The Mirage of Model Editing: Revisiting Evaluation in the Wild

Anonymous ACL submission

Abstract

Despite near-perfect results in artificial evaluations, the effectiveness of model editing in real-world applications remains unexplored. To bridge this gap, we propose to study model editing in question answering (QA) by establishing a rigorous evaluation practice to assess the effectiveness of editing methods in correcting LLMs' errors. It consists of QAEdit, a new benchmark derived from popular QA datasets, and a standardized evaluation framework. Our single editing experiments indicate that current editing methods perform substantially worse than previously reported (38.5% vs. ∼96%). Through module analysis and controlled experiments, we demonstrate that this performance decline stems from issues in evaluation practices of prior editing research. One key issue is the inappropriate use of *teacher forcing* in testing prevents error propagation by feeding ground truth tokens (inaccessible in real-world scenarios) as input. Furthermore, we simulate real-world deployment by sequential editing, revealing that current approaches fail drastically with only 1000 edits. Our analysis provides a fundamental reexamination of both the real-world applicability of existing model editing methods and their evaluation practices, and establishes a rigorous evaluation framework with key insights to advance reliable and practical model editing research[1].

1 Introduction

Model editing (Zhang et al., 2024; Wang et al., 2024c) has attracted widespread attention for its promising vision: enabling efficient and precise updates to specific knowledge within pretrained Large Language Models (LLMs) without retraining from scratch. Recent advances report near-perfect results on corresponding benchmarks (Meng et al., 2022; Wang et al., 2024b), suggesting substantial progress toward this goal. However, these results often come from artificial, oversimplified evaluation settings (e.g., identical prompts for editing and testing; more in §4) that may fail to capture real-world complexities. This disparity raises a critical question: *Can these promising results in the literature translate to practical applications?*

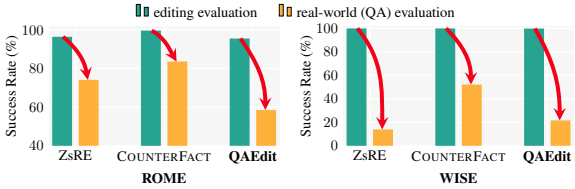


Figure 1: Comparison of editing and real-world (QA) evaluation for ROME and WISE on Llama-2-7b-chat.

To address this question, we propose to study model editing in QA tasks, which provide clear evaluation criteria and broad applicability. This adaptation involves two key components: a real-world dataset and realistic evaluation. For dataset, we create **QAEdit**, a tailored dataset derived from three widely-used QA datasets, enabling editing methods to inject answers from real-world QA tasks into LLMs. For evaluation, we shift from traditional editing evaluation to standard QA evaluation (Gao et al., 2024), assessing editing methods through the performance of edited LLMs on their previously incorrect questions.

Our initial study reveals that current advanced editing methods achieve only a **38.5**% average success rate on QAEdit, significantly lower than the results reported in previous studies. This raises a question: *Does the performance decline stem from QAEdit's real-world complexity, or from the shift of editing to real-world (i.e., QA) evaluation?*

To enable rigorous analysis, starting with single editing experiments, we evaluate six representative methods across three leading LLMs on QAEdit and two established editing benchmarks, using both evaluation frameworks. As illustrated in Figure 1, switching from editing to real-world evaluation

[1]Code and data released at https://anonymous.4open.science/r/12BC.

consistently leads to a significant performance decline across all datasets for each editing method, whether on real-world QA data or previous editing benchmarks. This dramatic performance gap raises two critical questions: *What differences between these frameworks drive such disparity, and which most accurately reflects editing effectiveness?*

To answer them, we carefully examine the experimental setups for both editing and QA evaluations in previous work. From this, we abstract four key modules (*input*, *generation strategy*, *output truncation*, and *metric*) and analyze their variations through controlled experiments. The results expose four critical limitations in editing evaluation: **{❶ input module**: using identical prompts for editing and testing overlooks the variability and unpredictability in real-world queries; **❷ generation strategy**: teacher forcing, which feeds the ground truth as input during decoding, artificially beautifies results by disregarding potential errors in the model's own outputs; **❸ output truncation**: using target answer length to truncate outputs conceals errors (e.g., repetition, irrelevant, or incorrect information) that would occur with natural stopping criteria; **❹ metric**: match ratio may inflate performance by rewarding partial matches of incorrect answers.**}** Among these issues, teacher forcing and target length truncation cause the most significant overestimation, as they rely on ground truth that is unavailable in real-world scenarios. This highlights that **editing evaluation, reliant on such idealized or even unrealistic conditions, fails to accurately measure true editing effectiveness**.

After uncovering evaluation issues through single editing analysis, we now return to our initial question: how do editing methods perform under realistic conditions? In practice, editing requests arrive continuously, making sequential editing a more genuine test of real-world applicability. Under real-world evaluation, our sequential editing experiments show that current methods catastrophically fail to scale, with average success rates dropping to $\sim$**10**% for only 1000 samples.

Our work, for the first time, exposes severe issues in current evaluations of model editing research and demonstrates substantial limitations of existing editing methods under real-world conditions. We hope this work will inspire more rigorous evaluation practices and motivate the development of algorithms that can truly fulfill the promise of model editing: to reliably and scalably update knowledge in LLMs *for real-world applications*.

