Multi-granularity Temporal Knowledge Editing over Large Language Models

Simiao Zhao[1*], **Ning Pang**[2,3*], **Zhen Tan**[1], **Yanli Hu**[1], **Weidong Xiao**[1], **Xiang Zhao**[4†]
[1] National Key Laboratory of Information Systems Engineering, National University of Defense Technology, China
[2] Aviation University of Air Force, China
[3] Jilin Provincial Key Laboratory of Unmanned Aerial Vehicle Intelligent Application, China
[4] National Key Laboratory of Big Data and Decision, National University of Defense Technology, China
{simiao_zhao2001, pangning14, tanzhen08a, huyanli, xiangzhao}@nudt.edu.cn, wilsonshaw@vip.sina.com

Abstract

The evolving worldly dynamics necessitate continuous revision and updating of knowledge within Large Language Models (LLMs), driving the development of Knowledge Editing (KE) techniques. Recently, a novel paradigm of Temporal Knowledge Editing (TKE) has been proposed, emphasizing that models deployed in dynamic environments should integrate new information while retaining historical knowledge. However, we observe that current definitions and methods for TKE are insufficient, as they do not effectively capture or adapt to the fine-grained temporal dynamics inherent in real-world knowledge evolution. In this paper, we introduce the notion of multi-granularity TKE, encompassing temporal knowledge across yearly, monthly, and daily granularities, and propose a corresponding dataset, named **MTKE**. We argue that comprehending and retaining knowledge across different temporal granularities is crucial for LLMs to accurately reflect real-world changes. The key challenge lies in integrating new temporal knowledge at various granularities while also preserving relevant historical knowledge, thus ensuring LLMs maintain a consistent and accurate understanding over time. To achieve this, we propose a **S**parse **P**arameter-**I**njected **K**nowledge **E**diting method, dubbed SPIKE, which anchors both temporal knowledge and subject positions within the model. Experiments demonstrate that our method effectively preserves historical knowledge performance while accurately incorporating dynamic temporal knowledge across multi-granularity temporal scenarios.

Datasets — https://github.com/LLMs-TKE/MTKE

Introduction

Large Language Models (LLMs) are widely acknowledged for their substantial factual knowledge gained from pretraining on large corpora (Brown et al. 2020; Ouyang et al. 2022; Touvron et al. 2023; Achiam et al. 2023). However, this knowledge tends to remain static once pre-training is complete. As the real world constantly evolves, much of the knowledge within these models becomes outdated, leading to discrepancies between the model's outputs and current facts. Fortunately, Knowledge Editing (KE) techniques provide a means to update existing knowledge and correct errors within the model without requiring costly retraining (Nonaka, Toyama, and Konno 2000; Roddick and Spiliopoulou 2002; Hoffart et al. 2011; Gottschalk and Demidova 2018; Chen, Liao, and Zhao 2023).

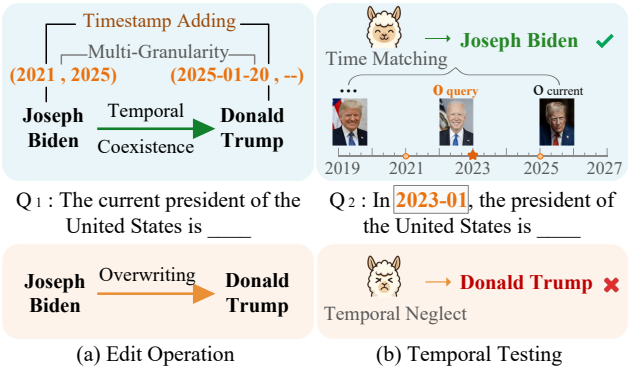


Figure 1: Illustration of TKE in LLMs. The upper row shows multi-granularity TKE, allowing factual coexistence with fine-grained timestamps, while the lower row shows traditional editing, which overwrites historical context with new facts. Both paradigms can correctly answer the current knowledge query (Q1), but only fine-grained TKE methods can correctly answer the historical knowledge query (Q2).

Most prevailing efforts in KE, including intrinsic knowledge editing methods (Huang et al. 2023; Wang et al. 2025) and external knowledge retrieval methods (Zheng et al. 2023; Mitchell et al. 2022), strive to effectively memorize new knowledge while keeping non-relevant knowledge unchanged (Pan et al. 2024). Nevertheless, these approaches are limited to static knowledge editing (Yin et al. 2024), as they largely neglect the crucial temporal dimension. In the dynamic world, factual knowledge evolves along a timeline. Therefore, the overlook of temporal information limits the model's ability to accurately reason about temporally contextualized knowledge, especially when multiple factual versions coexist (Zhao et al. 2025). To address the limitation, the concept of temporal knowledge editing (TKE) is introduced, and a corresponding solution AToKe, which considers temporal factors, is proposed (Yin et al. 2024).

However, akin to previous KE studies (Huang et al. 2024),

*These authors contributed equally.
†Corresponding author

for the evaluation questions, Yin et al. (2024) still fail to rigorously verify whether LLMs contain accurate historical knowledge while lacking the latest knowledge before editing. Therefore, the current evaluation datasets and metrics are inadequate for accurately assessing the effectiveness of TKE methods in integrating temporal knowledge.

