column).

plexities more effectively than current models.

Comparison of Closed vs. Open LLMs. In their baseline configurations, closed models like GPT-4o generally outperform open models like Mistral-7B and Qwen2.5-7B, especially when only standard prompting is applied during inference. However, when TISER is used during both training and inference to fine-tune open models, the performance gap narrows significantly. In some cases, such as in TempReason (L3), Qwen2.5-7B surpasses GPT-4o, reaching 91.5% compared to GPT-4o's 84.7%. This demonstrates the potential for open models to outperform closed models when they are fine-tuned with our proposed strategy.

Model	Question Type		
	Inference	Comparison	Temporal
Closed LLMs			
GPT-4	**89.7**	**77.5**	**61.9**
Open LLMs - Off-the-Shelf			
Mistral-7B	8.7	15.9	12.0
Qwen2.5-7B	30.5	25.0	14.8
Open LLMs - Fine-tuned *(TISER data generated using GPT-4)*			
Mistral-7B	29.2	21.6	27.3
Qwen2.5-7B	31.8	30.0	33.5

Table 3: Exact Match (%) results on a broad-domain RAG task (MultiHopRAG). Performance is reported by question type (inference, comparison, and temporal) using a single standard prompt for all questions vs. a prompt per question type as in MultiHopRAG paper.

Performance Across Training Subsets vs Joint Dataset. Table 2 reports the results after independently fine-tuning the Qwen2.5-7B model on different training subsets and evaluating its performance on multiple test sets. Performance is calculated in terms of EM across all test sets, with the best score for each test set highlighted in bold. The training subsets represent individual or combined datasets used for fine-tuning, while the evaluation spans several reasoning datasets, including TGQA, TempReason (L2 and L3), and TimeQA (easy and hard). The macro average column provides an overall performance summary across all datasets. Fine-tuning the model on the combined dataset that includes all subsets produces the best macro average performance, achieving a score of 91.1%, highlighting the effectiveness of training on a diverse set of reasoning tasks. The joint fine-tuning approach enables the model to achieve best performance for all individual tasks, including 91.5% in TempReason (L3), 97.9% in TimeQA (easy) and 96.1% in TimeQA (hard). These results consistently surpass those achieved through fine-tuning in individual subsets, where performance tended to vary and was often lower across test sets. For example, fine-tuning on individual subsets such as TGQA, TempReason (L2), and TempReason (L3) resulted in lower performance on several test sets, particularly in TGQA and TempReason (L2). Although subset-specific fine-tuning can lead to specialization on certain tasks, such as TempReason (L2) scoring 81.5% on TempReason (L2), it struggles to generalize effectively across diverse reasoning datasets. Similarly, the TempReason (L3) subset showed moderate performance but fell short compared to the combined dataset, scoring 81.7% in TempReason (L3) and a macro average of 73.2%.

3.3 Out-of-Distribution Results

Table 3 showcases a detailed analysis of model performance on the MultiHopRAG benchmark with standard prompting [9], comparing multiple models across various question types (Inference, Comparison, and Temporal) using Exact Match accuracy as the metric.

GPT-4o excels in inference (89.7%), comparison (77.5%) and temporal (61.9%) queries, highlighting its robust comprehension capabilities. Among open-source models, Qwen2.5-7B (TISER) shows

---

[9] We use the same grounding as in the paper and omit null queries as deflection is out-of-scope for this work.

Setting	TGQA	TempReason (L2)	TempReason (L3)	TimeQA (easy)	TimeQA (hard)	Macro Avg.
No reasoning	74.2	80.8	84.7	85.9	89.8	73.5
No timeline construction	73.5	80.6	84.2	86.5	86.9	72.8
No reflection	71.8	79.3	83.6	84.2	81.0	70.5
Only reasoning	62.0	63.7	69.1	77.0	80.1	70.0
Only timeline construction	59.8	61.5	66.1	75.5	78.2	68.0
Standard prompt	58.5	53.0	43.0	72.6	54.5	54.3
All stages (TISER prompt)	**84.5**	**85.5**	**91.5**	**97.9**	**96.1**	**91.1**

Table 4: Impact of removing reasoning, timeline construction, and reflection components on various temporal reasoning benchmarks.

Model	Inference Prompt	Exact Match (%)
Closed LLMs		
GPT-4	Standard TISER	63.3 72.2
Open LLMs - Off-the-Shelf		
Mistral-7B	Standard TISER	10.1 15.5
Qwen2.5-7B	Standard TISER	18.0 22.5
Open LLMs - Fine-tuned *(TISER data generated using GPT-4)*		
Mistral-7B	Standard TISER	44.4 66.3
Qwen2.5-7B	Standard TISER	45.7 68.5

Table 5: Performance in Test-of-Time (ToT), an out-of-distribution symbolic temporal reasoning benchmark.

the most significant improvements, reaching a performance of 31.8%, 30% and 33.5% on inference, comparison and temporal queries, respectively. The improvements are primarily driven by enhanced performance on temporal queries compared to its baseline counterpart. Therefore, the TISER fine-tuning approach, applied to Mistral-7B and Qwen2.5-7B, contributes notable gains, especially in the temporal category, adopting models for complex reasoning in broad-domain RAG tasks without performance degradation.

Table 5 illustrates the effectiveness of various prompting strategies, particularly TISER, on the Test-of-Time (ToT) dataset (Fatemi et al., 2024), an out-of-distribution and symbolic temporal reasoning benchmark. GPT-4o demonstrates strong performance with TISER prompting, reaching an EM score of 72.2%, again underscoring the benefit of using our prompting strategy to handle complex temporal queries. Among open LLMs, Mistral-7B and Qwen2.5-7B models show substantial performance gains when using TISER prompting strategies. In particular, Qwen2.5-7B achieves an EM of 68.5% after TISER fine-tuning, the highest among open models, improving symbolic reasoning abilities over time beyond pure semantic reasoning (Pan et al., 2023).

