SPAN: Benchmarking and Improving Cross-Calendar Temporal Reasoning of Large Language Models

Zhongjian Miao[1], **Hao Fu**[1*†], **Chen Wei**[1]

[1]Li Auto Inc.
{miaozhongjian, chenwei10}@lixiang.com

Abstract

Temporal reasoning is a fundamental capability for large language models (LLMs) to understand real-world dynamics. Existing research on temporal reasoning has predominantly focused on the Gregorian calendar. However, as many countries and regions concurrently adopt multiple calendar systems, temporal reasoning across calendars becomes crucial for LLMs in global and multicultural contexts. Unfortunately, cross-calendar temporal reasoning remains underexplored, with no dedicated benchmark available to evaluate this capability. To bridge this gap, we introduce **SPAN**, a cro**S**s-calendar tem**P**oral re**A**soning be**N**chmark, which requires LLMs to perform intra-calendar temporal reasoning and inter-calendar temporal conversion. SPAN features ten cross-calendar temporal reasoning directions, two reasoning types, and two question formats across six calendars. To enable time-variant and contamination-free evaluation, we propose a template-driven protocol for dynamic instance generation that enables assessment on a user-specified Gregorian date. We conduct extensive experiments on both open- and closed-source state-of-the-art (SOTA) LLMs over a range of dates spanning 100 years from 1960 to 2060. Our evaluations show that these LLMs achieve an average accuracy of only 34.5%, with none exceeding 80%, indicating that this task remains challenging. Through in-depth analysis of reasoning types, question formats, and temporal reasoning directions, we identify two key obstacles for LLMs: *Future-Date Degradation* and *Calendar Asymmetry Bias*. To strengthen LLMs' cross-calendar temporal reasoning capability, we further develop an LLM-powered `Time Agent` that leverages tool-augmented code generation. Empirical results show that `Time Agent` achieves an average accuracy of 95.31%, outperforming several competitive baselines, highlighting the potential of tool-augmented code generation to advance cross-calendar temporal reasoning. We hope this work will inspire further efforts toward more temporally and culturally adaptive LLMs.

Code, dataset, and extended version —
https://github.com/miaozhongjian/span.git

Introduction

Recent advances in large language models (LLMs) have led to substantial progress across a range of reasoning tasks (Qiao et al. 2023; Xu et al. 2024; Li et al. 2024; Fan et al. 2025). Among these, temporal reasoning, as a fundamental capability for LLMs to understand real-world dynamics, has garnered increasing attention, leading to the development of multiple evaluation benchmarks (Chen, Wang, and Wang 2021; Liska et al. 2022; Kasai et al. 2023; Tan, Ng, and Bing 2023a; Wei et al. 2023; Tan, Ng, and Bing 2023b; Fatemi et al. 2024; Wang and Zhao 2024a; Chu et al. 2024; Wang and Zhao 2024b; Ge et al. 2025; Wei et al. 2025; Saxena, Gema, and Minervini 2025a). Nevertheless, most existing efforts focus on temporal reasoning within the Gregorian calendar, neglecting the exploration of LLMs' capability to reason across calendar systems. In practice, numerous countries and regions employ an alternative calendar system alongside the widely-used Gregorian calendar system (Taqizadeh 1939; Graumann 2015). For instance, both Saudi Arabia and China use the Gregorian calendar for official purposes, but Saudi Arabia follows the Islamic calendar for religious observances while China relies on the Chinese lunar calendar for traditional cultural activities. These real-world practices underscore the necessity for reasoning and converting dates across calendar systems, *i.e.,* cross-calendar temporal reasoning, which is crucial for global and multicultural applications of LLMs.

Despite its practical significance, the cross-calendar temporal reasoning capability of LLMs remains unexplored. Critically, there is currently no benchmark for assessing this capability, which hinders further research progress. Moreover, existing temporal reasoning benchmarks are inherently time-invariant, limiting evaluation to a single, fixed reference date when the benchmark was created. This leads to two drawbacks: (1) ***Temporal Scope Limitation.*** Given the dynamic nature of time, robust temporal reasoning evaluation should ensure a broad temporal scope by enabling assessments at arbitrary reference dates, which is not yet supported by existing benchmarks. (2) ***Data Contamination.*** The ground truth in certain temporal reasoning benchmarks is typically deterministic and publicly available through platforms such as Hugging Face and GitHub. Such accessibility may lead to evaluation data being absorbed into training corpora, potentially biasing evaluations (Sainz et al. 2023; Deng et al. 2024; Balloccu et al. 2024).

In response to these challenges, we introduce **SPAN**, a cro**S**s-calendar tem**P**oral re**A**soning be**N**chmark. Unlike prior

* Corresponding Author.
† Work done when Hao Fu was at Li Auto.

Gregorian-centric temporal reasoning benchmarks, SPAN requires LLMs to perform temporal reasoning within one calendar based on a reference date, and then convert the result to its equivalent in another calendar. To mitigate data contamination and time-invariant problems, we propose a novel evaluation protocol that dynamically generates instances by instantiating a set of question–code template pairs. By utilizing these carefully-designed template pairs, SPAN generates diverse evaluation instances covering three key dimensions: (1) ***Temporal Reasoning Directions***. SPAN supports ten cross-calendar reasoning directions between the Gregorian and five other calendars: Chinese Lunar, Shaka, Hebrew, Islamic, and Persian. (2) ***Reasoning Types***. SPAN encompasses date- and festival-based reasoning tasks that perform cross-calendar conversions for general dates and festival dates, respectively. (3) ***Question Formats***. SPAN comprises both polar and content questions, where the answers to polar questions are binary judgments (*i.e.,* "Yes" or "No"), whereas content questions require calendar-specific dates as answers. During evaluation, these question-code template pairs are instantiated to generate questions and corresponding code snippets, which are then executed to automatically derive the answers. Overall, SPAN offers a dynamic mechanism to generate varied evaluation instances, enabling robust assessment of LLMs' cross-calendar temporal reasoning.

