

# RAG or Learning? Understanding the Limits of LLM Adaptation under Continuous Knowledge Drift in the Real World

Hanbing Liu<sup>1\*†</sup> Lang Cao<sup>2\*</sup> Yang Li<sup>3‡</sup>

<sup>1</sup>Tsinghua University <sup>2</sup>University of Illinois Urbana-Champaign

<sup>3</sup>School of Artificial Intelligence, Chinese University of Hong Kong (Shenzhen)

liuhb24@mails.tsinghua.edu.cn langcao2@illinois.edu yangli@cuhk.edu.cn

## Abstract

Large language models (LLMs) acquire most of their knowledge during pretraining, which ties them to a fixed snapshot of the world and makes adaptation to continuously evolving knowledge challenging. As facts, entities, and events change over time, models may experience *continuous knowledge drift*, resulting not only in outdated predictions but also in temporally inconsistent reasoning. Although existing approaches, such as continual finetuning, knowledge editing, and retrieval-augmented generation (RAG), aim to update or supplement model knowledge, they are rarely evaluated in settings that reflect chronological, evolving, and real-world knowledge evolution. In this work, we introduce a new benchmark of real-world dynamic events, constructed from time-stamped evidence that captures how knowledge evolves over time, which enables systematic evaluation of model adaptation under continuous knowledge drift. The benchmark reveals that most existing methods, including vanilla RAG and several learning-based approaches, struggle under this setting, exposing critical limitations such as catastrophic forgetting and temporal inconsistency. To mitigate these limitations, we propose a time-aware retrieval baseline, *Chronos*, which progressively organizes retrieved evidence into an Event Evolution Graph to enable more temporally consistent understanding in LLMs without additional training. Overall, this work provides a foundation for analyzing and advancing LLM adaptation to continuous knowledge drift in realistic settings. Code is available at <https://github.com/hbing-1/chronos>.

## 1 Introduction

Large language models (LLMs) have shown strong performance across a wide range of tasks, includ-

\*Equal contribution. Author order may be swapped.

†Affiliated with the Shenzhen Key Laboratory of Ubiquitous Data Enabling, Tsinghua Shenzhen International Graduate School, Tsinghua University.

‡Corresponding author.

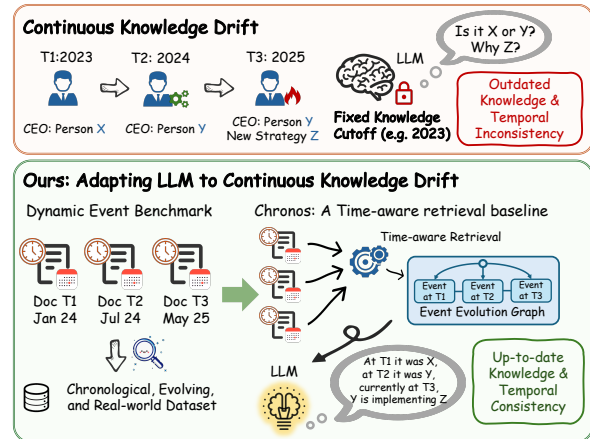


Figure 1: Illustration of continuous knowledge drift and our proposed method. Real-world knowledge evolves over time (e.g., changes in company CEO), while LLMs with a fixed knowledge cutoff may produce outdated or temporally inconsistent answers. We introduce a dynamic event benchmark composed of time-stamped documents that reflect such evolution, and a time-aware retrieval baseline (*Chronos*) that progressively organizes retrieved evidence into an Event Evolution Graph, enabling more temporally consistent reasoning.

ing question answering and reasoning, by acquiring extensive linguistic and factual knowledge during pretraining (Achiam et al., 2023; He et al., 2024; Sun et al., 2025; Liu et al., 2025a). However, this knowledge is inherently limited to a fixed snapshot of the world, commonly referred to as a knowledge cutoff (Chang et al., 2024; Thede et al., 2025). In contrast, real-world knowledge evolves continuously: organizations reorganize, officials change roles, policies are revised, and entity attributes are updated over time (Wang et al., 2025). As a result, deployed LLMs may gradually become misaligned with the current state of the world, a phenomenon we refer to as *continuous knowledge drift*.

Beyond outdated facts, continuous knowledge drift also poses challenges for temporal reasoning (Chu et al., 2024; Wang and Zhao, 2024; Qiu et al., 2024). As knowledge evolves, correct reasoning

requires models to not only access up-to-date information, but also preserve past knowledge and maintain temporal consistency in how events and facts relate and change over time. This involves capturing their ordering, dependencies, and state transitions. Without such temporal consistency, models may produce inconsistent or logically incorrect outputs even when relevant evidence is available (Guo et al., 2025). These challenges naturally motivate the central question of this work:

*How can LLMs adapt to continuously evolving real-world knowledge while maintaining temporally consistent understanding?*

A natural approach to addressing knowledge drift is continual pretraining or finetuning (Shi et al., 2025; Zheng et al., 2025a,b). While effective in some settings, such methods are computationally expensive, require repeated retraining as knowledge evolves, and are prone to catastrophic forgetting (Ke et al., 2023). Knowledge editing offers a more targeted alternative by modifying a small subset of model parameters to update specific facts (Meng et al., 2022a,b; Mitchell et al., 2021; Li et al., 2024b; Wang et al., 2024). However, existing editing methods are primarily designed to overwrite a model’s current beliefs, without explicitly preserving past knowledge. Moreover, they typically operate on one fact at a time, whereas real-world knowledge updates often arrive in batches. When applied repeatedly, individual edits may interfere with each other, and models lack a principled mechanism to distinguish past from current information.

Retrieval-augmented generation (RAG) provides a complementary paradigm by incorporating external knowledge at inference time without modifying model parameters, and has shown strong performance on a wide range of knowledge-intensive tasks (Fan et al., 2024; Yu et al., 2024). However, existing RAG approaches are rarely evaluated in settings that reflect chronological, evolving, and real-world knowledge evolution. Most evaluations focus on isolated or static updates, and seldom require reasoning over sequences of evolving events. As a result, it remains unclear whether retrieval-based methods can support temporally consistent understanding under continuous real-world knowledge drift. Additional discussion of related work is provided in Appendix B.

Addressing this gap requires evaluation settings

that reflect how knowledge evolves over time. However, existing benchmarks are largely static or centered on single-fact updates, and seldom require handling ordered event sequences or enforcing temporal consistency (Thede et al., 2025; Levy et al., 2017; Jang et al., 2022). In addition, many commonly used benchmarks rely on knowledge that is already outdated by one or more years. To bridge this gap, we introduce a new benchmark of real-world dynamic events from 2024 to 2025, spanning ten domains. The benchmark collects time-stamped evidence describing both newly emerging and historically changing facts, thereby reflecting real-world continuous knowledge drift and enabling the study of how models adapt to evolving knowledge over time.

Building on this benchmark, we further introduce *Chronos*, a time-aware retrieval framework designed for continuously evolving knowledge, as illustrated in Figure 1. *Chronos* incorporates temporal signals into retrieval and progressively organizes evidence through an *Event Evolution Graph*, which explicitly encodes how entities and events change over time. By structuring evidence in this way, *Chronos* enables more temporally consistent understanding without modifying model parameters.

Overall, our contributions are threefold:

1. **Problem formulation.** We formalize the problem of adapting LLMs to continuously evolving real-world knowledge, highlighting the challenge of maintaining temporally consistent understanding as knowledge changes over time.
2. **A new benchmark.** We introduce a new benchmark of real-world dynamic events. The benchmark consists of time-stamped evidence reflecting how knowledge changes in practice, and enables systematic evaluation of models under continuous knowledge drift.
3. **A strong baseline.** We propose *Chronos*, a retrieval-based framework that constructs an Event Evolution Graph from retrieved evidence to model how entities and events evolve over time, which enables more temporally consistent understanding, and serves as a strong baseline for the proposed benchmark.

## 2 Benchmark Design

Our benchmark consists of two main components: a collection of knowledge quadruples and five categories of question–answer pairs. These question

types define distinct evaluation tasks that assess models from multiple perspectives, evaluating both their ability to access up-to-date factual knowledge and to maintain temporal consistency when reasoning under continuous knowledge drift.

## 2.1 Knowledge Quadruple Curation

We first define the temporal scope of all knowledge to span from *January 1, 2024* to *October 31, 2025*. Accordingly, all evaluated LLMs are assumed to have a knowledge cutoff prior to 2024, ensuring that the benchmark tests their ability to handle post-cutoff information.

We manually define ten topical domains, including *Corporate Leadership Changes*, *Sports Coaching Changes and Player Transfers*, and *Economic Indicators and Policy Changes*, among others. Entities within these domains are highly dynamic, undergoing frequent updates and continuously giving rise to new events, making them particularly suitable for evaluating a model’s ability to track and reason over evolving knowledge across time.