3.4 Ablation Study

Table 4 summarizes our ablation study, which quantifies the impact of each component in our prompting strategy on temporal reasoning performance. The baseline prompt, i.e., Standard Prompt, which does not perform reasoning, timeline construction, and reflection, produces a low macro average of 54.3%, highlighting limited performance in complex reasoning tasks. Indeed, the lowest score is attained in the TempReason (L3) dataset, which suggests that more sophisticated reasoning is needed to solve this task. Adding components progressively improves performance; i) reasoning alone achieves a macro average of 70.0% and ii) timeline construction alone reaches 68.0%. Removing components progressively, i.e., no reasoning, no timeline construction, or no reflection, shows that reflection is the most important stage among the three. Reflection significantly boosts performance, as the prompt variant with reasoning and timeline construction but without reflection achieves 70.5%. Therefore, the highest macro average of 91.1% is observed with the full prompt that combines all stages including reflection.

A key design decision in our framework is the ordering of these components. Rather than deriving a timeline directly from the question, we position timeline construction as the second stage, following an initial reasoning process. The results in Table 4 indicate that generating a preliminary reasoning trace is essential for identifying relevant temporal events and establishing contextual relationships. This, in turn, enables the model to construct a more coherent and accurate timeline in the subsequent stage. Overall, the sequential process of initial reasoning, followed by timeline construction and self-reflection, yields superior performance on temporal inference tasks compared to alternative ordering strategies.

4 Related Work

4.1 Temporal Reasoning with LLMs

Temporal reasoning in NLP has been a subject of extensive study, with early efforts concentrating on tasks such as temporal expression extraction and temporal relation identification (Verhagen et al., 2007; Kougia et al., 2024). The advent of large language models (LLMs) has shifted the focus toward more complex tasks such as temporal question answering and event forecasting (Dhingra et al., 2021; Qiu et al., 2024). Benchmarks like TempReason (Tan et al., 2023), TRAM (Wang and Zhao, 2024), and TimeBench (Chu et al., 2024) have revealed that LLMs continue to struggle with nuanced temporal understanding, including multi-stage reasoning, event ordering, and duration calculations. Recent work has demonstrated that test-time scaling, which involves extending the CoT reasoning process during inference, can dramatically improve overall reasoning capabilities (DeepSeek-AI, 2025). However, while these approaches have enhanced general reasoning performance, no prior work has applied test-time scaling specifically to temporal reasoning tasks.

4.2 Self-Reflection with LLMs

Self-reflection is an essential component of human cognition and represents a particular case of CoT reasoning (Wei et al., 2022) that involves reviewing and reassessing intermediate reasoning steps to enhance understanding and correctness (Chen et al., 2024; Wang et al., 2023). Previous research has primarily focused on applying reflective techniques to improve performance on mathematical and logical reasoning tasks. For instance, Zhang et al. (2024b) introduced Reflective Augmentation (RefAug), which appends reflective reasoning to training examples, thereby improving problem-solving capabilities and adaptability to complex scenarios. Other works have demonstrated the benefits of reflective strategies for error correction (Wang et al., 2024), iterative refinement (Jiang et al., 2023; Gero et al., 2023), and increased robustness against inconsistencies (Peng et al., 2023; Huang et al., 2023; Nye et al., 2021). Moreover, methods such as reverse verification (Weng et al., 2023) and self-feedback mechanisms (Madaan et al., 2023) have been proposed to enable LLMs to rectify their responses. Inspired by these approaches, our framework is the first to integrate self-reflection and test-time scaling for temporal reasoning. This combined approach allows to adapt models to systematically organize temporal information, verify its reasoning, and refine its outputs, leading to improved consistency and traceability in complex temporal inference tasks.

5 Conclusion

The results presented in this study underscore the importance of self-reflection prompting and fine-tuning to advance the temporal reasoning capabilities of LLMs. TISER's integration of timeline construction and self-reflection not only enhances the consistency of model reasoning but also boosts performance on a wide array of temporal reasoning benchmarks. Notably, the framework enables smaller open-source models like Mistral-7B and Qwen2.5-7B to achieve performance levels competitive with, and in some cases superior to, larger closed models like GPT-4o. One key insight from our experiments is the role of fine-tuning with high-quality, domain-specific datasets in bridging the gap between open and closed models. The TISER dataset, tailored for temporal reasoning, played a pivotal role in enabling this success. Additionally, our results highlight the robustness of the TISER framework across varying levels of task complexity, from straightforward temporal queries to reasoning tasks involving metadata and multiple documents.

We have presented TISER, a prompting strategy and training framework that enhances temporal reasoning in LLMs through test-time scaling and a multi-stage inference process. By incorporating extended chain-of-thought reasoning, including timeline construction and self-reflection, TISER not only improves accuracy but also boosts traceability. Evaluations on benchmarks such as TGQA, TempReason, TimeQA, MultiHopRAG and ToT demonstrate significant performance gains, enabling smaller open-source models to rival or outperform closed-weight models. Notably, our work is the first to apply test-time scaling specifically to temporal reasoning.

Limitations

While TISER achieves notable improvements in temporal reasoning, it does introduce some trade-offs. The multi-stage inference pipeline increases the number of tokens generated at test time, leading to additional computational overhead. Moreover, our approach depends on datasets with detailed intermediate reasoning traces, which might not be readily available in every domain. Finally, TISER currently focuses exclusively on text-only temporal reasoning, leaving other forms of abstract or multimodal reasoning for future exploration. These limitations represent opportunities for further refinement.

