

TRACE: A Fine-Grained Benchmark for Temporal Reasoning and Chronological Knowledge Editing over Large Language Models

Xiliang Liu¹ Zhiwen Xie^{1,2,†} Derek F. Wong^{1,†}

¹NLP²CT Lab, Faculty of Information Science and Computing, University of Macau

²School of Computer Science, Central China Normal University

nlp2ct.xiliang@gmail.com, zwxie@ccnu.edu.cn, derekfw@um.edu.mo

Abstract

Knowledge editing algorithms have emerged as an efficient paradigm for updating parametric facts in Large Language Models (LLMs) without full retraining. However, existing benchmarks primarily focus on isolated, static facts. While a few temporal datasets exist, they are frequently bottlenecked by coarse temporal granularity, severe relation imbalance, and a critical lack of *a priori* knowledge validation, which can conflate inherent model deficits with editing failures. To address these structural limitations, we introduce **TRACE**, a fine-grained *Temporal Reasoning And Chronological Editing* benchmark. Using exact-match *a priori* probing, TRACE dynamically constructs model-specific, non-mutually exclusive chronological event chains. Featuring over 16,000 customized test cases across two distinct operational setups (*Modify* and *Insert*), TRACE provides a comprehensive 7-dimensional metric matrix. It extends simple fact recall with relation-constrained counterfactual injection and sequential temporal logic probes that test timeline integrity. Our evaluation of 6 representative baselines (FT, IKE, ROME, GRACE, LoRA, WISE) on **Llama-3.2-3B** and **GPT-J-6B** identifies widespread *temporal interference* in the studied settings: editing a single timestamp often sharply degrades recall of adjacent historical and future facts. The consistently weak bidirectional inference after editing further suggests that the evaluated methods tend to introduce shallow, syntax-sensitive patches rather than integrate the edit into a chronological network. Our code, the EasyEdit snapshot used for the experiments, and the benchmark are publicly available.¹

1 Introduction

Large Language Models (LLMs) inevitably internalize outdated or incorrect factual knowledge during pre-training (Zhao et al., 2026b). To rectify

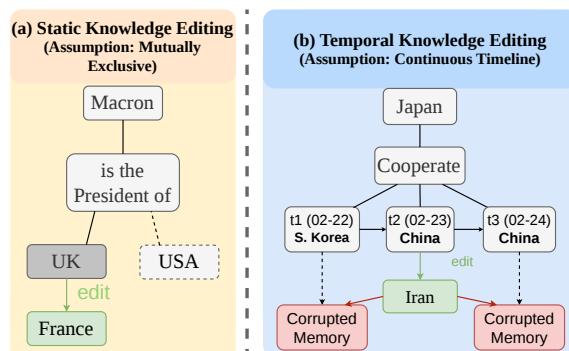


Figure 1: Static vs. Temporal Knowledge Editing.

these parametric flaws without the prohibitive overhead of global retraining, knowledge editing has emerged as a rapidly evolving alternative (Sinitsin et al., 2020; Meng et al., 2022; Park et al., 2025). Advanced paradigms, ranging from locate-and-edit layer modifications (Meng et al., 2023) to external memory mapping (Hartvigsen et al., 2023), have recently been extended to tackle massive-scale concurrent edits (Wan et al., 2026) and logical ripple-effect propagation (Dong et al., 2025). Despite these advancements, they primarily demonstrate precision in rewriting isolated, static (*Subject, Relation, Object*) triplets, implicitly assuming facts exist in a vacuum.

However, real-world knowledge is seldom static; it is intrinsically dynamic and structured along a continuous timeline (Jin et al., 2019). Consequently, historical facts exist as temporal quadruples: (*Subject, Relation, Object, Timestamp*). While existing methods claim high precision on static benchmarks like CounterFact (Meng et al., 2022), their architectural assumptions heavily rely on the mutual exclusivity of counterfactual flips (Figure 1). When confronted with chronological chains—where an entity frequently interacts with varying objects across different dates—the temporal alignment and historical locality of these algorithms remain largely uncharted.

[†]Corresponding author.

¹<https://github.com/NLP2CT/TRACE>

While a few pioneering benchmarks (Yin et al., 2024; Lee and Park, 2024; Zhao et al., 2026a) attempt to evaluate temporal editing, they exhibit critical methodological limitations. Existing datasets are frequently bottlenecked by coarse year-level granularity, severe relation imbalance, and an exclusive focus on present-day updates. Crucially, they lack rigorous *a priori* probing, conflating the model’s pre-existing knowledge deficits with editing failures.

In this work, we bridge this gap by introducing the **TRACE** (Temporal Reasoning And Chronological Editing) benchmark, a multi-dimensional evaluation framework explicitly designed to stress-test knowledge editing under complex, fine-grained chronological chains. Sourced from the ICEWS database (2005-2015), TRACE compiles sequential event histories. We implement a model-specific *a priori* probing pipeline that retains facts recalled by an exact probe, thereby reducing failures caused by missing pre-training knowledge. This filter establishes reliability for the tested prompt but does not by itself guarantee paraphrase generality or whole-chain reasoning. Furthermore, we formulate a 7-dimensional metric system across *Modify* and *Insert* operational setups. Beyond standard success metrics, our framework introduces sequential logic probes to deduce neighboring historical or future entities, utilizing a *Relation-Constrained Pooling* mechanism to reduce semantic leakage.

We execute an evaluation of 6 prominent baselines (FT, IKE (Zheng et al., 2023), ROME (Meng et al., 2022), GRACE (Hartvigsen et al., 2023), LoRA (Hu et al., 2022), and WISE (Wang et al., 2024)) on **Llama-3.2-3B** (Grattafiori et al., 2024) and **GPT-J-6B** (Wang and Komatsuzaki, 2021). Results reveal three recurring patterns in these models: (1) **Reliability–Generality Misalignment**: near-perfect reliability can mask substantial sensitivity to descriptive paraphrasing. (2) **Severe Temporal Interference**: altering a single temporal anchor often drives adjacent historical and future memory retention close to 0% for weight-updating methods. (3) **Chronological Disconnection**: low sequential inference scores indicate that the tested methods often fail to integrate counterfactual injections into the surrounding timeline.

Our core contributions are three-fold:

1. We present **TRACE**, the first fine-grained temporal knowledge editing benchmark equipped

with a strict *a priori* probing filter and relation-constrained counterfactual synthesis.

2. We propose a robust 7-dimensional metric matrix that pioneers sequential logic chains to explicitly diagnose temporal alignment, syntax invariance, and memory continuity.
3. We conduct a cross-model evaluation, providing evidence of widespread limitations among the tested editing techniques in managing temporal reasoning for 3B–6B models.