For each topical domain, we first identify a set of core entities whose attributes are prone to frequent change over time, and treat each as a subject. We then track the evolution of these entities using Wikipedia data from 2024 to 2025, recording all observable state changes and newly emerging events together with their corresponding timestamps. Each change event is extracted using large language models and represented as a knowledge quadruple of the form  $(subject, relation, object, timestamp)$ , which explicitly captures the temporal dynamics of real-world knowledge.

## 2.2 Question-Answer Pair Construction

To evaluate whether models can retain historical knowledge, and understand up-to-date knowledge and correctly reason over temporally evolving information, we design five types of QA tasks that can be grouped into three major categories.

1. **Historical QA.** This category evaluates whether a model can correctly recall historical facts that occurred before 2024. Under our setting, all information required to answer these questions originates from events prior to 2024.
2. **Contemporary QA.** This category evaluates whether a model can correctly answer questions involving up-to-date knowledge occurring between *2024-01-01* and *2025-10-31*, which we treat as contemporary knowledge. It is further

Table 1: An illustration of the dataset structure. Each instance includes time-stamped knowledge quadruples together with questions covering historical, contemporary, and commonsense settings, designed to evaluate reasoning over evolving knowledge.

<b>Knowledge</b>	
⟨World’s Richest Person, held by, Elon Musk, 2024-01-01⟩	
⟨World’s Richest Person, held by, Bernard Arnault, 2024-01-28⟩	
⟨World’s Richest Person, held by, Jeff Bezos, 2024-03-05⟩	
⟨World’s Richest Person, held by, Bernard Arnault, 2024-04-02⟩	
⟨World’s Richest Person, held by, Elon Musk, 2024-06-08⟩	
⟨World’s Richest Person, held by, Larry Ellison, 2025-09-10⟩	
⟨World’s Richest Person, held by, Elon Musk, 2025-09-11⟩	
⟨Oracle stock price, surged to, USD 328, 2025-09-10⟩	
<b>Historical Question</b>	
Who was the richest person in the world in March 2014?	
<b>Contemporary Questions</b>	
C1: Who was the world’s richest person on August 20, 2025?	
C2: Who were the world’s richest people at any point during 2024?	
C3: Which company’s stock surge led to Elon Musk losing his position as the world’s richest person on September 10, 2025?	
<b>Commonsense Question</b>	
What is the brightest star in the night sky?	
A. Sirius B. Sun C. Polaris D. North Star	
<b>Answers</b>	
Historical: Bill Gates	Commonsense: A C1: Elon Musk
C2: Elon Musk, Bernard Arnault, Jeff Bezos	C3: Oracle

divided into three subtypes that require different forms of temporal reasoning:

- (a) **Single-timestamp QA (C1).** Each question can be answered using a single knowledge quadruple associated with one specific timestamp.
  - (b) **Multi-timestamp QA (C2).** Each question requires reasoning over multiple temporally distinct facts associated with the same entity.
  - (c) **Multi-source QA (C3).** Each question requires integrating information from multiple entities and multiple timestamps, testing the model’s ability to combine evidence across sources while preserving temporal consistency.
3. **Commonsense QA.** This category evaluates time-invariant commonsense reasoning ability and is not directly related to the other four question types or the knowledge base used in our benchmark. It serves to verify that incorporating temporally evolving knowledge does not degrade a model’s general-purpose reasoning performance.

For QA construction, we primarily rely on large

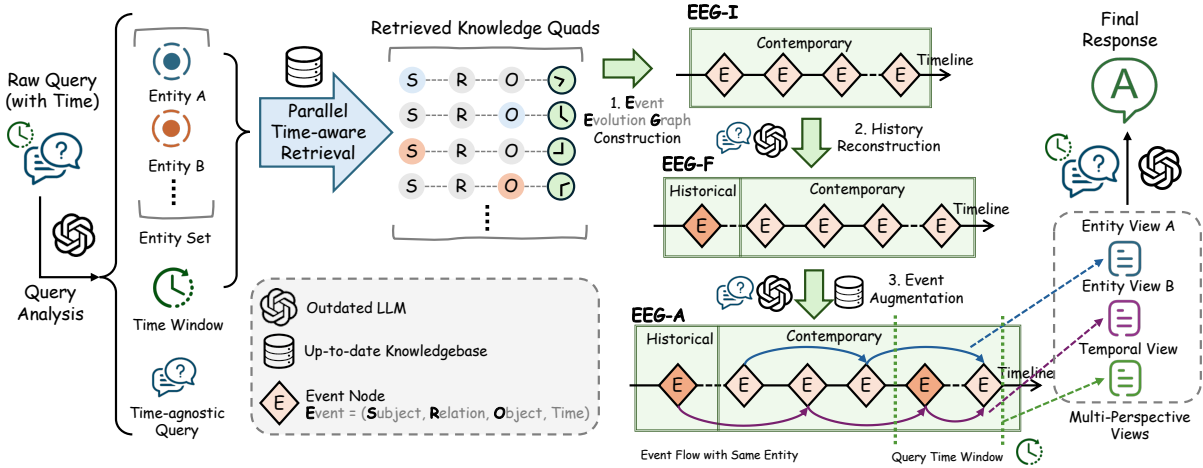


Figure 2: Overview of *Chronos*. The framework first performs query analysis to extract relevant entities and the associated time window. A time-aware retriever then collects temporally relevant knowledge quadruples from an up-to-date knowledge base. These facts are organized into an Event Evolution Graph (EEG), which models how entity states evolve over time by sorting events along a timeline and linking events that share common entities. The EEG is progressively refined, and its resulting multi-perspective views enable the model to reason over evolving knowledge and produce temporally consistent responses.

language models to generate candidate questions. For *Historical QA*, questions are generated such that the relevant events occur strictly before 2024. For *CI*, each question is derived directly from a single knowledge quadruple. For *C2*, multiple quadruples associated with the same entity are merged to construct questions requiring temporal aggregation. For *C3*, quadruples from different but semantically related entities are combined, ensuring that correct answers require multi-source reasoning across both entities and timestamps. An illustration of the dataset structure is provided in Table 1.

All generated questions undergo manual verification to ensure clarity, correctness, and alignment with the intended evaluation setting. For *Commonsense QA*, we directly adopt the TruthfulQA dataset (Lin et al., 2022) as the evaluation data.

This construction process ensures that the benchmark captures realistic knowledge evolution while maintaining high data quality and enabling comprehensive and systematic evaluation of model robustness under temporal change. Dataset statistics are provided in Appendix C.

### 3 Methodology

We propose *Chronos*, a time-aware retrieval-augmented framework that serves as a strong baseline for the proposed benchmark. Its overall architecture is illustrated in Figure 2.

#### 3.1 Task Formulation

Given an input query  $q_{\text{raw}}$ , which may contain explicit or implicit temporal constraints, we assume access to an up-to-date knowledge base  $\mathcal{K}$  consisting of a set of time-stamped quadruples  $\{k_i\}$ , where each quadruple is defined as  $k_i = (s_i, r_i, o_i, t_i)$ . Here,  $s_i$  denotes the subject,  $r_i$  the relation,  $o_i$  the object, and  $t_i$  the timestamp associated with the fact. All timestamps fall within a fixed time window of newly acquired knowledge,  $T_0 \leq t_i \leq T_1$ .

We further assume an outdated large language model  $\mathcal{M}$  whose pretraining data only covers information available up to time  $T_0$ , and therefore lacks up-to-date knowledge  $\mathcal{K}$ . Under this setting, we study the problem of answering queries under continuous knowledge drift and the goal is to enable  $\mathcal{M}$  to produce the correct final answer  $A$  for diverse types of queries.

#### 3.2 Query Analysis

Given an input query  $q_{\text{raw}}$ , the model  $\mathcal{M}$  is prompted to perform query analysis:

$$\mathcal{E}, q_0, t_q = \mathcal{M}(q_{\text{raw}} \mid \mathcal{P}_1), \quad (1)$$

where  $\mathcal{E} = \{e\}$  denotes the set of extracted entities,  $q_0$  is a rewritten, time-agnostic version of the query with all temporal expressions removed, and  $t_q = [t_s, t_e]$  is the time window of the query.  $\mathcal{P}_1$  denotes the prompt used for this task.

It decomposes the raw query into explicit semantic components, enabling more precise and controllable retrieval. In addition, explicitly modeling temporal constraints is crucial because temporal information retrieval remains a long-standing challenge (Abdallah et al., 2025; Kanhabua and Anand, 2016). Dense text embeddings often struggle to accurately encode temporal expressions, making it difficult to distinguish events occurring at different points in time. By explicitly representing the time window, our approach enables time-aware retrieval that better aligns retrieved evidence with the intended temporal context of the query.

### 3.3 Time-aware Retrieval

For each extracted entity  $e$ , we treat it as a query  $q_e$  and associate it with a time window  $t_q = [t_s, t_e]$ . We perform time-aware retrieval in parallel for all entities using dense semantic embeddings.