We conduct comprehensive experiments on both open- and closed-source state-of-the-art LLMs over a range of dates spanning a 100-year period from 1960 to 2060. We demonstrate that these LLMs achieve an average accuracy of only 34.5% across the evaluation dates, with none exceeding 80% accuracy. We also investigate the impact of reasoning types, question formats, and temporal reasoning directions, identifying two key obstacles for LLMs: *Future-Date Degradation*, where LLMs struggle with cross-calendar temporal reasoning under future reference dates, and *Calendar Asymmetry Bias*, an asymmetry where reasoning from the Gregorian calendar is more accurate than the reverse. To advance LLMs' cross-calendar temporal reasoning capabilities, we develop an LLM-powered `Time Agent` with tool-augmented code generation. Empirical results demonstrate that our method achieves an average accuracy of 95.31%, surpassing competitive baselines. Our contributions are summarized as follows:

- We introduce **SPAN**, a benchmark assessing LLMs' cross-calendar temporal reasoning across ten directions, two reasoning types, and two question formats.
- We propose a template-driven protocol for dynamic instance generation, enabling broad temporal evaluation while mitigating data contamination.
- We conduct comprehensive experiments and in-depth analyses, revealing that LLMs struggle to reason across calendars and identifying the obstacles of *Future-Date Degradation* and *Calendar Asymmetry Bias*.
- We develop an LLM-powered *Time Agent* using tool-augmented code generation, which achieves an average accuracy of 95.31%, surpassing multiple strong baselines and demonstrating the potential of tool-augmented code generation for the cross-calendar temporal reasoning task.

Related Work

Temporal Reasoning Benchmarks for LLMs. Temporal reasoning, the capability to perceive and understand the world's dynamics, is essential for advancing LLMs toward artificial general intelligence (AGI). Early benchmarks such as *TimeQA* (Chen, Wang, and Wang 2021) and *TimeDial* (Qin et al. 2021) concentrate on time-sensitive question answering and temporal commonsense in dialogue contexts. Recently, a new wave of benchmarks has emerged, targeting more fine-grained, complex temporal reasoning scenarios (Chen, Wang, and Wang 2021; Liska et al. 2022; Kasai et al. 2023; Tan, Ng, and Bing 2023a; Wei et al. 2023; Tan, Ng, and Bing 2023b; Wang and Zhao 2024a; Chu et al. 2024; Wang and Zhao 2024b; Fatemi et al. 2024; Ge et al. 2025; Wei et al. 2025; Saxena, Gema, and Minervini 2025a). For example, *TimeBench* (Chu et al. 2024) introduces a hierarchical benchmark for evaluating a wide spectrum of temporal reasoning capabilities, ranging from temporal relation classification to duration estimation and multi-hop time-aware question answering, spanning symbolic, commonsense, and event-centric tasks. *StreamingQA* (Liska et al. 2022) and *RealtimeQA* (Kasai et al. 2023) extend the focus to dynamic settings where new information arrives continuously, and LLMs are expected to revise their responses over time. Additionally, *Lost in Time* (Saxena, Gema, and Minervini 2025b) explores how multimodal LLMs process time-related visual inputs, including analogue clocks and yearly calendars. These benchmarks collectively enable systematic evaluation of LLMs' temporal reasoning across diverse tasks, complexities, and modalities.

Improving Temporal Reasoning Capabilities of LLMs. In pursuit of human-level temporal reasoning, recent research has proposed various strategies to enhance LLMs (Wei et al. 2023; Tan, Ng, and Bing 2023b; Yang et al. 2023; Jain et al. 2023; Su et al. 2024; Xiong et al. 2024; Yang et al. 2024b). For instance, *RemeMo* (Yang et al. 2023) improves LLMs' temporal understanding during pre-training by organizing events or sentences according to their temporal order, facilitating the model's capture of complex temporal relationships. *TG-LLM* (Xiong et al. 2024) applies supervised fine-tuning on a synthetic temporal graph dataset, enabling the model to convert text into structured graphs for more effective temporal reasoning. Additionally, *TempReason* (Yang et al. 2024b) leverages reinforcement learning to encourage the model to generate temporally coherent predictions, thereby improving the performance across diverse temporal reasoning tasks. These efforts advance the temporal reasoning capabilities of LLMs across various tasks.

SPAN: A Cross-Calendar Temporal Reasoning Benchmark for Large Language Models

In this section, we first formalize the cross-calendar temporal reasoning task. Then, we describe the data acquisition and template design, including cross-calendar data collection, a description of the newly developed cross-calendar conversion interface, and the construction of question–code template pairs. Finally, we present a novel evaluation protocol for dynamic instance generation.