Furthermore, the current formulation of TKE focuses on the coarse-grained temporal setting, which causes the model to confuse different factual versions and struggle with precise temporal alignment. As shown in Figure 1, when asked "Who is the president in 2023-01?", a model that has been edited only at the year level may exhibit temporal confusion, misinterpreting the query as referring to current time, such as 2025, and return "Donald Trump" instead of the correct answer "Joseph Biden". The challenge becomes even more pronounced for the query "Who is the president in 2025?", where the year marks a critical transition. Specifically, Donald Trump's term begins on 2025-01-20, meaning that without day- or month-level precision, the model cannot determine whether he is already in office at a given point in that year. These examples highlight the need for multi-granularity temporal modeling, as accurate temporal alignment and fact differentiation cannot be achieved with coarse-grained annotations (Zeng, Zhou, and Zhao 2024).

Confronted with these issues, we propose a novel KE task called **Multi-Granularity Temporal Knowledge Editing**, in which models must retain and reason about knowledge that evolves across multiple temporal granularities. To support this task, we construct the **MTKE** benchmark, which features a strict filtering mechanism to ensure high-quality evaluation samples, a wide range of relational types, and fine-grained temporal coverage with daily, monthly, and yearly timestamps. To tackle the challenges of multi-granularity temporal knowledge editing, we propose a lightweight **S**parse **P**arameter-**I**njected **K**nowledge **E**diting framework, namely **SPIKE**. SPIKE injects temporal knowledge by anchoring temporal markers (timestamps) and entities (subjects) in the model as semantic pivots, enabling the model to guide its outputs toward the intended knowledge when these pivots are encountered in context. This framework supports selective updates and the coexistence of multiple knowledge versions, together enabling precise reasoning over temporally evolving facts across granularities, which is essential for fine-grained temporal editing.

In this paper, our contributions are at least three-fold:

- We systematically identify two key limitations in current TKE methods: inadequate hallucination control which results in unreliable evaluation results, and lack of multi-granularity support which impedes accurate and faithful modeling of temporal knowledge.
- We are the first to propose the multi-granularity TKE benchmark **MTKE**, which features rigorous hallucination filtering, broad relational coverage, and fine-grained temporal annotations.
- A lightweight framework **SPIKE** is proposed for multi-granularity temporal knowledge editing, enabling selective updates, multi-version coexistence, and precise temporal reasoning across different temporal granularities.

Related Work

Datasets for Temporal Knowledge Editing

Existing Knowledge Editing (KE) datasets, such as **WikiData$_{\text{Recent}}$** (Cohen et al. 2024), **ZsRE** (Yao et al. 2023), **WikiBio** (Hartvigsen et al. 2023), and **COUNTERFACT** (Meng et al. 2022a), predominantly focus on static factual triples. Although these datasets facilitate general factual editing, their lack of temporal annotations prevents them from accurately reasoning about temporally contextualized knowledge, rendering them fundamentally unsuitable for Temporal Knowledge Editing (TKE).

To address the lack of temporal dynamics, **AToKe** (Yin et al. 2024) is introduced as the first benchmark for evaluating whether language models can effectively integrate temporally updated knowledge while preserving historical facts. However, AToKe only considers year-level temporal knowledge, thereby overlooking the need for multi-granularity temporal reasoning in real-world scenarios. To address this limitation, we introduce a novel benchmark **MTKE** featuring multi-granularity temporal labels, which is more suitable for evaluating knowledge editing in truly dynamic and realistic scenarios.

Methods for Knowledge Editing

Large Language Models (LLMs) have demonstrated remarkable performance across a broad range of NLP tasks (Yang et al. 2024). However, as time passes, the factual knowledge embedded within LLMs inevitably becomes outdated. Given that fully retraining these models to update obsolete information is computationally prohibitive, Knowledge Editing (KE) techniques have emerged as an efficient alternative for modifying model knowledge without exhaustive retraining. To systematically address this challenge, recent research (Pan et al. 2024) broadly categorizes KE into two main types: intrinsic editing (Huang et al. 2023; Wang et al. 2025) and external resorting (Zheng et al. 2023). Intrinsic methods, such as ROME (Meng et al. 2022a), MEMIT (Meng et al. 2022b), and MEND (Mitchell et al. 2021), directly modify model parameters to encode new knowledge. In contrast, external methods leverage auxiliary memory or prompt-based mechanisms at inference time (Zheng et al. 2023; Jiang et al. 2024).

Despite the advances in KE, these methods have yet to address how to preserve historical knowledge, which has been argued to be crucial for adapting to swift knowledge evolution in real-world (Yin et al. 2024). Recently, Yin et al. (2024) propose the METO framework, which introduces temporal elements into edited knowledge tuples and, for the first time, establishes a methodological foundation for Temporal Knowledge Editing (TKE). However, research on knowledge editing for temporally annotated knowledge at multiple granularities remains unexplored, which is critical for incorporating more precise temporal knowledge. To bridge this gap, we propose a lightweight framework **SPIKE** for multi-granularity temporal knowledge editing that allows precise reasoning over temporally evolving facts across granularities.

The MTKE Dataset

Analysis of Temporal Knowledge Editing

Temporal Knowledge Editing refers to revising and updating the knowledge within Large Language Models (LLMs) while considering the temporal context of that knowledge. Instead of directly overwriting historical information, TKE retains previously valid facts as historical knowledge and integrates new facts as temporally updated information, thereby enabling the coexistence of multiple factual versions over time. By organizing relevant knowledge along a timeline, the edited model can accurately predict facts for different time points.