Let  $\mathcal{K}_{q_e} = \{(k_i, sim_i)\}_{i=1}^N$  denote the top- $N$  candidate knowledge items retrieved based on semantic similarity, where  $sim_i \in [0, 1]$  is the similarity score between the query  $q_e$  and the knowledge item  $k_i$ . Each knowledge item  $k_i$  is associated with a timestamp  $t_i$ .

To incorporate temporal relevance, we define the temporal distance between the timestamp  $t_i$  and the query time window  $t_q$  as:

$$\Delta(t_i, t_q) = \begin{cases} t_s - t_i, & \text{if } t_i < t_s, \\ t_i - t_e, & \text{if } t_i > t_e, \\ 0, & \text{if } t_s \leq t_i \leq t_e. \end{cases} \quad (2)$$

This definition assigns zero distance to events occurring within the target time interval and penalizes events outside the interval proportionally to their temporal deviation.

We then convert this distance into a temporal relevance score using an exponential decay function:

$$\text{TimeScore}(k_i) = \exp\left(-\frac{\Delta(t_i, t_q)}{\tau}\right), \quad (3)$$

where  $\tau$  is a decay hyperparameter controlling how rapidly relevance decreases as the timestamp moves away from the query interval.

Finally, we compute a unified retrieval score by combining semantic similarity and temporal relevance:

$$\text{Score}(k_i) = \alpha sim_i + (1 - \alpha) \text{TimeScore}(k_i), \quad (4)$$

where  $\alpha \in [0, 1]$  balances the contribution of semantic similarity and temporal proximity.

All candidate knowledge items are ranked in descending order according to  $\text{Score}(k_i)$ , and the top- $n$  items are selected to form the time-aware retrieved set:

$$\mathcal{K}_r = \text{Top-}n(\mathcal{K}_{q_e}, \text{Score}). \quad (5)$$

This time-aware retrieval mechanism prioritizes evidence that is both semantically relevant and temporally aligned with the query, providing a principled foundation for subsequent steps.

### 3.4 Event Evolution Graph Construction

Given a set of retrieved knowledge quadruples  $\mathcal{K}_r = \{(s_i, r_i, o_i, t_i)\}_{i=1}^n$ , we construct an *Event Evolution Graph (EEG)* to explicitly model how entities, their relations, and associated events evolve over time. The EEG is built through three progressive stages, denoted as *EEG-I*, *EEG-F*, and *EEG-A*, corresponding to the **Initial**, **Full**, and **Augmented** graphs, respectively.

**Step 1: Initial Temporal Ordering.** We first construct an initial temporal graph by organizing all retrieved events according to their timestamps. Each knowledge quadruple  $(s_i, r_i, o_i, t_i)$  is treated as an *event node*  $E$  and positioned along a global timeline, forming a linear structure that reflects the chronological progression of facts. Formally, we define the initial event evolution graph as

$$\mathcal{G}_I = (\mathcal{K}_r, \rightarrow_t), \quad (6)$$

where  $\mathcal{K}_r$  denotes the set of retrieved events, and  $\rightarrow_t$  denotes directed temporal edges induced by their chronological ordering. Specifically, we add a temporal edge  $k_i \rightarrow_t k_j$  only between two consecutive events  $k_i, k_j \in \mathcal{K}_r$  such that  $t_i < t_j$ . The resulting graph  $\mathcal{G}_I$ , referred to as *EEG-I*, captures the temporal ordering among retrieved events.

**Step 2: History Reconstruction.** To enable reasoning over historical context beyond retrieved facts, we further construct a more complete view of event evolution by incorporating reconstructed historical events.

Specifically, conditioned on the time-agnostic query  $q_0$  and its associated time window  $t_q$ , we prompt the model  $\mathcal{M}$  to infer historical events that are relevant to the query. This process yields a set of historical knowledge quadruples:

$$\mathcal{K}_H = \mathcal{M}(q_0, t_q \mid \mathcal{P}_2), \quad (7)$$

where each element in  $\mathcal{K}_{\mathcal{H}}$  is a time-stamped quadruple  $(s_i, r_i, o_i, t_i)$ . Here,  $\mathcal{P}_2$  denotes the prompt used to elicit historical knowledge.

These reconstructed quadruples are then merged with the retrieved events and inserted into the global timeline according to their inferred timestamps, yielding full graph *EEG-F*  $\mathcal{G}_F$ :

$$\mathcal{G}_F = (\mathcal{G}_I \cup \mathcal{K}_{\mathcal{H}}, \rightarrow_t), \quad (8)$$

where  $\rightarrow_t$  denotes directed temporal edges induced by chronological ordering.

**Step 3: Event Augmentation and Entity-Centric Linking.** To augment the graph, we enrich it with additional relevant events. Specifically, conditioned on the raw query  $q_{\text{raw}}$  and the current full graph  $\mathcal{G}_F$ , we use prompt  $\mathcal{P}_3$  to instruct the model  $\mathcal{M}$  to either (i) directly generate additional knowledge quadruples or (ii) produce a follow-up query  $q_1$  that targets potentially missing but relevant information, which is then retrieved to obtain additional knowledge quadruples. Obtained quadruples are merged into the existing graph, forming an augmented knowledge set  $\mathcal{K}_{\text{aug}}$ , which improves the coverage and completeness.

While temporal ordering captures the global progression of all events, effective reasoning often requires tracking how the *same entity* evolves across multiple events over time. Therefore, we further augment the graph through an entity-centric linking mechanism.

For each entity  $e$ , we define the set of all events in which it appears as either subject or object:

$$\mathcal{K}_e = \{k_i \in \mathcal{K}_{\text{aug}} \mid s_i = e \vee o_i = e\}. \quad (9)$$

These events are sorted in ascending order of their timestamps, and consecutive events are connected by entity-specific directed edges. Formally, for any two events  $k_i, k_j \in \mathcal{K}_e$ , we add an entity edge  $k_i \rightarrow_e k_j$  if  $k_j$  immediately follows  $k_i$  in time for the same entity. It yields a temporal chain that captures the evolution of each entity over time.

The resulting augmented graph, denoted as *EEG-A*  $\mathcal{G}_A$ , integrates three complementary components: (i) temporal edges that encode the global chronological order of events, (ii) historical and augmented events that provide broader contextual coverage, and (iii) entity-centric edges that model how individual entities evolve across time.

### 3.5 Prediction with Multi-Perspective Views

Given the augmented event evolution graph  $\mathcal{G}_A$ , we derive multiple complementary subgraph-based views to support robust reasoning. Specifically, we define two types of views: (i) **Temporal view.**  $\Phi_{\text{temp}}(\mathcal{G}_A)$  denotes a subgraph that captures the global chronological structure of events within the queried time window. It is constructed by selecting events and temporal edges from  $\mathcal{G}_A$  whose timestamps fall within the query scope, thereby representing the overall temporal progression of relevant facts. (ii) **Entity-centric views.** For each entity  $e_l$ ,  $\Phi_{\text{ent}}^l(\mathcal{G}_A)$  denotes a subgraph that captures the evolution of that entity over time. This view consists of all events involving entity  $e_l$ , together with the entity-specific temporal edges that link these events in chronological order.

Each view is serialized into text and used as contextual evidence for reasoning. The final prediction is obtained by conditioning the model on the input query together with these multi-perspective representations:

$$A = \mathcal{M}\left(q_{\text{raw}}, \Phi_{\text{temp}}(\mathcal{G}_A), \{\Phi_{\text{ent}}^l(\mathcal{G}_A)\}_{l=1}^L \mid \mathcal{P}_4\right), \quad (10)$$

where  $A$  denotes the predicted answer and  $\mathcal{P}_4$  denotes the prompt used for inference.

By explicitly structuring both global temporal evolution and entity-specific trajectories prior to generation, this formulation enables the model to reason over evolving knowledge more reliably and to produce temporally consistent answers without any additional parameter updates.

## 4 Experiments

### 4.1 Setup

We evaluate several representative methods commonly used to address knowledge drift, together with the new baseline introduced in this work. We consider three large language models (LLMs) covering both proprietary and open-source settings: *GPT-4o* (gpt-4o-2024-11-20), *Claude* (claude-3-haiku-20240307), and *LLaMA-3.1* (Meta-Llama-3.1-8B-Instruct). These models span different architectures, training scales, and accessibility levels.

Their respective knowledge cutoff dates are October 2023 (*GPT-4o*), August 2023 (*Claude*), and December 2023 (*LLaMA-3.1*), all preceding our benchmark time range starting from January 2024. This setup ensures that none of the models has di-

Table 2: Performance comparison across historical, contemporary, and commonsense question settings. Higher values indicate better accuracy. Best results for each model are shown in **bold**, and second-best results are underlined. **Green** values indicate improvements over the strongest corresponding baseline. *Chronos* achieves consistent gains across multiple evaluation settings, while other baselines exhibit varying limitations.