Reasoning Type	Question Format	Question Template
Date-based	Content question	*Today's date on the $\{c_s\}$ calendar is "$\{d^r_{c_s}\}$". What was the $\{c_t\}$ calendar date $\{n_d\}$ days ago?*
		Today's date on the $\{c_s\}$ calendar is "$\{d^r_{c_s}\}$". What is the $\{c_t\}$ calendar date $\{n_d\}$ days later?
		Today's date on the $\{c_s\}$ calendar is "$\{d^r_{c_s}\}$". What was the $\{c_t\}$ calendar date $\{n_w\}$ weeks ago?
		Today's date on the $\{c_s\}$ calendar is "$\{d^r_{c_s}\}$". What is the $\{c_t\}$ calendar date $\{n_w\}$ weeks later?
	Polar question	*Today's date on the $\{c_s\}$ calendar is "$\{d^r_{c_s}\}$". Was the $\{c_t\}$ calendar date $\{n_d\}$ days ago equivalent to the date "$\{d^e_{c_t}\}$"?*
		Today's date on the $\{c_s\}$ calendar is "$\{d^r_{c_s}\}$". Is the $\{c_t\}$ calendar date $\{n_d\}$ days later equivalent to the date "$\{d^e_{c_t}\}$"?
		Today's date on the $\{c_s\}$ calendar is "$\{d^r_{c_s}\}$". Was the $\{c_t\}$ calendar date $\{n_w\}$ weeks ago equivalent to the date "$\{d^e_{c_t}\}$"?
		Today's date on the $\{c_s\}$ calendar is "$\{d^r_{c_s}\}$". Is the $\{c_t\}$ calendar date $\{n_w\}$ weeks later equivalent to the date "$\{d^e_{c_t}\}$"?
Festival-based	Content question	*Today's date on the $\{c_s\}$ calendar is "$\{d^r_{c_s}\}$". What was the $\{c_t\}$ calendar date of the $\{c_s\}$ festival "$\{f_{c_s}\}$" $\{n_d\}$ ago?*
		Today's date on the $\{c_s\}$ calendar is "$\{d^r_{c_s}\}$". What is the $\{c_t\}$ calendar date of the $\{c_s\}$ festival "$\{f_{c_s}\}$" $\{n_y\}$ later?
	Polar question	*Today's date on the $\{c_s\}$ calendar is "$\{d^r_{c_s}\}$". Was the $\{c_t\}$ calendar date of the $\{c_s\}$ festival "$\{f_{c_s}\}$" $\{n_y\}$ years ago equivalent to the date "$\{d^e_{c_t}\}$"?*
		Today's date on the $\{c_s\}$ calendar is "$\{d^r_{c_s}\}$". Is the $\{c_t\}$ calendar date of the $\{c_s\}$ festival "$\{f_{c_s}\}$" $\{n_y\}$ years later equivalent to the date "$\{d^e_{c_t}\}$"?

Table 1: The question templates. We use curly-brace placeholders to denote variables: c_s and c_t denote the source and target calendars; f_{c_s} denotes the festival in the source calendar; $d^r_{c_s}$ denotes the reference date in the source calendar; $d^e_{c_t}$ denotes the expected date in the target calendar (used for polar questions); n_d, n_w, and n_y denote temporal offsets in days, weeks, and years, respectively.

Task Formulation

Typically, the cross-calendar temporal reasoning task involves both intra-calendar date reasoning and inter-calendar date conversion, requiring LLMs to understand temporal relationships within individual calendars and to grasp temporal alignments between different calendars. Formally, this process can be represented as follows:

$$d^r_{c_s} \xrightarrow{\texttt{Reasoning}} d_{c_s} \xrightarrow{\texttt{Converting}} d^t_{c_t} \quad (1)$$

where $d^r_{c_s}$ and d_{c_s} represent the reference date and reasoning date in the source calendar c_s, respectively; $d^t_{c_t}$ denotes the target date derived by converting $d^i_{c_s}$ to the target calendar c_t. Note that Equation (1) is symmetric, enabling the source and target calendars to be interchanged.

In this work, we focus on a practical and representative cross-calendar temporal reasoning scenario, namely, reasoning between the Gregorian calendar and other calendar systems, which is prevalent in many countries and regions (Taqizadeh 1939; Graumann 2015). Accordingly, in our formulation, when either the source calendar c_s or the target calendar c_t is set to the Gregorian calendar, the other is set to a non-Gregorian calendar.

Data Collection and Template Design

Cross-Calendar Data Collection. SPAN covers dates from three major calendar families: solar (*i.e.*, the Gregorian, Persian and Shaka calendars), lunar (*i.e.*, the Islamic calendar), and lunisolar (*i.e.*, the Hebrew and Chinese lunar calendars). Calendar dates are collected using Python libraries and `Wikipedia`[1]. Specifically, the `datetime` library[2] is employed to enumerate Gregorian dates, and these dates are subsequently converted to their equivalent Islamic, Hebrew, Shaka, and Persian calendar dates using the `convertdate` library[3], while the conversions to the Chinese lunar calendar are handled using the `LunarCalendar` library[4]. Additionally, festival dates for each calendar are sourced and curated from `Wikipedia`. Festivals are listed in Appendix

[1] https://www.wikipedia.org

[2] https://docs.python.org/3/library/datetime.html

[3] https://pypi.org/project/convertdate

[4] https://pypi.org/project/LunarCalendar

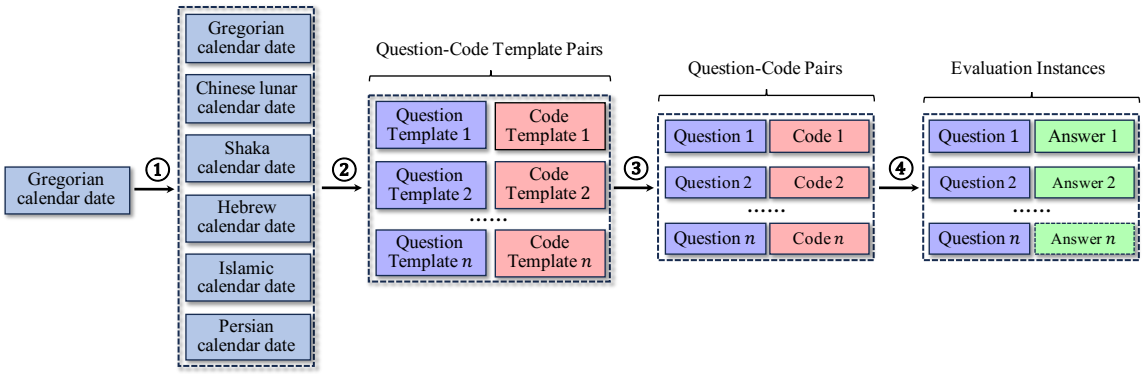

Figure 1: Overview of the proposed evaluation protocol. Given a user-specified Gregorian date as input, the process proceeds through four stages: ① ***Calendar Conversion***. The Gregorian date is converted into its equivalents in five calendars via our `search_calendar` interface, yielding $(c_s, d^r_{c_s}, f_{c_s})$ pairs, with $d^r_{c_s}$ and f_{c_s} denoting the reference date and the festival in the source calendar c_s, respectively. ② ***Template Matching***. These pairs are further utilized to construct $(c_s, d^r_{c_s}, f_{c_s}, c_t)$ pairs. Here, c_s and c_t are selected specifically to ensure one is a Gregorian calendar and the other is a non-Gregorian calendar. These pairs are matched against all question–code template pairs to generate candidate pairs. ③ ***Template Instantiation***. For each candidate question-code template pair, we manually specify the remaining variables $(d^e_{c_t}, n_d, n_w, n_y)$. Afterwards, question–code pairs are generated by filling the template placeholders with all variables. ④ ***Code Execution***. Finally, we execute each code snippet to generate the gold answer.