Our main contributions are as follows.

- We introduce QAEdit, a benchmark tailored for real-world QA tasks, and establish a more practical evaluation protocol.
- We reveal a significant gap between the performance reported in literature and that observed in real-world scenarios.
- We demonstrate that published results are inflated and identify the critical issues and underlying causes in current evaluation practices.
- We expose the severe scalability challenges of current editing methods in practical applications through sequential editing experiments.

2 Related Works

2.1 Model Editing Methodologies

Existing model editing methods can be categorized into the following four types:

Extension based. These methods update LLMs by adding trainable parameters to encode new knowledge, e.g., additional neurons in FFN (Dong et al., 2022; Huang et al., 2023) or specialized memory modules (Hartvigsen et al., 2023; Wang et al., 2024b), while preserving pretrained weights.

Fine-tuning Based. Fine-tuning offers a straightforward approach to update LLMs' knowledge but faces catastrophic forgetting. Recent works mitigate this by constraining parameter changes (Zhu et al., 2020) or leveraging Parameter-Efficient Fine-Tuning (PEFT) (Han et al., 2024) to limit modification scope (Yu et al., 2024; Wang et al., 2024a).

Meta Learning. Employing meta-learning, KE (De Cao et al., 2021), MEND (Mitchell et al., 2022), and MALMEN (Tan et al., 2024) train hypernetworks to predict effective gradients or parameter alterations for knowledge integration.

Locate-Then-Edit. Based on the investigation of knowledge mechanisms in LLMs (Geva et al., 2021, 2022), KN (Dai et al., 2022), ROME (Meng et al., 2022), and PMET (Li et al., 2024b) utilize knowledge attribution and causal tracing to pinpoint target knowledge to specific parameters, then perform localized editing. Furthermore, MEMIT (Meng et al., 2023) and EMMET (Gupta et al., 2024c) extend this for massive editing in a batch.

2.2 Evaluation of Model Editing

Current evaluation of model editing primarily focuses on editing effectiveness and side effects on model capabilities.

```
"Edit Prompt"      : "To whom was Grete Stern married?",
"Edit Target"      : "Horacio Coppola",
"Subject"          : "Grete Stern",
"Rephrased Prompt" : "Who was the spouse of Grete Stern?",
"Locality Prompt"  : "When was the clock tower built in London?",
"Locality Answer"  : "1859"
```

Figure 2: An example from QAEdit.

Effectiveness of Editing. The effectiveness of editing is typically evaluated from four key properties using artificial benchmarks and simplified evaluation settings: i) *reliability*, success rate of editing; ii) *generalization*, adaptability of edited knowledge to paraphrased prompts; iii) *locality*, impact on irrelevant knowledge; iv) *portability*, applicability of edited knowledge in factual reasoning. We refer readers to Yao et al. (2023) for details. In addition to these basic metrics, domain-specific editing tasks have been introduced, e.g., privacy preservation (Wu et al., 2023), bias mitigation (Chen et al., 2024b), and harm injection (Chen et al., 2024a).

Side Effects of Editing. Recent research has also examined the potential side effects of editing on LLMs (Hoelscher-Obermaier et al., 2023; Li et al., 2024c). While locality shares similar objectives, its limited evaluation scope fails to capture the full extent of editing side effects. Recent studies (Yang et al., 2024a; Gu et al., 2024; Gupta et al., 2024b) have revealed that model editing can significantly compromise LLMs' downstream tasks capabilities, motivating a growing research to mitigate such side effects (Ma et al., 2024; Fang et al., 2025).

Discussion. Distinct from aforementioned two aspects of evaluations, this paper presents the first comprehensive evaluation of model editing effectiveness in real-world scenarios. With similar motivations, AKEW (Wu et al., 2024) proposed a new task of unstructured text editing. In contrast, our study rethinks SOTA editing techniques on real-world setting, revealing their limited practical effectiveness and uncovering the pitfalls of traditional editing evaluation.

3 QAEdit

While existing works report remarkable success of model editing on artificial benchmarks (Meng et al., 2022; Wang et al., 2024b), its efficacy in real-world scenarios remains unproven. Here, we propose to study it through QA for its fundamental, universal, and representative nature. Specifically, we apply editing methods to correct LLMs' errors in QA tasks and assess the improvement by re-evaluating edited LLMs on a standard QA evaluation framework, lm-evaluation-harness (Gao et al., 2024).

Method	FT-M	MEND	ROME	MEMIT	GRACE	WISE	Avg.
Accuracy	0.611	0.333	0.585	0.552	0.012	0.216	0.385

Table 1: Accuracy of edited Llama-2-7b-chat on questions it failed before editing in QAEdit.

Since existing editing benchmarks are not derived from or aligned with mainstream QA tasks, we introduce QAEdit, a tailored benchmark to rigorously assess model editing in real-world QA. Specifically, QAEdit is constructed from three widely-used QA datasets with broad real-world coverage: Natural Questions (Kwiatkowski et al., 2019), TriviaQA (Joshi et al., 2017), and SimpleQA (Wei et al., 2024). Details about these datasets are provided in Appendix A.1.