Method	Historical	Contemporary			Commonsense	Overall
		C1	C2	C3		
<i>Direct Generation</i>						
LLaMA-3.1	45.95	11.31	1.38	1.69	45.53	21.17
Claude	<u>55.86</u>	10.53	2.75	3.39	62.55	27.02
GPT-4o	<u>61.26</u>	14.81	3.21	5.08	67.32	30.34
<i>Parametric Updating Methods</i>						
ROME <sub>LLaMA-3.1</sub>	10.81	2.92	0.92	0.00	44.92	11.91
MEMIT <sub>LLaMA-3.1</sub>	25.23	7.99	1.38	0.00	42.59	15.44
WISE <sub>LLaMA-3.1</sub>	<u>46.85</u>	11.11	1.83	1.69	46.51	21.60
LoRA FT <sub>LLaMA-3.1</sub>	35.14	71.15	4.13	10.17	40.27	32.17
<i>Retrieval-based Methods</i>						
Vanilla RAG <sub>LLaMA-3.1</sub>	10.81	<u>80.31</u>	39.91	16.95	45.41	38.68
Vanilla RAG <sub>Claude</sub>	27.03	84.02	44.50	23.73	63.53	48.56
Vanilla RAG <sub>GPT-4o</sub>	9.91	86.74	50.46	25.42	64.87	47.48
ReAct RAG <sub>LLaMA-3.1</sub>	6.31	64.52	<u>47.25</u>	<u>32.20</u>	<u>47.98</u>	<u>39.65</u>
ReAct RAG <sub>Claude</sub>	18.92	<u>91.62</u>	<u>59.63</u>	<u>47.46</u>	<u>67.07</u>	<u>56.94</u>
ReAct RAG <sub>GPT-4o</sub>	7.21	<u>93.37</u>	<u>56.42</u>	<u>45.76</u>	<u>69.89</u>	<u>54.53</u>
<i>Chronos (Ours)</i>						
Chronos <sub>LLaMA-3.1</sub>	<b>50.45</b> <sup>+3.6</sup>	<b>92.40</b> <sup>+12.1</sup>	<b>63.76</b> <sup>+16.5</sup>	<b>61.02</b> <sup>+28.8</sup>	<b>50.55</b> <sup>+2.6</sup>	<b>63.64</b> <sup>+24.0</sup>
Chronos <sub>Claude</sub>	<b>57.66</b> <sup>+1.8</sup>	<b>93.76</b> <sup>+2.1</sup>	<b>66.97</b> <sup>+7.3</sup>	<b>74.58</b> <sup>+27.1</sup>	<b>72.34</b> <sup>+5.3</sup>	<b>73.06</b> <sup>+16.1</sup>
Chronos <sub>GPT-4o</sub>	<b>63.96</b> <sup>+2.7</sup>	<b>96.30</b> <sup>+2.9</sup>	<b>66.51</b> <sup>+10.1</sup>	<b>71.19</b> <sup>+25.4</sup>	<b>71.36</b> <sup>+1.5</sup>	<b>73.86</b> <sup>+19.3</sup>

rect access to the evaluated knowledge during pre-training. Answers are evaluated using exact-match accuracy for all QA tasks. More details of the experimental setup can be found in Appendix D.

## 4.2 Baselines

We evaluate the following baselines. *Direct generation* assesses pretrained models without any external knowledge updates. *Parametric methods* include representative knowledge editing and fine-tuning approaches, such as ROME (Meng et al., 2022a), MEMIT (Meng et al., 2022b), WISE (Wang et al., 2024), and LoRA-based fine-tuning (Hu et al., 2022), which update model parameters to incorporate new information. All of these are implemented on *LLaMA-3.1*. *Retrieval-based methods* augment generation with external knowledge at inference time, including vanilla RAG (Lewis et al., 2020) and ReAct-style RAG (Yao et al., 2022), which incorporates self-reflective reasoning to iteratively guide retrieval. These methods are instantiated with different backbone models.

## 4.3 Main Results

Table 2 reports the performance of different adaptation paradigms under historical, contemporary, and commonsense evaluation settings. Several key

observations can be drawn from the results.

Direct generation by LLMs fails to cope with continuous knowledge drift, as such models lack access to up-to-date information. Their reasonable performance on historical and commonsense questions reflects intact general reasoning ability and thus provides a baseline for assessing the impact of newly emerging knowledge in other methods.

**Takeaway #1: Parametric updating exhibits limited scalability under evolving knowledge.**

Parametric updating methods exhibit limited robustness under continuous knowledge drift, with some showing clear degradation in basic reasoning ability, suggesting catastrophic forgetting.

Among these methods, LoRA fine-tuning shows slightly better performance, while other parametric approaches fail to deliver meaningful improvements. However, even LoRA fine-tuning still harms performance on historical and commonsense questions. Although prior work has shown that knowledge editing and fine-tuning can yield improvements in isolated settings, their effectiveness remains limited when adapting to continuous knowledge drift.

These results suggest that parameter-level up-

dates struggle to preserve previously acquired knowledge while incorporating new information. Moreover, because parametric methods typically update the model either one instance at a time or in a batch-wise manner, newly acquired knowledge can overwrite both the model’s original old knowledge and earlier updates, leading to additional forgetting over time. An error case analysis and further discussion of knowledge editing methods are provided in Appendix E.

**Takeaway #2: Traditional RAG Lacks Flexibility in Complex Temporal Reasoning.**

Vanilla RAG and ReAct-style RAG show substantial improvements on contemporary queries, but their performance degrades on historical questions. Their effectiveness also remains limited in complex temporal reasoning settings.

The poor performance in C2 and C3 can be attributed to the need to aggregate multiple pieces of evidence. These settings require reasoning over information from multiple timestamps or entities and often involve ordering events across time. A single query is insufficient to capture such complexity. Even though ReAct-style RAG can retrieve additional evidence through iterative querying, the resulting retrieved content is often noisy and unstructured, making it difficult for the model to form a coherent temporal understanding. While ReAct-style RAG performs well on commonsense queries due to its multi-step reasoning process, this advantage does not directly translate to complex temporal reasoning scenarios.

**Takeaway #3: Explicitly modeling event evolution enables more reliable reasoning under continuous knowledge drift.**

*Chronos* consistently delivers strong and stable performance across settings and backbone models, particularly in scenarios that require aggregating and ordering multiple time-stamped facts, while maintaining performance on historical knowledge and commonsense reasoning.

This suggests that explicitly organizing retrieved evidence into an *Event Evolution Graph* (EEG), rather than merely retrieving relevant documents, helps preserve previously acquired knowledge while supporting temporally coherent reasoning. By structuring evidence along its temporal evolu-

Table 3: Ablation study of *Chronos*. Each value reports accuracy, followed by its change relative to the full model. Green values indicate performance improvements, while red values indicate performance drops.

Method	Hist.	Contemporary			Comm.
		C1	C2	C3	
<b>Chronos (Ours)</b>	<b>63.96</b>	<b>96.30</b>	<b>66.51</b>	<b>71.19</b>	<b>71.36</b>
- Time-aware Retr.	62.16 <sub>-1.8</sub>	92.01 <sub>-4.3</sub>	57.80 <sub>-8.7</sub>	59.32 <sub>-11.9</sub>	71.76 <sub>+0.4</sub>
- History Recon.	6.31 <sub>-57.7</sub>	97.66 <sub>+1.4</sub>	67.43 <sub>+0.9</sub>	72.89 <sub>+1.7</sub>	71.69 <sub>+0.3</sub>
- Event Augment.	62.16 <sub>-1.8</sub>	92.53 <sub>-3.8</sub>	55.05 <sub>-11.5</sub>	52.54 <sub>-18.7</sub>	70.50 <sub>-0.9</sub>
- Temporal View	61.26 <sub>-2.7</sub>	90.25 <sub>-6.1</sub>	58.26 <sub>-8.3</sub>	55.93 <sub>-15.3</sub>	70.65 <sub>-0.7</sub>
- Entity View	60.36 <sub>-3.6</sub>	89.86 <sub>-6.4</sub>	49.54 <sub>-17.0</sub>	50.85 <sub>-20.3</sub>	71.32 <sub>-0.0</sub>

tion and serializing it into multi-perspective views, the model reduces interference between old and new information, leading to more stable and reliable behavior under continuous knowledge drift. A case study is shown in Appendix F.

**4.4 Ablation Study**

Table 3 presents an ablation study demonstrating that all components contribute to the overall performance improvements of *Chronos*. The variations observed on commonsense QA are negligible.