We formally represent temporal knowledge as a quintuple (s, r, o, t_s, t_e), where s denotes the subject, r the relation, o the object, and $t = [t_s, t_e)$ specifies the period of temporal validity for the fact. For a given subject–relation pair (s, r), the associated object may change over different time periods, resulting in a series of factual instances. Building on previous studies, we define the temporal editing operation as $e : (s, r, o, t_s, t_e) \rightarrow (s, r, o', t'_s, t'_e)$, where the object associated with a given (s, r) transitions from o to o', valid within a new time period $t' = [t'_s, t'_e)$. It is important to note that the previous factual instance (s, r, o, t_s, t_u) is preserved as historical knowledge, ensuring the coexistence of multiple factual versions at different time points within the LLM, rather than merely overwriting past information.

That is, Temporal Knowledge Editing (TKE) not only involves incorporating the new object o' associated with (s, r) at its corresponding timestamp $t_{\text{new}} = [t'_s, t'_e)$, but also ensures that the previously valid object o remains accessible when queried within its temporal context $t_{\text{old}} = [t_s, t_e)$. This dual requirement sets TKE apart from conventional knowledge editing and introduces unique challenges, such as temporal alignment, factual coexistence, and reasoning across multiple temporal facts.

To systematically evaluate these challenges, we define two sub-settings within MTKE:

- **MTKE-SG (Single Granularity)**: Both the historical and updated facts are annotated using timestamps at the same granularity (e.g., year→year or month→month). This setting evaluates whether the model can perform consistent edits when the temporal granularity remains consistent between the two versions.
- **MTKE-MG (Multiple Granularities)**: The history and updated facts are annotated at different temporal granularities (e.g., year→day), requiring the model to integrate and reason across different temporal granularities.

While the recent benchmark AToKe (Yin et al. 2024) introduces temporal annotations in knowledge editing, it has two main limitations. First, without rigorous data filtering to verify the absence of target facts and retention of historical facts, spurious edits may occur, undermining evaluation reliability. Second, it concentrates on coarse-grained timestamps, limiting fine-grained temporal reasoning. Therefore, we propose MTKE to address these limitations by enforcing multi-granularity supervision and a rigorous filtering procedure, providing a more realistic and reliable benchmark for temporal knowledge editing.

Temporal Knowledge Collection

We construct MTKE using real-world temporal knowledge sources to support both single and multiple granularity settings. MTKE is composed of numerous quintuples (s, r, o, t_s, t_e), extracted from real-world temporal knowledge bases, including GDELT (Schrodt 2010), ICEWS14 (García-Durán, Dumančić, and Niepert 2018), ICEWS05-15 (García-Durán, Dumančić, and Niepert 2018), and YAGO15k (Liu et al. 2019), which provide broad temporal coverage and highly structured factual data. To ensure that the dataset is well-suited for TKE, we focus on sustained and non-transient relation types, and manually filter out bursty or isolated events that cannot support meaningful temporal evolution. By chronologically aggregating all quintuples that share the same (s, r) pair based on their temporal annotations, we construct fact chains that capture the evolution of factual knowledge over time, ultimately yielding a curated set of 17,060 chains that serve as a comprehensive benchmark for temporal knowledge editing tasks.

Each fact chain traces the evolution of a specific (s, r) pair over time. We then reformat these chains into a standard editing data structure by annotating both pre-edit and post-edit facts along with precise temporal information. For each instance, we generate both fill-in-the-blank prompts and time-sensitive questions automatically using GPT-3.5-turbo to facilitate comprehensive model evaluation.

For alias-aware evaluation, we construct canonical alias sets for each object entity using a four-step pipeline: (1) extracting raw alias terms; (2) normalizing them by lowercasing and removing symbols; (3) expanding the alias set via Wikidata; and (4) merging and filtering candidates based on LLM outputs and embedding-based soft matching. By harmonizing surface forms through this process, we ensure robust and accurate answer matching during evaluation.

Construction and Statistics of MTKE

We construct MTKE by explicitly annotating each fact with its precise temporal granularity, emphasizing fine-grained temporal reasoning, which distinguishes MTKE from previous datasets. After filtering temporal knowledge from multiple data sources, the finalized MTKE dataset consists of 17,060 event chains, encompassing temporal knowledge editing scenarios at yearly, monthly and daily granularity.

Table 1 provides an illustrative overview of the core structure of the MTKE dataset. To address potential biases in TKE evaluation, we introduce a dual-verification filtering procedure for candidate samples. For each sample, we assess the model's responses to multiple paraphrased prompts for both the historical and target facts, as exemplified by (a, b, c, d) in Table 1. A sample is retained only if the model consistently answers all questions about the historical fact correctly (indicating robust retention) and fails to answer all questions about the target fact (demonstrating absence of prior knowledge). This dual-verification pipeline ensures that editing tasks are both necessary and well-defined, thereby providing a more reliable and rigorous foundation for evaluation.

Specifically, for evaluation with GPT-J (6B), Table 2 shows that our filtering substantially reduces the number of

BK	**(USA, President_of, object, time_since, time_until)** – (Donald_Trump, 2017-01-20, 2021-01-20) – (Joseph_Biden, 2021-01-20, 2025-01-20) – (Donald_Trump, 2025-01-20, –)
SG	– (Joseph_Biden, 2021, 2025)-year → (Donald_Trump, 2025, –)-year
MG	– (Joseph_Biden, 2021, 2025)-year → (Donald_Trump, 2025-01-20, –)-day
Q	a. From 2021 to 2025, the president of the USA is <blank>. b. Who is the president of the United States from 2021 to 2025? c. During the period of 2021 to 2025, the individual holding the position of president of the United States is <blank>. d. Who is the current president of the United States? e. Who is the president of the United States on 2025-01-20? f. Who served as president of the United States in January, 2023?
A	a. Joseph_Biden (ALL) b. Joseph_Biden (ALL) c. Joseph_Biden (ALL) d. Donald_Trump (ALL) e. Temporal Confusion (SG); Joseph_Biden (MG) f. Joseph_Biden (ALL) —- fuzzy matching strategy for granularity rollback