References

Wenhu Chen, Xinyi Wang, and William Yang Wang. 2021. A dataset for answering time-sensitive questions. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*.

Zhuo Chen, Zhao Zhang, Zixuan Li, Fei Wang, Yutao Zeng, Xiaolong Jin, and Yongjun Xu. 2024. Self-improvement programming for temporal knowledge graph question answering. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 14579–14594, Torino, Italia. ELRA and ICCL.

Zheng Chu, Jingchang Chen, Qianglong Chen, Weijiang Yu, Haotian Wang, Ming Liu, and Bing Qin. 2024. TimeBench: A comprehensive evaluation of temporal reasoning abilities in large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1204–1228, Bangkok, Thailand. Association for Computational Linguistics.

DeepSeek-AI. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning.

Bhuwan Dhingra, Jeremy R. Cole, Julian Martin Eisenschlos, Daniel Gillick, Jacob Eisenstein, and William W. Cohen. 2021. Time-aware language models as temporal knowledge bases. *Transactions of the Association for Computational Linguistics*, 10:257–273.

Bahare Fatemi, Mehran Kazemi, Anton Tsitsulin, Karishma Malkan, Jinyeong Yim, John Palowitch, Sungyong Seo, Jonathan Halcrow, and Bryan Perozzi. 2024. Test of time: A benchmark for evaluating llms on temporal reasoning.

Zelalem Gero, Chandan Singh, Hao Cheng, Tristan Naumann, Michel Galley, Jianfeng Gao, and Hoifung Poon. 2023. Self-verification improves few-shot clinical information extraction. In *ICML 3rd Workshop on Interpretable Machine Learning in Healthcare (IMLH)*.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*.

Jie Huang and Kevin Chen-Chuan Chang. 2023. Towards reasoning in large language models: A survey. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 1049–1065, Toronto, Canada. Association for Computational Linguistics.

Wenke Huang, Guancheng Wan, Mang Ye, and Bo Du. 2023. Federated graph semantic and structural learning. In *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence*, IJCAI '23.

Shuyang Jiang, Yuhao Wang, and Yu Wang. 2023. Selfevolve: A code evolution framework via large language models.

Vasiliki Kougia, Anastasiia Sedova, Andreas Joseph Stephan, Klim Zaporojets, and Benjamin Roth. 2024. Analysing zero-shot temporal relation extraction on clinical notes using temporal consistency. In *Proceedings of the 23rd Workshop on Biomedical Natural Language Processing*, pages 72–84, Bangkok, Thailand. Association for Computational Linguistics.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS '20, Red Hook, NY, USA. Curran Associates Inc.

Xingxuan Li, Liying Cheng, Qingyu Tan, Hwee Tou Ng, Shafiq Joty, and Lidong Bing. 2023. Unlocking temporal question answering for large language models using code execution. *arXiv preprint arXiv:2305.15014*.

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. 2023. Self-refine: Iterative refinement with self-feedback. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Bonan Min, Hayley Ross, Elior Sulem, Amir Pouran Ben Veyseh, Thien Huu Nguyen, Oscar Sainz, Eneko Agirre, Ilana Heintz, and Dan Roth. 2023. Recent advances in natural language processing via large pre-trained language models: A survey. *ACM Comput. Surv.*, 56(2).

Niklas Muennighoff, Zitong Yang, Weijia Shi, Xiang Lisa Li, Li Fei-Fei, Hannaneh Hajishirzi, Luke Zettlemoyer, Percy Liang, Emmanuel Candès, and Tatsunori Hashimoto. 2025. s1: Simple test-time scaling.

Maxwell Nye, Anders Johan Andreassen, Guy Gur-Ari, Henryk Michalewski, Jacob Austin, David Bieber, David Dohan, Aitor Lewkowycz, Maarten Bosma, David Luan, et al. 2021. Show your work: Scratchpads for intermediate computation with language models. *arXiv preprint arXiv:2112.00114*.

Liangming Pan, Alon Albalak, Xinyi Wang, and William Wang. 2023. Logic-LM: Empowering large language models with symbolic solvers for faithful logical reasoning. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 3806–3824, Singapore. Association for Computational Linguistics.

Baolin Peng, Michel Galley, Pengcheng He, Hao Cheng, Yujia Xie, Yu Hu, Qiuyuan Huang, Lars Liden, Zhou Yu, Weizhu Chen, and Jianfeng Gao. 2023. Check your facts and try again: Improving large language models with external knowledge and automated feedback.

Xinying Qian, Ying Zhang, Yu Zhao, Baohang Zhou, Xuhui Sui, Li Zhang, and Kehui Song. 2024. TimeR^4 : Time-aware retrieval-augmented large language models for temporal knowledge graph question answering. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 6942–6952, Miami, Florida, USA. Association for Computational Linguistics.

Yifu Qiu, Zheng Zhao, Yftah Ziser, Anna Korhonen, Edoardo Ponti, and Shay Cohen. 2024. Are large language model temporally grounded? In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 7064–7083, Mexico City, Mexico. Association for Computational Linguistics.

Zhaochen Su, Jun Zhang, Tong Zhu, Xiaoye Qu, Juntao Li, Min zhang, and Yu Cheng. 2024. Timo: Towards better temporal reasoning for language models. In *First Conference on Language Modeling*.