A. Finally, the collected dates are aligned to construct cross-calendar entries, each containing equivalent dates across the aforementioned six calendars.

Cross-Calendar Conversion Interface. We develop a unified interface, `search_calendar`, which converts a date or festival in the given calendar to its equivalents in multiple target calendars. The interface supports two parameter schemas: {`calendar_name`, `year`, `month`, `day`} to specify a date, and {`calendar_name`, `year`, `festival_name`} to specify a festival in the given calendar. Upon invocation, `search_calendar` returns a cross-calendar entry containing equivalent dates across six calendars, enabling subsequent temporal computations. In this work, we employ this interface in both the code templates and the proposed `Time Agent`. We provide a detailed description of `search_calendar` in Appendix D.

Question Template Design. As shown in Table 1, we design a set of question templates, which are organized into two categories: (1) ***Date-Based Question Templates***, which apply temporal offsets in days or weeks to a reference date and convert the resulting date to its equivalent in the target calendar. (2) ***Festival-Based Question Templates***, which apply a year offset to a reference year, identify the corresponding festival date for the resulting year, and convert it to its equivalent in the target calendar. For each category, two question formats are included: polar questions (answered with "Yes" or "No") and content questions (answered with specific dates). The question templates collectively define eight variables, as detailed below. c_s and c_t denote the source and target calendars; $d^r_{c_s}$ denotes the reference date in the source calendar; $d^e_{c_t}$ denotes the expected date in the target calendar (used for polar questions); f_{c_s} denotes the festival in the source calendar; n_d, n_w, and n_y denote temporal offsets in days, weeks, and years, respectively. Among these, $(c_s, c_t, d^r_{c_s}, f_{c_s})$ are determined during evaluation, whereas $(d^e_{c_t}, n_d, n_w, n_y)$ are manually configurable.

Code Template Design. For each question template, we design a corresponding code template in Python for answer generation. Code templates mirror the date-related variables defined in the corresponding question templates but differ in how these variables are represented. Question templates express variables in natural language, whereas code templates represent them as code snippets. For example, the natural language expression "{n_d} days later" corresponds to the code snippet "`+ timedelta(days=`{n_d}`)`". We divide the code templates into two categories based on the reasoning direction: *Gregorian-to-Others* and *Others-to-Gregorian*, indicating conversions from the Gregorian calendar to other calendar systems and vice versa, respectively. For the *Gregorian-to-Others* category, we first use the `datetime` library to compute the reasoning date in the Gregorian calendar based on the given reference date, and then convert this result to the target calendar using our `search_calendar` interface. Conversely, for the *Others-to-Gregorian* category, we first convert the reference date to the Gregorian calendar using our `search_calendar` interface, and then perform further computations with the `datetime` library.

Evaluation Protocol for Dynamic Instance Generation

Building upon our prior data preparation, we propose a novel evaluation protocol for on-the-fly instance generation, as illustrated in Figure 1. This protocol takes a user-specified Gregorian date as input and produces a diverse set of instances for cross-calendar temporal reasoning. Specifically, this protocol consists of the following stages: ① ***Calendar Conversion***. We utilize the `search_calendar` interface

to convert the user-specific Gregorian date into its equivalents in the Chinese lunar, Islamic, Hebrew, Shaka, and Persian calendars. This yields a set of $(c_s, d^r_{c_s}, f_{c_s})$ pairs, where $d^r_{c_s}$ and f_{c_s} denote the reference date and the festival in the source calendar c_s, respectively. ② ***Template Matching***. Building on these pairs, we further construct $(c_s, d^r_{c_s}, f_{c_s}, c_t)$ pairs, where c_s and c_t are selected such that one is the Gregorian calendar and the other is a non-Gregorian calendar. These pairs are then matched with all question-code template pairs to yield candidate pairs for subsequent instantiation. ③ ***Template Instantiation***. For each question-code candidate template pair, we manually specify the remaining variables $(d^e_{c_t}, n_d, n_w, n_y)$ from a predefined set (see the *Experimental Setup* for details). These variables are then substituted into the corresponding placeholders in the question and code templates, producing paired questions and executable code. ④ ***Code Execution***. Finally, each code snippet is executed, and its output serves as the ground truth answer for the corresponding evaluation instance.

Experimental Setup

Evaluation Models. We evaluate six open- and closed-source LLMs. Specifically, the open-source models include `Llama-3.3-70B-Instruct` (Grattafiori et al. 2024), `DeepSeek-V3-1226` (Liu et al. 2024), and `Qwen-2.5-72B-Instruct` (Yang et al. 2024a), which are accessed via Hugging Face and deployed using the vLLM framework (Kwon et al. 2023); the closed-source models include `GPT-4o` (Hurst et al. 2024), `Claude-3.7-Sonnet` (Anthropic 2025), and `Gemini-1.5-Pro` (Team et al. 2024), which are accessed through their APIs. All models are evaluated using greedy decoding with a temperature of 0.0 to ensure reproducibility.

Evaluation Metric. To enable reliable and scalable evaluation, we adopt GPT-4o as an automatic evaluator. Given a question, a model-generated response, and the gold answer, GPT-4o is prompted to assess the correctness of the response. Accuracy is then computed over the entire test set based on GPT-4o's judgments. The complete evaluation prompt is detailed in Appendix E, and we also present the agreement between the GPT-4o evaluator and human annotations in Appendix C.

