

AdapTime: Enabling Adaptive Temporal Reasoning in Large Language Models

Yimin Deng^{1,2*}, Yejing Wang², Zhenxi Lin³, Zichuan Fu², Guoshuai Zhao^{1†}, Derong Xu², Yefeng Zheng⁴, Xiangyu Zhao^{2†}, Xian Wu^{3†}, Li Zhu^{1†}, Xueming Qian¹

¹Xi'an Jiaotong University, ²City University of Hong Kong,

³Tencent Jarvis Lab, ⁴Westlake University

dymanne@stu.xjtu.edu.cn, guoshuai.zhao@xjtu.edu.cn, xianzhao@cityu.edu.hk

kevinxwu@tencent.com, zhuli@xjtu.edu.cn

Abstract

Large language models have demonstrated strong reasoning capabilities in general knowledge question answering. However, their ability to handle temporal information remains limited. To address this limitation, existing approaches often involve external tools or manual verification and are tailored to specific scenarios, leading to poor generalizability. Moreover, these methods apply a fixed pipeline to all questions, overlooking the fact that different types of temporal questions require distinct reasoning strategies, which leads to unnecessary processing for simple cases and inadequate reasoning for complex ones. To this end, we propose AdapTime, an adaptive temporal reasoning method that dynamically executes reasoning steps based on the input context. Specifically, it involves three temporal reasoning actions: reformulate, rewrite and review, with an LLM planner guiding the reasoning process. AdapTime integrates seamlessly with state-of-the-art LLMs and significantly enhances their temporal reasoning capabilities without relying on external support. Extensive experiments demonstrate the effectiveness of our approach. The code is available at <https://github.com/Applied-Machine-Learning-Lab/ACL2026-AdapTime>.

1 Introduction

Recent years, large language models (LLMs) have demonstrated remarkable reasoning capabilities in general question answering (QA) tasks (Shao et al., 2023; Kamaloo et al., 2023; Dong et al., 2024a; Jia et al., 2025; Zhang et al., 2026). However, they still face significant challenges in handling temporal questions (Son and Oh, 2023; Xiong et al., 2024; Deng et al., 2025). Temporal reasoning tasks are concerned with understanding documents con-

Story: Terence Cooper (born 12 July 1944) is an English former football player and manager born in Brotherton, West Riding of Yorkshire. He was a full-back in the Leeds United team of the 1960s and 1970s. He worked in the England team of the 1970s and 1980s. ...

Question: Which position did Terence Cooper hold between **Mar 1966** and **Oct 1969**?

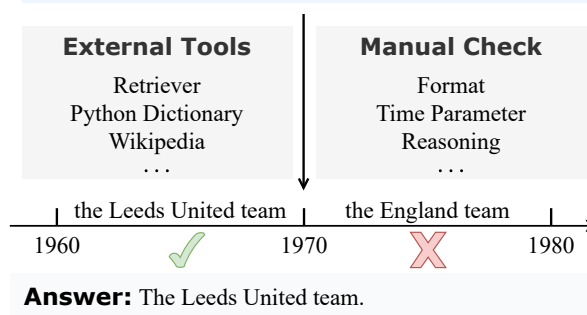


Figure 1: An example of temporal reasoning in question answering.

taining temporal information and answering time-sensitive questions. As shown in Figure 1, answering a question such as “Which position did Terence Cooper hold between Mar 1966 and Oct 1969?” requires both retrieving relevant facts and aligning them with the temporal context. In such cases, LLMs often generate incorrect answers due to a poor understanding of temporal expressions (e.g., “between Mar 1966 and Oct 1969”) or difficulty in identifying the order of events (Chu et al., 2023). Therefore, enhancing the temporal reasoning capabilities of LLMs remains a critical challenge.

Recent approaches typically focus on developing reasoning pipelines involve analyzing temporal questions, extracting timelines, and verifying generated answers (Xiong et al., 2024; Bazaga et al., 2025). While these reasoning strategies enhance the model’s ability to handle temporal information, they often rely on external support. As shown in Figure 1 and Table 1, one kind of approach makes use of external tools, such as applying retrievers to obtain compressed input, predefined Python dictionaries to filter time-sensitive expressions, or search

*Work was conducted at Tencent Jarvis Lab.

† Corresponding authors.

| | Reformulate | Rewrite | Review | Manual | External Tool |
|----------|-------------|---------|--------|--------|---------------|
| QAaP | × | ✓ | ✓ | ✓ | ✓ |
| Time-CoT | ✓ | ✓ | × | ✓ | × |
| Event-AL | × | ✓ | ✓ | × | ✓ |
| TG-LLM | × | ✓ | × | ✓ | × |
| TISER | × | ✓ | ✓ | ✓ | × |
| AdapTime | ✓ | ✓ | ✓ | × | × |

Table 1: The differences between our AdapTime and prior works.

engines to retrieve knowledge from sources like Wikipedia. For example, QAaP (Zhu et al., 2023) converts questions into code and selects the optimal answer from a set of candidate entities via code execution, relying on a execution engine and manually constructed Check and Match functions. Step-back (Zheng et al., 2024) combines retrieval-augmented generation with step-back prompting, requiring a retriever to obtain relevant facts. Event-AL (Wu et al., 2024) constructs a task-specific execution function and employs a Python-based solver to identify the answer. The other kind of approach involves manual correction during the reasoning process, including constructing timelines, verifying temporal parameters, or defining reasoning trajectories. Time-CoT (Yin and Hu, 2024) addresses temporal reasoning by explicitly identifying implicit temporal parameters, organizing standard question entities along a timeline, with manual supplementation of timeline information. TG-LLM (Xiong et al., 2024) transforms textual inputs into temporal graphs (TGs) through a pipeline that includes the manual extraction of hard examples, followed by reasoning over the TGs. TISER (Bazaga et al., 2025) utilizes these TGs instead of raw temporal documents as context. Although these designs can enhance temporal reasoning performance, they rely on external components that are often tailored to specific settings. Such reliance hinders their adaptability to diverse scenarios. To address this issue, we abstract temporal reasoning into three core actions (reformulate, rewrite, and review) that LLMs can autonomously perform.

Moreover, different types of temporal questions may require different reasoning steps. Simple temporal questions typically locate the answer directly from the text, while complex questions often require reasoning over multiple events and their sequential order, demanding more elaborate inference. To accommodate the varying levels of reasoning complexity, each of the aforementioned actions should be applied in different scenarios.

Therefore, enabling LLMs to select the appropri-

ate action and adapt the internal reasoning strategy within that action according to the scenario and question is a key challenge in temporal reasoning. To address this problem, we propose AdapTime, an adaptive temporal reasoning framework that dynamically selects and executes reasoning steps based on the input context and task requirements. AdapTime begins with the raw document/question and follows a multi-stage reasoning pipeline consisting of *reformulate*, *rewrite*, and *review*. Rather than executing all steps in sequence, the model leverages the planning ability of LLMs to dynamically determine which steps to perform. This process is guided by the semantic characteristics of the question, the structural features of the input context, and the model’s confidence in its intermediate reasoning. AdapTime adapts its reasoning strategy to each instance, enabling more accurate temporal understanding across diverse question types.

To summarize, our contributions are as follows:

- We propose AdapTime, a novel approach for adaptive temporal reasoning, which supports a multi-stage reasoning process consisting of *reformulate*, *rewrite*, and *review*.
- We enable LLMs to autonomously plan and control the reasoning process, without relying on handcrafted rules, annotations, or external tools.
- We validate the effectiveness of our approach through comprehensive experiments on two benchmark datasets under four temporal settings.