Table 1: An example illustrates the construction of the MTKE dataset, which consists of two sub-datasets: MTKE-SG and MTKE-MG. In SG, old and new knowledge pairs share the same temporal granularity, whereas in MG, they correspond to different granularities. Q denotes the input prompt (query) presented to the model, and A represents the ideal answer corresponding to the label in query (Q).

samples, whereas the filtered MTKE dataset provides a pool of valid data that is approximately three times as large as that of AToKe, offering a richer basis for evaluation. In addition, the accuracy changes on the AToKe dataset, as defined in Table 3, further illustrate both the effectiveness of our filtering method and the presence of data bias in the original dataset. After filtering, ACC$current$ decreases and ACC$history$ increases substantially, confirming that data bias in the original dataset significantly influenced the evaluation of model performance. We will compare the filtered MTKE data with the model performance of Yin et al. (2024) in Table 4.

Besides, MTKE features two granularity settings: MTKE-SG and MTKE-MG. As shown in example (e) in Table 1, when the event time falls on a critical boundary (e.g., 2025-01-20), the MG setting allows for finer-grained factual differentiation, whereas the SG setting is prone to temporal confusion. This observed discrepancy underscores the need for robust multi-granularity reasoning in TKE. In addition, when encountering unseen temporal knowledge like example (f), our granularity fallback mechanism (detailed in Section *The SPIKE Model*) enables fuzzy temporal matching, guiding the model toward more accurate responses.

Dataset	Original	Filtered
AToKe-SE	8819	144
AToKe-ME	8819	112
MTKE-SG	6873	467
MTKE-MG	6347	362

Table 2: Comparison of AToKe and MTKE sample counts before and after applying the filtering mechanism.

Dataset	ACC$_{current}$	ACC$_{history}$
AToKe-SE (Original)	99.95	20.25
AToKe-SE (Filtered)	90.97	29.86
AToKe-ME (Original)	99.93	23.22
AToKe-ME (Filtered)	87.50	34.82

Table 3: Accuracy comparison on the AToKe dataset before and after filtering under our SPIKE method.

It is important to note that MTKE is constructed with broad and balanced relation coverage. Recall that previous datasets AToKe-SE and AToKe-ME (Yin et al. 2024) construct single-hop and multi-hop reasoning with the same set of 8,819 event chains and include only 9 relation types, the relation type distributions of which are highly imbalanced. To be specific, the <playsFor> relation accounts for 83.43% of all instances, while 5 relation types each constitute less than 0.5% of the data. Such an imbalance causes evaluation metrics to be dominated by performance on a single relation type, making it difficult to fairly assess the effectiveness of editing methods across diverse semantic relations. In contrast, MTKE covers 44 distinct relation types, with attention paid to balancing the sample distribution among them. After the selection, MTKE retains 6,873 valid MTKE-SG samples and 6,347 MTKE-MG samples. In the SG subset, the most frequent relation type accounts for only 29.90% of samples, and in the MG subset, the highest proportion is 33.18%. Additionally, 37 relation types each account for between 0.5% and 10% of the dataset, indicating that the majority of relation types are represented in a well-balanced manner.

Evaluation Metrics

Existing methods for evaluating temporal knowledge editing remain in the early stages of development. Prior research, such as that by Yin et al. (2024), has primarily focused on assessing the accuracy of responses to temporally anchored queries. In this context, we present a comprehensive overview of the evaluation metrics utilized in this study.

Soft Recall measures the degree to which the model's

confidence in the updated fact improves following the editing operation. This metric is defined as:

$$\text{SR} = \frac{1}{N}\sum_{i=1}^{N}\max\left(0, P^i_{\text{after-target}} - P^i_{\text{before-target}}\right), \quad (1)$$

where N is the number of evaluation samples. $P^i_{\text{before-target}}$ and $P^i_{\text{after-target}}$ respectively denote the model's confidence in the i-th updated fact before and after editing. A higher SR indicates better incorporation of newly injected knowledge into the model's internal representation.

In contrast, **Negative Soft Recall** measures the degree to which the model's confidence in historical knowledge diminishes due to the edit. It is defined as:

$$\text{Neg SR} = \frac{1}{N}\sum_{i=1}^{N}\max\left(0, P^i_{\text{before-historical}} - P^i_{\text{after-historical}}\right), \quad (2)$$

where $P^i_{\text{before-historical}}$ and $P^i_{\text{after-historical}}$ denote the model's confidence in the i-th historical fact before and after editing. A higher Neg SR indicates greater degradation of prior knowledge, highlighting the extent to which editing erases or disrupts historical information. Controlling the negative impact is especially important in intrinsic editing to enable the coexistence of new and old knowledge within the model.

Temporal Consistency (TC) measures whether the model correctly associates o with t and o' with t' without confounding the temporal contexts. The definitions of o, t, o', and t' are provided in Section *Analysis of Temporal Knowledge Editing*. We define TC as:

$$\text{TC} = \frac{C(o@t) + C(o'@t')}{N_{\text{count}}}, \quad (3)$$

where N_{count} is the total number of predictions involving o and o' at either t or t':

$$N_{\text{count}} = C(o@t) + C(o@t') + C(o'@t) + C(o'@t'). \quad (4)$$

A higher TC value indicates stronger alignment between each factual version and its appropriate timestamp, reflecting better temporal discrimination.