2 Related Work

2.1 Knowledge Editing in LLMs

Knowledge editing seeks to modify specific factual representations within an LLM’s parameters efficiently without global retraining (Sinitisin et al., 2020; Yao et al., 2023). Existing approaches can be broadly bifurcated into two distinct paradigms. The first incorporates *parameter-locating methods* that directly pinpoint and modify specific weight matrices, notably the causal layers within Feed-Forward Networks (FFNs). ROME (Meng et al., 2022) utilizes a causal trace analysis to treat downstream MLP layers as key-value storages, while MEMIT (Meng et al., 2023) extends this fundamental concept to multi-layer, multi-fact batched scaling. LoRA-based editing approaches (Hu et al., 2022) optimize localized low-rank adapter matrices to alter weights under localized constraints. Pushing the boundaries of scalability and logic, recent advancements in 2025 and 2026 have expanded these weight-modifying mechanisms to support massive-scale concurrent updates via dynamic weight generation (Wan et al., 2026) and to propagate logical ripple effects across sequential entity dependencies (Dong et al., 2025).

The second paradigm focuses on *meta-learning and external memory expansion*, which sidesteps direct weight contamination. IKE (Zheng et al., 2023) leverages in-context learning by prefixing structurally relevant factual demonstrations during the inference phase. GRACE (Hartvigsen et al., 2023) caches counterfactual updates in an external codebook that dynamically intercepts and substitutes token representations when activation vectors fall within trained codebook radii. WISE (Wang et al., 2024) actively separates editing and core parameters to minimize routing conflicts. This non-parametric paradigm has been further augmented

by recent frameworks designed to ensure context-robust representations against semantically diverse prompts (Park et al., 2025). While these methods perform strongly on static datasets like CounterFact (Meng et al., 2022) or MEND (Mitchell et al., 2022), their resistance to chronological cross-contamination across continuous historical timelines remains underexplored.

2.2 Temporal Knowledge and Chronological Reasoning

Modeling time-evolving realities has long been investigated within the Temporal Knowledge Graph (TKG) community (Jin et al., 2019; Goel et al., 2020). Benchmarks like ICEWS (Boschee et al., 2015) and GDELT (Leetaru and Schrodt, 2013) have been extensively used to evaluate graph networks on predicting future link states or tracking continuous entity transitions. With the ascension of generative AI, researchers began evaluating the zero-shot temporal awareness of pretrained LLMs (Ge et al., 2024; Dhingra et al., 2022), observing that these foundational models inherently suffer from massive temporal confusion and chronological decay when retrieving dynamically shifting facts.

To evaluate editing efficacy in dynamic contexts, several pioneering benchmarks have been recently introduced, though they retain methodological limitations. The ATOKE benchmark (Yin et al., 2024) pioneered temporal evaluation but suffers from severe relation imbalance, as a single relation (*playsFor*) accounts for over 83% of the data, and uses only coarse, year-level granularity. Furthermore, it exclusively targets present-knowledge updates, neglecting the modification of historical timelines. Subsequent works like TTKEB (Lee and Park, 2024) focused on temporal reasoning but have limited sample sizes and no prior knowledge validation. Other datasets (Ge et al., 2024; Cheng et al., 2024) have explored side-effects on non-temporal data, yet they largely bypass the insertion of new counterfactuals into past timelines. Even the recent MTKE dataset (Zhao et al., 2026a), which introduces finer granularity, does not directly address both error correction and the injection of new knowledge into established historical sequences.

TRACE differs from prior work in three respects. First, it offers broad relation diversity and **fine-grained temporal resolution** (exact dates) for high-frequency real-world events. Second, TRACE extends present-day updates with explicit *Modify*

and *Insert* operations over historical sequences. Third, it introduces **sequential temporal probing**. By anchoring on the counterfactual injection, these metrics test whether an edit is connected to the surrounding chronological chain or remains an isolated patch. The automated *a priori* probing pipeline then constructs a separate exact-match-filtered benchmark for each target model, reducing evaluation noise from facts that the model cannot recall under the reliability prompt.

3 The TRACE Benchmark

To systematically evaluate the robustness of Large Language Models (LLMs) and knowledge editing algorithms in dynamic temporal contexts, we construct the **TRACE** (Temporal Reasoning And Chronological Editing) benchmark. Our pipeline features a rigorous end-to-end framework: it first probes the target model to retain only known temporal facts, organizes these facts into chronological event chains, and then samples counterfactual edits under the *Modify* and *Insert* settings. Finally, it converts each edit and its surrounding temporal context into prompts spanning seven evaluation dimensions, as illustrated in Figure 2.

3.1 A Priori Knowledge Probing

A prerequisite for evaluating knowledge editing is ensuring the target model has already internalized the pre-edit facts. Otherwise, editing failures cannot be distinguished from the model’s inherent knowledge deficits. We extract factual tuples, specifically (*Subject, Relation, Object, Time*), from the ICEWS database (2005-2015).²

To transform these structured quadruplets into fluent natural language queries, we leverage Gemini 3.1 Pro to generate linguistic mappings for all 251 distinct relations present in the database. Utilizing these generated relation mappings, probing prompts are constructed in batch via a standardized template format: "On {Time}, {Subject} {Verb}", where {Verb} denotes the relation-specific natural language description.

Human quality control. Two authors independently audited all 251 relation mappings across the verb, past_verb, noun, and ing_form slots. The audit found no temporally or chronologically ambiguous mappings; 4 of 251 mappings (1.6%) contained minor grammatical or stylistic stiffness and

²<https://github.com/andybega/icews>

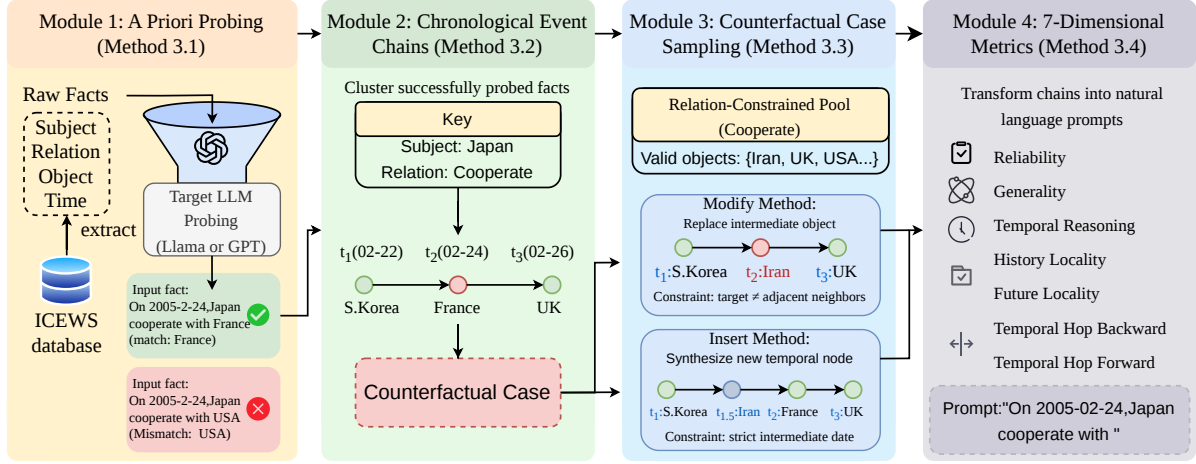


Figure 2: The end-to-end construction pipeline of the TRACE benchmark. (1) **A Priori Probing**: Filtering ICEWS facts to ensure target model comprehension. (2) **Chain Construction**: Linking facts into chronological sequences. (3) **Counterfactual Sampling**: Utilizing a relation-constrained pool to generate compatible counterfactual objects (e.g., Iran) for *Modify* and *Insert* strategies. (4) **Metrics Generation**: Creating prompts across seven exploration dimensions to evaluate the editor.

were corrected. We additionally sampled 100 complete prompts from the final datasets, all of which were judged fluent and natural by both annotators. Some prompts retain entity-level disambiguation tags inherited from ICEWS (e.g., *High Ranking Military Personnel (Japan)*), which aid grounding but can make phrasing less idiomatic. This spot check supports the quality of the generated prompts but does not rule out all template artifacts.