2 Related Work

2.1 LLM Reasoning

Large language models have made significant progress in reasoning with the ability to adapt to various downstream tasks (Wang et al., 2023a; Laban et al., 2023; Qiu et al., 2024; Gao et al., 2025b; Liu et al., 2025b). They can generalize to new tasks through few-shot in-context learning (Li et al., 2023; Dong et al., 2024b). Recent advances in chain-of-thought prompt (Wei et al., 2022) strategies further demonstrate its potential in handling complex reasoning (Gao et al., 2025a; Fu et al., 2025). By integrating multiple intermediate steps, existing methods effectively optimize the answer generation process and achieve improved reasoning performance (Yeo et al.; Wang et al., 2023b; Liu et al., 2025a; Wang et al., 2025). The inherent reasoning capability of LLMs has also been leveraged to enhance QA frameworks (Li et al., 2024; Stechly et al., 2024; Jia et al., 2024). How-

ever, previous methods often overlook the dynamic evolution of knowledge over time, and addressing temporal questions remains for further exploration.

2.2 Temporal Reasoning

In recent years, a series of approaches have been proposed that employ time-sensitive reasoning pipelines to answer temporal questions. QAaP (Zhu et al., 2023) represents the question as a Python dictionary with predefined keys: subject, relation, object, and time. It then extracts relevant events from the document and represents them using the same format, involving executing Python code and manually defining functions. Time-CoT (Yin and Hu, 2024) extracts standard question entities without temporal information (e.g., “Who was president of the US?”) from the question, and then searches the document for temporal parameters that match these standard entities. When the temporal parameters are implicit, they are manually matched. TG-LLM (Xiong et al., 2024) represents text as temporal graphs (TGs) in the form of “(John Thompson was born in Weston) starts at 1921”, consisting of a subject, relation, object, start/end, and timestamp. As LLMs may fail to produce accurate TGs, manual correction is required for some samples. Event-AL (Wu et al., 2024) extracts relation-centric temporal events from text in the form of (s, r, o, t) . It also requires invoking an external Python interpreter. These methods involve one or two of the three actions we define (reformulate, rewrite, and review), as shown in Table 1, and have improved the model’s temporal reasoning capabilities. However, the fixed reasoning pipeline and the reliance on handcrafted rules, external tools, or manual check restrict the model’s generalizability and adaptability across diverse temporal questions.

3 Methods

In this section, we first present the problem definition of temporal reasoning, then introduce each component of our proposed method, and conclude with the integration of these components into a complete reasoning process.

3.1 Problem Formulation

The temporal question answering (TQA) task aims to answer a time-constrained question Q with a specific time expression. The dataset is composed of tuples in the form $\{C, Q, A\}$, where C denotes a document containing a series of temporal facts,

Q represents temporal questions and A is the corresponding answer. The time expression in Q can be either explicit, such as in the question “Who was the president of the U.S. in 1971?”, or implicit, as in “Who was the U.S. president **before** Trump?”.

3.2 Overall Framework

In this section, we provide a brief overview of our framework, which performs temporal reasoning through a flexible combination of three core operations: reformulate, rewrite, and review, as illustrated in Figure 2. The model begins with the original document and the temporal question, and under the guidance of an LLM planner, it adaptively executes appropriate reasoning operations. First, based on the formulation of the question, the planner decides whether to decompose it into simple sub-questions. Then, if the temporal information in the context is not clear, the model transforms the relevant parts of the document into a structured format. Finally, when the model has low confidence in its answer, it extracts supporting sentences from the original document and verifies the answer. The final answer is generated after this adaptive multi-stage reasoning process.

3.3 Reasoning Action

In this section, we provide a detailed introduction to the three proposed actions: *reformulate*, *rewrite*, and *review*, and explain how each contributes to the reasoning process.

3.3.1 Reformulate

Temporal questions often involve complex time expression and multi-hop reasoning across events, making them difficult to answer directly. Therefore, we introduce a *reformulate* module that decomposes the question into a set of sub-questions, enabling more precise and interpretable reasoning.

We leverage the inherent reasoning capabilities of LLMs to perform question decomposition in a prompt-based manner. Formally, given an input question Q and a context document C , the Reformulate module prompts the LLM to produce a set of sub-questions $Q = \{q_1, q_2, \dots, q_n\}$, such that the answer to Q can be aggregated from the answers to each $q_i \in Q$. This decomposition is adaptive and varies according to the type of the question.

For instance, as shown in Figure 2, the model handles a question involving a temporal constraint over a person’s career history. Instead of answering directly, it first extracts relevant events (positions

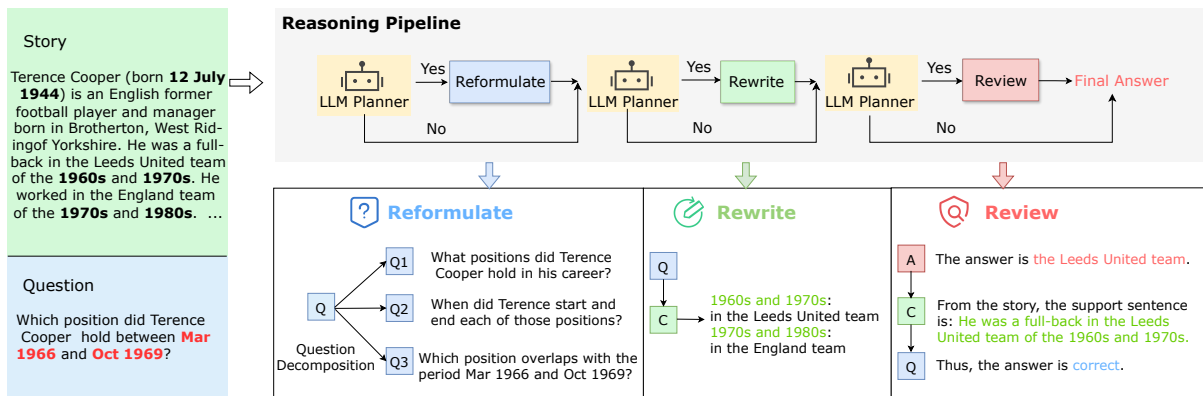


Figure 2: The overall architecture of our model.

held), then estimates their temporal spans, and finally infers the answer based on the specified time expression. This decomposition is induced by the model’s own understanding of temporal semantics, requiring no additional rule definitions or tools and thus allowing it to adapt to different settings.

The Reformulate module reduces ambiguity by isolating temporally relevant variables, allowing the model to better focus on the essential elements of the question. It also enables modular interaction with downstream components, such as retrieval and verification, which benefit from working with more focused and interpretable reasoning units. In our ablation studies, we observe that this module significantly boosts performance, especially on multi-hop and implicit temporal questions.

3.3.2 Rewrite

Natural language texts are usually diverse in format and often express temporal information implicitly through tense, adverbials, or discourse structure. Such implicitness makes it challenging to accurately identify event orders, durations, or overlaps.

To facilitate more accurate and consistent temporal reasoning, we introduce a *rewrite* module. By rewriting the text into an explicit temporal format, such as code, timeline, or temporal graphs, we make the temporal structure more accessible for reasoning. This transformation is performed by an LLM prompted to reorganize and rephrase the input into a time-anchored structure. For example, implicit expressions like “during his presidency” or “after the war ended” are rewritten into explicit temporal references grounded in calendar dates or relative sequences. This format allows downstream modules to directly operate on time-anchored information. By aligning events with specific time points, the model can better capture their temporal order, which supports more accurate reasoning

over temporal constraints.