Evaluation Setup. The variables (c_s, $d^r_{c_s}$, f_{c_s} c_t) are dynamically determined by our evaluation protocol, while the remaining variables ($d^e_{c_t}$, n_d, n_w, n_y) are manually specified as follows. The offset variables n_d and n_w are set to integers in the range [1, 10], and n_y in the range [1, 5]. For polar questions, the variable $d^e_{c_t}$ is set to the answer date, ensuring that all polar questions yield "Yes" as the gold answer. Additionally, we set the user-specified Gregorian dates (evaluation dates) to July 1st of every fifth year from 1960 to 2060, resulting in 21 evaluation dates over 100 years. At each of the 21 evaluation dates, 1,780 instances are generated, comprising 800 date-based and 980 festival-based questions, yielding a total of 37,380 evaluation instances. Finally, all numeric month values in the generated instances are replaced with their corresponding textual month names according to their respective calendars.

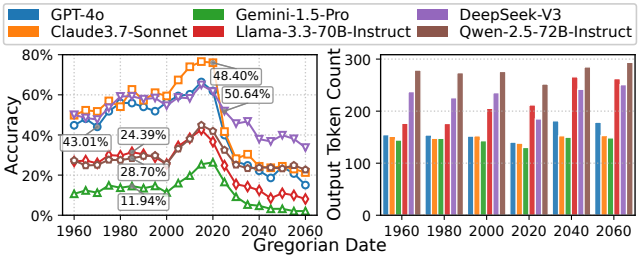

Figure 2: Left: Accuracy of LLMs across evaluation dates ranging from July 1st, 1960 to July 1st, 2060 at five-year intervals (July 1st omitted for clarity). The average accuracy over time is annotated for each model. Right: Average output token counts of LLMs at sampled evaluation dates. To ensure comparability, model outputs are tokenized using OpenAI's `tiktoken` tokenizer with the `o200k_base` encoding.

Main Results

The left part of Figure 2 shows accuracy at five-year intervals from July 1st, 1960 to July 1st, 2060. The following conclusions are drawn based on empirical evidence:

Cross-Calendar Temporal Reasoning Remains Challenging. As shown in the right part of Figure 2, we observe that LLMs tend to produce lengthy outputs (150∼300 tokens), indicating that they perform explicit reasoning rather than directly predicting dates for cross-calendar temporal questions. Despite these reasoning attempts, the average accuracy across LLMs and evaluation dates remains only 34.5%, with none exceeding 80% accuracy across evaluation dates. This underscores substantial room for improvement in LLMs' cross-calendar temporal reasoning capabilities.

Accuracy Stratification in Large Language Models. We find that the closed-source LLMs (*e.g.*, `GPT-4o` and `Claude-3.7-Sonnet`) generally outperform the open-source ones (*e.g.*, `Llama-3.3-70B-Instruct` and `Qwen-2.5-72B-Instruct`). Encouragingly, the open-source `DeepSeek-V3` achieves accuracy on par with leading closed-source LLMs, whereas the closed-source `Gemini-1.5-Pro` demonstrates the poorest performance, with accuracy consistently below 30% across evaluation dates. These observations reflect the strengths of commercial LLMs and progress in the open-source community.

Accuracy Exhibits Significant Temporal Dependency. Across all LLMs, accuracies tend to improve for past dates but drop sharply for future ones. This temporally-dependent pattern of accuracy also persists across different reasoning types, question formats, and temporal reasoning directions (see the *Analysis* section for details). These findings suggest that LLMs have a limited capability to perform cross-calendar temporal reasoning for future dates, a phenomenon we term ***Future-Date Degradation***.

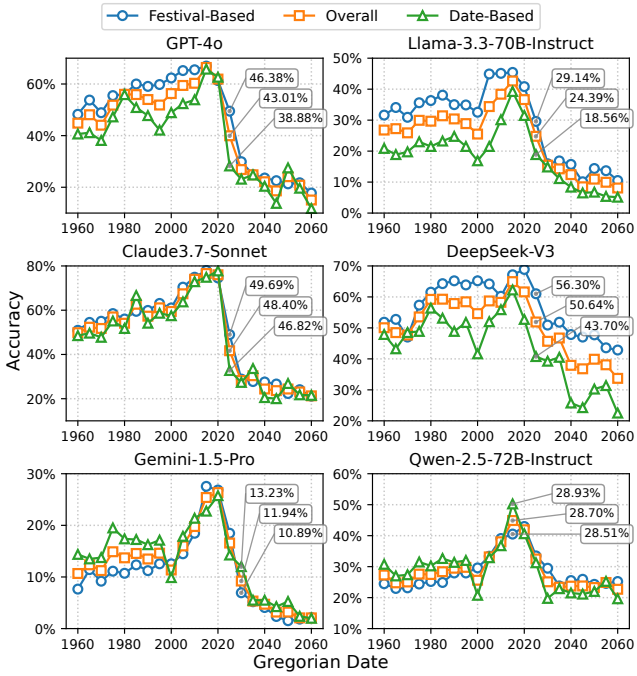

Figure 3: Accuracy of date-based and festival-based cross-calendar temporal reasoning over the evaluation dates from July 1st, 1960 to July 1st, 2060 at five-year intervals (July 1st omitted for clarity). The average accuracy over time for each reasoning type is annotated.