Following prompt synthesis, we sequentially query the target model by conducting batched probing with left-padding and greedy decoding (temperature = 0.0). Only facts successfully recalled by the model, for which the generated text exactly and unconditionally matches the ground-truth target object, are retained. This establishes an exact-prompt reliability baseline; it does not imply that the retained facts generalize to paraphrases or that the model can reason over the full chain.

3.2 Chronological Event Chains

Isolated facts are insufficient for rigorous temporal evaluation. We cluster the successfully probed facts using the tuple (*Subject, Relation*) as the primary key and sort them chronologically to construct continuous event chains. During this process, we deliberately retain redundant nodes with identical consecutive objects (defining them as *Soft Cases*), as they are crucial for evaluating the model’s and editor’s resistance to temporal bleeding across sustained state durations.

Dataset Statistics	Llama-3.2-3B	GPT-J-6B
Total Valid Chains	633	918
Max Chain Length	322	438
Chain Length Mode	3	2
Unique Relations	98	100
Unique Entities	290	489
- Subjects	268	467
- Objects	56	64

Table 1: Comparative statistics of the generated TRACE temporal chains across two target models. Differences in volume and length reflect the model-specific pre-training distributions captured by our *a priori* probing.

To accommodate our dual evaluation strategies, we implement method-specific length constraints. For the *Modify Method*, chains shorter than 3 nodes are strictly discarded to ensure sufficient bidirectional context (e_1, e_2, e_3) for temporal hop reasoning. Conversely, for the *Insert Method*, chains containing at least 2 adjacent nodes (e_1, e_2) are retained, providing both boundaries for inserting a novel intermediate event.

As detailed in Table 1, the scale and distribution of the constructed chronological chains vary significantly across different language models. For instance, while Llama-3.2-3B yields 633 valid temporal chains spanning 98 unique relations, GPT-J-6B produces 918 chains covering 100 relations. This variance illustrates the effect of model-specific *a priori* probing: TRACE shapes each evaluation dataset to facts recalled by that model, thereby reducing noise from missing pre-training knowledge

under the tested prompt.

3.3 Counterfactual Case Sampling

To systematically construct counterfactual nodes, we employ two distinct evaluation strategies—the *Insert Method* and the *Modify Method*. These strategies are specifically designed to simulate the injection of novel knowledge and the correction of erroneous parametric memories, respectively:

- **Modify Method:** Given a chain of consecutive nodes e_1 , e_2 , and e_3 , we replace the object of the intermediate node e_2 with a counterfactual entity. We enforce temporal gaps between nodes to avoid ambiguity.
- **Insert Method:** We synthesize an entirely new counterfactual event on an intermediate date between two existing adjacent nodes e_1 and e_2 , requiring the model to absorb the new knowledge without corrupting prior memory.

To ensure semantic and grammatical compatibility during this counterfactual synthesis, we introduce a *Relation-Constrained Pool*. For any specific relation, this pool archives all valid objects that have historically co-occurred with that relation across the entire database. When generating a counterfactual target, we randomly sample an object from this relation-specific pool, strictly ensuring that the sampled object is entirely distinct from the existing objects in the adjacent nodes (e.g., replacing a president with another politician, rather than a location).

Finally, we address the inherent length imbalance across chronological event chains, where some highly active entities may span hundreds of chronological nodes. To prevent these hyper-active chains from dominating the benchmark and skewing the data distribution, we implement a tunable threshold to restrict the maximum number of sampling operations permitted per chain. This length-capping mechanism effectively balances the overall dataset volume and guarantees a uniform, diverse distribution of evaluation samples.

3.4 7-Dimensional Evaluation Metrics

To transform the structured factual quadruplets into fluent natural language probes, we implement a systematic prompt generation pipeline. First, we preprocess all entities by stripping structural underscores to ensure natural phrasing (e.g., converting `South_Korea` to `South Korea`). For each specific relation, we meticulously design diverse linguistic mappings, encompassing verb phrases, noun

Metric	Input Prompt & Expected [Target]
<i>Context Anchor (Input Setup)</i>	
Chain	Japan $\rightarrow$ Cooperate $\rightarrow \dots \rightarrow e_1$: S.Korea (02-22) $\rightarrow e_2$: China (02-23) $\rightarrow e_3$: China (02-24)
Edit Target	Update e_2 object from <i>China</i> to <i>Iran</i> .
<i>TRACE 7-Dimensional Probing Metrics</i>	
(1) Rel.	On 2005-02-23, Japan engaged in diplomatic cooperation with [Iran]
(2) Gen.	The specific entity associated with the diplomatic cooperation of Japan on 2005-02-23 was [Iran]
(3) Reason	Following the activity involving South Korea on 2005-02-22, Japan next engaged in diplomatic cooperation with [Iran]
(4) Hist.	On 2005-02-22, Japan engaged in diplomatic cooperation with [South Korea]
(5) Fut.	On 2005-02-24, Japan engaged in diplomatic cooperation with [China]
(6) Hop B.	Before engaging in diplomatic cooperation with Iran on 2005-02-23, Japan previously engaged in diplomatic cooperation with [South Korea]
(7) Hop F.	After engaging in diplomatic cooperation with Iran on 2005-02-23, Japan next engaged in diplomatic cooperation with [China]

Table 2: Illustrative prompt inputs and expected target answers across the 7 evaluation metrics in our TRACE benchmark, anchoring on the counterfactual node e_2 .

phrases, and past-tense variations. Based on these linguistic anchors, we leverage Gemini 3.1 Pro to systematically formulate specific prompt templates for our evaluation metrics. Notably, to rigorously test syntax invariance, we construct three distinct paraphrase templates for the Generality metric, dynamically sampling one at random during the generation process. For a complete list of the formalized prompt templates utilized across all metrics, please refer to Appendix C.

As illustrated with a running example in Table 2, we provide a comprehensive assessment across 7 probing dimensions for each instance:

- **Reliability (Rel.):** Evaluates whether the target model can successfully recall the newly injected counterfactual fact when queried with the exact target timestamp (t_2). This serves as the fundamental verification that the parameter modification was successful.
- **Generality (Gen.):** Assesses the robustness of the edited knowledge against syntactic variations. We query the model using semantically equivalent but structurally diverse paraphrased templates to ensure the fact is genuinely internalized into the parametric memory, rather than merely overfitting to a specific

Metric	Llama-3.2-3B		GPT-J-6B	
	Modify	Insert	Modify	Insert
Total Samples	5,025	4,862	2,832	3,316
Unique Relations	47	47	45	46
Total Unique Entities	286	285	338	450
- Unique Subjects	264	263	309	424
- Unique Objects	54	55	60	60
Max Samples / Chain (κ)	32	32	10	10
Dominant Relation Ratio	20.68%	20.54%	38.94%	37.69%

Table 3: Comprehensive dataset statistics of the final TRACE benchmark across different target architectures and operational evaluation setups. κ denotes the maximum sampling threshold per chronological chain.

prompt string.