In essence, the Rewrite module enables the system to handle complex temporal contexts with greater clarity and consistency. By making temporal relations explicit, the Rewrite module provides a normalized and interpretable representation of temporal information. This clarity supports downstream modules such as reasoning and answering components, enabling them to carry out temporal inference with greater effectiveness and robustness. Empirically, we find that this module significantly improves performance on questions requiring fine-grained temporal understanding, particularly in scenarios with multiple overlapping events.

3.3.3 Review

When generating time-sensitive answers, the model may incorrectly locate the answer within the context, especially in the presence of ambiguous or overlapping temporal information. Such mistakes can lead to factual inconsistencies, hallucinations, or misaligned reasoning steps. Therefore, verifying the generated answer is a crucial step in ensuring accuracy and consistency (Peng et al., 2025).

To address this, we introduce a *review* module that performs a thorough validation of the model’s answer by prompting it to:

- Retrieve supporting statements from the document that justify the predicted answer.
- Evaluate whether the supporting information is consistent with the question.
- Revise the answer if conflicts are detected.

This module is particularly useful in implicit temporal reasoning, where the answer depends on correctly understanding multiple time-sensitive facts. By explicitly asking the model to justify its output, we introduce a review mechanism that improves robustness and interpretability.

3.4 LLM Planner

Temporal questions differ significantly in their complexity and the type of reasoning they require. Using a fixed, uniform reasoning strategy for all questions is inadequate. To accommodate this variability, we introduce a Planner module that dynamically selects the appropriate reasoning strategy based on each question and context.

The Planner is implemented as an LLM prompted to perform step-by-step decision-making. Given the reasoning context C and the question Q , it first analyzes the temporal structure and reasoning demand, then decides which reasoning action is most suitable. Specifically, it decides whether to apply *reformulate*, when the question involves multiple implicit steps that can be broken down into simpler sub-questions; *rewrite*, when the temporal information in the document is implicit or scattered and needs to be transformed into a structured form; or *review*, when the model lacks confidence in its answer and needs to verify it. This decision process is carried out in natural language, where the model is instructed to justify its choice and, if applicable, generate the corresponding sub-steps for execution.

By explicitly planning the reasoning process, this module allows the system to adapt to the nature of each question and enhances reasoning ability by allowing the model to focus on relevant temporal operations. Overall, the Planner acts as a controller that orchestrates different reasoning actions, leading to robust temporal understanding.

3.5 Reasoning Pipeline

Based on the given temporal document and question, the model follows the strategies provided by the LLM planner to perform step-by-step reasoning and obtain the final answer. To better illustrate the reasoning process of our proposed approach, we present the pipeline in Algorithm 1. First, we initialize the answer variable (lines 1–2), then invoke the Planner to decide whether to decompose the question into sub-questions (lines 4–9). Next, the LLM planner determines whether to rewrite the context into a temporally structured form (lines 12–16). Then each sub-question is answered using the context C (lines 18–20), and the answers are aggregated (line 21). Finally, the model may optionally review and revise the answer (lines 23–25) before returning the final output (line 27).

Algorithm 1 Temporal QA with Adaptive Reasoning

Require: Temporal question Q , document context C

Ensure: Final answer A

```
1:  $A \leftarrow \text{null}$ 
2: // Step 1: Reformulate
3:  $d_1 \leftarrow \text{Planner}(Q, C, \text{step} = \text{Reformulate})$ 
4: if  $d_1 == \text{execute}$  then
5:    $Q_1, \dots, Q_n \leftarrow \text{Reformulate}(Q)$ 
6: else
7:    $Q_1 \leftarrow Q$ 
8:    $n \leftarrow 1$ 
9: end if
10: // Step 2: Rewrite
11:  $d_2 \leftarrow \text{Planner}(Q, C, \text{step} = \text{Rewrite})$ 
12: if  $d_2 == \text{execute}$  then
13:    $C \leftarrow \text{Rewrite}(C)$ 
14: else
15:    $C \leftarrow C$ 
16: end if
17: // Step 3: Answer
18: for  $i = 1$  to  $n$  do
19:    $A_i \leftarrow \text{Answer}(Q_i, C)$ 
20: end for
21:  $A \leftarrow \text{Aggregate}(A_1, \dots, A_n)$ 
22: // Step 4: Review
23:  $d_3 \leftarrow \text{Planner}(Q, C, A, \text{step} = \text{Review})$ 
24: if  $d_3 == \text{execute}$  then
25:    $A \leftarrow \text{Review}(A, Q, C)$ 
26: end if
27: return  $A$ 
```

4 Experiments

In this section, we conduct a series of experiments to evaluate the effectiveness of the proposed method and offer a detailed analysis.

4.1 Experimental Setup

4.1.1 Datasets

We evaluate our approach on two widely used temporal question answering datasets. TimeQA (Chen et al., 2021) is a human-annotated benchmark consisting of both easy and hard questions. Easy questions can be answered based on explicitly mentioned temporal expressions in the document, while hard questions require additional reasoning. TempReason (Tan et al., 2023) defines multiple levels of temporal understanding. We focus on the more challenging ones: aligning time expressions with

| Model | TimeQA | | | | TempReason | | | | Average [†] | |
|-----------------------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|---------------------------------|---------------------------------|
| | Easy-mode | | Hard-mode | | OBQA-L2 | | OBQA-L3 | | EM(%) | F1(%) |
| | EM(%) | F1(%) | EM(%) | F1(%) | EM(%) | F1(%) | EM(%) | F1(%) | | |
| T5-base [†] | 60.0 | 68.2 | 55.6 | 64.1 | 26.0 | 45.0 | 23.8 | 41.8 | 41.3 | 54.8 |
| T5-large [†] | 63.1 | 71.6 | 59.5 | 68.1 | 32.7 | 50.9 | 28.8 | 46.8 | 46.0 | 59.3 |
| REMEMO-base [†] | 61.4 | 70.4 | 58.2 | 67.3 | 33.6 | 51.6 | 28.5 | 44.9 | 45.4 | 58.6 |
| REMEMO-large [†] | 63.7 | 72.3 | 60.5 | 69.3 | 37.4 | 54.9 | 33.4 | 49.3 | 48.8 | 61.5 |
| GPT-4 [†] | 71.6 | 74.2 | 54.6 | 57.1 | 45.4 | 52.5 | 43.1 | 48.5 | 54.3 | 57.5 |
| QAaP [†] | 48.2 | 58.3 | 39.6 | 49.3 | - | - | - | - | - | - |
| TG-LLM [†] | 66.4 | 69.1 | 63.1 | 66.4 | 42.4 | 52.2 | 35.6 | 46.9 | 51.9 | 58.7 |
| LLaMA-3-8B-ICL | 1.1 | 3.1 | 1.7 | 3.6 | 3.8 | 10.0 | 1.8 | 10.5 | 2.1 | 6.8 |
| LLaMA-3-8B-CoT | 29.7 | 32.5 | 31.6 | 33.8 | 18.5 | 26.1 | 16.5 | 24.1 | 24.1 (+22.0 [†]) | 29.1 (+22.3 [†]) |
| LLaMA-3-8B-AdapTime* | 41.5 | 47.2 | 33.3 | 38.6 | 18.7 | 26.8 | 14.5 | 22.5 | 27.0 (+24.9[†]) | 33.8 (+27.0[†]) |
| Qwen-3-8B-ICL | 67.5 | 70.3 | 56.9 | 60.1 | 23.9 | 33.9 | 23.1 | 32.7 | 42.9 | 49.3 |
| Qwen-3-8B-CoT | 69.4 | 71.0 | 62.9 | 64.7 | 22.6 | 30.6 | 28.8 | 33.4 | 45.9 (+3.0 [†]) | 49.9 (+0.6 [†]) |
| Qwen-3-8B-AdapTime* | 72.7 | 74.1 | 66.5 | 68.2 | 29.1 | 37.9 | 28.8 | 33.8 | 49.3 (+6.4[†]) | 53.5 (+4.2[†]) |
| DeepSeek-V3-ICL | 80.8 | 82.9 | 68.8 | 71.6 | 45.1 | 50.8 | 43.6 | 48.3 | 59.6 | 63.4 |
| DeepSeek-V3-CoT | 85.3 | 86.7 | 75.6 | 77.0 | 44.8 | 49.1 | 47.0 | 50.4 | 63.2 (+3.6 [†]) | 65.8 (+2.4 [†]) |
| DeepSeek-V3-Step-back | 84.4 | 86.0 | 76.4 | 77.9 | 45.8 | 50.8 | 48.8 | 52.3 | 63.9 (+4.3 [†]) | 66.8 (+3.4 [†]) |
| DeepSeek-V3-Self-refinement | 77.6 | 80.4 | 76.4 | 78.4 | 44.3 | 47.2 | 41.1 | 42.3 | 60.1 (+0.5 [†]) | 62.3 (-1.1 [↓]) |
| DeepSeek-V3-AdapTime* | 85.4 | 86.6 | 77.7 | 79.2 | 48.0 | 52.1 | 49.8 | 53.2 | 65.1 (+5.5[†]) | 67.7 (+4.3[†]) |