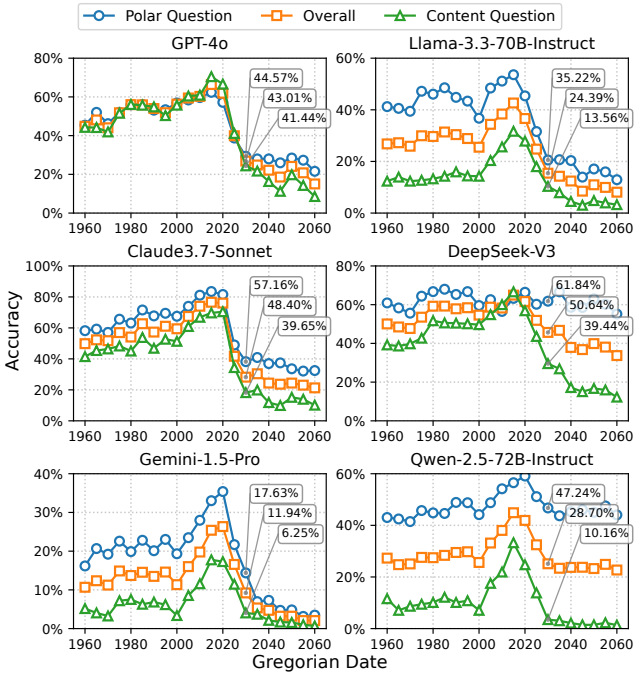

Figure 4: Accuracy of content question and polar question over the evaluation dates from July 1st, 1960 to July 1st, 2060 at five-year intervals (July 1st omitted for clarity). The average accuracy over time for each question format is annotated.

Experimental Results

Analysis

We empirically investigate the impact of various factors on LLMs' cross-calendar temporal reasoning capabilities, including reasoning types, question formats, and temporal reasoning directions.

Date-Based vs. Festival-Based. Figure 3 presents the accuracy of LLMs on two reasoning types. We observe that most LLMs perform better on festival-based reasoning, with gains between 2.87% and 12.60%, except for `Gemini-1.5-Pro` and `Qwen-2.5-72B-Instruct`. This likely stems from the prevalence of festival dates in pretraining data, which reduces the difficulty of festival-based reasoning.

Polar Question vs. Content Question. Figure 4 presents the accuracy of two question formats. We find that the accuracy on polar questions consistently exceeds that of content questions for all LLMs, with gains ranging from 3.13% to 37.08% (18.86% on average). This gap reflects the lower complexity of polar questions, whose answers are binary, resulting in a 50% chance of correctness even without precise temporal reasoning.

Gregorian-to-Others vs. Others-to-Gregorian. Figure 5 presents accuracy for two group of temporal reasoning directions: *Gregorian-to-Others* and *Others-to-Gregorian*. We see that all LLMs perform better on the *Gregorian-to-Others* group, with accuracy gains ranging from 3.97% to 17.49%, particularly among higher-performing models like `DeepSeek-V3` (17.49%), `GPT-4o` (15.82%), and `Claude-3.7-Sonnet` (15.76%). We refer to this discrepancy as ***Calendar Asymmetry Bias*** in LLMs, conjecturing that it likely stems from the prevalence of Gregorian-origin expressions in pretraining data. Since modern web documents and textual resources predominantly use Gregorian timestamps as the primary temporal anchor, LLMs are more frequently exposed to conversions originating from the Gregorian calendar, while receiving limited exposure to the reverse direction.

Improving Cross-Calendar Temporal Reasoning of LLMs

In this section, we present a practical method to improve LLM performance on SPAN. Our experiments indicate that LLMs face challenges with direct reasoning in this task. Consequently, we rely on an external tool for date conversion and defer enhancing LLMs' intrinsic reasoning to future work, emphasizing a pragmatic solution here.

Our Method. We develop an LLM-powered `Time Agent` that combines LLM code-generation capabilities with our cross-calendar conversion interface `search_calendar`. In particular, it follows a three-step agentic workflow: ① GPT-4o is guided with the description of `search_calendar` in a few-shot prompting, enabling it to generate executable code snippets for solving the given question. ② The gen-

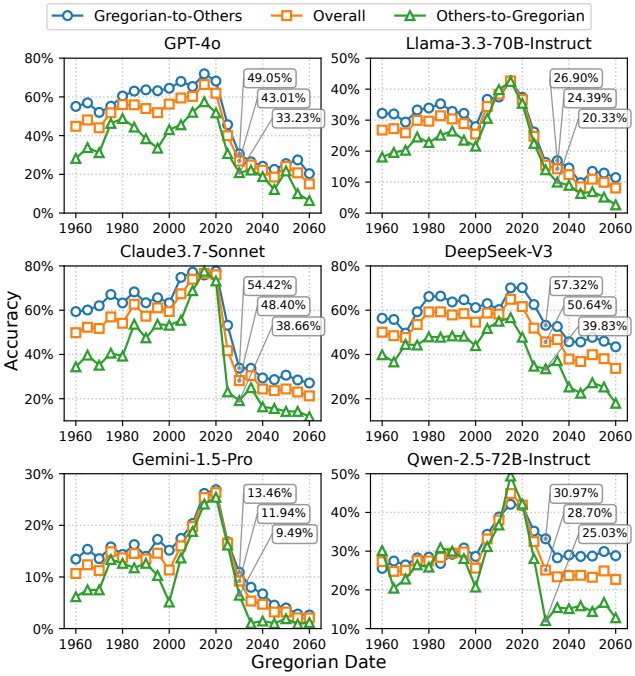

Figure 5: Accuracy of *Gregorian-to-Others* and *Others-to-Gregorian* cross-calendar temporal reasoning over the evaluation dates from July 1st, 1960 to July 1st, 2060 at five-year intervals (July 1st omitted for clarity). The average accuracy over time for each group's temporal reasoning directions is annotated.

erated code snippet is executed via a code interpreter, and we obtain the execution results. ③ The execution results are appended to the dialogue context as additional input, upon which GPT-4o produces the final answer. The complete prompts of `Time Agent` are provided in Appendix F.

Baselines. We compare our method with the following competitive baselines:

- `Previous Best Results`: The best results across all LLMs for each evaluation date, illustrated in Figure 2.
- `OpenAI-o1`: A state-of-the-art closed-source reasoning model with high token consumption, requiring extensive reasoning chains to process questions.
- `GPT-4o w/ RAG`: The GPT-4o model augmented with a retrieval-augmented generation (RAG) system, leveraging the Bing Search[5] for external retrieval.
- `GPT-4o`: The GPT-4o model, which is used to assess the effects of introducing RAG.