- **Temporal Reasoning (Temp. Reasoning):** Measures the model’s capacity for forward chronological deduction. By providing the historical context of the prior node (e_1 at t_1), we evaluate whether the model can smoothly transition along the timeline to predict the newly edited counterfactual object at t_2 .
- **History Locality (Hist.) & Future Locality (Fut.):** These metrics quantify temporal interference by testing whether the factual integrity of the adjacent historical node (e_1 at t_1) and future node (e_3 at t_3) is preserved after modifying the intermediate temporal anchor (t_2).
- **Temporal Hop Backward (Hop B.) & Temporal Hop Forward (Hop F.):** These metrics stress-test the structural integrity of the surrounding chronological chain. Using the newly injected counterfactual entity and its explicit timestamp as the reasoning anchor (e.g., “Before/After interacting with [Counterfactual Object] on t_2 ”), we force the model to deduce the adjacent historical or future entities. This exposes whether the editing algorithm weaves the new fact into the chronology or merely introduces an isolated patch that disrupts it.

3.5 Dataset Statistics

Following the end-to-end prompt generation and filtration pipeline, we compile the final evaluation instances for TRACE. To demonstrate that the construction pipeline can be applied to different target models, we generate model-specific benchmark variants for **Llama-3.2-3B** and **GPT-J-6B**. Each variant comprises two distinct operational splits: the *Modify Method* and the *Insert Method*.

As summarized in Table 3, the resulting TRACE benchmark provides a substantial evaluation scale,

yielding a total of 9,887 instances for Llama-3.2-3B and 6,148 instances for GPT-J-6B. A prominent architectural highlight of our sampling pipeline is the inclusion of a tunable hyperparameter, denoted as the maximum sampling constraint per chronological chain (κ). To evaluate the benchmark’s behavior under varying structural densities, we configure $\kappa = 32$ for Llama-3.2-3B and enforce a stricter cap of $\kappa = 10$ for GPT-J-6B.

Restricting κ in the GPT-J setup promotes uniform entity distribution at the cost of total sample scale. Crucially, this sampling strategy guarantees a significantly more balanced relation distribution compared to prior benchmarks. For instance, while the ATOKE dataset is severely bottlenecked by a single relation accounting for over 83% of its instances, the dominant relation in TRACE constitutes at most 38.94% across all experimental configurations. Coupled with our *Relation-Constrained Pooling* mechanism, this structural diversity ensures that semantic integrity and syntactic variations are rigorously maintained. A complete, verbatim evaluation instance sampled from the final TRACE benchmark is illustrated in Appendix E.

4 Results and Discussion

4.1 Experimental Setup

We implement our experiments with the open-source EasyEdit framework.³ The exact EasyEdit snapshot and configuration files used in our experiments are packaged with the TRACE code linked in the abstract. All evaluations are conducted on a server equipped with a single NVIDIA RTX 5090 GPU (32GB VRAM), an Intel Xeon Platinum 8470Q CPU, and 120GB of system RAM, operating on Ubuntu 22.04 with CUDA 12.8. We evaluate two target models in the 3B–6B range: **Llama-3.2-3B** (Grattafiori et al., 2024) and **GPT-J-6B** (Wang and Komatsuzaki, 2021).

We evaluate 6 representative baselines: FT, IKE (Zheng et al., 2023), ROME (Meng et al., 2022), GRACE (Hartvigsen et al., 2023), LoRA (Hu et al., 2022), and WISE (Wang et al., 2024). We use the official EasyEdit default configurations for these methods; only the number of optimization iterations is adjusted to manage computational cost and avoid loss stagnation, without changing the core update rules or search radii. We use greedy decoding (temperature = 0.0) for all probes. The exact

³<https://github.com/zjunlp/EasyEdit>

Metric	Reliability Acc $\uparrow$	Generality Acc $\uparrow$	Temp. Reasoning $\uparrow$	History Score $\uparrow$	Future Score $\uparrow$	Hop Backward $\uparrow$	Hop Forward $\uparrow$
Pre-Edit	0.9747	0.1220	0.4354	0.9767	0.9767	0.1013	0.1391
FT	0.6951	0.6424	0.7928	0.0107	0.0115	0.0107	0.0123
IKE	0.9940	0.9918	0.8100	0.0105	0.0109	0.0591	0.0374
ROME	0.9936	0.6595	0.9510	0.0100	0.0096	0.0219	0.0362
GRACE	0.8392	0.0109	0.0024	0.6191	0.6207	0.0903	0.1763
LoRA	0.9893	0.9813	0.9853	0.0048	0.0048	0.0046	0.0048
WISE	0.3075	0.6878	0.4677	0.3502	0.3343	0.0472	0.0547

Table 4: Macro-evaluation of 6 representative baselines under **Modify** on **Llama-3.2-3B** ($n = 5, 025$). Pre-Edit uses the unedited model and original factual chains.

Metric	Reliability Acc $\uparrow$	Generality Acc $\uparrow$	Temp. Reasoning $\uparrow$	History Score $\uparrow$	Future Score $\uparrow$	Hop Backward $\uparrow$	Hop Forward $\uparrow$
Pre-Edit	0.0000	0.0000	0.0000	0.9897	0.9903	0.0000	0.0000
FT	0.8235	0.7515	0.8824	0.0113	0.0111	0.0362	0.0251
IKE	1.0000	0.9977	0.9755	0.0298	0.0181	0.2211	0.1989
ROME	0.9998	0.6900	0.9589	0.0265	0.0263	0.0847	0.0895
GRACE	0.8538	0.0162	0.0132	0.8544	0.8610	0.2018	0.3159
LoRA	0.9905	0.9817	0.9866	0.0051	0.0051	0.0047	0.0051
WISE	0.3784	0.7427	0.5705	0.4665	0.4490	0.0886	0.0866

Table 5: Macro-evaluation results of 6 representative model editing baselines under the **Insert Method** setup using **Llama-3.2-3B** ($n = 4, 862$). Pre-Edit Reliability, Generality, Temporal Reasoning, and Hop scores are zero because the newly inserted intermediate anchor is absent from the original timeline.

model-specific optimization and method parameters are reported in Appendix B; detailed results appear in Tables 4–7.

We additionally attempted to run AlphaEdit (Fang et al., 2025) through its official EasyEdit integration. In our setup, optimization collapsed: the negative log-likelihood stagnated and gradients flattened on the chronological edit prompts. Because we cannot rule out parameter mismatch, framework regression, or model/configuration incompatibility, we exclude these invalid runs from quantitative comparison and do not interpret them as evidence about AlphaEdit itself.

4.2 Intrinsic Temporal Weakness of Pre-Edit Models

The Modify Pre-Edit rows in Tables 4 and 6 evaluate the unedited models on their original factual chains. Llama-3.2-3B and GPT-J-6B exhibit high exact-prompt *Reliability* (97.47% and 100.00%), but their *Generality*, *Temporal Reasoning*, and bidirectional *Temporal Hop* scores are lower. For example, Llama-3.2-3B obtains 12.20% Generality and 10.13%/13.91% backward/forward Hop accuracy.