Temporal Target Recall (TTR) measures the degree to which the model's predictions are confined to the temporally relevant scope of the edited knowledge. It is defined as:

$$\text{TTR} = \frac{N_{\text{count}}}{N}, \quad (5)$$

where N_{count} is as defined in Eq. 4, and N denotes the total number of model predictions for all evaluated temporal queries, including both relevant and irrelevant outputs. A higher TTR value indicates that the model remains focused on relevant temporal knowledge and avoids producing irrelevant outputs outside the edited scope.

Harmonic Temporal Consistency (HTC) assesses the model's ability to simultaneously maintain temporal alignment and restrict its predictions to the relevant time interval after editing. It is defined by combining TC and TTR to provide a unified view of both alignment and relevance.

$$\text{HTC} = \frac{2 \cdot \text{TTR} \cdot \text{TC}}{\text{TTR} + \text{TC}}. \quad (6)$$

HTC penalizes the imbalance between temporal precision and output focus. For example, a model with high TC but low TTR may correctly assign facts to the appropriate times but also generate information unrelated to the relevant time interval. Conversely, a model with high TTR but low TC indicates good temporal focus but poor alignment. Therefore, a higher HTC reflects a more balanced and robust understanding of temporal knowledge after editing.

The SPIKE Model

SPIKE is a temporally aware knowledge editing framework that integrates new knowledge while preserving historical facts. As illustrated in Figure 2, SPIKE optimizes hidden representations at multiple anchor positions (start time, end time, and subject) within input sequences to support temporally precise knowledge editing. The optimized representations yield sparse parameter increments (deltas), which are stored in a key table for efficient knowledge updates. During inference, relevant deltas are selectively injected into specific model layers to enable precise parameter editing. SPIKE uses multi-granularity matching and a fallback mechanism, progressively relaxing the time interval until a suitable match is found. If a match is found, the corresponding increments are injected; otherwise, inference defaults to the original model to ensure historical consistency. This integrated design enables efficient, accurate, and robust temporal knowledge editing across diverse scenarios.

Loss Function Design and Sequential Prediction

The optimization process of SPIKE is driven by a loss function that jointly updates hidden states at three anchor positions (start time, end time, and subject) to support accurate target object generation. The loss function is defined as a weighted sum of the losses at the start time, end time, and subject anchor positions. Building upon Meng et al. (2022a), we introduce three anchor positions and calculate a joint loss over these anchors. Following the sequential order of start time, end time, and subject, we compute the hidden state increment for each anchor by minimizing the cross-entropy loss between the model's prediction and the ground-truth target object o'. The total loss L_{total} is thus given by:

$$L_{\text{total}} = \lambda_1 L_{\text{time\_start}} + \lambda_2 L_{\text{time\_end}} + \lambda_3 L_{\text{subject}}, \quad (7)$$

where each L term denotes the cross-entropy loss at the corresponding anchor position.

Sparse Knowledge Injection for Precise Editing

After computing the total loss, SPIKE determines optimal hidden state increments (Δh) for each anchor position by minimizing L_{total}:

$$\Delta h = \arg\min_{\Delta h}\left(L_{\text{total}}\right). \quad (8)$$

These increments are represented as sparse vectors and stored with keys composed primarily of subject and relation, along with associated target time intervals and granularity. In each Transformer layer of the LLM, the feed-forward network (FFN) consists of a fully-connected weight

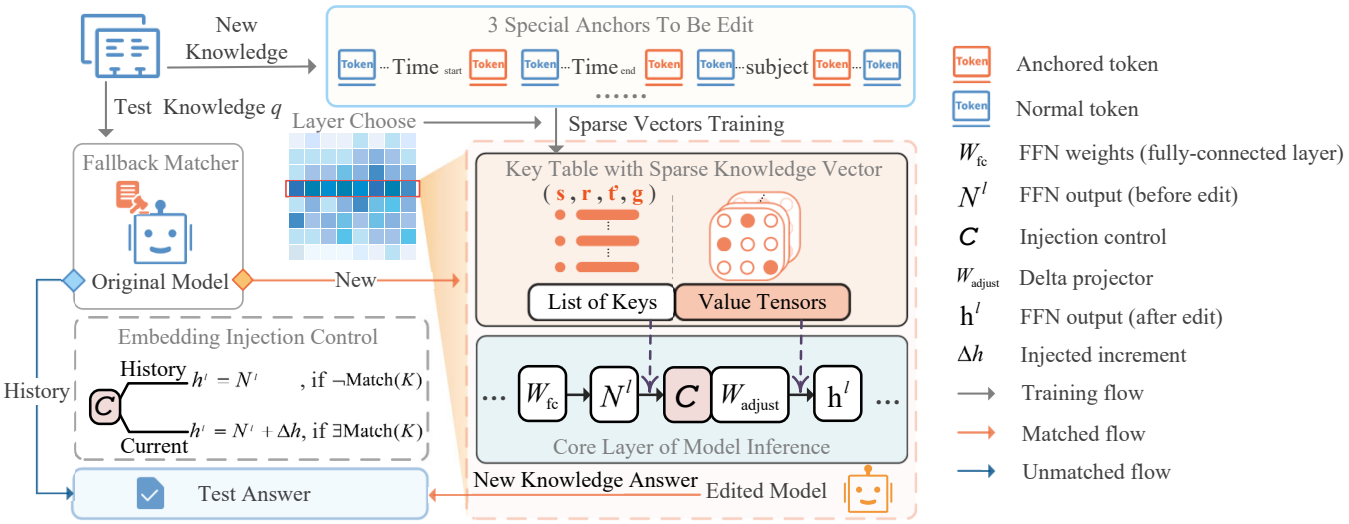


Figure 2: Overview of the SPIKE architecture. Knowledge editing is performed via anchor-based sparse vector injection and multi-granularity matching, enabling efficient updates while preserving historical information.