While these benchmarks provide questions and answers as *edit prompts* and *targets* respectively, they lack essential fields that mainstream editing methods require for editing and evaluation. To obtain required *subjects* for editing, we employ GPT-4 (gpt-4-1106-preview) to extract them directly from the questions. To align with the previous editing evaluation protocol, we assess: reliability using original *edit prompts*; generalization through GPT-4 *paraphrased prompts*; and locality using *unrelated QA pairs* from ZsRE locality set[2].

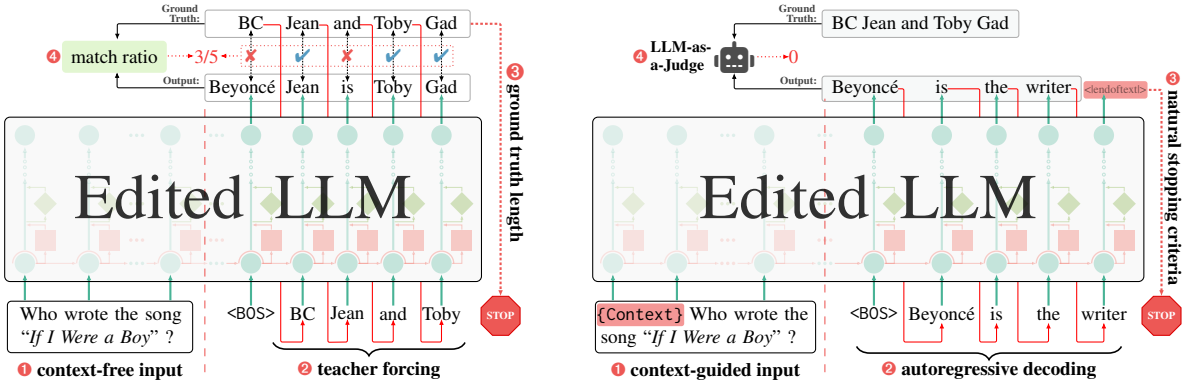
As a result, QAEdit contains 19,249 samples across ten categories, ensuring diverse coverage of QA scenarios. Figure 2 shows a QAEdit entry with all fields. Dataset construction and dataset statistics are detailed in Appendix A.2.

As a preliminary study, we conduct single-edit experiments on Llama-2-7b-chat's failed questions in QAEdit (detailed in §5). As shown in Table 1, after applying SOTA editing methods, the edited models achieve only 38.5% average accuracy under QA evaluation, far below previously reported results (Meng et al., 2023; Wang et al., 2024b). This raises a critical question: *Is the performance degradation attributed to the real-world complexity of QAEdit, or to real-world QA evaluation?*

4 A Tale of Two Evaluation Frameworks

To identify the cause of this performance gap and guide further investigation, we first delve into the experimental setup of both editing and real-world evaluations. We abstract them into four key modules: *input*, *generation strategy*, *output truncation*, and *metric*. This modular paradigm enables systematic comparison between the two evaluation

[2]We exclude portability evaluation as it concerns reasoning rather than our focus on knowledge updating in real-world.



(a) editing evaluation framework (b) real-world evaluation framework

Figure 3: Illustration of editing and real-world evaluation frameworks, each comprising four key modules: ❶ *input*, ❷ *generation strategy*, ❸ *output truncation*, and ❹ *metric*, for measuring reliability, generalization, and locality.

Module	editing	real-world
Input	context-free	context-guided
Generation Strategy	teacher forcing	autoregressive decoding
Output Truncation	ground truth length	natural stopping criteria
Metric	match ratio	LLM-as-a-Judge

Table 2: Key settings of editing and real-world evaluation across all four modules.

frameworks, as shown in Figure 3.

As shown in Figure 3a, we formalize previous works' evaluation pipeline (Yao et al., 2023; Wang et al., 2024b) as **editing evaluation** framework, which implements four modules as follows: i) *input*: using only question without additional context; ii) *generation strategy*: employing teacher forcing to feed ground truth tokens as input during generation; iii) *output truncation*: truncating output to match the length of target answer; iv) *metric*: using token-level match ratio between the target and generated answer as accuracy.

We define **real-world evaluation** framework based on the standard QA evaluation protocol (Gao et al., 2024), which implements these modules differently (Figure 3b): i) *input*: prefixing question with contexts like task instructions; ii) *generation strategy*: adopting autoregressive decoding, where each output serves as input for subsequent generation; iii) *output truncation*: using predefined stop tokens (e.g., ".", "\n", and "<|endoftext|>") as signal to terminate generation; iv) *metric*: employing LLMs as binary judgment based on question, target and generated answers[3]. Notably, we employ LLM-as-a-Judge (Li et al., 2024a) instead of exact match as our evaluation metric, as it has become standard practice in QA evaluation and our human validation confirms its superior alignment with human judgment.