Time-aware retrieval helps retrieve more temporally relevant evidence, enabling more precise extraction of information within the target time window. While historical reconstruction may introduce some interference when answering about contemporary queries, this effect is expected and difficult to completely avoid. However, historical reconstruction plays a crucial role in improving performance on historical questions. Event augmentation further enriches the EEG by supplementing additional relevant knowledge, leading to more informative representations. Moreover, the temporal-view and entity-view components provide two different perspectives for serializing the EEG, jointly offering the model a more structured and informative context. Both views are necessary and mutually complementary for effective reasoning.

These components help the method progressively construct and refine the EEG, enabling more robust and temporally consistent reasoning under continuous knowledge drift.

**5 Conclusion**

This paper studies the limitations of current adaptation paradigms for LLMs under continuous knowledge drift. We introduce a new benchmark of real-world dynamic events that captures how knowledge evolves over time and enables systematic evaluation of temporal consistency in model reasoning.

Our results show that existing approaches struggle to maintain coherent understanding as knowledge evolves. To address this, we propose a time-aware retrieval baseline based on an Event Evolution Graph, enabling temporally consistent reasoning without modifying model parameters. We believe this work provides a useful foundation for future research on adapting LLMs to continuously evolving real-world knowledge.

## Limitations

To focus on recent and realistic knowledge changes, our benchmark is constructed under the assumption that the underlying language models have a knowledge cutoff before 2024, while the evaluation data cover up-to-date knowledge from 2024 to 2025. In real-world applications, however, the temporal scope of relevant knowledge may vary across domains and deployment settings, and should be adjusted accordingly. Moreover, practical industrial scenarios often involve more complex and heterogeneous knowledge dynamics than those captured in our benchmark, calling for richer and more challenging test cases in future work.

## Acknowledgments

This work is supported by the Natural Science Foundation of China (Grant 62371270).

## References

- Abdelrahman Abdallah, Bhawna Piryani, Jonas Wallat, Avishek Anand, and Adam Jatowt. 2025. Extending dense passage retrieval with temporal information. *arXiv e-prints*, pages arXiv–2502.
- Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, and 1 others. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.
- Hoyeon Chang, Jinho Park, Seonghyeon Ye, Sohee Yang, Youngkyung Seo, Du-Seong Chang, and Minjoon Seo. 2024. How do large language models acquire factual knowledge during pretraining? *Advances in neural information processing systems*, 37:60626–60668.
- Zheng Chu, Jingchang Chen, Qianglong Chen, Weijiang Yu, Haotian Wang, Ming Liu, and Bing Qin. 2024. Timebench: A comprehensive evaluation of temporal reasoning abilities in large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1204–1228.
- Nicola De Cao, Wilker Aziz, and Ivan Titov. 2021. Editing factual knowledge in language models. *arXiv preprint arXiv:2104.08164*.
- Guanting Dong, Jiajie Jin, Xiaoxi Li, Yutao Zhu, Zhicheng Dou, and Ji-Rong Wen. 2025. Rag-critic: Leveraging automated critic-guided agentic workflow for retrieval augmented generation. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3551–3578.
- Wenqi Fan, Yujuan Ding, Liangbo Ning, Shijie Wang, Hengyun Li, Dawei Yin, Tat-Seng Chua, and Qing Li. 2024. A survey on rag meeting llms: Towards retrieval-augmented large language models. In *Proceedings of the 30th ACM SIGKDD conference on knowledge discovery and data mining*, pages 6491–6501.
- Jiacheng Guo, Yue Wu, Jiahao Qiu, Kaixuan Huang, Xinzhe Juan, Ling Yang, and Mengdi Wang. 2025. Temporal consistency for llm reasoning process error identification. *arXiv preprint arXiv:2503.14495*.
- Jinwen He, Yujia Gong, Zijin Lin, Cheng’an Wei, Yue Zhao, and Kai Chen. 2024. Llm factoscope: Uncovering llms’ factual discernment through measuring inner states. In *Findings of the Association for Computational Linguistics ACL 2024*, pages 10218–10230.
- Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, and 1 others. 2022. Lora: Low-rank adaptation of large language models. *ICLR*, 1(2):3.
- Joel Jang, Seonghyeon Ye, Changho Lee, Sohee Yang, Joongbo Shin, Janghoon Han, Gyeonghun Kim, and Minjoon Seo. 2022. Temporalwiki: A lifelong benchmark for training and evaluating ever-evolving language models. *arXiv preprint arXiv:2204.14211*.
- Soyeong Jeong, Jinheon Baek, Sukmin Cho, Sung Ju Hwang, and Jong C Park. 2024. Adaptive-rag: Learning to adapt retrieval-augmented large language models through question complexity. *arXiv preprint arXiv:2403.14403*.
- Pengcheng Jiang, Lang Cao, Ruike Zhu, Minhao Jiang, Yunyi Zhang, Jimeng Sun, and Jiawei Han. 2025a. Ras: Retrieval-and-structuring for knowledge-intensive llm generation. *arXiv preprint arXiv:2502.10996*.
- Pengcheng Jiang, Jiacheng Lin, Lang Cao, R. Tian, S. Kang, Z. Wang, Jimeng Sun, and Jiawei Han. 2025b. Deepretrieval: Hacking real search engines and retrievers with large language models via reinforcement learning. *arXiv preprint arXiv:2503.00223*.
- Yuxin Jiang, Yufei Wang, Chuhan Wu, Wanjun Zhong, Xingshan Zeng, Jiahui Gao, Liangyou Li, Xin Jiang, Lifeng Shang, Ruiming Tang, and 1 others. 2024. Learning to edit: Aligning llms with knowledge editing. *arXiv preprint arXiv:2402.11905*.

- Bernal Jimenez Gutierrez, Yiheng Shu, Yu Gu, Michihiro Yasunaga, and Yu Su. 2024. Hipporag: Neurobiologically inspired long-term memory for large language models. *Advances in Neural Information Processing Systems*, 37:59532–59569.
- Nattiya Kanhabua and Avishek Anand. 2016. Temporal information retrieval. In *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval*, pages 1235–1238.
- Zixuan Ke, Yijia Shao, Haowei Lin, Tatsuya Konishi, Gyuhak Kim, and Bing Liu. 2023. Continual pre-training of language models. *arXiv preprint arXiv:2302.03241*.
- Omer Levy, Minjoon Seo, Eunsol Choi, and Luke Zettlemoyer. 2017. Zero-shot relation extraction via reading comprehension. *arXiv preprint arXiv:1706.04115*.
- Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, and 1 others. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in neural information processing systems*, 33:9459–9474.
- Chen-An Li and Hung-Yi Lee. 2024. Examining forgetting in continual pre-training of aligned large language models. *arXiv preprint arXiv:2401.03129*.
- Hongyu Li, Liang Ding, Meng Fang, and Dacheng Tao. 2024a. Revisiting catastrophic forgetting in large language model tuning. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 4297–4308.
- Xiaopeng Li, Shasha Li, Shezheng Song, Jing Yang, Jun Ma, and Jie Yu. 2024b. Pmet: Precise model editing in a transformer. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 18564–18572.
- Stephanie Lin, Jacob Hilton, and Owain Evans. 2022. Truthfulqa: Measuring how models mimic human falsehoods. In *Proceedings of the 60th annual meeting of the association for computational linguistics (volume 1: long papers)*, pages 3214–3252.
- Hanbing Liu, Lang Cao, Yuanyi Ren, Mengyu Zhou, Haoyu Dong, Xiaojun Ma, Shi Han, and Dongmei Zhang. 2025a. Bingo: Boosting efficient reasoning of llms via dynamic and significance-based reinforcement learning. *arXiv preprint arXiv:2506.08125*.
- Hanbing Liu, Huaze Tang, Yanru Wu, Yang Li, and Xiao-Ping Zhang. 2025b. Reinforced domain selection for continuous domain adaptation. In *ICASSP 2025-2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5. IEEE.
- Hanbing Liu, Jingge Wang, Xuan Zhang, Ye Guo, and Yang Li. 2024. Enhancing continuous domain adaptation with multi-path transfer curriculum. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pages 286–298. Springer.
- Yun Luo, Zhen Yang, Fandong Meng, Yafu Li, Jie Zhou, and Yue Zhang. 2025. An empirical study of catastrophic forgetting in large language models during continual fine-tuning. *IEEE Transactions on Audio, Speech and Language Processing*.
- Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. 2022a. Locating and editing factual associations in gpt. *Advances in neural information processing systems*, 35:17359–17372.
- Kevin Meng, Arnab Sen Sharma, Alex J Andonian, Yonatan Belinkov, and David Bau. 2022b. Mass-editing memory in a transformer. In *The Eleventh International Conference on Learning Representations*.
- Eric Mitchell, Charles Lin, Antoine Bosselut, Chelsea Finn, and Christopher D Manning. 2021. Fast model editing at scale. *arXiv preprint arXiv:2110.11309*.
- Jie Ouyang, Tingyue Pan, Mingyue Cheng, Ruiran Yan, Yucong Luo, Jiaying Lin, and Qi Liu. 2025. Hoh: A dynamic benchmark for evaluating the impact of outdated information on retrieval-augmented generation. *arXiv preprint arXiv:2503.04800*.
- Hongjin Qian, Peitian Zhang, Zheng Liu, Kelong Mao, and Zhicheng Dou. 2024. Memorag: Moving towards next-gen rag via memory-inspired knowledge discovery. *arXiv preprint arXiv:2409.05591*, 1.
- Yifu Qiu, Zheng Zhao, Yftah Ziser, Anna Korhonen, Edoardo Ponti, and Shay B Cohen. 2024. Are large language model temporally grounded? In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 7064–7083.
- Kartik Sharma, Peeyush Kumar, and Yunqing Li. 2025. Og-rag: Ontology-grounded retrieval-augmented generation for large language models. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 32950–32969.
- Haizhou Shi, Zihao Xu, Hengyi Wang, Weiyi Qin, Wenyan Wang, Yibin Wang, Zifeng Wang, Sayna Ebrahimi, and Hao Wang. 2025. Continual learning of large language models: A comprehensive survey. *ACM Computing Surveys*, 58(5):1–42.
- Jiankai Sun, Chuanyang Zheng, Enze Xie, Zhengying Liu, Ruihang Chu, Jianing Qiu, Jiaqi Xu, Mingyu Ding, Hongyang Li, Mengzhe Geng, and 1 others. 2025. A survey of reasoning with foundation models: Concepts, methodologies, and outlook. *ACM Computing Surveys*, 57(11):1–43.