Within the two studied models, this misalignment suggests that successful exact recall need not imply robust paraphrase invariance or a well-connected chronological representation. It also motivates comparing post-edit Hop scores against both native and explicit-context controls rather than attributing every failure to editing alone.

For several parameter-updating methods, native Hop performance falls further after editing. FT,

ROME, and LoRA, for example, often approach zero on these metrics. This pattern indicates disruption of residual chronological associations.

4.3 Reliability–Generality Misalignment

Across both target models, several baselines exhibit a substantial gap between *Reliability* and *Generality*. As shown in Tables 4 and 6, ROME, IKE, and LoRA can exceed 99% Reliability after editing, yet paraphrases can make the same edit less stable.

On GPT-J-6B (Table 6), ROME achieves 99.96% Reliability but 68.54% Generality. GRACE obtains 100.00% Reliability, whereas its Generality and Temporal Reasoning scores are 0.42% and 0.49%. Across the evaluated models and prompts, these gaps indicate varying degrees of template sensitivity rather than uniform semantic integration.

4.4 Severe Temporal Interference

The Pre-Edit baselines retain most adjacent historical and future nodes (History/Future Locality near 100% for GPT-J-6B and above 97% for Llama-3.2-3B). After counterfactual editing, however, several methods exhibit severe *temporal interference*.

In the Modify setup (Tables 4 and 6), Locality for FT, ROME, and LoRA falls close to zero. On GPT-J-6B, for example, LoRA retains only 0.56% History and Future Locality, while ROME retains approximately 1%. Similar reductions occur in the Insert setup (Tables 5 and 7), suggesting that these updates often over-activate the new object beyond its target timestamp.

To quantify this error pattern, we define the *Edit-*

Metric	Reliability Acc $\uparrow$	Generality Acc $\uparrow$	Temp. Reasoning $\uparrow$	History Score $\uparrow$	Future Score $\uparrow$	Hop Backward $\uparrow$	Hop Forward $\uparrow$
Pre-Edit	1.0000	0.3157	0.5265	1.0000	1.0000	0.5173	0.4562
FT	0.8598	0.8510	0.8796	0.0286	0.0268	0.0141	0.0159
IKE	0.9982	0.9848	0.6691	0.0286	0.0244	0.3284	0.2800
ROME	0.9996	0.6854	0.9944	0.0117	0.0106	0.0240	0.0265
GRACE	1.0000	0.0042	0.0049	0.9040	0.9036	0.3983	0.4061
LoRA	0.8362	0.8358	0.8351	0.0056	0.0056	0.0053	0.0053
WISE	0.8690	0.9177	0.9149	0.1875	0.1462	0.0095	0.0088

Table 6: Macro-evaluation of 6 representative baselines under **Modify** on **GPT-J-6B** ($n = 2, 832$). Pre-Edit uses the unedited model and original factual chains.

Metric	Reliability Acc $\uparrow$	Generality Acc $\uparrow$	Temp. Reasoning $\uparrow$	History Score $\uparrow$	Future Score $\uparrow$	Hop Backward $\uparrow$	Hop Forward $\uparrow$
Pre-Edit [†]	0.0000	0.0000	0.0000	0.9967	0.9967	0.2379	0.2711
FT	0.8824	0.8673	0.8981	0.0139	0.0145	0.0081	0.0057
IKE	0.9991	0.9855	0.9077	0.0446	0.0332	0.3456	0.2820
ROME	0.9994	0.6888	0.9967	0.0087	0.0097	0.0280	0.0286
GRACE	1.0000	0.0027	0.0033	0.9331	0.9439	0.4086	0.4180
LoRA	0.8426	0.8411	0.8417	0.0048	0.0051	0.0048	0.0057
WISE	0.8776	0.9246	0.9343	0.1758	0.1451	0.0081	0.0075

Table 7: Macro-evaluation results under the **Insert Method** setup using **GPT-J-6B** ($n = 3, 316$). Reliability, Generality, and Temporal Reasoning are zero before editing because the inserted anchor is absent. For [†], the two Pre-Edit Hop scores are oracle-context controls obtained by giving the unedited model the ground-truth chain; the remaining Pre-Edit fields use the standard unedited prompts.

Object Dominance (EOD) Rate as the proportion of incorrect History, Future, or Hop predictions that output the edited object e_2 . Under FT on Llama-3.2-3B Insert, EOD ranges from 83.0% to 85.1% across these four metrics. Under ROME on GPT-J-6B Insert, backward and forward Hop EOD reaches 61.90% and 63.27%. Such target over-prediction resembles previously reported edit side effects (Hoelscher-Obermaier et al., 2023) and may help explain the observed interference. EOD is a diagnostic association, not a causal account of the underlying parameters; complete results appear in Appendix Table 8.

4.5 Failure in Sequential Temporal Inference

The *Hop Backward* and *Hop Forward* metrics anchor the query on the newly injected fact and ask for an adjacent entity. Their interpretation depends on the base model’s ability to answer the same temporal question. In the Modify setup, the ordinary Pre-Edit row provides a native control: GPT-J-6B obtains 51.73%/45.62% backward/forward accuracy before editing, but after ROME editing, it dropped to 2.40%/2.65%.

For Insert, the new anchor is absent from the original timeline, so the standard unedited Hop score would be zero. We therefore add an oracle-context control for GPT-J-6B by explicitly providing the ground-truth chain to the unedited model. It achieves 23.79%/27.11%, whereas ROME after editing obtains 2.80%/2.86%. The oracle result shows that the base model can solve a non-trivial

subset given the relevant facts, whereas the post-edit reduction indicates disruption but does not rule out prompt or reasoning effects.

Many Hop scores remain low for the tested parameter-updating methods, although IKE and GRACE sometimes perform better with different trade-offs. Alongside EOD and the example in Appendix D, this suggests that edits integrate poorly with the surrounding timeline.

5 Conclusion

We introduce **TRACE**, a fine-grained benchmark for temporal reasoning under chronological knowledge editing. Its exact-match *a priori* probing, non-mutually exclusive event chains, and seven metrics separate target recall from paraphrase robustness, adjacent-fact retention, and temporal hops. Across six successfully executed baselines on **Llama-3.2-3B** and **GPT-J-6B**, high reliability frequently coexists with severe temporal interference and weak bidirectional Hop performance. Oracle-context controls show that intrinsic reasoning limitations explain part, but not all, of the gap, while EOD analysis identifies edited-object over-prediction as a widespread error pattern. These findings are limited to the studied models, methods, and prompts, but motivate editing techniques that connect an update to its chronological context rather than treating it as an isolated patch.

Limitations

TRACE is a fine-grained temporal knowledge editing benchmark with several limitations.

First, our study covers two architectures and only the 3B–6B parameter range. The behavior of editing methods on larger models and closed-source systems remains to be tested; our conclusions should therefore not be generalized to all LLMs or editing algorithms.