Discussion. Table 2 details the key differences between these evaluation frameworks. Editing evaluation has two types of critical limitations compared to real-world evaluation: i) **oversimplification**: context-free input overlooks the complexity and variability of practical queries, and match ratio rewards partial matches of incorrect answers; ii) **unreasonableness**: teacher forcing generation and corresponding truncation to the target length leak ground truth information that should remain inaccessible during testing. These artificial settings result in a significant gap between research on editing and its practical applications.

5 Analysis on Benchmark & Evaluation

The preliminary analysis and theoretical comparison in §3 and §4 reveal a notable disparity between editing and real-world evaluation. To rigorously address the question raised in §3—whether the performance gap stems from differences in dataset or evaluation—we conduct systematic single-edit experiments, where each edit is independently applied to the original model from scratch.

5.1 Experimental Setup

This section outlines the experimental setup used in all subsequent experiments, unless stated otherwise. Due to space limitations, further details are provided in Appendix A.4.

Editing Methods. To ensure comprehensive coverage, we employ six diverse and representative editing techniques across four categories: extension based (**GRACE**, Hartvigsen et al., 2023 and **WISE**, Wang et al., 2024b), fine-tuning based (**FT-M**, Zhang et al., 2024), meta-learning (**MEND**, Mitchell et al., 2022), and locate-then-edit (**ROME**, Meng et al., 2022 and **MEMIT**, Meng et al., 2023).

[3]Detailed prompt is provided in Appendix A.3.

	Method	ZsRE				COUNTERFACT				QAEdit			
		Reliability		Generalization		Reliability		Generalization		Reliability		Generalization	
		Edit.	Real.	Edit.	Real.	Edit.	Real.	Edit.	Real.	Edit.	Real.	Edit.	Real.
Llama-2-7b-chat	FT-M	1.000	0.562	0.950	0.470	1.000	0.867	0.503	0.426	1.000	0.611	0.966	0.560
	MEND	0.967	0.288	0.949	0.244	0.997	0.478	0.425	0.183	0.942	0.333	0.900	0.328
	ROME	0.964	0.741	0.811	0.656	0.996	0.836	0.452	0.420	0.955	0.585	0.744	0.411
	MEMIT	0.950	0.685	0.858	0.634	0.997	0.797	0.513	0.460	0.929	0.552	0.791	0.450
	GRACE	0.986	0.033	0.319	0.029	0.998	0.013	0.114	0.008	0.983	0.012	0.383	0.087
	WISE	0.999	0.139	0.973	0.081	0.999	0.521	0.612	0.104	0.998	0.216	0.877	0.122
Mistral-7b	FT-M	1.000	0.441	0.824	0.358	1.000	0.733	0.330	0.220	1.000	0.562	0.862	0.503
	MEND	0.977	0.719	0.963	0.657	0.820	0.431	0.355	0.149	0.903	0.544	0.895	0.516
	ROME	0.757	0.608	0.717	0.573	0.965	0.866	0.466	0.488	0.845	0.555	0.735	0.435
	MEMIT	0.868	0.707	0.842	0.670	0.962	0.887	0.539	0.583	0.850	0.563	0.788	0.485
	GRACE	0.995	0.035	0.350	0.029	1.000	0.011	0.110	0.006	0.991	0.018	0.421	0.080
	WISE	0.948	0.033	0.903	0.025	0.868	0.129	0.420	0.027	0.979	0.024	0.906	0.064
Llama-3-8b	FT-M	1.000	0.706	0.995	0.698	1.000	0.916	0.588	0.613	1.000	0.560	0.988	0.576
	ROME	0.996	0.820	0.971	0.789	0.999	0.877	0.422	0.491	0.987	0.691	0.865	0.570
	MEMIT	0.982	0.803	0.961	0.781	0.998	0.882	0.516	0.557	0.967	0.649	0.886	0.566
	GRACE	0.999	0.036	0.261	0.032	1.000	0.008	0.008	0.005	0.999	0.018	0.366	0.103
	WISE	0.859	0.091	0.825	0.075	0.807	0.212	0.508	0.075	0.910	0.121	0.876	0.138
	Average	0.956	0.438	0.792	0.400	0.965	0.557	0.405	0.283	0.956	0.389	0.779	0.351

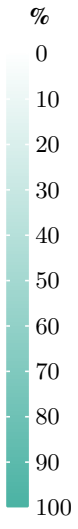


Table 3: Comparison between editing evaluation (**Edit.**) and real-world evaluation (**Real.**). Cell background shading indicates performance drop from Edit. to Real., with darker shades indicating greater decreases.

All methods are implemented using `EasyEdit`[4]. Due to the inconsistent keys implementation in ROME, we adopt its refined variant R-ROME (Gupta et al., 2024a; Yang et al., 2024b) instead.

Edited LLMs. In line with prior research (Wang et al., 2024b; Fang et al., 2025), we test three leading open-source LLMs: **Llama-2-7b-chat** (Touvron et al., 2023), **Mistral-7b** (Jiang et al., 2023), and **Llama-3-8b** (Meta, 2024). Greedy decoding is used for all models, aligning with prior research. Results for MEND with Llama-3-8b are excluded due to architectural incompatibility.