Table 2: Main results using different models and strategies. We report exact match (EM) and token-level F1 scores. In line with previous work, we randomly sampled 1,000 examples under each task. Results with * are averaged over three random runs ($p < 0.05$ under t-test). The last two columns show average scores across four tasks, and their absolute improvements over the corresponding ICL baselines. Results with [†] are reported in the original papers.

events (L2) and reasoning about temporal relations between events (L3). To ensure a balanced distribution across different types of questions, we randomly sample 1,000 test instances for each type.

4.1.2 Evaluation Metrics

We use Exact Match (EM) and F1 score to evaluate model performance. EM measures the percentage of predictions that exactly match the ground truth, while F1 captures the token-level overlap between predictions and ground truth, computed as the harmonic mean of precision and recall.

4.1.3 Baselines

To better analyze our method and enable a broad comparison, we include two lines of recent temporal reasoning approaches as baselines. One line enhances temporal reasoning through fine-tuning, manual verification, or external tools. We introduce a set of embedding-based models (Tan et al., 2023; Yang et al., 2023), including T5-base, T5-large, REMEMO-base, and REMEMO-large, as well as a competitive LLM baseline, GPT-4. The results of T5-base and T5-large are based on vanilla

checkpoints without any continual pretraining on temporal data, using only standard supervised fine-tuning. (Tan et al., 2023; Yang et al., 2023) We also include QAaP (Zhu et al., 2023) and TG-LLM (Xiong et al., 2024), the previous state-of-the-art methods for temporal reasoning. The other explores the reasoning capabilities of LLMs using prompting-based techniques in-context learning (ICL) and chain-of-thought (CoT) (Wei et al., 2022). Step-back (Zheng et al., 2024) is a prompting technique that encourages LLMs to perform higher-level abstraction before answering. Self-Refinement (Madaan et al., 2023) is an agentic method that enables the model to reflect on its initial output and iteratively revise it.

4.1.4 Implementation Details

We adopt three competitive open-source LLMs as backbones: LLaMA-3.1-8B-Instruct, Qwen-3-8B, and DeepSeek-V3-0324. The reasoning process only relies on the inherent capabilities of the models, without incorporating any external tools, fine-tuning, or manual corrections. During inference,

we adopt a decoding strategy with top- $k = 10$ and temperature of 0.7. We set the batch size to 1 for single-instance evaluation and limit the output to a maximum of 512 new tokens. The prompt templates used are provided in Appendix A.

4.2 Main Results

The experimental results in Table 2 show the effectiveness of our method AdapTime. We have the following observations:

- AdapTime consistently outperforms both ICL and generalized CoT. ICL often fails to capture complex temporal dependencies, as it lacks explicit reasoning mechanisms, and CoT may generate overly generic reasoning paths that are not well-adapted to temporal questions. In contrast, AdapTime leverages structured decomposition to produce more targeted and accurate inference.
- Adaptime consistently improves performance across LLMs of different structures or scales, including LLaMA-3-8B, Qwen-3-8B, and DeepSeek-V3. This indicates that the method is model-agnostic and exhibits strong generalizability, making it applicable to a wide range of language models. Moreover, the gains come from our proposed framework rather than model-specific capabilities. It achieves better performance on DeepSeek-V3 compared to the previous state-of-the-art method. Qwen-3-8B equipped with AdapTime even surpass the larger closed-source model GPT-4 on TimeQA-Easy/Hard.
- AdapTime shows greater improvements on more challenging benchmarks, especially those requiring multi-hop or temporally complex reasoning, since it explicitly guides the model through intermediate reasoning steps. This highlights its strength in handling difficult question types that involve complex temporal structures.

4.3 Ablation Study

To assess the effectiveness of our method and each component, we conduct a detailed ablation study on DeepSeek-V3, as shown in Table 3.

First, we compare the baseline ICL (i.e., in-context learning) with three variants that incorporate our proposed reasoning actions: *reformulate*, *rewrite*, and *review*. Each action leads to consistent performance improvements. In particular, the rewrite step contributes the most to performance gains, especially in the TimeQA-Easy and OBQA-L2 datasets, highlighting the effectiveness of trans-

| Model | TimeQA | | | | TempReason | | | |
|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|
| | Easy-mode | | Hard-mode | | OBQA-L2 | | OBQA-L3 | |
| | EM | F1 | EM | F1 | EM | F1 | EM | F1 |
| ICL | 80.8 | 82.9 | 68.8 | 71.6 | 45.1 | 50.8 | 43.6 | 48.3 |
| w/ reform. | 84.3 | 85.8 | 77.0 | 78.9 | 46.7 | 50.8 | 49.3 | 51.9 |
| w/ rewrite | 86.4 | 87.6 | 76.0 | 77.8 | 46.9 | 52.1 | 48.5 | 51.9 |
| w/ review | 85.8 | 86.9 | 75.9 | 77.4 | 46.8 | 50.6 | 48.1 | 51.4 |
| w/o reform. | 85 | 86.6 | 76.8 | 78.5 | 46.9 | 51.8 | 48.9 | 51.8 |
| w/o rewrite | 84.8 | 86.5 | 76.5 | 77.9 | 46.1 | 49.8 | 47.6 | 50.7 |
| w/o review | 84.8 | 86.2 | 77.2 | 78.7 | 47.0 | 51.2 | 49.0 | 51.6 |
| w/o Planner | 85.3 | 86.7 | 77.1 | 78.8 | 47.1 | 51.3 | 48.9 | 52.0 |
| AdapTime | 85.4 | 86.6 | 77.7 | 79.2 | 48.0 | 52.1 | 49.8 | 53.2 |

Table 3: Ablation study on DeepSeek-V3. All metrics are reported as percentages.

forming text into a time-sensitive format.