Second, TRACE inherits the ontology of ICEWS and therefore concentrates on text-only political events and international relations. Corporate, scientific, conversational, and multimodal timelines are not represented. In addition, the rigid ISO date format (e.g., 2005-02-24) is less idiomatic than natural-language dates, and template generation may still introduce wording artifacts despite our manual validation. Both of these choices may affect the quality of the model’s responses.

Finally, the *Modify* split uses overlapping sliding windows (e.g., $e_1 \rightarrow e_2 \rightarrow e_3$ followed by $e_2 \rightarrow e_3 \rightarrow e_4$) to preserve sample volume. TRACE can evaluate a single isolated edit produced by methods such as ROME, but overlapping instances create incompatible constraints for coherent batch, sequential, or lifelong evaluation. Evaluating batched editing algorithms like MEMIT requires disjoint windows and a dedicated update protocol. WikiFactDiff (Ammar Khodja et al., 2024), which distinguishes new, obsolete, and static facts, and WikiBigEdit (Thede et al., 2025), which targets large-scale real-world lifelong edits, provide relevant foundations for further expansion.

Acknowledgments

This work was supported in part by the Science and Technology Development Fund of Macau SAR (Grant Nos. FDCT/0007/2024/AKP, EF2024-00185-FST), the UM and UMDF (Grant Nos. MYRG-GRG2024-00165-FST-UMDF, MYRG-GRG2025-00236-FST), the Tencent AI Lab Rhino-Bird Research Program (Grant No. EF2023-00151-FST), the Stanley Ho Medical Development Foundation (Grant No. SHMDF-AI/2026/001), the National Natural Science Foundation of China (Grant No. 62266013, 62402194), the Macao Young Scholars Program (Grant No. AM2024015), and the China Postdoctoral Science Foundation under Grant Number 2024M751062.

References

- Hichem Ammar Khodja, Frédéric Béchet, Quentin Brabant, Alexis Nasr, and Gwénolé Lecorvé. 2024. [WikiFactDiff: A large, realistic, and temporally adaptable dataset for atomic factual knowledge update in causal language models](#). In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 17614–17624, Torino, Italia. ELRA and ICCL.
- Elizabeth Boschee, Jennifer Lautenschlager, Sean O’Brien, Steve Shellman, James Starz, and Michael Ward. 2015. [ICEWS coded event data](#). *Harvard Dataverse*.
- Keyuan Cheng, Gang Lin, Haoyang Fei, Yuxuan zhai, Lu Yu, Muhammad Asif Ali, Lijie Hu, and Di Wang. 2024. [Multi-hop question answering under temporal knowledge editing](#). *Preprint*, arXiv:2404.00492.
- Bhuwan Dhingra, Jeremy R. Cole, Julian Martin Eisenschlos, Daniel Gillick, Jacob Eisenstein, and William W. Cohen. 2022. [Time-aware language models as temporal knowledge bases](#). *Transactions of the Association for Computational Linguistics*, 10:257–273.
- Zilu Dong, Xiangqing Shen, Zinong Yang, and Rui Xia. 2025. [ChainEdit: Propagating ripple effects in LLM knowledge editing through logical rule-guided chains](#). In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 13558–13571, Vienna, Austria. Association for Computational Linguistics.
- Junfeng Fang, Houcheng Jiang, Kun Wang, Yunshan Ma, Jie Shi, Xiang Wang, Xiangnan He, and Tat-Seng Chua. 2025. [AlphaEdit: Null-space constrained knowledge editing for language models](#). In *The Thirteenth International Conference on Learning Representations*.
- Xiou Ge, Ali Mousavi, Edouard Grave, Armand Joulin, Kun Qian, Benjamin Han, Mostafa Arefiyan, and Yunyao Li. 2024. [Time sensitive knowledge editing through efficient finetuning](#). In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 583–593, Bangkok, Thailand. Association for Computational Linguistics.
- Rishab Goel, Seyed Mehran Kazemi, Marcus Brubaker, and Pascal Poupart. 2020. [Diachronic embedding for temporal knowledge graph completion](#). In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 3988–3995.
- Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, and 542 others. 2024. [The llama 3 herd of models](#). *Preprint*, arXiv:2407.21783.

- Tom Hartvigsen, Swami Sankaranarayanan, Hamid Palangi, Yoon Kim, and Marzyeh Ghassemi. 2023. [Aging with grace: Lifelong model editing with discrete key-value adaptors](#). In *Advances in Neural Information Processing Systems*, volume 36, pages 47934–47959. Curran Associates, Inc.
- Jason Hoelscher-Obermaier, Julia Persson, Esben Kran, Ioannis Konstas, and Fazl Barez. 2023. [Detecting edit failures in large language models: An improved specificity benchmark](#). In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 11548–11559, Toronto, Canada. Association for Computational Linguistics.
- Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. [LoRA: Low-rank adaptation of large language models](#). In *International Conference on Learning Representations*.
- Woojeong Jin, Changlin Zhang, Pedro A. Szekely, and Xiang Ren. 2019. [Recurrent event network for reasoning over temporal knowledge graphs](#). *ArXiv*, abs/1904.05530.
- Jae-Wook Lee and Ki-Nam Park. 2024. [A time-sensitive temporal knowledge editing benchmark](#). In *Proceedings of the 8th International Conference on Interdisciplinary Research on Computer Science, Psychology, and Education*.
- Kalev Leetaru and Philip A. Schrodt. 2013. [Gdelt: Global data on events, location, and tone](#). *ISA Annual Convention*.
- Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. 2022. [Locating and editing factual associations in gpt](#). In *Advances in Neural Information Processing Systems*, volume 35, pages 17359–17372. Curran Associates, Inc.
- Kevin Meng, Arnab Sen Sharma, Alex Andonian, Yonatan Belinkov, and David Bau. 2023. [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. 2022. [Fast model editing at scale](#). In *International Conference on Learning Representations*.
- Haewon Park, Gyubin Choi, Minjun Kim, and Yohan Jo. 2025. [Context-robust knowledge editing for language models](#). In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 10360–10385, Vienna, Austria. Association for Computational Linguistics.
- Anton Sinitin, Vsevolod Plokhotnyuk, Dmitriy Pyrkin, Sergei Popov, and Artem Babenko. 2020. [Editable neural networks](#). *Preprint*, arXiv:2004.00345.
- Lukas Thede, Karsten Roth, Matthias Bethge, Zeynep Akata, and Thomas Hartvigsen. 2025. [WikiBigEdit: Understanding the limits of lifelong knowledge editing in LLMs](#). In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pages 59326–59354. PMLR.
- Wentao Wan, Qiqing Lao, Zhiwei Xie, Hefeng Wu, Runnan Lin, Liang Lin, and Keze Wang. 2026. [Massive editing for large language models based on dynamic weight generation](#). In *International Conference on Learning Representations*, volume 2026, pages 74467–74494.
- Ben Wang and Aran Komatsuzaki. 2021. GPT-J-6B: A 6 Billion Parameter Autoregressive Language Model. <https://github.com/kingoflolz/mesh-transformer-jax>.
- 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](#). In *Advances in Neural Information Processing Systems*, volume 37, pages 53764–53797. Curran Associates, Inc.
- Yunzhi Yao, Peng Wang, Bozhong Tian, Siyuan Cheng, Zhoubo Li, Shumin Deng, Hua-jun Chen, and Ningyu Zhang. 2023. [Editing large language models: Problems, methods, and opportunities](#). In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 10222–10240, Singapore. Association for Computational Linguistics.
- Xunjian Yin, Jin Jiang, Liming Yang, and Xiaojun Wan. 2024. [History matters: Temporal knowledge editing in large language model](#). In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 19413–19421.
- Simiao Zhao, Ning Pang, Zhen Tan, Yanli Hu, Weidong Xiao, and Xiang Zhao. 2026a. [Multi-granularity temporal knowledge editing over large language models](#). In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pages 28742–28750.
- Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Zican Dong, Yupeng Hou, Beichen Zhang, Yingqian Min, Junjie Zhang, Peiyu Liu, Xiaolei Wang, Yifan Du, Chen Yang, Yushuo Chen, Zhipeng Chen, Jinhao Jiang, Ruiyang Ren, Yifan Li, Xinyu Tang, and 4 others. 2026b. [A survey of large language models](#). *Frontiers of Computer Science*, 20(12):2012627.
- Ce Zheng, Lei Li, Qingxiu Dong, Yuxuan Fan, Zhiyong Wu, Jingjing Xu, and Baobao Chang. 2023. [Can we edit factual knowledge by in-context learning?](#) In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 4862–4876, Singapore. Association for Computational Linguistics.