Qingyu Tan, Hwee Tou Ng, and Lidong Bing. 2023. Towards benchmarking and improving the temporal reasoning capability of large language models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 14820–14835, Toronto, Canada. Association for Computational Linguistics.

Yixuan Tang and Yi Yang. 2024. Multihop-RAG: Benchmarking retrieval-augmented generation for multi-hop queries. In *First Conference on Language Modeling*.

Siddharth Vashishtha, Adam Poliak, Yash Kumar Lal, Benjamin Van Durme, and Aaron Steven White. 2020. Temporal reasoning in natural language inference. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4070–4078, Online. Association for Computational Linguistics.

Marc Verhagen, Robert Gaizauskas, Frank Schilder, Mark Hepple, Graham Katz, and James Pustejovsky. 2007. SemEval-2007 task 15: TempEval temporal relation identification. In *Proceedings of the Fourth International Workshop on Semantic Evaluations (SemEval-2007)*, pages 75–80, Prague, Czech Republic. Association for Computational Linguistics.

Leandro von Werra, Younes Belkada, Lewis Tunstall, Edward Beeching, Tristan Thrush, Nathan Lambert, Shengyi Huang, Kashif Rasul, and Quentin Gallouédec. 2020. Trl: Transformer reinforcement learning. `https://github.com/huggingface/trl`.

Haoyu Wang, Tao Li, Zhiwei Deng, Dan Roth, and Yang Li. 2024. Devil's advocate: Anticipatory reflection for LLM agents. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 966–978, Miami, Florida, USA. Association for Computational Linguistics.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2023. Self-consistency improves chain of thought reasoning in language models. In *The Eleventh International Conference on Learning Representations*.

Yuqing Wang and Yun Zhao. 2024. TRAM: Benchmarking temporal reasoning for large language models. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 6389–6415, Bangkok, Thailand. Association for Computational Linguistics.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, brian ichter, Fei Xia, Ed Chi, Quoc V Le, and Denny Zhou. 2022. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*, volume 35, pages 24824–24837. Curran Associates, Inc.

Yifan Wei, Yisong Su, Huanhuan Ma, Xiaoyan Yu, Fangyu Lei, Yuanzhe Zhang, Jun Zhao, and Kang Liu. 2023. Menatqa: A new dataset for testing the temporal comprehension and reasoning abilities of large language models. *arXiv preprint arXiv:2310.05157*.

Yixuan Weng, Minjun Zhu, Fei Xia, Bin Li, Shizhu He, Shengping Liu, Bin Sun, Kang Liu, and Jun Zhao. 2023. Large language models are better reasoners with self-verification. In *The 2023 Conference on Empirical Methods in Natural Language Processing*.

Siheng Xiong, Ali Payani, Ramana Kompella, and Faramarz Fekri. 2024. Large language models can learn temporal reasoning. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 10452–10470, Bangkok, Thailand. Association for Computational Linguistics.

Xinliang Frederick Zhang, Nicholas Beauchamp, and Lu Wang. 2024a. Narrative-of-thought: Improving temporal reasoning of large language models via recounted narratives. In *Findings of the Association* *for Computational Linguistics: EMNLP 2024*, pages 16507–16530, Miami, Florida, USA. Association for Computational Linguistics.

Zhihan Zhang, Tao Ge, Zhenwen Liang, Wenhao Yu, Dian Yu, Mengzhao Jia, Dong Yu, and Meng Jiang. 2024b. Learn beyond the answer: Training language models with reflection for mathematical reasoning. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 14720–14738, Miami, Florida, USA. Association for Computational Linguistics.

Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, Yifan Du, Chen Yang, Yushuo Chen, Zhipeng Chen, Jinhao Jiang, Ruiyang Ren, Yifan Li, Xinyu Tang, Zikang Liu, Peiyu Liu, Jian-Yun Nie, and Ji-Rong Wen. 2024. A survey of large language models.

Ben Zhou, Daniel Khashabi, Qiang Ning, and Dan Roth. 2019. “going on a vacation” takes longer than “going for a walk”: A study of temporal commonsense understanding. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3363–3369, Hong Kong, China. Association for Computational Linguistics.

Appendix

A Dataset Quality Statistics

Table 6 and Table 7 provides statistics on the quality of dataset generation with GPT-4o and DeepSeek V2.5 as generation models, respectively, focusing specifically on the correctness of the answers provided by a GPT-4o model for various temporal reasoning tasks. The 'Correctness' row shows the percentage of instances where the model-generated answer was correct, with very high accuracy across all datasets. Only instances for which the generated answers were correct were kept in the joint training dataset (TISER dataset), which contains a total of 62,648 training instances and 22,030 test instances. This filtering ensures that the training data used for fine-tuning the models are of high quality.

	TGQA	**TempReason (L2)**	**TempReason (L3)**	**TimeQA (easy)**	**TimeQA (hard)**	**TISER dataset**
Correctness (%)	91.8	99.6	99.5	98.5	98.1	97.5
Train instances	2790	16000	13000	14300	14700	60790
Test instances	3320	5400	4430	3000	3080	22030

Table 6: Statistics on dataset generation quality. The generator model is GPT-4o. For our joint training dataset, we keep exclusively the instances for which the final answer was correct.

	TGQA	**TempReason (L2)**	**TempReason (L3)**	**TimeQA (easy)**	**TimeQA (hard)**	**TISER dataset**
Correctness (%)	77.2	77.7	87.8	95.2	93.9	86.4
Train instances	2790	16000	13000	14300	14700	60790
Test instances	3320	5400	4430	3000	3080	22030

Table 7: Statistics on dataset generation quality when the generator model is DeepSeek V2.5. For our joint training dataset, we keep exclusively the instances for which the final answer was correct.