Second, we removed one action at a time from AdapTime. The removal of any single component results in a slight performance drop, confirming that each action plays an important role in the reasoning process. Notably, removing the rewrite module leads to the most significant performance drop over these datasets, suggesting its critical role in aligning temporal context across tasks.

Finally, we eliminated the LLM planner and forced the model to follow a fixed sequence of reasoning steps. Performance drops consistently across all settings except TimeQA-Easy without the LLM planner. This indicates that adopting appropriate reasoning steps is crucial, as different tasks may require different reasoning processes.

4.4 Computational Cost Analysis

To quantify cost, we report the average total number of tokens (input + output) per instance across methods on the TempReason-L3 dataset in Table 4. Although AdapTime uses slightly more tokens due to the inclusion of reasoning instructions and intermediate steps, the increase is modest and much smaller than that of iterative or multi-call methods such as self-refinement. Moreover, since the temporal context is often lengthy, the majority of token cost comes from the input text, which constitutes a fixed overhead across methods.

4.5 Planner Comparison

To further investigate planner quality, we conducted an experiment where we sampled 1,000 high-quality planning trajectories from DeepSeek-V3 outputs on the training set, and used them to

| | Avg. Total Tokens |
|-----------------|-------------------|
| ICL | 4345.04 |
| CoT | 4584.17 |
| Self-refinement | > 10000 |
| AdapTime | 4873.64 |

Table 4: Computational Cost Analysis.

| Method | TimeQA-Easy | | TimeQA-Hard | |
|--------------------|-------------|-------------|-------------|-------------|
| | EM(%) | F1(%) | EM (%) | F1(%) |
| CoT | 29.7 | 32.5 | 31.6 | 33.8 |
| AdapTime-Finetuned | 31.0 | 40.5 | 23.4 | 32.0 |
| AdapTime(Ours) | 41.5 | 47.2 | 33.3 | 38.6 |

Table 5: Performance of different planner. We use LLaMA-3-8B as the backbone model.

fine-tune a LLaMA-3-8B model as a supervised planner. Surprisingly, the fine-tuned planner underperformed compared to our original prompt-based adaptive planner as shown in Table 5. These results suggest that fine-tuning a planner may introduce overfitting or poor generalization. In contrast, our prompt-based planner leverages the in-context reasoning capability of strong LLMs, offering better generalization and flexibility.

4.6 Rule Distribution

We hypothesize that different types of temporal questions require distinct reasoning steps. In our ablation studies, we have already demonstrated the effectiveness of using an LLM planner to define temporal reasoning strategy. To further assess the motivation of the action combination generated by the planner and to understand what types of steps it produces for different question types, we analyzed the action distributions across multiple datasets. The results in Figure 3 show that our method is capable of adaptively choosing reasoning actions:

- For questions with clear structure, the model performs decomposition to ensure that each sub-question is simple and unambiguous. For example, in TimeQA, questions often involve events at a specific time point. These can be decomposed into: {(1) What are the possible answers regardless of time? and (2) Which answer holds true at the given time?} In contrast, for less clearly structured questions, the model adopts a more conservative strategy: it first reconstructs the timeline to derive the answer, and then verifies the result. For instance, in TempReason, L2 and L3 questions require flexible multi-step inference over temporal spans and event order.

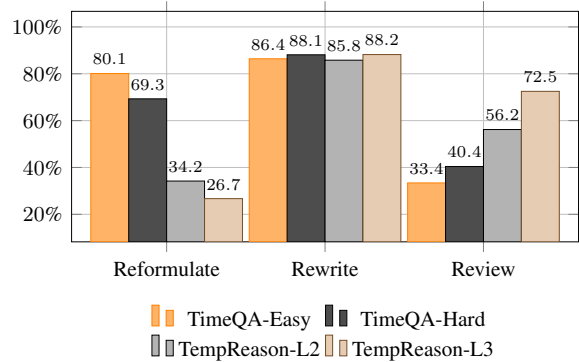


Figure 3: Comparison of the proportion of each operation in different datasets.

- In all settings, the rewrite action is frequently chosen. This confirms that rewriting natural language context into a time-sensitive format is essential for effective temporal reasoning.
- On more challenging datasets such as TempReason-L2/L3, the model tends to be less confident in its initial predictions and therefore prefers to select the review action as a form of self-verification.

Overall, the results of our action distribution analysis confirm that the LLM planner generates appropriate reasoning strategies tailored to different types of temporal questions, which in turn improves overall model performance.

4.7 Open-domain Temporal Reasoning

We further evaluate AdapTime on an open-domain setting using 1,000 questions from the ArchivalQA benchmark (Wang et al., 2022). As shown in Table 6, AdapTime significantly outperforms the baseline methods in both accuracy and F1.

Our method is modular and can be combined with retrievers such as BM25. While retrieval quality would affect planner decisions and downstream reasoning (Xu et al., 2025a,b), AdapTime’s adaptive mechanism remains effective in handling noisy or incomplete evidence through selective verification and timeline reconstruction. We applied BM25 on TempReason-L3, retrieving the most relevant sentences as context. We then applied AdapTime using these retrieved contexts. Results in 7 show that AdapTime consistently improves reasoning performance over retrieved content. This confirms that AdapTime is compatible with retrievers and can improve performance in retrieval-augmented, open-domain settings. It also shows that our method remains effective when operating

| | ACC | F1 |
|-----------------|-------------|-------------|
| Deepseek-V3 | 19.7 | 18.6 |
| Step-back | 29.5 | 27.8 |
| Self-refinement | 28.2 | 24.3 |
| AdapTime | 32.2 | 30.5 |

Table 6: Experimental results on the open-domain temporal QA benchmark ArchivalQA.

| | EM | F1 |
|---------------|-------------|-------------|
| BM25 | 44.7 | 47.7 |
| BM25+AdapTime | 46.7 | 49.9 |

Table 7: Performance with retriever integration on the TempReason-I3 dataset.

on retrieved, potentially noisy input, rather than clean gold contexts.

4.8 Case Study

We conducted a case study to analyze the reasoning trajectories generated by our method. Unlike standard prompting methods that often struggle with implicit temporal structures, AdapTime explicitly guides the model through intermediate reasoning stages, enhancing its ability to handle chronological dependencies and integrate commonsense knowledge with textual evidence. This highlights the model’s capacity for adaptive and interpretable reasoning, which is particularly valuable in temporally rich QA scenarios. A detailed example is provided in the Appendix C, illustrating how our model performs reasoning and demonstrating its superior ability to interpret temporal information compared to other methods.

5 Conclusion

In this paper, we propose a novel temporal reasoning approach AdapTime that enables LLMs to adaptively perform time-aware inference. AdapTime consistently outperforms strong baselines, exhibiting robust generalizability and significant gains on temporally complex tasks. Moreover, it can be seamlessly integrated with state-of-the-art LLMs. In the future, we plan to incorporate more temporal actions and planners to further enhance the model’s capability for robust temporal reasoning.

Limitations

AdapTime addresses the limitations of existing methods, which often rely on external support or rigid reasoning pipelines and struggle to generalize across diverse temporal tasks. However, relying

solely on LLMs as planners can be unstable in practice, leading to inconsistent planning outcomes across runs or tasks, as their behavior may exhibit a degree of randomness. Moreover, the current action set can be further expanded to include more fine-grained temporal operations or practical tools to improve precision and applicability. In future work, we plan to explore more temporal actions, integrate symbolic or external components, and develop techniques to improve the reliability and controllability of LLM-based planners.