A Complete Edit-Object Dominance Results

Table 8 reports the EOD Rate for the four dimensions whose incorrect outputs have a meaningful edited-object alternative. Reliability, Generality, and Temporal Reasoning are omitted because the edited object is their intended target rather than an error category.

B Editing Hyperparameters

Table 9 transcribes the algorithmic settings from the configuration files used in our experiments. “Steps” denotes num_steps for FT and LoRA, v_num_grad_steps for ROME, and n_iter for GRACE and WISE. IKE performs retrieval-based in-context editing and therefore has no gradient steps or edit learning rate. Device identifiers, local paths, cache directories, and architecture template strings are omitted because they do not alter the optimization setting. The same configuration is used for the Modify and Insert splits.

C Formal Prompt Templates of the TRACE Benchmark

To facilitate full reproducibility and provide transparent insights into our evaluation framework, we detail the exact prompt templates utilized across our 7-dimensional metrics in Table 10.

During the automated prompt synthesis, structural variables are dynamically instantiated based on the specific factual quadruplets. Notably, the *Generality* metric uniformly samples from a pre-defined pool of linguistically diverse templates (shown in Part I of Table 10), requiring the model to generalize across varying noun forms and syntactic structures.

D Case Study: Qualitative Analysis of Algorithmic Failures

To provide an intuitive, qualitative view of the patterns discussed in Section 4 (e.g., reliability–generality misalignment and sequential inference failure), we present a concrete case study drawn from our evaluations. Through this representative case study, we analyze the editing effect of the GRACE editor on prompts constructed based on the GPT-J-6B model and *Modify* setup.

As illustrated in Table 11, we isolate a chronological chain tracking the accusations leveled by the *UN Security Council*. The temporal anchor at

t_2 (2008-05-15) is modified from its original object (*Iran*) to a counterfactual target (*Colombia*).

Analysis of Reliability–Generality Misalignment: GRACE operates by caching activation vectors in an external codebook. Under the exact Reliability formulation, the cache returns the target (*Colombia*). Under the descriptive Generality paraphrase, however, the edited model produces the same incorrect continuation (“*the UN Security Council, not the Colombia*”) as the pre-edit model. For this example, the algorithm fails to generalize or integrate the new knowledge into the temporal timeline, merely overfitting to the exact prompt string of metric (1).

Analysis of Locality vs. Reasoning: Unlike the low Locality observed for several parameter-updating methods, GRACE preserves the *History Locality* at t_1 (*Iran*) in this case. Yet when asked to reason backward from the injected target (*Colombia*), the post-edit model returns the same incorrect sequence (“*The United States...*”) as the pre-edit model. Thus, locality preservation in this instance does not imply successful temporal integration.

This case study illustrates, but does not by itself establish, the broader pattern: the GRACE edit succeeds on the exact trigger while failing to generalize or support a temporal hop.

E Complete Evaluation Instance of the TRACE Benchmark

To provide a transparent overview of the dataset’s underlying structure, Table 12 presents a complete, concrete evaluation instance sampled from the TRACE benchmark under the *Modify* operational setup.

Llama-3.2-3B					
Split	Method	Hist.	Fut.	Hop B.	Hop F.
Modify	FT	0.6934	0.6924	0.7962	0.7449
Modify	IKE	0.6066	0.6058	0.3082	0.3180
Modify	ROME	0.9934	0.9936	0.4195	0.4249
Modify	GRACE	0.3819	0.3814	0.0118	0.0210
Modify	LoRA	0.9898	0.9900	0.9800	0.9794
Modify	WISE	0.2953	0.3148	0.7270	0.7032
Insert	FT	0.8348	0.8305	0.8515	0.8344
Insert	IKE	0.9137	0.9183	0.4032	0.3674
Insert	ROME	0.9996	1.0000	0.6479	0.6250
Insert	GRACE	0.8460	0.8269	0.2602	0.1158
Insert	LoRA	0.9907	0.9903	0.9804	0.9797
Insert	WISE	0.3797	0.3987	0.7495	0.7210

GPT-J-6B					
Split	Method	Hist.	Fut.	Hop B.	Hop F.
Modify	FT	0.8575	0.8567	0.9244	0.9171
Modify	IKE	0.5569	0.5841	0.6057	0.6596
Modify	ROME	0.9993	0.9989	0.6378	0.6627
Modify	GRACE	1.0000	1.0000	0.1338	0.0993
Modify	LoRA	0.8363	0.8374	0.8392	0.8378
Modify	WISE	0.9148	0.9305	0.9950	0.9932
Insert	FT	0.8807	0.8800	0.9362	0.9299
Insert	IKE	0.8545	0.8986	0.6138	0.6926
Insert	ROME	0.9994	0.9994	0.6190	0.6327
Insert	GRACE	1.0000	1.0000	0.1285	0.1093
Insert	LoRA	0.8409	0.8409	0.8421	0.8411
Insert	WISE	0.9283	0.9418	0.9957	0.9945

Table 8: Edit-Object Dominance (EOD) Rate among incorrect predictions. Hist., Fut., Hop B., and Hop F. denote History, Future, Hop Backward, and Hop Forward, respectively. Higher values indicate that a larger fraction of errors reproduce the edited object e_2 rather than the expected adjacent entity.