B Description and Examples of Benchmark Datasets

Below is a brief description of each dataset used to create the full TISER dataset, as well as the additional datasets used for out-of-domain performance evaluation. Example questions for in-domain and out-of-distribution questions are provided in Tables 8 and 9, respectively.

1. **Test-of-Time (ToT)**: A benchmark for evaluating LLMs on temporal reasoning. It is designed to assess the temporal reasoning capabilities of AI models on two main dimensions:
   (a) **ToT-semantic**: Measures the semantics and logic of time understanding.
   (b) **ToT-arithmetic**: Measures the ability to carry out time arithmetic operations.
2. **TGQA**: A synthetic dataset for temporal reasoning question-answering with ground-truth temporal graphs. It contains:
   - A temporal graph.
   - A story-based description of the temporal graph (generated by GPT-3.5).
   - A question based on the story.
   - The answer to the question.
3. **TempReason**: A temporal reasoning dataset for question answering over time events. In TempReason, the L2 questions require Time-Event relations, whereas L3 questions require Event-Event relations.
4. **TimeQA**: A benchmark for time-sensitive question answering. In TimeQA easy mode questions the query time expression is explicitly mentioned in the story. In the hard mode obtaining the answer needs inference based on the temporal relation between the query time expression and the one mentioned in the story.

5. **MultiHopRAG**: A QA dataset to evaluate retrieval and reasoning across documents with metadata in RAG pipelines. It contains 2556 queries, with evidence for each query distributed across 2 to 4 documents. The queries also involve document metadata, reflecting complex scenarios commonly found in real-world RAG applications.

Context	**Question**	**Dataset**
George Washington (February 22, 1732–December 14, 1799) was an American Founding Father, military officer, politician, and statesman who served as the first ...	What position did George Washington hold in June 1775?	TimeQA (easy)
George Washington (February 22, 1732–December 14, 1799) was an American Founding Father, military officer, politician, and statesman who served as the first ...	George Washington took which position before 1778?	TimeQA (hard)
Lionel Andrés "Leo" Messi (born 24 June 1987) is an Argentine professional footballer who plays as a forward for and captains both Major League Soccer club...	What team did Leo Messi play for in 2010?	TempReason (L2)
Lionel Andrés "Leo" Messi (born 24 June 1987) is an Argentine professional footballer who plays as a forward for and captains both Major League Soccer club...	What team did Leo Messi play for after Barcelona?	TempReason (L3)
["(Lucas Prescott was born in Northampton, Massachusetts) starts at 1908", "(Megan Peterson was born in Harmonyville, Florida) starts at 1921", "(Megan Peterson was married to Lucas Prescott) starts at 1946", "(Lucas Prescott was married to Megan Peterson) starts at 1946", "(Lucas Prescott died in Oceanview) starts at 1990", "(Megan Peterson was married to Lucas Prescott) ends at 1990", "(Lucas Prescott was married to Megan Peterson) ends at 1990", "(Megan Peterson died in Millwood Town, Austin) starts at 1997"]	Given the following five events: (Lucas Prescott was married to Megan Peterson), (Lucas Prescott was born in Northampton, Massachusetts), (Megan Peterson was married to Lucas Prescott), (Megan Peterson died in Millwood Town, Austin), (Megan Peterson was born in Harmonyville, Florida). Which event is the first one in chronological order?	TGQA

Table 8: Examples of in-domain questions from the TimeQA, TempReason, and TGQA datasets.

Context	Question	Dataset
Here is a set of temporal facts: E11 was the R17 of E69 from 1946 to 1950. E11 was the R11 of E69 from 1953 to 1958. E11 was the R63 of E69 from 1957 to 1958. E16 was the R63 of E11 from 1936 to 1946. E9 was the R17 of E11 from 1938 to 1941. E9 was the R11 of E11 from 1940 to 1942. E16 was the R88 of E11 from 1946 to 1959. E9 was the R17 of E11 from 1957 to 1961. E16 was the R96 of E11 from 1962 to 1963. E74 was the R88 of E92 from 1935 to 1937. E74 was the R7 of E92 from 1937 to 1943. E16 was the R96 of E42 from 1937 to 1939. E16 was the R11 of E42 from 1943 to 1948. E16 was the R96 of E42 from 1950 to 1958. E16 was the R11 of E42 from 1962 to 1965. E42 was the R35 of E33 from 1941 to 1942. E42 was the R90 of E33 from 1942 to 1943. E42 was the R11 of E33 from 1945 to 1954. E11 was the R11 of E74 from 1941 to 1946. E41 was the R53 of E74 from 1941 to 1946. E22 was the R85 of E25 from 1936 to 1949. E22 was the R35 of E25 from 1938 to 1940. E30 was the R7 of E25 from 1944 to 1949. E37 was the R85 of E25 from 1945 to 1949. E37 was the R53 of E25 from 1946 to 1947. E30 was the R35 of E25 from 1946 to 1953. E16 was the R88 of E25 from 1948 to 1951. E22 was the R63 of E25 from 1949 to 1953. E33 was the R11 of E25 from 1949 to 1959. E37 was the R7 of E25 from 1950 to 1953. E16 was the R85 of E25 from 1957 to 1962. E30 was the R35 of E25 from 1958 to 1964. E9 was the R53 of E25 from 1961 to 1964.	Immediately after E22, which entity was the R88 of E25?	Test-of-Time
The company did accomplish that goal, but in doing so, it inadvertently and profoundly changed how the internet looked. (2023-11-01T13:00:00+00:00). This whole line of argument is called 'market definition'. If the court decides the relevant market is phones and app stores, not specifically Android ones, then Google on the road to victory. (2023-11-05T11:00:00+00:00). The case, filed by Arkansas-based publisher Helena World Chronicle, argues that Google phone off news publisher content, their readers and ad revenue through anticompetitive means. (2023-12-15T17:56:02+00:00).	Was there no change in the portrayal of Google's influence on the digital ecosystem between the report from The Verge on Google's impact on the internet's appearance published on November 1, 2023, and the report from TechCrunch on a class action antitrust suit against Google published later?	MultiHopRAG

Table 9: Examples of out-of-distribution questions from the Test-of-Time (ToT) and MultiHopRAG datasets.