matrix W_{fc}, which produces the standard hidden representation N^l. To enable knowledge editing, we introduce an additional adjustment matrix W_{adjust}, which is used to project the retrieved increment vectors for precise injection. During inference, SPIKE retrieves relevant increments from the key table, processes them through an embedding injection controller C, and projects them with W_{adjust}. The resulting vectors are then injected into the original hidden representations at the output of the FFN sub-layer:

$$h^l = N^l + W_{adjust}(C(\Delta h)), \tag{9}$$

where N^l denotes the output of W_{fc} at layer l, and h^l is the edited hidden state after injection. This targeted mechanism ensures minimal yet precise parameter updates, effectively guiding the model to generate temporally coherent and semantically accurate target objects o'.

Temporal Matching and Fallback Mechanism

Real-world temporal knowledge spans multiple granularities, which requires flexible representation and retrieval. SPIKE addresses this by maintaining a multi-resolution key table K, where each entry corresponds to an edited knowledge tuple $(\text{subject}, \text{time}, \text{relation})$ and the time component is stored at multiple granularities g (e.g., day, month, year). This enables adaptive matching for fine-grained temporal knowledge editing. The temporal matching and fallback mechanism is summarized in Algorithm 1.

Specifically, given a query q, SPIKE first attempts to exactly match the subjects, then the start and end times at their original granularity. If no exact temporal match is found, a fallback mechanism progressively relaxes the time granularity (e.g., from 2025-01-20 to 2025-01, then 2025). At each fallback step, the relation similarity threshold increases by δ (set as $\delta = 0.05$) to penalize looser matches. This process continues until a match is found or the coarsest granularity is reached. The final match is then determined by relation similarity. This mechanism ensures reliable and flexible retrieval of knowledge across different temporal granularities.

Algorithm 1: Matching with Fallback Mechanism

Input: query text q, key table K, initial threshold θ
Output: the matching result (Boolean)
current_subject s = extract_subject(q);
if s in K.subjects **then**
  current_time t = extract_time(q);
  if t in K.times **then**
    rel_sim = compute_sim(K.relations, q);
    if rel_sim $\geq \theta$ **then**
      return True
  foreach t_f in granularity_fallback(t) **do**
    $\theta = \theta + \delta$;
    if t_f in K.times **then**
      rel_sim = compute_sim(K.relations, q);
      if rel_sim $\geq \theta$ **then**
        return True
return False

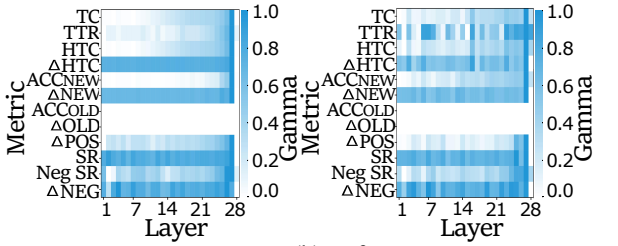
Experiments

Experimental settings. To evaluate the effectiveness of our approach in TKE, we conduct experiments using GPT-J (6B) as the base editing model. We compare our method against three representative METO-based baselines, including MEND†, ROME†, and MEMIT†, all of which have demonstrated strong performance on TKE (Yin et al. 2024).

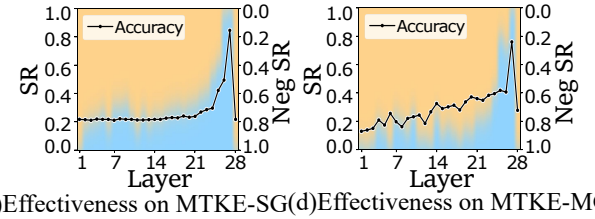
Injection sensitivity across layers. By analyzing the injection sensitivity, we identify the most effective editing layer (Figure 3). As shown in Figures 3a and 3b, shifting

Method	MTKE-SG					MTKE-MG				
	$ACC_{current}$	ACC_{hitory}	SR	Neg SR	HTC	$ACC_{current}$	ACC_{hitory}	SR	Neg SR	HTC
MEND†	81.80	21.84	0.3408	0.0219	0.5362	79.56	26.52	0.3217	0.0298	0.5392
ROME†	**91.22**	19.06	**0.4820**	**0.0144**	0.5734	**92.27**	19.89	**0.4368**	**0.0174**	0.5762
MEMIT†	85.43	26.55	0.3974	0.0196	0.5673	86.19	29.83	0.4023	0.0232	0.5876
SPIKE$_{\text{ours}}$	$84.37_{\downarrow 6.85}$	$\mathbf{85.01}_{\uparrow 58.46}$	$0.3552_{\downarrow 0.1268}$	$0.0208_{\uparrow 0.0064}$	$\mathbf{0.7926}_{\uparrow 0.2192}$	$75.69_{\downarrow 16.58}$	$\mathbf{80.66}_{\uparrow 50.83}$	$0.3406_{\downarrow 0.0962}$	$0.0308_{\uparrow 0.0134}$	$\mathbf{0.6904}_{\uparrow 0.1028}$
SPIKE$^{\ominus}$	50.11	82.66	0.2714	0.0254	0.6884	42.82	80.39	0.1976	0.0219	0.6640

Table 4: Comparison of knowledge editing methods on the MTKE-SG and MTKE-MG benchmarks. MEND†, ROME†, and MEMIT† are METO-based baselines. *SPIKE*$_{\text{ours}}$ is our proposed model, and *SPIKE*$^{\ominus}$ is an ablation variant without the time-anchored mechanism. Improvements and degradations over the best METO-based baseline are denoted with ↑ and ↓.