Acknowledgments

This work is in part funded by the National Key Research and Development Program of China (2023YFC3321600); in part by National Natural Science Foundation of China (Grant No. 62372364) and the Technical Innovation Guidance Plan of Shaanxi Province, China (Grant No. 2024QCY-KXJ-199); in part by National Natural Science Foundation of China (No.62502404), Hong Kong Research Grants Council (Research Impact Fund No.R1015-23, Collaborative Research Fund No.C1043-24GF, General Research Fund No. 11218325), Institute of Digital Medicine of City University of Hong Kong (No.9229503), Huawei (Huawei Innovation Research Program), Tencent (Tencent Rhino-Bird Focused Research Program, Tencent University Cooperation Project), Didi (CCF-Didi Gaia Scholars Research Fund), Kuaishou (CCF-Kuaishou Large Model Explorer Fund No. 2025008, Kuaishou University Cooperation Project), and Bytedance.

References

- Adrián Bazaga, Rexhina Blloshmi, Bill Byrne, and Adrià de Gispert. 2025. Learning to reason over time: Timeline self-reflection for improved temporal reasoning in language models. *arXiv preprint arXiv:2504.05258*.
- Wenhu Chen, Xinyi Wang, and William Yang Wang. 2021. A dataset for answering time-sensitive questions. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks (NeurIPS Datasets and Benchmarks)*.
- Zheng Chu, Jingchang Chen, Qianglong Chen, Weijiang Yu, Haotian Wang, Ming Liu, and Bing Qin. 2023. Timebench: A comprehensive evaluation of temporal reasoning abilities in large language models. *arXiv preprint arXiv:2311.17667*.
- Yimin Deng, Yuxia Wu, Yejing Wang, Guoshuai Zhao, Li Zhu, Qidong Liu, Derong Xu, Zichuan Fu, Xian

- Wu, Yefeng Zheng, Xiangyu Zhao, and Xueming Qian. 2025. [A multi-expert structural-semantic hybrid framework for unveiling historical patterns in temporal knowledge graphs](#). In *Findings of the Association for Computational Linguistics, ACL 2025, Vienna, Austria, July 27 - August 1, 2025*, Findings of ACL, pages 20553–20565. Association for Computational Linguistics.
- Junnan Dong, Qinggang Zhang, Chuang Zhou, Hao Chen, Daochen Zha, and Xiao Huang. 2024a. Cost-efficient knowledge-based question answering with large language models. *Advances in Neural Information Processing Systems*, 37:115261–115281.
- Qingxiu Dong, Lei Li, Damai Dai, Ce Zheng, Jingyuan Ma, Rui Li, Heming Xia, Jingjing Xu, Zhiyong Wu, Baobao Chang, et al. 2024b. A survey on in-context learning. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 1107–1128.
- Zichuan Fu, Xiangyang Li, Chuhan Wu, Yichao Wang, Kuicai Dong, Xiangyu Zhao, Mengchen Zhao, Huifeng Guo, and Ruiming Tang. 2025. [A unified framework for multi-domain CTR prediction via large language models](#). *ACM Trans. Inf. Syst.*, 43(5):117:1–117:33.
- Jingtong Gao, Bo Chen, Xiangyu Zhao, Weiwen Liu, Xiangyang Li, Yichao Wang, Wanyu Wang, Huifeng Guo, and Ruiming Tang. 2025a. [Llm4rerank: Llm-based auto-reranking framework for recommendations](#). In *Proceedings of the ACM on Web Conference 2025, WWW 2025, Sydney, NSW, Australia, 28 April 2025- 2 May 2025*, pages 228–239. ACM.
- Jingtong Gao, Ling Pan, Yejing Wang, Rui Zhong, Chi Lu, Qingpeng Cai, Peng Jiang, and Xiangyu Zhao. 2025b. [Navigate the unknown: Enhancing LLM reasoning with intrinsic motivation guided exploration](#). *CoRR*, abs/2505.17621.
- Pengyue Jia, Yiding Liu, Xiangyu Zhao, Xiaopeng Li, Changying Hao, Shuaiqiang Wang, and Dawei Yin. 2024. [MILL: mutual verification with large language models for zero-shot query expansion](#). In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers), NAACL 2024, Mexico City, Mexico, June 16-21, 2024*, pages 2498–2518. Association for Computational Linguistics.
- Pengyue Jia, Derong Xu, Xiaopeng Li, Zhaocheng Du, Xiangyang Li, Yichao Wang, Yuhao Wang, Qidong Liu, Maolin Wang, Huifeng Guo, Ruiming Tang, and Xiangyu Zhao. 2025. [Bridging relevance and reasoning: Rationale distillation in retrieval-augmented generation](#). In *Findings of the Association for Computational Linguistics, ACL 2025, Vienna, Austria, July 27 - August 1, 2025*, Findings of ACL, pages 4242–4256. Association for Computational Linguistics.
- Ehsan Kamalloo, Nouha Dziri, Charles Clarke, and Davood Rafiei. 2023. Evaluating open-domain question answering in the era of large language models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5591–5606.
- Philippe Laban, Wojciech Kryściński, Divyansh Agarwal, Alexander Richard Fabbri, Caiming Xiong, Shafiq Joty, and Chien-Sheng Wu. 2023. Summedits: Measuring llm ability at factual reasoning through the lens of summarization. In *Proceedings of the 2023 conference on empirical methods in natural language processing*, pages 9662–9676.
- Tianle Li, Xueguang Ma, Alex Zhuang, Yu Gu, Yu Su, and Wenhui Chen. 2023. Few-shot in-context learning on knowledge base question answering. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6966–6980.
- Zhenyu Li, Sunqi Fan, Yu Gu, Xiuxing Li, Zhichao Duan, Bowen Dong, Ning Liu, and Jianyong Wang. 2024. [Flexkbqa: A flexible llm-powered framework for few-shot knowledge base question answering](#). In *Proceedings of the AAAI conference on artificial intelligence*, volume 38, pages 18608–18616.
- Qidong Liu, Xian Wu, Wanyu Wang, Yejing Wang, Yuanshao Zhu, Xiangyu Zhao, Feng Tian, and Yefeng Zheng. 2025a. [Llmemb: Large language model can be a good embedding generator for sequential recommendation](#). In *Thirty-Ninth AAAI Conference on Artificial Intelligence, Thirty-Seventh Conference on Innovative Applications of Artificial Intelligence, Fifteenth Symposium on Educational Advances in Artificial Intelligence, AAAI 2025, Philadelphia, PA, USA, February 25 - March 4, 2025*, pages 12183–12191. AAAI Press.
- Qidong Liu, Xiangyu Zhao, Yuhao Wang, Yejing Wang, Zijian Zhang, Yuqi Sun, Xiang Li, Maolin Wang, Pengyue Jia, Chong Chen, Wei Huang, and Feng Tian. 2025b. [Large language model enhanced recommender systems: Methods, applications and trends](#). In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining, V.2, KDD 2025, Toronto ON, Canada, August 3-7, 2025*, pages 6096–6106. ACM.
- Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegrefe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, et al. 2023. Self-refine: Iterative refinement with self-feedback. *Advances in Neural Information Processing Systems*, 36:46534–46594.
- Jingyu Peng, Maolin Wang, Xiangyu Zhao, Kai Zhang, Wanyu Wang, Pengyue Jia, Qidong Liu, Ruocheng Guo, and Qi Liu. 2025. [Stepwise reasoning disruption attack of llms](#). In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2025, Vienna, Austria, July 27 - August 1, 2025*, pages 5040–5058. Association for Computational Linguistics.