Editing Datasets. We employ QAEdit along with two prevalent benchmarks, ZsRE (Levy et al., 2017) and COUNTERFACT (Meng et al., 2022), for a rigorous investigation. For QAEdit, we evaluate the edited LLMs using only samples that their unedited counterparts initially answered incorrectly. This yields evaluation sets of 12,715, 10,213, 10,467 samples for Llama-2-7b-chat, Mistral-7b, and Llama-3-8b, respectively. For ZsRE and COUNTERFACT, we use their established test sets, each with 10,000 records.

5.2 Results & Analysis

The experimental results are presented in Table 3. Due to the minor side effects in single editing scenarios, the consistently favorable locality results are moved to Appendix A.5.

Benchmark Perspective: QAEdit exhibits moderately lower editing reliability compared to ZsRE and CounterFact, reflecting its diverse and challenging nature as a real-world benchmark. However, this modest gap is insufficient to explain the significant discrepancy observed in our earlier analysis.

Method Perspective: i) Recent state-of-the-art methods, GRACE and WISE, exhibit the most significant decrease, with both reliability and generalization dropping below 5%. This decline mainly stems from their edited models generating erroneous information after producing the correct answers, detailed in §6.3. ii) In comparison, traditional methods like FT-M and ROME exhibit superior stability and preserve a certain level of effectiveness in real-world evaluation.

Evaluation Perspective: i) Performance on each benchmark drops sharply from editing evaluation (∼96%) to real-world evaluation (e.g., 43.8% on ZsRE and 38.9% on QAEdit), indicating that **editing evaluation substantially overestimates the effectiveness of editing methods**. ii) Unlike editing evaluation, which reports consistently near-perfect results across all methods and benchmarks, real-world evaluation effectively distinguishes them, providing valuable insights for future research.

6 Controlled Study of Editing Evaluation

This section presents controlled experiments to systematically investigate how different module variations in editing evaluation (outlined in §4) contribute to performance overestimation. Due to resource and space limitations, we conduct experiments on Llama-3-8b with 3,000 randomly sam-

[4] https://github.com/zjunlp/EasyEdit

Input	FT-M	ROME	MEMIT	GRACE	WISE
Context-free	1.000	0.985	0.965	0.998	0.908
Context-guided	0.937	0.930	0.907	0.412	0.838

Table 4: Reliability score for different input formats on Llama-3-8b under teacher forcing generation, truncation at ground truth length, and match ratio metric.

Generation Strategy	FT-M	ROME	MEMIT	GRACE	WISE
❶ *context-free*, ❸ ground truth length, ❹ match ratio					
Teacher forcing	1.000	0.985	0.965	0.998	0.908
Autoregressive decoding	1.000	0.967	0.929	0.996	0.765
❶ *context-guided*, ❸ ground truth length, ❹ match ratio					
Teacher forcing	0.937	0.930	0.907	0.412	0.838
Autoregressive decoding	0.800	0.851	0.786	0.036	0.592

Table 5: Reliability of different generation strategies on Llama-3-8b under two prompt strategies.

pled QAEdit instances, where the findings generalizable across other LLMs and datasets.

6.1 Input

This subsection empirically isolates how idealistic prompts may lead to overestimated results in editing evaluation. Specifically, we compare context-free prompts with real-world input formats that include task instructions, while keeping all other modules identical. Detailed prompts are provided in Appendix A.6.

Table 4 shows that incorporating task instruction degrades performance across all editing methods, with GRACE showing the most significant decline due to its weak generalization. This trend contrasts with the behavior of original Llama-3-8b, where task instructions usually improve results (Grattafiori et al., 2024). These findings reveal that **using identical prompts for editing and testing in current editing evaluation, while yielding optimistic results, may fail to reflect editing effectiveness under diverse real-world inputs**.

6.2 Generation Strategy

Here, we examine how teacher forcing in the generation strategy contributes to the inflated results in editing evaluation. We compare reliability of teacher forcing and autoregressive decoding under two distinct input formats, while keeping all other modules consistent.

As depicted in Table 5, switching from teacher forcing to autoregressive decoding consistently leads to performance degradation across all methods, with lower-performing methods exhibiting more substantial decline. The underlying reason for this phenomena is that teacher forcing prevents error propagation by feeding ground truth tokens as input, while autoregressive decoding allows errors to cascade. Although teacher forcing is beneficial for stabilizing LLM training, it should be avoided during testing, where ground truth is unavailable. Our results demonstrate that **inappropriate use of teacher forcing in evaluation artificially elevates editing performance, especially for methods with poor real-world performance**.

Truncation Strategy	FT-M	ROME	MEMIT	GRACE	WISE
❶ *context-free*, ❷ autoregressive decoding, ❹ LLM-as-a-Judge					
Ground truth length	1.000	0.954	0.886	0.992	0.700
Natural stop criteria	0.202	0.478	0.461	0.301	0.046
❶ *context-guided*, ❷ autoregressive decoding, ❹ LLM-as-a-Judge					
Ground truth length	0.751	0.783	0.704	0.003	0.482
Natural stop criteria	0.528	0.556	0.529	0.000	0.108

Table 6: Reliability score under different answer truncation strategies on Llama-3-8b.