Results. As depicted in Figure 6, `Time Agent` demonstrates consistently high accuracy across all evaluation dates, achieving an average accuracy of 95.31% while maintaining a minimal average output token count. The few errors in `Time Agent` primarily stem from occasional code execution failures or subtle logical bugs, which can be further

[5]https://www.microsoft.com/en-us/bing

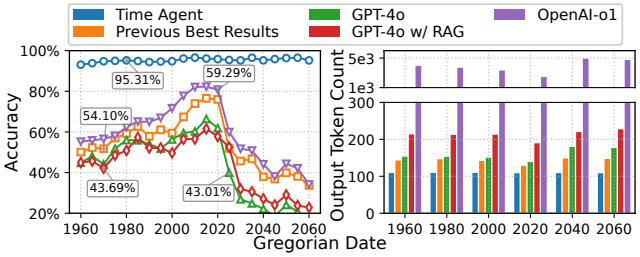

Figure 6: Left: Accuracy of each method over the evaluation dates at five-year intervals from July 1st, 1960 to July 1st, 2060. Right: Average output token count of each method at sampled evaluation dates. Model outputs are tokenized using the `tiktoken` library with the `o200k_base` encoding.

mitigated through enhanced prompt engineering. By comparison, `OpenAI-o1` attains the second-highest average accuracy of 59.29%, though it remains significantly below that of `Time Agent` and incurs substantially greater token usage. Additionally, we observe that `GPT-4o w/ RAG` achieves an average accuracy of 43.69%, and slightly outperforms `GPT-4o` on future evaluation dates, with an average improvement of merely 0.68%. However, this marginal gain requires generating more output tokens compared to `GPT-4o`, indicating that RAG provides negligible benefit for the cross-calendar temporal reasoning task. In summary, our results show that reasoning models and RAG are insufficient for the cross-calendar temporal reasoning task. By contrast, tool-augmented code generation provides a robust and efficient solution.

Conclusion

In this work, we introduce **SPAN**, a benchmark to evaluate the cross-calendar temporal reasoning capabilities of LLMs. Unlike prior benchmarks that are Gregorian-centric and time-invariant, SPAN enables cross-calendar temporal reasoning across ten reasoning directions, two reasoning types, and two question formats. To support evaluation at different temporal contexts, we design a novel template-driven protocol for time-variant evaluation. Based on this protocol, we perform extensive empirical evaluations on dates ranging from 1960 to 2060, revealing significant limitations in LLMs' capability to reason across calendars, with the challenges including *Future-Date Degradation* and *Calendar Asymmetry Bias*. Furthermore, we demonstrate that an LLM-powered `Time Agent` with tool-augmented code generation achieves impressive performance on the cross-calendar temporal reasoning task.

In future work, we plan to extend SPAN to encompass a wider spectrum of calendar systems, enabling more comprehensive coverage of global temporal contexts. Moreover, we will adapt existing Gregorian-based temporal reasoning tasks (*e.g.*, event ordering and duration estimation) to cross-calendar settings, facilitating a systematic evaluation of LLMs' generalization capabilities and advancing their robustness in diverse temporal reasoning scenarios.

Acknowledgments

This work was independently conducted at Li Auto. We sincerely thank Li Auto for its generous support, which was essential to the successful completion of this work.

References

Anthropic. 2025. Claude 3.7 sonnet and claude code. *https://www.anthropic.com/news/claude-3-7-sonnet*.

Balloccu, S.; Schmidtová, P.; Lango, M.; and Dušek, O. 2024. Leak, Cheat, Repeat: Data Contamination and Evaluation Malpractices in Closed-Source LLMs. In *Proceedings of ECAL*.

Chen, W.; Wang, X.; and Wang, W. Y. 2021. A Dataset for Answering Time-Sensitive Questions. In *Proceedings of NeurIPS*.

Chu, Z.; Chen, J.; Chen, Q.; Yu, W.; Wang, H.; Liu, M.; and Qin, B. 2024. TimeBench: A Comprehensive Evaluation of Temporal Reasoning Abilities in Large Language Models. In *Proceedings of ACL*.

Deng, C.; Zhao, Y.; Tang, X.; Gerstein, M.; and Cohan, A. 2024. Investigating Data Contamination in Modern Benchmarks for Large Language Models. In *Proceedings of ACL*.

Fan, Y.; Mu, Y.; Wang, Y.; Huang, L.; Ruan, J.; Li, B.; Xiao, T.; Huang, S.; Feng, X.; and Zhu, J. 2025. SLAM: Towards Efficient Multilingual Reasoning via Selective Language Alignment. In *Proceedings of COLING*.

Fatemi, B.; Kazemi, M.; Tsitsulin, A.; Malkan, K.; Yim, J.; Palowitch, J.; Seo, S.; Halcrow, J.; and Perozzi, B. 2024. Test of time: A benchmark for evaluating llms on temporal reasoning. In *Proceedings of ICLR*.

Ge, Y.; Romeo, S.; Cai, J.; Shu, R.; Sunkara, M.; Benajiba, Y.; and Zhang, Y. 2025. TReMu: Towards Neuro-Symbolic Temporal Reasoning for LLM-Agents with Memory in Multi-Session Dialogues. *arXiv preprint arXiv:2502.01630*.

Grattafiori, A.; Dubey, A.; Jauhri, A.; Pandey, A.; Kadian, A.; Al-Dahle, A.; Letman, A.; Mathur, A.; Schelten, A.; Vaughan, A.; et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Graumann, G. 2015. The problem field of calendars in different cultures. *LUMAT: International Journal on Math, Science and Technology Education*.

Hurst, A.; Lerer, A.; Goucher, A. P.; Perelman, A.; Ramesh, A.; Clark, A.; Ostrow, A.; Welihinda, A.; Hayes, A.; Radford, A.; et al. 2024. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*.