C Example of TISER Prompt

Example TISER Prompt and Question

You are an AI assistant that uses a Chain of Thought (CoT) approach with reflection to answer queries. Follow these steps:

1. Reason through the problem step by step within the `<reasoning>` tags.
2. Given your previous reasoning, identify relevant temporal events in the given context for answering the given question within `<timeline>` tags. Assume relations in the context are unidirectional.
3. Reflect on your reasoning and the timeline to check for any errors or improvements within the `<reflection>` tags.
4. Make any necessary adjustments based on your reflection. If there is additional reasoning required, go back to Step 1 (reason through the problem step-by-step), otherwise move to the next step (Step 5).
5. Provide your final, concise answer within the `<answer>` tags. If the answer is a number, just output the number, nothing else. Otherwise, output the entity or event without any additional comments.

Additional Instructions:

- **[IF during data generation]** Perform your reasoning knowing that the answer is: `X`.
- The `<reasoning>`, `<reflection>`, and `<timeline>` sections are for your internal reasoning process. All the reflection and the timeline have to be contained inside the thinking section.
- Do not use enumerations or lists when writing; use plain text such as paragraphs.
- The response to the query must be entirely contained within the `<answer>` tags.

Response Format:

```
<reasoning>
[Your step-by-step reasoning goes here. This is your internal thought process.]
<timeline> [Relevant temporal events for answering the given question.]</timeline>
<reflection>
[Your reflection on your reasoning, checking for errors or changes required.]
</reflection>
[Any adjustments to your thinking based on your reflection.]
</reasoning>
<answer> [Your final, concise answer to the query.]</answer>
```

Question: Which event happened in 1990?
Temporal Context: (Lucas Prescott was born in Northampton, Massachusetts) starts at 1908. (Megan Peterson was born in Harmonyville, Florida) starts at 1921. (Lucas Prescott was married to Megan Peterson) starts at 1946. (Lucas Prescott died in Oceanview) starts at 1990. (Lucas Prescott was married to Megan Peterson) ends at 1990. (Megan Peterson died in Millwood Town, Austin) starts at 1997.

D Example of Standard Prompt

Example Standard Prompt and Question

You are an AI assistant that has to respond to questions given a context.
Question: Which event happened in 1990?
Temporal Context: (Lucas Prescott was born in Northampton, Massachusetts) starts at 1908. (Megan Peterson was born in Harmonyville, Florida) starts at 1921. (Lucas Prescott was married to Megan Peterson) starts at 1946. (Lucas Prescott died in Oceanview) starts at 1990. (Lucas Prescott was married to Megan Peterson) ends at 1990. (Megan Peterson died in Millwood Town, Austin) starts at 1997.

E Example of correct DeepSeek and GPT4 reasoning

Example Correct DeepSeek V2.5 Reasoning

Question: Given the following five events: (Amy Johnson died in San Francisco), (Olivia Price was married to Amy Johnson), (Amy Johnson was born in Willowdale, Kansas), (Olivia Price was born in Harrisonburg), (Amy Johnson was married to Olivia Price). Which event is the second one in chronological order?
Expected Answer: Olivia Price was born in Harrisonburg
Temporal Context: (Amy Johnson was born in Willowdale, Kansas) starts at 1880. (Olivia Price was born in Harrisonburg) starts at 1890. (Amy Johnson was married to Olivia Price) starts at 1914. (Olivia Price was married to Amy Johnson) starts at 1914. (Amy Johnson died in San Francisco) starts at 1964. (Amy Johnson was married to Olivia Price) ends at 1964. (Olivia Price was married to Amy Johnson) ends at 1964. (Olivia Price died in Harrisonburg, Virginia) starts at 1971
Model Answer: <reasoning>To determine the second event in chronological order, we need to first identify the order of the events based on their starting points. The events provided are: 1. (Amy Johnson was born in Willowdale, Kansas) starts at 1880. 2. (Olivia Price was born in Harrisonburg) starts at 1890. 3. (Amy Johnson was married to Olivia Price) starts at 1914. 4. (Olivia Price was married to Amy Johnson) starts at 1914. 5. (Amy Johnson died in San Francisco) starts at 1964.Since the events (Amy Johnson was married to Olivia Price) and (Olivia Price was married to Amy Johnson) both start at the same time (1914), they are concurrent and can be considered as one event for the purpose of chronological ordering. The chronological order of the events based on their starting points is: 1. (Amy Johnson was born in Willowdale, Kansas) starts at 1880. 2. (Olivia Price was born in Harrisonburg) starts at 1890. 3. (Amy Johnson was married to Olivia Price) and (Olivia Price was married to Amy Johnson) start at 1914. 4. (Amy Johnson died in San Francisco) starts at 1964. The second event in this chronological order is (Olivia Price was born in Harrisonburg). <timeline>1. (Amy Johnson was born in Willowdale, Kansas) starts at 1880. 2. (Olivia Price was born in Harrisonburg) starts at 1890. 3. (Amy Johnson was married to Olivia Price) and (Olivia Price was married to Amy Johnson) start at 1914. 4. (Amy Johnson died in San Francisco) starts at 1964. </timeline><reflection>The reasoning correctly identifies the chronological order of the events based on their starting points. The timeline accurately reflects the sequence of events, and the second event in the timeline is indeed (Olivia Price was born in Harrisonburg). No errors or improvements are necessary.</reflection></reasoning><answer>Olivia Price was born in Harrisonburg</answer>