Meaningless Repetition	
`Input Prompt`	Who got the first Nobel Prize in physics?
`Target Answer`	Wilhelm Conrad Röntgen
`Natural Stop`	Wilhelm Conrad Röntgen Wilhelm Conrad Röntgen Wilhelm Conrad Röntgen ...
Irrelevant Information	
`Input Prompt`	Who was the first lady nominated member of the Rajya Sabha?
`Target Answer`	Mary Kom
`Natural Stop`	Mary Kom is the first woman boxer to qualify for the Olympics
Incorrect Information	
`Input Prompt`	When does April Fools' Day end at noon?
`Target Answer`	April 1st
`Natural Stop`	April 1st ends at noon on April 2nd

Table 7: Examples of additionally generated content beyond ground truth length under natural stop criteria.

6.3 Output Truncation

Besides leaking ground truth tokens, teacher forcing also implicitly controls output length by aligning with ground truth length. However, this is not applicable in real-world scenarios where ground truth is unavailable. In practice, during inference, generation typically terminates based on predefined stop tokens, e.g., "`<|endoftext|>`" (Gao et al., 2024). Here, we analyze these two truncation strategies by employing GPT-4o-mini as a binary judge to assess correctness (detailed in §6.4), since length discrepancies between generated and target answers preclude the use of match ratio metric.

As shown in Table 6, truncation based on natural stop criteria significantly reduces editing performance across all methods. To identify the underlying causes, we analyze the content truncated at

Method	Llama-2-7b-chat				Mistral-7b				Llama-3-8b			
	Reliability		Locality		Reliability		Locality		Reliability		Locality	
	Edit.	Real.	Edit.	Real.	Edit.	Real.	Edit.	Real.	Edit.	Real.	Edit.	Real.
FT-M	0.973	0.531	0.420	0.072	0.960	0.454	0.573	0.204	0.925	0.229	0.127	0.004
MEND	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	–	–	–	–
ROME	0.114	0.001	0.028	0.001	0.059	0.001	0.052	0.028	0.034	0.001	0.020	0.000
MEMIT	0.057	0.002	0.030	0.000	0.058	0.002	0.031	0.000	0.000	0.000	0.000	0.000
GRACE	0.370	0.015	1.000	1.000	0.416	0.018	1.000	1.000	0.368	0.022	1.000	1.000
WISE	0.802	0.195	0.676	0.184	0.735	0.060	0.214	0.003	0.526	0.072	0.743	0.104
Average	0.386	0.124	0.359	0.210	0.494	0.089	0.312	0.206	0.371	0.065	0.378	0.222

Table 8: Results of sequential editing on QAEdit under editing evaluation (**Edit.**) and real-world evaluation (**Real.**).

Metric	FT-M	ROME	MEMIT	GRACE	WISE
❶ *context-free*, ❷ autoregressive decoding, ❸ ground truth length					
Match ratio	1.000	0.967	0.929	0.996	0.765
LLM-as-a-Judge	1.000	0.954	0.886	0.992	0.700
❶ *context-guided*, ❷ autoregressive decoding, ❸ ground truth length					
Match ratio	0.800	0.851	0.786	0.036	0.592
LLM-as-a-Judge	0.751	0.783	0.704	0.003	0.482

Table 9: Reliability score derived from different metric judgment on Llama-3-8b.

both the ground truth length and the natural stop criteria. Our analysis reveals that, under natural stop criteria, the edited models typically generate content beyond the ground truth length, introducing *meaningless repetition* and *irrelevant or incorrect information*, as evidenced in Table 7.

These findings demonstrate that **irrational truncation in editing evaluation masks subsequent errors that emerge in real-world scenarios, resulting in overestimated performance**. As shown in Table 6, although context-guided prompting enhances generation termination, it still fails to address the fundamental limitations. Such pitfalls in current approaches, overlooked by traditional evaluation, highlight the need to explore more effective ways to express edited knowledge.

6.4 Metric

As explained in §4, the match ratio metric could lead to inflated performance. To quantify this effect, we compare match ratio against LLM-as-a-Judge, specifically using GPT-4o-mini. Since match ratio requires length parity with targets, we autoregressively generate sequences to target length for both metircs to ensure fair comparison.

The results presented in Table 9 reveal that **the match ratio metric indeed overestimates the performance of edited models**. Moreover, a lower match ratio often indicates a smaller proportion of fully correct answers, resulting in worse performance in LLM evaluation.

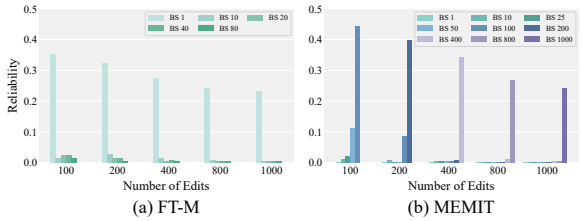
7 (Sequential) Editing in the Wild

Although our analysis via single editing reveals limitations in current editing evaluation, such isolated editing fails to capture the continuous, large-scale demands of editing in real-world scenarios. Therefore, we now address our primary research question: testing model editing under real-world evaluation via sequential editing, a setup that better reflects practical requirements.