- Lukas Thede, Karsten Roth, Matthias Bethge, Zeynep Akata, and Thomas Hartvigsen. 2025. Wikibigedit: Understanding the limits of lifelong knowledge editing in llms. In *Forty-second International Conference on Machine Learning*.
- Mingyang Wang, Alisa Stoll, Lukas Lange, Heike Adel, Hinrich Schütze, and Jannik Strötgen. 2025. Bring your own knowledge: A survey of methods for llm knowledge expansion. *arXiv preprint arXiv:2502.12598*.
- Peng Wang, Zexi Li, Ningyu Zhang, Ziwen Xu, Yunzhi Yao, Yong Jiang, Pengjun Xie, Fei Huang, and Hua-jun Chen. 2024. Wise: Rethinking the knowledge memory for lifelong model editing of large language models. *Advances in Neural Information Processing Systems*, 37:53764–53797.
- Yuqing Wang and Yun Zhao. 2024. Tram: Benchmarking temporal reasoning for large language models. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 6389–6415.
- Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik R Narasimhan, and Yuan Cao. 2022. React: Synergizing reasoning and acting in language models. In *The eleventh international conference on learning representations*.
- Yue Yu, Wei Ping, Zihan Liu, Boxin Wang, Jiaxuan You, Chao Zhang, Mohammad Shoeybi, and Bryan Catanzaro. 2024. Rankrag: Unifying context ranking with retrieval-augmented generation in llms. *Advances in Neural Information Processing Systems*, 37:121156–121184.
- Junhao Zheng, Shengjie Qiu, Chengming Shi, and Qianli Ma. 2025a. Towards lifelong learning of large language models: A survey. *ACM Computing Surveys*, 57(8):1–35.
- Junhao Zheng, Chengming Shi, Xidi Cai, Qiuke Li, Duzhen Zhang, Chenxing Li, Dong Yu, and Qianli Ma. 2025b. Lifelong learning of large language model based agents: A roadmap. *arXiv preprint arXiv:2501.07278*.

## Contents of Appendix

---

<b>A Ethical Considerations</b>	<b>12</b>
<b>B Related Work</b>	<b>12</b>
<b>C Statistics of the Benchmark Dataset</b>	<b>13</b>
<b>D Details of Experiment Setup</b>	<b>13</b>
<b>E Error Case Analysis of Knowledge Editing</b>	<b>14</b>
<b>F Case Study</b>	<b>14</b>
<b>G Prompt Design</b>	<b>14</b>

---

### A Ethical Considerations

This work focuses on benchmarking and modeling the ability of large language models to reason over continuously evolving real-world knowledge. All data used in our benchmark are derived from publicly available sources and do not contain private, sensitive, or personally identifiable information. The benchmark is constructed to evaluate temporal reasoning and knowledge consistency rather than to profile individuals or make consequential decisions.

While our framework aims to improve temporal consistency under knowledge drift, it does not guarantee complete correctness, and outdated or inaccurate outputs may still occur, especially in high-stakes settings. Accordingly, systems based on this approach should be deployed with appropriate human oversight.

Overall, we view this work as a methodological step toward better understanding and evaluating temporal adaptation in large language models, rather than as a standalone solution for real-world deployment.

### B Related Work

**Continual Learning of LLMs.** Prior work has explored adapting language models to evolving knowledge through continual learning, where model parameters are updated as new data becomes available (Shi et al., 2025; Zheng et al., 2025a,b; Hu et al., 2022; Liu et al., 2025b). Continual pre-training is one strategy for adapting LLMs to new domains or data streams, but it has been shown to incur substantial computational cost and can lead to catastrophic forgetting of previously learned

information (Ke et al., 2023; Li and Lee, 2024). Empirical studies further indicate that sequential fine-tuning and continual updates are prone to degrade performance on previously learned knowledge (Luo et al., 2025; Li et al., 2024a; Liu et al., 2024), highlighting the challenge of retaining historical knowledge alongside new information.

More recently, knowledge editing methods have been proposed as a lightweight alternative that directly modifies a small subset of model parameters to update specific facts (De Cao et al., 2021; Jiang et al., 2024; Meng et al., 2022a,b; Mitchell et al., 2021; Li et al., 2024b). Representative locate-then-edit approaches such as ROME and MEMIT identify and modify weights corresponding to targeted facts for efficient single or batched updates (Meng et al., 2022a,b), while other variants such as PMET offer subject-centric editing with precise control over localized changes (Li et al., 2024b). Extensions toward sequential or lifelong editing, such as WISE, introduce dual memory schemes and knowledge sharding to support repeated edits while balancing reliability, generalization, and locality (Wang et al., 2024). While effective for isolated edits, existing editing methods primarily focus on overwriting a model’s current beliefs rather than representing multiple historical states of knowledge. As a result, repeated edits may interfere with one another, and edited models typically lack mechanisms to retain or reason about past information alongside newly updated facts.

**Retrieval-augmented Generation.** Retrieval-augmented generation (RAG) has emerged as an effective approach for enabling large language models to access external information beyond their fixed pretraining data (Lewis et al., 2020; Jiang et al., 2025b,a; Jimenez Gutierrez et al., 2024), thereby mitigating issues related to outdated or incomplete knowledge without retraining the model itself (Fan et al., 2024; Yu et al., 2024; Dong et al., 2025; Yao et al., 2022; Lewis et al., 2020). Recent work has explored mechanisms to adapt RAG systems more effectively to evolving information. For example, Adaptive-rag proposes strategies that adapt retrieval and generation to varying query complexities, enabling dynamic adjustment between iterative and single-step retrieval processes (Jeong et al., 2024). HoH introduce a dynamic benchmark highlighting how current RAG approaches struggle to handle outdated informa-

tion in retrieval and generation, demonstrating the need for improved update handling (Ouyang et al., 2025). On the modeling front, variants such as OG-RAG ground retrievals using ontological structures to better capture domain knowledge and improve contextual accuracy (Sharma et al., 2025). Other work, such as MemoRAG, incorporates long-term memory into the RAG pipeline to support improved retrieval and generation performance, particularly in complex tasks where conventional RAG may fall short (Qian et al., 2024). Despite these advances, most of these approaches assume a largely static retrieval corpus and do not explicitly model the temporal evolution of knowledge, limiting their ability to adapt to continuously drifting real-world information, which motivates our temporal retrieval framework.

### C Statistics of the Benchmark Dataset

The ten topical domains used during dataset construction are as follows:

1. *Politics & Government Leadership*
2. *Corporate Leadership & Executive Roles*
3. *Sports, Coaching Changes & Player Transfers*
4. *Regulation, Compliance, & Financial Penalties*
5. *Mergers, Acquisitions, & Strategic Partnerships*
6. *Product Releases & Version Updates*
7. *Natural Disasters & Climate Events*
8. *Public Health Events & Disease Outbreaks*
9. *Scientific Discoveries & Research Publications*
10. *Economic Indicators & Policy Changes*

After construction, each topical domain contains between 10 and 20 core subjects, and each subject is associated with approximately 3 to 7 temporally ordered events. In total, the benchmark comprises 513 knowledge quadruples. It includes 111 historical questions and 817 commonsense questions adopted from TruthfulQA (Lin et al., 2022). For contemporary questions, we construct three sub-categories: 513 single-timestamp questions (C1), 218 multi-timestamp questions (C2), and 59 multi-source questions (C3), each designed to evaluate increasingly complex forms of temporal reasoning.