Example Correct GPT-4o Reasoning

Question: Given the following five events: (Amy Johnson died in San Francisco), (Olivia Price was married to Amy Johnson), (Amy Johnson was born in Willowdale, Kansas), (Olivia Price was born in Harrisonburg), (Amy Johnson was married to Olivia Price). Which event is the second one in chronological order?
Expected Answer: Olivia Price was born in Harrisonburg
Temporal Context: (Amy Johnson was born in Willowdale, Kansas) starts at 1880. (Olivia Price was born in Harrisonburg) starts at 1890. (Amy Johnson was married to Olivia Price) starts at 1914. (Olivia Price was married to Amy Johnson) starts at 1914. (Amy Johnson died in San Francisco) starts at 1964. (Amy Johnson was married to Olivia Price) ends at 1964. (Olivia Price was married to Amy Johnson) ends at 1964. (Olivia Price died in Harrisonburg, Virginia) starts at 1971
Model Answer: <reasoning>To determine the second event in chronological order, I need to first identify the sequence of all given events based on their start times. The events are: Amy Johnson was born in Willowdale, Kansas (1880), Olivia Price was born in Harrisonburg (1890), Amy Johnson was married to Olivia Price (1914), Olivia Price was married to Amy Johnson (1914), and Amy Johnson died in San Francisco (1964). The first event is Amy Johnson's birth in 1880. The second event, based on the start times, is Olivia Price's birth in 1890. The marriages occur simultaneously in 1914, and Amy Johnson's death is the last event in 1964.<timeline>1. Amy Johnson was born in Willowdale, Kansas (1880).2. Olivia Price was born in Harrisonburg (1890).3. Amy Johnson was married to Olivia Price (1914).4. Olivia Price was married to Amy Johnson (1914).5. Amy Johnson died in San Francisco (1964).</timeline><reflection>The reasoning correctly identifies the sequence of events based on their start times. The timeline confirms that Olivia Price's birth in 1890 is indeed the second event chronologically. There are no errors in the sequence, and the reflection confirms the accuracy of the reasoning.</reflection></reasoning><answer>Olivia Price was born in Harrisonburg</answer>

F Example of incorrect DeepSeek reasoning generation

Example Incorrect DeepSeek Reasoning

Question: Which team did Taylor Graham play for in May 2007?
Expected Answer: Seattle Sounders
Temporal Context: 2003 - 2004 : Taylor Graham's team is (Kansas City Wizards). 2006 - 2007 : Taylor Graham's team is (New York Red Bulls). 2007 - 2008 : Taylor Graham's team is (Seattle Sounders). 2009 - 2011 : Taylor Graham's team is (Seattle Sounders FC)
Model Answer: <reasoning>Taylor Graham's team affiliations are provided for different time periods. The question asks for the team he played for in May 2007. The relevant time period for this query is 2006 - 2007, where Taylor Graham's team is stated to be the New York Red Bulls. However, there is an additional time period provided for 2007 - 2008, where his team is the Seattle Sounders. This suggests a transition from the New York Red Bulls to the Seattle Sounders during the year 2007. Since the question specifies May 2007, it is necessary to determine whether the transition occurred before or after this month.<timeline>2006 - 2007: Taylor Graham's team is New York Red Bulls.2007 - 2008: Taylor Graham's team is Seattle Sounders.</timeline><reflection>The timeline indicates a change in teams during 2007. The question asks for the team in May 2007, which falls within the first half of the year. Given the provided context, it is reasonable to assume that the transition from the New York Red Bulls to the Seattle Sounders occurred sometime after May 2007. Therefore, Taylor Graham would still be playing for the New York Red Bulls in May 2007.</reflection><answer>New York Red Bulls</answer>

G Example of incorrect GPT-4o reasoning

Example Incorrect GPT-4o Reasoning

Question: Which team did Conor Sammon play for in Feb 2014?
Expected Answer: Ipswich Town
Temporal Context: 2005 - 2007 : Conor Sammon's team is (UCD). 2008 - 2011 : Conor Sammon's team is (Kilmarnock). 2011 - 2012 : Conor Sammon's team is (Wigan Athletic). 2012 - 2014 : Conor Sammon's team is (Derby County). 2014 - 2015 : Conor Sammon's team is (Ipswich Town). 2015 - 2016 : Conor Sammon's team is (Sheffield United)
Model Answer: <reasoning>To determine which team Conor Sammon played for in February 2014, I need to examine the timeline of his team associations. According to the temporal context provided, Conor Sammon was associated with Derby County from 2012 to 2014. The next team he played for was Ipswich Town, from 2014 to 2015. Since February 2014 falls within the year 2014, I need to determine whether it is before or after the transition from Derby County to Ipswich Town. The transition to Ipswich Town occurs in 2014, but without a specific month, it is reasonable to assume that the transition happened after February 2014. Therefore, in February 2014, Conor Sammon was still playing for Derby County.<timeline>Conor Sammon's team from 2012 to 2014 is Derby County. The transition to Ipswich Town occurs in 2014, but likely after February.</timeline><reflection>Upon reflection, the reasoning aligns with the temporal context provided. The transition to Ipswich Town is noted to occur in 2014, but without a specific month, it is logical to assume that in February 2014, he was still with Derby County. The initial assumption that Ipswich Town was the correct answer was incorrect based on the timeline analysis.</reflection>The reasoning and timeline analysis confirm that Conor Sammon was playing for Derby County in February 2014.</reasoning><answer>Derby County</answer>