- Jielin Qiu, Andrea Madotto, Zhaojiang Lin, Paul A Crook, Yifan Xu, Babak Damavandi, Xin Luna Dong, Christos Faloutsos, Lei Li, and Seungwhan Moon. 2024. Snapntell: Enhancing entity-centric visual question answering with retrieval augmented multi-modal llm. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 247–266.
- Zhenwei Shao, Zhou Yu, Meng Wang, and Jun Yu. 2023. Prompting large language models with answer heuristics for knowledge-based visual question answering. In *Proceedings of the IEEE/CVF Conference on computer vision and pattern recognition*, pages 14974–14983.
- Jungbin Son and Alice Oh. 2023. Time-aware representation learning for time-sensitive question answering. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 70–77.
- Kaya Stechly, Karthik Valmeekam, and Subbarao Kambhampati. 2024. Chain of thoughtlessness? an analysis of cot in planning. *Advances in Neural Information Processing Systems*, 37:29106–29141.
- 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.
- Boshi Wang, Xiang Yue, and Huan Sun. 2023a. Can chatgpt defend its belief in truth? evaluating llm reasoning via debate. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 11865–11881.
- Hanbing Wang, Xiaorui Liu, Wenqi Fan, Xiangyu Zhao, Venkataramana Kini, Devendra Pratap Yadav, Fei Wang, Zhen Wen, and Hui Liu. 2025. Rethinking large language model architectures for sequential recommendations. In *Proceedings of the 14th International Joint Conference on Natural Language Processing and the 4th Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics, IJCNLP-AAACL 2025, Mumbai, India, December 20-24, 2025*, pages 3376–3391. The Asian Federation of Natural Language Processing and The Association for Computational Linguistics.
- Jiexin Wang, Adam Jatowt, and Masatoshi Yoshikawa. 2022. Archivalqa: A large-scale benchmark dataset for open-domain question answering over historical news collections. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 3025–3035.
- Yuhao Wang, Xiangyu Zhao, Bo Chen, Qidong Liu, Huifeng Guo, Huanshuo Liu, Yichao Wang, Rui Zhang, and Ruiming Tang. 2023b. PLATE: A prompt-enhanced paradigm for multi-scenario recommendations. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2023, Taipei, Taiwan, July 23-27, 2023*, pages 1498–1507. ACM.
- Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837.
- Shaojuan Wu, Jitong Li, Xiaowang Zhang, and Zhiyong Feng. 2024. An event-based abductive learning for hard time-sensitive question answering. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 1105–1115.
- Siheng Xiong, Ali Payani, Ramana Kompella, and Faramarz Fekri. 2024. Large language models can learn temporal reasoning. *arXiv preprint arXiv:2401.06853*.
- Derong Xu, Pengyue Jia, Xiaopeng Li, Yingyi Zhang, Maolin Wang, Qidong Liu, Xiangyu Zhao, Yichao Wang, Huifeng Guo, Ruiming Tang, Enhong Chen, and Tong Xu. 2025a. Align-grag: Reasoning-guided dual alignment for graph retrieval-augmented generation. *CoRR*, abs/2505.16237.
- Derong Xu, Xinhang Li, Ziheng Zhang, Zhenxi Lin, Zhihong Zhu, Zhi Zheng, Xian Wu, Xiangyu Zhao, Tong Xu, and Enhong Chen. 2025b. Harnessing large language models for knowledge graph question answering via adaptive multi-aspect retrieval-augmentation. In *Thirty-Ninth AAAI Conference on Artificial Intelligence, Thirty-Seventh Conference on Innovative Applications of Artificial Intelligence, Fifteenth Symposium on Educational Advances in Artificial Intelligence, AAAI 2025, Philadelphia, PA, USA, February 25 - March 4, 2025*, pages 25570–25578. AAAI Press.
- Sen Yang, Xin Li, Lidong Bing, and Wai Lam. 2023. Once upon a time in graph: Relative-time pretraining for complex temporal reasoning. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 11879–11895.
- Edward Yeo, Yuxuan Tong, Xinyao Niu, Graham Neubig, and Xiang Yue. Demystifying long chain-of-thought reasoning in llms. In *ICLR 2025 Workshop on Deep Generative Model in Machine Learning: Theory, Principle and Efficacy*.
- Baosheng Yin and Naiyu Hu. 2024. Time-cot for enhancing time reasoning factual question answering in large language models. In *2024 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE.
- Yingyi Zhang, Junyi Li, Wenlin Zhang, Pengyue Jia, Xianneng Li, Yichao Wang, Derong Xu, Yi Wen, Huifeng Guo, Yong Liu, and Xiangyu Zhao. 2026. Evoking user memory: Personalizing LLM via recollection-familiarity adaptive retrieval. In *The*

Fourteenth International Conference on Learning Representations.

Huaixiu Steven Zheng, Swaroop Mishra, Xinyun Chen, Heng-Tze Cheng, Ed H Chi, Quoc V Le, and Denny Zhou. 2024. Take a step back: Evoking reasoning via abstraction in large language models. In *The Twelfth International Conference on Learning Representations*.

Xinyu Zhu, Cheng Yang, Bei Chen, Siheng Li, Jian-Guang Lou, and Yujiu Yang. 2023. Question answering as programming for solving time-sensitive questions. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 12775–12790.

A Prompt

In this section, we present the prompt templates used for each type of action in our AdapTime framework. As described in the methods section, AdapTime decomposes the reasoning process into three structured operations: Reformulate, Rewrite, and Review. To guide LLMs to perform these operations in a controlled and interpretable manner, we design simple yet effective prompt templates for each action type. These templates are instantiated dynamically during multi-step reasoning, depending on the needs of each case.

The Reformulate action aims to simplify the original complex question into a set of easier, focused sub-questions that are easier for the model to answer individually. This step encourages the model to disentangle temporal or logical dependencies embedded in the original query, facilitating more accurate downstream reasoning.

Reformulate Template

Break the question QUESTION down into several simple sub-questions and answer each sub-question. Then return the final answer.

The Rewrite action prompts the model to explicitly construct a timeline from the narrative context relevant to the question. This is particularly useful for questions that require temporal grounding, such as identifying when an event happened, or what happened at a particular point in time. By asking the model to generate a timeline and align it with the question, we encourage temporal abstraction and normalization of narrative information.

Rewrite Template

In the context of STORY and the QUESTION, generate timeline for what the question concerns and answer each sub-question. Then return the final answer.

The Review action serves as a verification step. After an initial answer is generated, the model is asked to re-examine the original context and identify support sentences to justify the answer. If evidence is lacking or inconsistent, the model is encouraged to reconsider and revise its answer, enhancing factual consistency and robustness. This step mimics human self-checking behavior and helps reduce hallucinated or unsupported answers.

| Method | TempReason-L2 | | TempReason-L3 | |
|-------------------|---------------|-------|---------------|-------|
| | EM(%) | F1(%) | EM (%) | F1(%) |
| AdapTime-Original | 50.9 | 55.2 | 52.5 | 55.1 |
| AdapTime-Variant1 | 50.3 | 54.9 | 52.8 | 56.0 |
| AdapTime-Variant2 | 50.6 | 55.2 | 52.1 | 55.5 |
| AdapTime-Variant3 | 50.2 | 54.6 | 53.3 | 56.5 |

Table 8: Performance of different prompt variants on TempReason-L2 and TempReason-L3.