Jain, R.; Sojitra, D.; Acharya, A.; Saha, S.; Jatowt, A.; and Dandapat, S. 2023. Do language models have a common sense regarding time? revisiting temporal commonsense reasoning in the era of large language models. In *Proceedings of EMNLP*.

Kasai, J.; Sakaguchi, K.; Takahashi, Y.; Le Bras, R.; Asai, A.; Yu, X. V.; Radev, D.; Smith, N. A.; Choi, Y.; and Inui, K. 2023. REALTIME QA: what's the answer right now? In *Proceedings of NeurIPS*.

Kwon, W.; Li, Z.; Zhuang, S.; Sheng, Y.; Zheng, L.; Yu, C. H.; Gonzalez, J.; Zhang, H.; and Stoica, I. 2023. Efficient memory management for large language model serving with pagedattention. In *Proceedings of SOSP*.

Li, Z.; Jiang, G.; Xie, H.; Song, L.; Lian, D.; and Wei, Y. 2024. Understanding and Patching Compositional Reasoning in LLMs. In *Proceedings of ACL (Findings)*.

Liska, A.; Kocisky, T.; Gribovskaya, E.; Terzi, T.; Sezener, E.; Agrawal, D.; D'Autume, C. D. M.; Scholtes, T.; Zaheer, M.; Young, S.; et al. 2022. Streamingqa: A benchmark for adaptation to new knowledge over time in question answering models. In *Proceedings of ICML*.

Liu, A.; Feng, B.; Xue, B.; Wang, B.; Wu, B.; Lu, C.; Zhao, C.; Deng, C.; Zhang, C.; Ruan, C.; et al. 2024. Deepseek-v3 technical report. *arXiv preprint arXiv:2412.19437*.

Qiao, S.; Ou, Y.; Zhang, N.; Chen, X.; Yao, Y.; Deng, S.; Tan, C.; Huang, F.; and Chen, H. 2023. Reasoning with Language Model Prompting: A Survey. In *Proceedings of ACL*.

Qin, L.; Gupta, A.; Upadhyay, S.; He, L.; Choi, Y.; and Faruqui, M. 2021. TIMEDIAL: Temporal Commonsense Reasoning in Dialog. In *Proceedings of ACL*.

Sainz, O.; Campos, J.; García-Ferrero, I.; Etxaniz, J.; de Lacalle, O. L.; and Agirre, E. 2023. NLP Evaluation in trouble: On the Need to Measure LLM Data Contamination for each Benchmark. In *Proceedings of EMNLP (Findings)*.

Saxena, R.; Gema, A. P.; and Minervini, P. 2025a. Lost in Time: Clock and Calendar Understanding Challenges in Multimodal LLMs. In *Proceedings of ICLR*.

Saxena, R.; Gema, A. P.; and Minervini, P. 2025b. Lost in Time: Clock and Calendar Understanding Challenges in Multimodal LLMs. *arXiv preprint arXiv:2502.05092*.

Su, Z.; Zhang, J.; Zhu, T.; Qu, X.; Li, J.; Cheng, Y.; et al. 2024. Timo: Towards Better Temporal Reasoning for Language Models. In *Proceedings of COLM*.

Tan, Q.; Ng, H. T.; and Bing, L. 2023a. Towards Benchmarking and Improving the Temporal Reasoning Capability of Large Language Models. In *Proceedings of ACL*.

Tan, Q.; Ng, H. T.; and Bing, L. 2023b. Towards Benchmarking and Improving the Temporal Reasoning Capability of Large Language Models. In *Proceedings of ACL*.

Taqizadeh, S. 1939. Various Eras and Calendars used in the Countries of Islam. *Bulletin of the School of Oriental and African Studies*.

Team, G.; Georgiev, P.; Lei, V. I.; Burnell, R.; Bai, L.; Gulati, A.; Tanzer, G.; Vincent, D.; Pan, Z.; Wang, S.; et al. 2024. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *arXiv preprint arXiv:2403.05530*.

Wang, Y.; and Zhao, Y. 2024a. TRAM: Benchmarking Temporal Reasoning for Large Language Models. In *Proceedings of ACL (Findings)*.

Wang, Y.; and Zhao, Y. 2024b. TRAM: Benchmarking Temporal Reasoning for Large Language Models. In *Proceedings of ACL (Findings)*.

Wei, S.; Li, W.; Song, F.; Luo, W.; Zhuang, T.; Tan, H.; Guo, Z.; and Wang, H. 2025. TIME: A Multi-level Benchmark

for Temporal Reasoning of LLMs in Real-World Scenarios. *arXiv preprint arXiv:2505.12891*.

Wei, Y.; Su, Y.; Ma, H.; Yu, X.; Lei, F.; Zhang, Y.; Zhao, J.; and Liu, K. 2023. MenatQA: A New Dataset for Testing the Temporal Comprehension and Reasoning Abilities of Large Language Models. In *Proceedings of EMNLP (Findings)*.

Xiong, S.; Payani, A.; Kompella, R.; and Fekri, F. 2024. Large Language Models Can Learn Temporal Reasoning. In *Proceedings of ACL*.

Xu, J.; Fei, H.; Pan, L.; Liu, Q.; Lee, M.-L.; and Hsu, W. 2024. Faithful Logical Reasoning via Symbolic Chain-of-Thought. In *Proceedings of ACL*.

Yang, A.; Yang, B.; Zhang, B.; Hui, B.; Zheng, B.; Yu, B.; Li, C.; Liu, D.; Huang, F.; Wei, H.; et al. 2024a. Qwen2. 5 technical report. *arXiv preprint arXiv:2412.15115*.

Yang, S.; Li, X.; Bing, L.; and Lam, W. 2023. Once Upon a Time in Graph: Relative-Time Pretraining for Complex Temporal Reasoning. In *Proceedings of EMNLP*.

Yang, W.; Li, Y.; Fang, M.; and Chen, L. 2024b. Enhancing Temporal Sensitivity and Reasoning for Time-Sensitive Question Answering. In *Proceedings of EMNLP (Findings)*.