H Example with Multiple Rounds of Reasoning with DeepSeek

Multiple Rounds of Reasoning with DeepSeek V2.5 Reasoning

<reasoning>To determine which event started first, I compare the two events: “Chris Evans was married to Oliver Jensen” and “Chris Evans was born in Bristol, Connecticut.” From the context, Chris Evans was married to Oliver Jensen starting in 1970. Chris Evans was born in Bristol, Connecticut starting in 1948. Comparing these two, the marriage starts in 1970, and the birth starts in 1948. However, because both events involve Chris Evans and have start times, I need to ensure I am not overlooking any subtle distinction, such as possibly overlapping or misaligned event names or entity relationships. Assuming all information is accurate and directly stated, I conclude that the marriage started first because it’s a significant relational milestone and possibly indicative of earlier events, whereas the birth date might not directly indicate involvement in a particular action.</reasoning><timeline>(Chris Evans was born in Bristol, Connecticut) starts at 1948. (Chris Evans was married to Oliver Jensen) starts at 1970.</timeline><reflection>Upon reviewing the timeline, I realize the conclusion is flawed. I initially suggested the marriage began first based on a false assumption about the importance or implication of events. The dates are explicitly clear: Chris Evans was born in 1948 and married in 1970. Birth precedes marriage in time. My earlier reasoning was incorrect in suggesting the marriage began first, it contradicts the temporal facts.</reflection><reasoning>With a corrected approach, I observe that Chris Evans’s birth is recorded as beginning in 1948 and the marriage to Oliver Jensen begins in 1970. The earlier of these two events is clearly the birth in 1948. The names or relational framing of the events do not affect their temporal ordering.</reasoning><timeline>(Chris Evans was born in Bristol, Connecticut) starts at 1948. (Chris Evans was married to Oliver Jensen) starts at 1970.</timeline><reflection>This reasoning is sound. The event starting in 1948 occurs earlier than the one beginning in 1970. There are no contradictions, and the timeline fully supports this answer.</reflection><answer>Chris Evans was born in Bristol, Connecticut</answer>

I Detailed fine-tuning process

In this appendix, we provide a detailed description of our fine-tuning process. Our approach employs a standard supervised fine-tuning (SFT) framework and Low-Rank Adaptation (LoRA) (Hu et al., 2022) to efficiently adapt the model to the structured output format defined by our TISER framework.

Our fine-tuning dataset is constructed according to the TISER framework (see Section 2.2). Each training sample consists of a question q, its gold-standard answer a, and a temporal context c. To enrich the dataset, we augment each sample with intermediate reasoning traces annotated using specific tags: `<reasoning>`, which contains a detailed, step-by-step CoT outlining the initial reasoning process; `<timeline>`, which provides an ordered summary of the temporal events extracted during reasoning; `<reflection>`, which serves as a self-reflective verification step that reviews and refines the initial reasoning; and finally, `<answer>`, which contains the final, concise response. These annotated outputs serve as the training targets, enabling the model to learn both the content and structure necessary for effective temporal reasoning.

We adapt our base model using LoRA (Hu et al., 2022), which is integrated into the attention layers of the pre-trained LLM so that only a small subset of parameters is updated during fine-tuning. Each training input is formatted to include the TISER tags, and the target output is the concatenation of the intermediate reasoning steps (i.e., `<reasoning>`, `<timeline>`, `<reflection>`) followed by the final answer (i.e., `<answer>`). We optimize the model using a standard cross-entropy loss between the generated output and the ground-truth sequence, updating only the LoRA parameters to efficiently adapt the model while preserving the majority of the pre-trained weights. Our fine-tuning is implemented using the Hugging Face TRL library (von Werra et al., 2020). Throughout the process, we monitor metrics such as validation loss, the accuracy of the final answer, and the coherence of the intermediate reasoning trace.

The overall training workflow proceeds in several stages. First, we generate the synthetic TISER dataset using off-the-shelf LLMs guided by our prompt framework. This dataset includes the detailed multi-stage reasoning traces that are central to our approach. Next, the base model is fine-tuned on this dataset using LoRA, during which the model learns to generate outputs that adhere strictly to the TISER format. The training objective reinforces the generation of a coherent chain-of-thought, an accurate timeline, and a reflective verification, culminating in the final answer. Finally, the correctness of the final answer is periodically evaluated on a held-out validation set. The best-performing checkpoint is saved.

J Token overhead analysis

To quantify the additional computational cost introduced by TISER, we measured the average token counts produced on our test set and compared them against standard prompting. Table 10 summarizes these results.

	Standard	**TISER**
Overall average	3.41	94.74
Reasoning stage	–	30.27
Timeline stage	–	23.62
Reflection stage	–	49.02

Table 10: Average token usage per response for standard versus TISER prompting.

While standard prompting yields only 3.41 tokens per response on average, TISER prompting generates substantially longer outputs, with 94.74 tokens on average, due to its multi-stage reasoning, timeline construction, and reflection steps. Breaking this down further, the reasoning, timeline, and reflection stages contribute approximately 30.27, 23.62, and 49.02 tokens, respectively. This increase in token usage corresponds to a significant improvement in task performance, boosting macro-average accuracy by +38.3% (e.g., Qwen2.5-7B before vs. after TISER fine-tuning).