Review Template

In the context of STORY and the QUESTION, after obtain the answer, given the support sentences in original story and check if the answer is correct. If yes, return the answer again. If not, think again and return the right answer.

We use a unified LLM prompt to guide the planner’s decision-making process. The prompt is as follows.

AdapTime Template

Let’s think step by step. First, if the question is complex, break the question down into several simple sub-questions. Then generate timeline for what the question concerns and answer each sub-question if the story’s timeline is unclear. Then return the final answer. After obtaining the answer, if you are not sure, look for support sentences in the original story and check if the answer is correct. If yes, return the answer again. If not, think again and return the right answer.

B Prompt Variation Experiments

To further verify the robustness of our proposed method, we conducted prompt variation experiments using the latest DeepSeek-V3.2-exp model on TempReason-L2 and TempReason-L3. We tested several semantically equivalent variations of the prompt, changing only the wording and structure while keeping the intended logic unchanged. Results in Table 8 show that reasoning performance remains stable across these variants, confirming that our method is not sensitive to specific prompt phrasing, and that the adaptive reasoning mechanism is the key to its effectiveness.

C Case Study

Table 9 and 10 present a representative example illustrating how our proposed framework, AdapTime, improves temporal reasoning compared to the traditional Chain-of-Thought (CoT) approach. The task involves identifying where Mikhail Lomonosov was educated in January

1736, given a long narrative containing multiple temporally-anchored events and institutions.

The traditional CoT method attempts to construct a linear timeline by extracting and interpreting key events. While it correctly identifies that Lomonosov had completed his studies at the Slavic Greek Latin Academy and briefly attended the Kyiv-Mohyla Academy in 1735, it struggles to pinpoint his exact status in January 1736. Due to ambiguity around the transition period before his scholarship to the St. Petersburg Academy, the model ultimately predicts "Unknown", failing to commit to a specific answer.

In contrast, AdapTime applies a structured multi-step reasoning strategy: Reformulate: It decomposes the original question into focused sub-questions, clarifying pre- and post-January 1736 educational phases. Timeline Construction: It builds an explicit timeline of Lomonosov's academic journey, aligning events with their corresponding years. Answer Selection: Based on the timeline, AdapTime identifies that by January 1736, Lomonosov had already completed his studies in Moscow and was transitioning to the Academic University at the St. Petersburg Academy of Sciences. Verification: It cross-verifies the answer with textual evidence, confirming that the scholarship was already granted in 1736 and that he was affiliated with the St. Petersburg Academy at that time. This structured process enables AdapTime to resolve temporal ambiguities and ground its reasoning in anchored timelines, leading to a correct and explainable answer, while the traditional CoT remains uncertain. This case demonstrates how AdapTime significantly enhances the model's ability to reason over complex temporal contexts, especially when precise date alignment is required.

| | |
|--|---|
| <p>Story</p> <p>Question</p> <p>Options</p> | <p><i>Mikhail Lomonosov was a Russian polymath, scientist and writer, who made important contributions to literature, education, and science. Among his discoveries were the atmosphere of Venus and the law of conservation of mass in chemical reactions. . . . In five years Lomonosov completed a twelve-year study course and in 1736, among 12 best graduates, was awarded a scholarship at the St. Petersburg Academy. He plunged into his studies and was rewarded with a four-year grant to study abroad, in Germany, first at the University of Marburg and then in Freiberg. The University of Marburg was among Europe’s most important universities in the mid-18th century due to the presence of the philosopher Christian Wolff, a prominent figure of the German Enlightenment. Lomonosov became one of Wolff’s students while at Marburg from November 1736 to July 1739. Both philosophically and as a science administrator, this connection would be the most influential of Lomonosov’s life. . . .</i></p> <p><i>Where was Mikhail Vassilyevich Lomonosov educated in Jan, 1736?</i></p> <p><i>“Slavic Greek Latin Academy”, “Kyiv-Mohyla Academy”, “Kyiv-Mohyla Academy”, “Academic University at the St. Petersburg Academy of Sciences”, “Slavic Greek Latin Academy”, “Unknown”</i></p> |
| <p>Step 1: Reformulate</p> <p>Step 2: Timeline</p> <p>Step 3: Answer</p> <p>Step 4: Verification</p> | <p>Sub-question 1: Where was Lomonosov educated before January 1736? - Lomonosov studied at the Slavic Greek Latin Academy in Moscow from 1730 to 1735. - In 1735, he was sent to the Kyiv-Mohyla Academy for a short period but quickly returned to Moscow. 2. Sub-question 2: Where was Lomonosov in January 1736? - In 1736, Lomonosov was among the 12 best graduates of the Slavic Greek Latin Academy and was awarded a scholarship at the St. Petersburg Academy. - By January 1736, he had already completed his studies in Moscow and was preparing to move to St. Petersburg.</p> <p>Timeline: - 1730–1735: Slavic Greek Latin Academy (Moscow). - 1735 (briefly): Kyiv-Mohyla Academy (Kiev). - 1736: Awarded scholarship to the Academic University at the St. Petersburg Academy of Sciences.</p> <p>Academic University at the St. Petersburg Academy of Sciences.</p> <p>The text states: In five years Lomonosov completed a twelve-year study course and in 1736, among 12 best graduates, was awarded a scholarship at the St. Petersburg Academy. - This confirms that by 1736, he was no longer at the Kyiv-Mohyla Academy or the Slavic Greek Latin Academy but was affiliated with the St. Petersburg Academy. Thus, the correct answer is (Academic University at the St. Petersburg Academy of Sciences).</p> |

Table 9: Case study demonstrating temporal reasoning using the AdapTime framework.

| | |
|-------------------------------|--|
| <p>Traditional CoT</p> | <p>To determine where Mikhail Lomonosov was educated in January 1736, let's analyze the timeline of his education:1. 1724–1730: Lomonosov lived in his village, Mishaninskaya, and pursued basic education locally. 2.1730: He left for Moscow and was admitted to the **Slavic Greek Latin Academy**, where he studied for several years. 3. 1735: After excelling at the Slavic Greek Latin Academy, he was sent to the Kyiv-Mohyla Academy for a short period but quickly returned to Moscow due to dissatisfaction. 4. 1736: By this time, Lomonosov had completed his studies at the Slavic Greek Latin Academy and was among the top graduates. In 1736, he was awarded a scholarship to the Academic University at the St. Petersburg Academy of Sciences, marking the next phase of his education. Since the question specifies January 1736, this falls after his time at the Slavic Greek Latin Academy and Kyiv-Mohyla Academy but before his enrollment at the Academic University in St. Petersburg. However, the text does not explicitly mention his whereabouts in January 1736, though it is likely he was still in Moscow transitioning to St. Petersburg. Given the options: - Slavic Greek Latin Academy: He had already completed his studies there by 1736. Kyiv-Mohyla Academy: He attended this briefly in 1735 but had returned to Moscow by 1736. Academic University at the St. Petersburg Academy of Sciences: His enrollment here began later in 1736, not January. Thus, the most accurate answer is (Unknown).</p> |
|-------------------------------|--|

Table 10: Case study demonstrating temporal reasoning using the traditional CoT.