### D Details of Experiment Setup

For generation, we set the temperature of all LLMs to 0 to ensure stable and deterministic outputs, while keeping all other hyperparameters at their

Table 4: An error case of parameter-based knowledge editing under continuous knowledge drift. While WISE produces the same answer for all time-specific queries due to overwriting effects, *Chronos* correctly returns different answers conditioned on the queried timestamp, demonstrating its ability to model temporal variation in evolving knowledge.

Query (C1)	WISE Output	Chronos Output
Who was the world’s richest person on Jan 1, 2024?	Elon Musk ✓	Elon Musk ✓
Who was the world’s richest person on Jan 28, 2024?	Elon Musk ×	Bernard Arnault ✓
Who was the world’s richest person on Mar 5, 2024?	Elon Musk ×	Jeff Bezos ✓
Who was the world’s richest person on Apr 2, 2024?	Elon Musk ×	Bernard Arnault ✓
Who was the world’s richest person on Jun 8, 2024?	Elon Musk ✓	Elon Musk ✓

default values. All prompts used in the experiments are provided in Appendix G.

For dense retrieval, we use `all-mpnet-base-v2` as a lightweight retriever across all retrieval-based methods, retrieving the top 4 documents for each query. The retrieval pipeline is implemented using LangChain<sup>1</sup>. For the ReAct baseline, the maximum number of reasoning steps is set to 3. We use the cosine similarity between the query and target document embeddings as the similarity score.

For knowledge-editing baselines, we perform edits on the target objects according to their corresponding time-stamped knowledge quadruples, with edits conditioned on temporal information. Specifically, for ROME, MEMIT, and WISE, we adopt the default hyperparameters reported in their original papers. For the LoRA-based fine-tuning baseline, we use a learning rate of  $1 \times 10^{-4}$ , a batch size of 8, and rank 8, selecting the best-performing checkpoint at epoch 5. All experiments are conducted on a single NVIDIA A800 80GB GPU.

For the Commonsense QA task, we adopt the multiple-choice questions (MC1-target) from TruthfulQA (Lin et al., 2022), using the V1 version as the evaluation dataset. All candidate answer options are explicitly provided in the prompt, and the model is instructed to select one option, allowing the responses to be evaluated using exact-match accuracy.

For our proposed baseline *Chronos*, we also retrieve the top 4 documents for each query. In the time-aware retrieval module, we set  $\alpha = 0.75$  and  $\tau_{\text{days}} = 180$ , which achieve the best performance in our validation experiments. We observe that the time-aware retrieval mechanism is not highly sensitive to these hyperparameters and can be tuned easily and intuitively within a reasonable range.

<sup>1</sup><https://www.langchain.com>

## E Error Case Analysis of Knowledge Editing

To better understand the limitations of parameter-based knowledge editing under continuous knowledge drift, we first highlight two fundamental challenges inherent to this paradigm. First, knowledge editing methods typically treat updated facts as static, rather than modeling them as temporally evolving entities. As a result, they lack mechanisms to represent multiple coexisting states of the same fact across different time points. Second, existing methods are designed to perform isolated edits and do not scale well to scenarios where many related facts must be updated jointly. When updates arrive sequentially, repeated parameter modifications can interfere with one another, leading to overwriting effects and the loss of previously edited knowledge.

To illustrate these limitations, we analyze a representative failure case of WISE (Wang et al., 2024). As shown in Table 4, the underlying knowledge concerns the identity of the world’s richest person at different time points. Although the dataset contains multiple time-stamped facts, WISE applies edits sequentially and updates model parameters in place. Consequently, later edits overwrite earlier ones, and the model collapses multiple temporal states into a single dominant answer. When queried with different time-specific questions (C1), the edited model repeatedly produces the same output corresponding to the most recent update, regardless of the queried timestamp.

This example directly reflects the two limitations outlined above. Because parameter-level editing does not explicitly model time, it cannot represent multiple coexisting states of the same fact, causing temporally distinct knowledge to collapse into a single outcome. Moreover, since edits are applied sequentially and independently, repeated updates interfere with one another, leading to overwriting effects when many related facts must be updated over time. Together, these behaviors illustrate why parameter-based editing struggles under continuous knowledge drift, and motivate the need for approaches that explicitly model temporal structure and preserve access to multiple historical states rather than encoding evolving knowledge implicitly in model parameters.

## F Case Study

Figure 3 presents a case study of *Chronos* for question answering under continuous knowledge drift.

The example involves a multi-source contemporary query that requires the model to jointly reason over changes in a company’s stock price and the corresponding changes in the world’s richest person. The figure illustrates the data flow through each stage of the framework, showing how evidence is progressively retrieved, augmented, and organized, ultimately leading to the correct answer.

## G Prompt Design

Figures 4, 5, and 6 present the prompt templates adopted by the LLM-based baseline methods. Figures 7, 8, 9, and 10 illustrate the prompts  $\mathcal{P}_1$ ,  $\mathcal{P}_2$ ,  $\mathcal{P}_3$ , and  $\mathcal{P}_4$  employed at different stages of our proposed method, *Chronos*.

**Input**

Raw Query: *Which company's stock surge led to Elon Musk losing his position as the world's richest person on September 10, 2025?*

---

**Query Analysis**

Entities: *company's stock, Elon Musk, the world's richest person*

Time window: *(2025-09-10, 2025-09-10)*

Time-agnostic Query: *Which company's stock surge led to Elon Musk losing his position as the world's richest person*

---

**Parallel Time-aware Retrieval**

*Elon Musk:*

*<World's Richest Person, held by, Elon Musk, 2024-01-01>*

*<World's Richest Person, held by, Elon Musk, 2024-06-08>*

*<World's Richest Person, held by, Elon Musk, 2025-09-11>*

...

*the world's richest person:*

*<World's Richest Person, held by, Bernard Arnault, 2024-01-28>*

*<World's Richest Person, held by, Jeff Bezos, 2024-03-05>*

*<World's Richest Person, held by, Bernard Arnault, 2024-04-02>*

*<World's Richest Person, held by, Larry Ellison, 2025-09-10>*

...

*company's stock:*

*<Oracle stock price, surged to, USD 328, 2025-09-10>*

...

---

**History Reconstruction**

*(no output, since the specified time window lies entirely after 2024)*

---

**Event Augmentation**

*<Larry Ellison, is cofounder of, Oracle, 1977-06-01>*

...

---

**Final Event Evolution Graph (Chronologically Ordered)**

*(1977-06-01) <Larry Ellison, is cofounder of, Oracle>*

...

*(2024-01-01) <World's Richest Person, held by, Elon Musk>*

*(2024-01-28) <World's Richest Person, held by, Bernard Arnault>*

*(2024-03-05) <World's Richest Person, held by, Jeff Bezos>*

*(2024-04-02) <World's Richest Person, held by, Bernard Arnault>*

*(2024-06-08) <World's Richest Person, held by, Elon Musk>*

*(2025-09-10) <World's Richest Person, held by, Larry Ellison>*

*(2025-09-10) <Oracle stock price, surged to, USD 328>*

*(2025-09-11) <World's Richest Person, held by, Elon Musk>*

...

---

**Final Response**

*On September 10, 2025, the record shows that Larry Ellison became the world's richest person. The evidence also states that on the same date, Oracle's stock price surged to USD 328. Since Larry Ellison is the cofounder of Oracle, the surge in Oracle's stock explains why his wealth overtook Elon Musk's, causing Musk to lose the top position.*


*Answer: Oracle (Correct *)

Figure 3: A case study of *Chronos* for question answering under continuous knowledge drift.

**Prompt of Direct Generation**

*# Task*  
Answer the question.

*# Question*  
**{question}**

*# Instructions*  
1. Use only your own knowledge (no external tools or retrieval).  
2. Return only the final answer as a word or noun phrase.  
3. If multiple answers are required, provide them as a single comma-separated list.  
4. Do NOT include explanations, reasoning, or any additional words.

Answer:

Figure 4: Prompt template used for direct generation baseline. Blue text indicates input variables.

**Prompt of RAG**

*# Task*  
Answer the question using *ONLY* the retrieved factual statements.

*# Question*  
**{query}**

*# Retrieved Facts*  
**{retrieved\_text}**

*# Instructions*  
1. Return *ONLY* the final answer as a word or noun phrase.  
2. If multiple answers are required, return a single comma-separated list (chronological order if applicable).  
3. Do NOT include explanations, reasoning, citations, or extra words.