7.1 Sample-wise Sequential Editing

Experimental Setup. Following established protocols (Huang et al., 2023; Hartvigsen et al., 2023), we evaluate editing methods with a batch size of 1, i.e., updating knowledge incrementally one sample at a time. We keep the same setup as in §5.1, but limit to 1000 samples per dataset, as existing methods perform significantly worse with more edits. For QAEdit, the chosen samples are incorrectly answered by all pre-edit LLMs. Given the notable side effects in sequential editing (Yang et al., 2024a), we focus on the evaluation of *reliability* and *locality*, with *generalization* results provided in Appendix A.7.

Results & Analysis. The results on QAEdit are shown in Table 8, with similar findings for ZsRE and COUNTERFACT in Appendix A.8. i) In real–world evaluation with sequential editing, all methods except FT-M exhibit nearly unusable performance (only 9.3% average reliability), with FT-M achieving a 40.5% average reliability. ii) The gap between editing and real-world evaluation further confirms the evaluation issues we discussed in §6. iii) The notably low average locality of 21.3% highlights the severe disruption to LLMs. While GRACE effectively preserves unrelated knowledge through external edit modules, it struggles with knowledge updating.



(a) FT-M (b) MEMIT

Figure 4: Impact of batch size (BS) when editing Llama-3-8b with FT-M and MEMIT on QAEdit.

7.2 Mini-Batch Sequential Editing

Real-world applications often batch multiple edits together for efficient processing of high-volume demands. Moreover, Pan et al. (2024) suggest increasing batch size may alleviate the side effects of sequential editing. Thus, this section investigates whether increasing the batch size could serve as a potential solution to the practical challenges faced by current editing methods.

Experiment Setup. Following the experimental setup in §7.1, we evaluate three batch-capable editing algorithms: FT-M, MEND, and MEMIT. Due to VRAM constraints (80GB A800), we empirically set the maximum testable batch sizes: 80 for FT-M, 16 for MEND, and 1000 for MEMIT.

Results & Analysis. Figure 4 presents the editing performance with varying batch sizes, evaluated across various-sized QAEdit subsets. Despite experimenting with various batch sizes, all methods show consistently limited performance, with the highest score below 30% for 1000 edits. The all-zero performance of MEND are provided in Appendix A.9. Notably, Figure 4 presents opposite trends: i) MEMIT achieves optimal performance only when editing all requests in a single batch, with performance decreasing sharply as batch size decreases. ii) In contrast, FT-M performs best at a batch size of 1 but degrades drastically as batch size increases. The divergence may arise from their distinct batch editing mechanisms: FT-M optimizes for aggregate batch-level loss, potentially compromising individual edit accuracy; whereas MEMIT estimates parametric changes individually before integration, facilitating effective batch edits.

Further Analysis. To gain insights into the poor final performance, we also investigate how editing effectiveness changes during continuous editing. Specifically, we randomly partition 100 QAEdit samples into 5 batches of 20 samples each. Using MEMIT on Llama-3-8b, we iteratively edit each batch while evaluating the edited model on each previously edited batch separately to track dynamics of editing effectiveness.

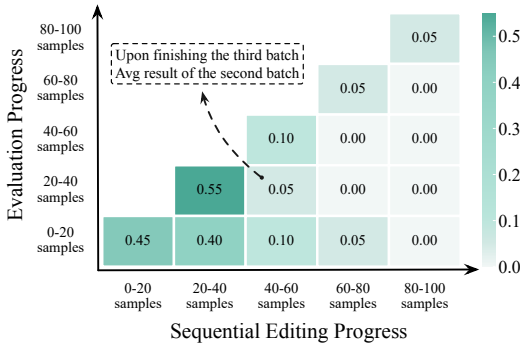


Figure 5: Reliability evolution of sequential editing on Llama-3-8b, with repeated evaluation of previous batches after each new edit batch (batch size = 20).

Figure 5 reveals two key insights: i) While the first batch exhibits high initial reliability, its performance declines sharply with subsequent editing, suggesting that new edits disrupt the knowledge injected in earlier batches. ii) As editing progresses, the effectiveness of MEMIT decreases rapidly. These findings reveal the key challenges of sequential editing: **progressive loss of previously edited knowledge coupled with decreasing effectiveness in editing new knowledge**.

8 Conclusion and Future Works

In this paper, we present the first systematic investigation that exposes the gap between theoretical advances and practical effectiveness of model editing by real-world QA evaluation. Our proposed QAEdit benchmark and real-world evaluation demonstrate that current model editing techniques exhibit significant limitations in practical scenarios, particularly under sequential editing. Furthermore, we reveal that this significant discrepancy from previously reported results stems from unrealistic evaluation adopted in prior model editing research. Through modular analysis and extensive controlled experiments, we uncover fundamental issues in current editing evaluation that inflate reported performance. This work establishes new evaluation standards for model editing and provides valuable insights that will inspire the development of more robust editing methods, ultimately enabling reliable and efficient knowledge updates in LLMs for real-world applications.

In future research, we aim to develop editing methods that can i) generalize robustly across diverse scenarios with reliable self-termination, and ii) support extensive sequential updates while maintaining the capabilities of edited LLMs.