(a)Performance on MTKE-SG (b)Performance on MTKE-MG



(c)Effectiveness on MTKE-SG(d)Effectiveness on MTKE-MG

Figure 3: Layer-wise analysis of GPT-J (6B). Subfigures (a) and (b) show how SPIKE's performance varies across layers on MTKE-SG and MTKE-MG. Subfigures (c) and (d) show the gain in target knowledge (blue, measured by SR), loss in historical knowledge (orange, measured by Neg SR), and layer-wise prediction accuracy.

the injection layer toward higher transformer blocks consistently improves temporal consistency, prediction accuracy, and target knowledge confidence, while reducing interference with historical knowledge. Although metrics fluctuate slightly across layers, their values stabilize and peak between layers 25 and 27 (Figures 3a and 3b). On MTKE-MG (Figure 3b), lower layers (6–8) exhibit higher TTR but lower TC and accuracy, indicating severe misalignment of temporal knowledge before and after editing. This dual impact is further illustrated in Figures 3c and 3d. Overall, editing in higher layers increasingly reinforces target knowledge and reduces disruption to preserved information, with performance on both MTKE-SG and MTKE-MG peaking at layers 25–27. This finding confirms layer 27 (the penultimate layer) as the optimal injection point.

Comparison with SOTA methods. As shown in Table 4, SPIKE consistently achieves higher historical accuracy and HTC than METO-based baselines on both MTKE-SG and MTKE-MG. At the same time, its $ACC_{current}$ is comparable to the best-performing baselines. These results indicate that SPIKE can edit temporal knowledge with high precision while minimizing disruption to prior information. While the METO framework adds temporal annotations to traditional editing models, its approach remains limited to static tuple-based prompts and one-shot overwriting (Figure 1), thus lacking both parameter-level changes and dedicated editing mechanisms for temporal knowledge. In contrast, SPIKE overcomes this limitation by enabling anchor-aware, fine-grained updates that preserve and integrate temporal context.

Specifically, SPIKE substantially outperforms all baselines in $ACC_{history}$ and harmonic temporal consistency (HTC). On MTKE-SG, historical accuracy surpasses the best baseline by 220.19% and HTC by 38.23%; on MTKE-MG, by 170.40% and 17.49%, respectively. These improvements stem from explicit temporal and entity anchoring, enhancing temporal sensitivity and reducing editing interference. Although SPIKE's $ACC_{current}$ is slightly below the top baseline, this trade-off reflects its selective injection strategy and the complexity of multi-granularity reasoning. While granularity fallback may cause some timestamp ambiguity, this approach enables more robust knowledge coexistence and avoids destructive overwriting.

We further validate this design with an ablation study. Removing the time-anchored mechanism in *SPIKE*$^{\ominus}$ leads to a substantial drop in $ACC_{current}$ and moderate declines in other metrics, confirming the effectiveness of SPIKE's multi-anchor strategy for temporally structured knowledge. Taken together, SPIKE demonstrates more balanced and temporally aware editing than existing approaches. It is the first model designed to address the challenges of temporal knowledge, enabling fine-grained updates without compromising historical fidelity.

Conclusion and Limitation

In this paper, we introduce the concept of multi-granularity TKE and present the benchmark dataset MTKE, which more accurately reflects real-world complexity and addresses the limited coverage of historical knowledge in existing approaches. We also mitigate hallucination issues through rigorous data filtering. Furthermore, our SPIKE model demonstrates strong performance for multi-granularity TKE. As this area is still in its early stages, many challenges remain and further exploration is needed.

Acknowledgments

We thank all the anonymous reviewers and meta reviewers for their valuable comments, as well as all of our team members for their support and assistance. This work was partially supported by National Natural Science Foundation of China (62272469, 72371245, 72471237, 72501299), and partially supported by the Postgraduate Scientific Research Innovation Project of Hunan Province (XJQY2024044).

References

Achiam, J.; Adler, S.; Agarwal, S.; Ahmad, L.; Akkaya, I.; Aleman, F. L.; Almeida, D.; Altenschmidt, J.; Altman, S.; Anadkat, S.; et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Brown, T.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J. D.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A.; et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33: 1877–1901.

Chen, Z.; Liao, J.; and Zhao, X. 2023. Multi-granularity temporal question answering over knowledge graphs. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 11378–11392.

Cohen, R.; Biran, E.; Yoran, O.; Globerson, A.; and Geva, M. 2024. Evaluating the ripple effects of knowledge editing in language models. *Transactions of the Association for Computational Linguistics*, 12: 283–298.

García-Durán, A.; Dumančić, S.; and Niepert, M. 2018. Learning sequence encoders for temporal knowledge graph completion. *arXiv preprint arXiv:1809.03202*.

Gottschalk, S.; and Demidova, E. 2018. Eventkg: A multilingual event-centric temporal knowledge graph. In *European semantic web conference*, 272–287. Springer.