Method	Edited layer/module	Steps	LR	Other configured settings
Llama-3.2-3B				
FT	Layer 21, down_proj.weight	25	5×10^{-4}	Batch size 1; max length 100; weight decay 0; KL factor 0; prompt_last objective; no norm constraint.
IKE	In-context demonstrations	–	–	$k = 16$ retrieved examples; all-MiniLM-L6-v2 sentence encoder.
ROME	Layer 5, down_proj	25	0.5	subject_last fact token; loss layer 27; weight decay 10^{-3} ; clamp 4; KL factor 0.0625; context lengths [(5, 10), (10, 10)]; Wikipedia moment statistics (10^5 samples, float32); no moment adjustment; fp16.
GRACE	Layer 27, down_proj.weight	50	1.0	$\epsilon = 1$; Euclidean distance; cold value initialization; SGD value training; early stopping; replace_last; coverage expansion; 8 perturbations; dropout 0.
LoRA	layers=[]; q_proj, v_proj	30	2×10^{-3}	Batch size 1; max length 100; rank 8; $\alpha = 32$; dropout 0.1; weight decay 0; KL factor 0; no norm constraint.
WISE	Layer 27, down_proj.weight	70	1.0	Mask ratio 0.2; norm constraint 1; activation margin [5, 20, 10]; activation ratio 0.88; only_label objective; save/merge frequencies 500/1000; TIES merge (density 0.53, weight 1); retrieval on, replay off.
GPT-J-6B				
FT	Layer 21, fc_out	25	5×10^{-4}	Batch size 1; max length 50; weight decay 0; KL factor 0; prompt_last objective; no norm constraint.
IKE	In-context demonstrations	–	–	$k = 16$ retrieved examples; all-MiniLM-L6-v2 sentence encoder.
ROME	Layer 5, fc_out	15	0.5	subject_last fact token; loss layer 27; weight decay 0.5; clamp 4; KL factor 0.0625; context lengths [(5, 10), (10, 10)]; Wikipedia moment statistics (10^5 samples, float32); no moment adjustment; fp16.
GRACE	Layer 25, fc_out.weight	25	1.0	$\epsilon = 1$; Euclidean distance; cold value initialization; SGD value training; early stopping; replace_last; coverage expansion; 8 perturbations; dropout 0.
LoRA	layers=[]; fc_in, fc_out	20	5×10^{-3}	Batch size 1; max length 50; rank 8; $\alpha = 32$; dropout 0.1; weight decay 0; KL factor 0; no norm constraint.
WISE	Layer 21, fc_out.weight	30	1.0	Mask ratio 0.2; norm constraint 1; activation margin [15, 40, 20]; activation ratio 0.88; only_label objective; save/merge frequencies 500/1000; TIES merge (density 0.53, weight 1); retrieval on, replay off.

Table 9: Model-specific hyperparameters used for the six successfully executed editing baselines. Values are transcribed from the archived EasyEdit configuration files; non-algorithmic runtime fields are omitted.

Part I: Dynamic Template Pool for the <i>Generality</i> Metric		
Template 1: Regarding the [Relation-noun] involving [Subject] on [Edit-Date], the target was		
Template 2: The specific entity associated with the [Relation-noun] of [Subject] on [Edit-Date] was		
Template 3: On [Edit-Date], the [Relation-noun] concerning [Subject] was directed at		
Part II: Templates for the 7-Dimensional Metrics		
Metric Dimension	Formal Prompt Template	Target Entity
(1) Reliability	On [Edit-Date], [Subject] [Relation-past]	[CF-Object]
(2) Generality	$\langle$ <i>Uniformly sampled template from Part I</i> $\rangle$	[CF-Object]
(3) Temporal Reasoning	Following the activity involving [Hist-Object] on [Hist-Date], [Subject] next [Relation-past]	[CF-Object]
(4) History Locality	On [Hist-Date], [Subject] [Relation-past]	[Hist-Object]
(5) Future Locality	On [Future-Date], [Subject] [Relation-past]	[Future-Object]
(6) Temporal Hop Backward	Question: Before [Relation-ing] [CF-Object] on [Edit-Date], who or what was the previous entity that [Subject] [Relation-past]? Answer:	[Hist-Object]
(7) Temporal Hop Forward	Question: After [Relation-ing] [CF-Object] on [Edit-Date], who or what was the next entity that [Subject] [Relation-past]? Answer:	[Future-Object]

Table 10: The formalized prompt templates utilized in the TRACE benchmark. Variables enclosed in brackets [] denote dynamic slots populated during evaluation. [Relation-past], [Relation-ing], and [Relation-noun] represent the past tense, present participle, and noun form of the specific ICEWS relation, respectively. [CF-Object] refers to the injected counterfactual target.

Context Chain: UN Security Council leveled accusations against $\rightarrow$ Iran (t_1 : 2007-09-07) $\rightarrow$ Iran (t_2 : 2008-05-15) $\rightarrow$ Iran (t_3 : 2009-03-10) Edit Operation: Modify the target at t_2 (2008-05-15) from <i>Iran</i> to Colombia .			
Metric Dimension	Evaluation Prompt	Pre-Edit Output	GRACE Output
(1) Reliability (Target: Colombia)	On 2008-05-15, UN Security Council leveled accusations against	Iran $\times$	Colombia $\checkmark$
(2) Generality (Target: Colombia)	Regarding the accusation involving UN Security Council on 2008-05-15, the target was	the UN Security... $\times$	the UN Security... $\times$
(4) History Locality (Target: Iran)	On 2007-09-07, UN Security Council leveled accusations against	Iran $\checkmark$	Iran $\checkmark$
(6) Temporal Hop Backward (Target: Iran)	Question: Before leveling accusations against Colombia on 2008-05-15, who or what was the previous entity that UN Security Council leveled accusations against? Answer:	The United States... $\times$	The United States... $\times$

Table 11: A representative case study from the TRACE benchmark evaluating the GPT-J-6B model edited via the GRACE algorithm. The text highlights actual verbatim outputs generated by the model. The symbol $\checkmark$ denotes a successful match with the expected target, while $\times$ denotes a failure.

Part I: Context & Edit Configuration (Metadata)		
Subject: China	Old Object: South Korea	Case Type: Hard
Relation: Criticize or denounce	New Object (Target): Sudan	Method: Modify
Edit Target Date (t_2): 2008-06-04	Adjacent Nodes: Japan (t_1 : 2008-05-06), Japan (t_3 : 2008-06-20)	
Part II: 7-Dimensional Evaluation Prompts		
Metric Dimension	Prompt Template	Expected Target
(1) Reliability	On 2008-06-04, China leveled criticism or denunciation against	Sudan
(2) Generality	On 2008-06-04, the criticism or denunciation concerning China was directed at	Sudan
(3) Temporal Reasoning	Following the activity involving Japan on 2008-05-06, China next leveled criticism or denunciation against	Sudan
(4) History Locality	On 2008-05-06, China leveled criticism or denunciation against	Japan
(5) Future Locality	On 2008-06-20, China leveled criticism or denunciation against	Japan
(6) Temporal Hop Backward	Question: Before leveling criticism or denunciation against Sudan on 2008-06-04, who or what was the previous entity that China leveled criticism or denunciation against? Answer:	Japan
(7) Temporal Hop Forward	Question: After leveling criticism or denunciation against Sudan on 2008-06-04, who or what was the next entity that China leveled criticism or denunciation against? Answer:	Japan

Table 12: A complete data instance sampled from the TRACE benchmark (Modify setup for GPT-J-6B). Part I illustrates the structural metadata of the counterfactual event chain. Part II demonstrates the corresponding natural language prompts generated to evaluate the metrics following the injection of the target entity (*Sudan*).