Answer:

Figure 5: Prompt template used for retrieval-augmented generation baseline. Blue text indicates input variables.

```
Prompt of ReAct RAG

# Task
You are answering a question using ONLY retrieved factual statements.
You may either:
- propose a new retrieval query to fetch more facts, OR
- provide the final answer.

# Question
{question}

# Retrieved Facts So Far
{context}

# Instructions
1. You must output JSON with keys: action, search_query, final_answer.
2. Decide action:
- Choose action="final" ONLY if the retrieved facts explicitly contain the answer.
- Otherwise choose action="search".

3. If action="search", search_query MUST be a short keyword query (NOT a sentence), 3–10 tokens, and MUST include:
- the main entity in the question (who/what the question is about),
- the relation type (e.g., held_by / head coach / CEO / capital / founded),
- any time constraint if mentioned (year/month/date).
Set final_answer to "".

4. If action="final", final_answer must be ONLY the answer (a word or noun phrase; if multiple, comma-separated in chronological order).
Set search_query to "".

5. Do NOT include explanations, reasoning, citations, or extra words.

6. If the question is commonsense and does not involve temporal information, answer it directly without retrieval.

# Output JSON
Return strictly JSON:
{"action":"search","search_query":"...", "final_answer":""}
or
{"action":"final","search_query":"","final_answer":"..."}

Now respond:
```

Figure 6: Prompt template used for ReAct-style retrieval-augmented generation baseline. Blue text indicates input variables.

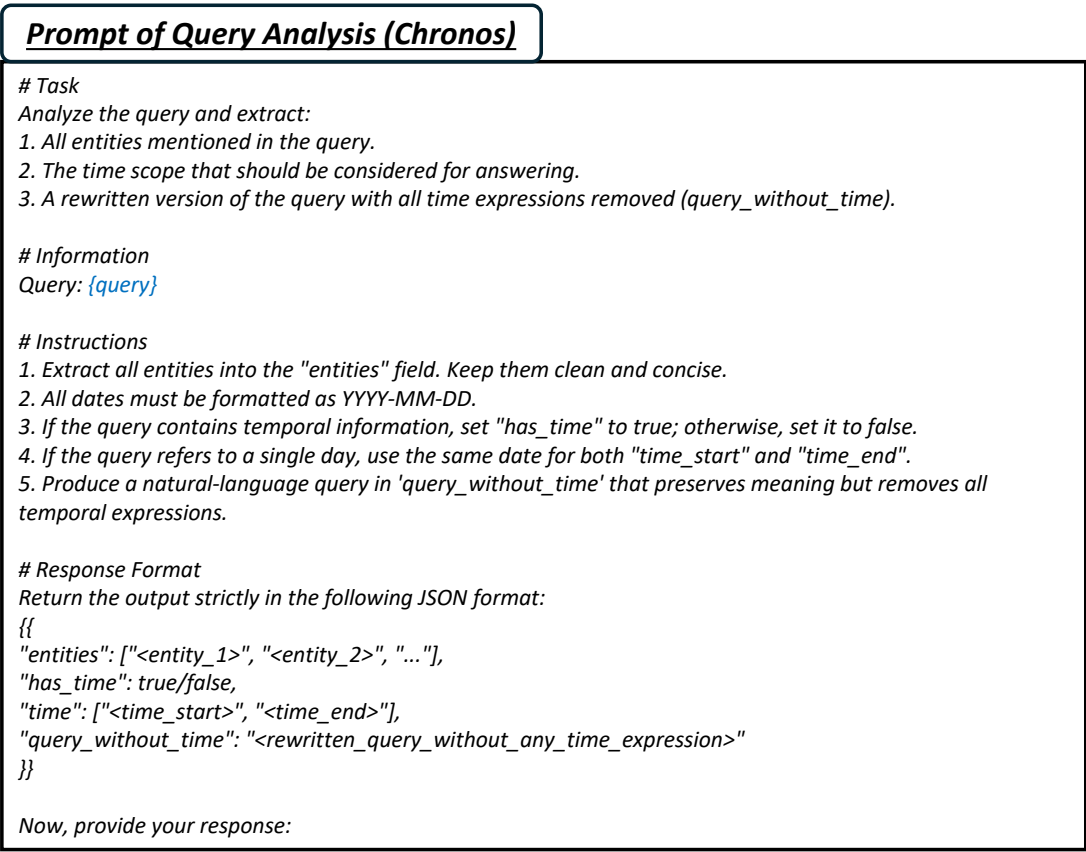


Figure 7: Prompt template used for query analysis in *Chronos*. Blue text indicates input variables.

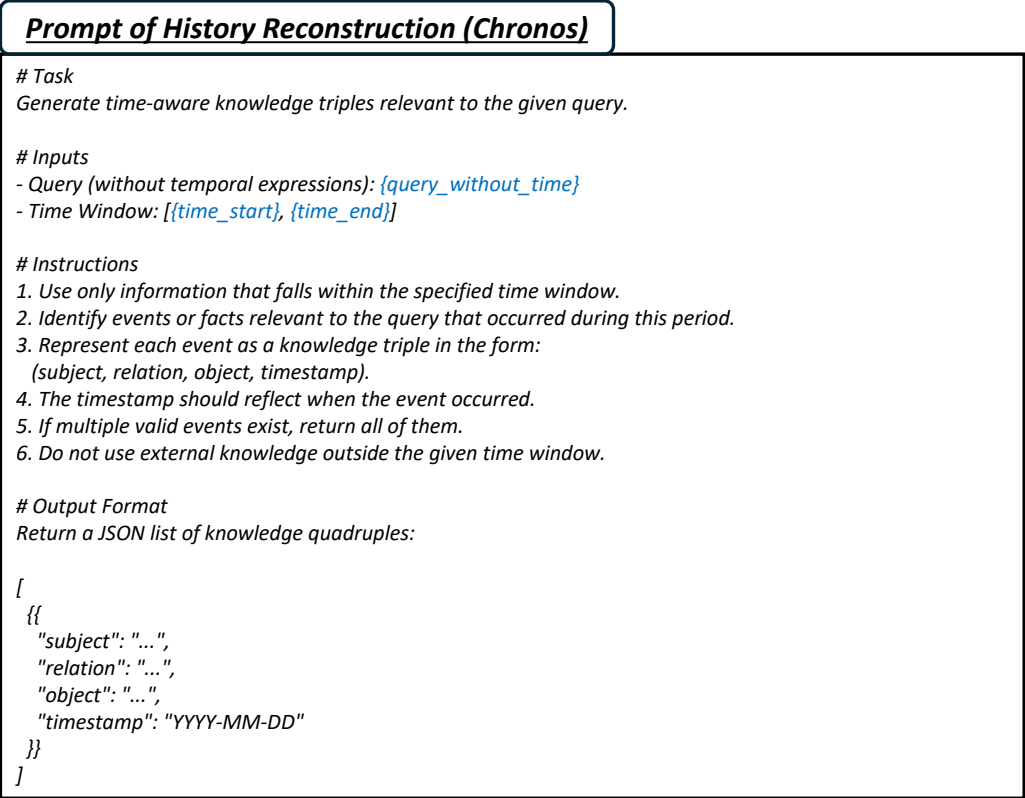


Figure 8: Prompt template used for history construction in *Chronos*. Blue text indicates input variables.

**Prompt of Event Augmentation (Chronos)**

```

# Task
Decide how to improve evidence coverage for answering the question.
You must choose exactly ONE action:
- retrieve: propose ONE focused search query to retrieve missing evidence.
- augment: directly propose additional knowledge quadruples to supplement the Event Evolution Graph (EEG).

# Question
{query}

# Current Evidence (from EEG)
{evidence_from_EEG}

# Proposed Answer
{initial_predicted_answer}

# Rules
1. Use ONLY the evidence above to judge whether something is missing. Do NOT use outside knowledge to invent facts.
2. If you choose action="retrieve":
   - search_query MUST be non-empty (a focused query string).
   - quads MUST be an empty list [].
3. If you choose action="augment":
   - quads MUST be non-empty (a list of knowledge quadruples).
   - search_query MUST be an empty string "".
4. Quadruple format:
   - timestamp MUST be in YYYY-MM-DD.
   - Each quad must be directly useful to answer the question or to connect key entities/events.
5. Output JSON ONLY. No explanations.

# Output JSON
{{
  "action": "retrieve" | "augment",
  "search_query": "",
  "quads": [{{"subject": "", "relation": "", "object": "", "timestamp": "YYYY-MM-DD"}}]
}}

```

Figure 9: Prompt template used for event augmentation in *Chronos*. Blue text indicates input variables.

**Prompt of Final Response (Chronos)**

```

# Task
Answer the question step by step using ONLY the provided information.

# Information
Question: {question}
Evidence: {evidence_from_EEG}

# Instructions
1. Use ONLY the information in Historical Context and Contemporary Information. Do NOT use outside knowledge.
2. You may reason freely.
3. The final line MUST be exactly: Answer: <final answer>
4. The final answer must be a word or noun phrase.
5. If multiple answers are required, output a single comma-separated list (noun phrases) in chronological order when applicable.
6. Do NOT write anything after the Answer line.

Now, answer:

```

Figure 10: Prompt template used for final response in *Chronos*. Blue text indicates input variables.