Hartvigsen, T.; Sankaranarayanan, S.; Palangi, H.; Kim, Y.; and Ghassemi, M. 2023. Aging with grace: Lifelong model editing with discrete key-value adaptors. *Advances in Neural Information Processing Systems*, 36: 47934–47959.

Hoffart, J.; Suchanek, F. M.; Berberich, K.; Lewis-Kelham, E.; De Melo, G.; and Weikum, G. 2011. YAGO2: exploring and querying world knowledge in time, space, context, and many languages. In *Proceedings of the 20th international conference companion on World wide web*, 229–232.

Huang, B.; Chen, C.; Xu, X.; Payani, A.; and Shu, K. 2024. Can Knowledge Editing Really Correct Hallucinations? *arXiv preprint arXiv:2410.16251*.

Huang, Z.; Shen, Y.; Zhang, X.; Zhou, J.; Rong, W.; and Xiong, Z. 2023. Transformer-patcher: One mistake worth one neuron. *arXiv preprint arXiv:2301.09785*.

Jiang, Y.; Wang, Y.; Wu, C.; Zhong, W.; Zeng, X.; Gao, J.; Li, L.; Jiang, X.; Shang, L.; Tang, R.; et al. 2024. Learning to edit: Aligning llms with knowledge editing. *arXiv preprint arXiv:2402.11905*.

Liu, Y.; Li, H.; Garcia-Duran, A.; Niepert, M.; Onoro-Rubio, D.; and Rosenblum, D. S. 2019. MMKG: multi-modal knowledge graphs. In *The semantic web: 16th international conference, ESWC 2019, portorož, Slovenia, June 2–6, 2019, proceedings 16*, 459–474. Springer.

Meng, K.; Bau, D.; Andonian, A.; and Belinkov, Y. 2022a. Locating and editing factual associations in gpt. *Advances in neural information processing systems*, 35: 17359–17372.

Meng, K.; Sharma, A. S.; Andonian, A.; Belinkov, Y.; and Bau, D. 2022b. Mass-editing memory in a transformer. *arXiv preprint arXiv:2210.07229*.

Mitchell, E.; Lin, C.; Bosselut, A.; Finn, C.; and Manning, C. D. 2021. Fast model editing at scale. *arXiv preprint arXiv:2110.11309*.

Mitchell, E.; Lin, C.; Bosselut, A.; Manning, C. D.; and Finn, C. 2022. Memory-based model editing at scale. In *International Conference on Machine Learning*, 15817–15831. PMLR.

Nonaka, I.; Toyama, R.; and Konno, N. 2000. SECI, Ba and leadership: a unified model of dynamic knowledge creation. *Long range planning*, 33(1): 5–34.

Ouyang, L.; Wu, J.; Jiang, X.; Almeida, D.; Wainwright, C.; Mishkin, P.; Zhang, C.; Agarwal, S.; Slama, K.; Ray, A.; et al. 2022. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35: 27730–27744.

Pan, K.; Fan, Z.; Li, J.; Yu, Q.; Fei, H.; Tang, S.; Hong, R.; Zhang, H.; and Sun, Q. 2024. Towards unified multimodal editing with enhanced knowledge collaboration. *Advances in Neural Information Processing Systems*, 37: 110290–110314.

Roddick, J. F.; and Spiliopoulou, M. 2002. A survey of temporal knowledge discovery paradigms and methods. *IEEE Transactions on Knowledge and data engineering*, 14(4): 750–767.

Schrodt, P. A. 2010. Automated production of high-volume, real-time political event data. In *Apsa 2010 annual meeting paper*.

Touvron, H.; Lavril, T.; Izacard, G.; Martinet, X.; Lachaux, M.-A.; Lacroix, T.; Rozière, B.; Goyal, N.; Hambro, E.; Azhar, F.; et al. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Wang, P.; Tang, Z.; Zhou, K.; Li, J.; Zhu, Q.; and Zhang, M. 2025. Revealing and mitigating over-attention in knowledge editing. *arXiv preprint arXiv:2502.14838*.

Yang, J.; Jin, H.; Tang, R.; Han, X.; Feng, Q.; Jiang, H.; Zhong, S.; Yin, B.; and Hu, X. 2024. Harnessing the power of llms in practice: A survey on chatgpt and beyond. *ACM Transactions on Knowledge Discovery from Data*, 18(6): 1–32.

Yao, Y.; Wang, P.; Tian, B.; Cheng, S.; Li, Z.; Deng, S.; Chen, H.; and Zhang, N. 2023. Editing large language models: Problems, methods, and opportunities. *arXiv preprint arXiv:2305.13172*.

Yin, X.; Jiang, J.; Yang, L.; and Wan, X. 2024. History matters: Temporal knowledge editing in large language model.

In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, 19413–19421.

Zeng, W.; Zhou, J.; and Zhao, X. 2024. Benchmarking Challenges for Temporal Knowledge Graph Alignment. In Serra, E.; and Spezzano, F., eds., *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management, CIKM 2024, Boise, ID, USA, October 21-25, 2024*, 3103–3112. ACM.

Zhao, R.; Zeng, W.; Zhang, W.; Zhao, X.; Tang, J.; and Chen, L. 2025. Towards Temporal Knowledge Graph Alignment in the Wild. *arXiv preprint arXiv:2507.14475*.

Zheng, C.; Li, L.; Dong, Q.; Fan, Y.; Wu, Z.; Xu, J.; and Chang, B. 2023. Can we edit factual knowledge by in-context learning? *arXiv preprint arXiv:2305.12740*.