Limitations

We acknowledge following limitations of our work:

- This work provides an existence proof of fundamental issues of evaluation in model editing, rather than attempting an exhaustive assessment of all existing approaches and LLMs. Due to resource constraints, we focus on representative methods and LLMs to demonstrate the issues and challenges, as exhaustive testing of all approaches is neither feasible nor necessary for establishing our findings.

- Our research makes the first systematic investigation into previously overlooked evaluation issues in model editing, prioritizing the identification and analysis of these fundamental challenges rather than solution development. Our work focuses on comprehensive analysis of these issues, uncovering their root causes and providing insights into factors affecting editing effectiveness. While presenting promising directions for future research, developing solutions to these challenges remains beyond our current scope.

- Our study focuses exclusively on parameter-based editing methods, without investigating in-context learning based *knowledge editing* approaches which leverage external information. While these approaches may achieve superior performance on QA tasks, our primary objective is not to advocate for any particular approach, but to critically revisit current practices in the field and provide insights for future development. We believe efficient parameter-based editing approaches have their unique advantages and represent a valuable direction worth pursuing, despite current challenges in real-world applications.

Ethics Statement

Data. All data used in our research are publicly available and do not raise any privacy concerns.

AI Writing Assistance. We employ LLMs to polish our original content, focusing on correcting grammatical errors and enhancing clarity, rather than generating new content or ideas.

References

Canyu Chen, Baixiang Huang, Zekun Li, Zhaorun Chen, Shiyang Lai, Xiongxiao Xu, Jia-Chen Gu, Jindong Gu, Huaxiu Yao, Chaowei Xiao, Xifeng Yan, William Yang Wang, Philip Torr, Dawn Song, and Kai Shu. 2024a. Can editing llms inject harm? *Preprint*, arXiv:2407.20224.

Ruizhe Chen, Yichen Li, Zikai Xiao, and Zuozhu Liu. 2024b. Large language model bias mitigation from the perspective of knowledge editing. In *ICLR 2024 Workshop on Secure and Trustworthy Large Language Models*.

Damai Dai, Li Dong, Yaru Hao, Zhifang Sui, Baobao Chang, and Furu Wei. 2022. Knowledge neurons in pretrained transformers. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics*, pages 8493–8502, Dublin, Ireland. Association for Computational Linguistics.

Nicola De Cao, Wilker Aziz, and Ivan Titov. 2021. Editing factual knowledge in language models. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6491–6506, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Qingxiu Dong, Damai Dai, Yifan Song, Jingjing Xu, Zhifang Sui, and Lei Li. 2022. Calibrating factual knowledge in pretrained language models. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 5937–5947, Abu Dhabi, United Arab Emirates. 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 model editing for language models. In *The Thirteenth International Conference on Learning Representations*.

Leo Gao, Jonathan Tow, Baber Abbasi, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Alain Le Noac'h, Haonan Li, Kyle McDonell, Niklas Muennighoff, Chris Ociepa, Jason Phang, Laria Reynolds, Hailey Schoelkopf, Aviya Skowron, Lintang Sutawika, Eric Tang, Anish Thite, Ben Wang, Kevin Wang, and Andy Zou. 2024. A framework for few-shot language model evaluation.

Mor Geva, Avi Caciularu, Kevin Wang, and Yoav Goldberg. 2022. Transformer feed-forward layers build predictions by promoting concepts in the vocabulary space. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 30–45, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Mor Geva, Roei Schuster, Jonathan Berant, and Omer Levy. 2021. Transformer feed-forward layers are key-value memories. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 5484–5495, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, and et al. 2024. The llama 3 herd of models. *Preprint*, arXiv:2407.21783.

Jia-Chen Gu, Hao-Xiang Xu, Jun-Yu Ma, Pan Lu, Zhen-Hua Ling, Kai-Wei Chang, and Nanyun Peng. 2024. Model editing harms general abilities of large language models: Regularization to the rescue. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 16801–16819, Miami, Florida, USA. Association for Computational Linguistics.

Akshat Gupta, Sidharth Baskaran, and Gopala Anumanchipalli. 2024a. Rebuilding ROME : Resolving model collapse during sequential model editing. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 21738–21744, Miami, Florida, USA. Association for Computational Linguistics.

Akshat Gupta, Anurag Rao, and Gopala Anumanchipalli. 2024b. Model editing at scale leads to gradual and catastrophic forgetting. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 15202–15232, Bangkok, Thailand. Association for Computational Linguistics.

Akshat Gupta, Dev Sajnani, and Gopala Anumanchipalli. 2024c. A unified framework for model editing. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 15403–15418, Miami, Florida, USA. Association for Computational Linguistics.

Zeyu Han, Chao Gao, Jinyang Liu, Jeff Zhang, and Sai Qian Zhang. 2024. Parameter-efficient fine-tuning for large models: A comprehensive survey. *Preprint*, arXiv:2403.14608.

Thomas Hartvigsen, Swami Sankaranarayanan, Hamid Palangi, Yoon Kim, and Marzyeh Ghassemi. 2023. Aging with GRACE: Lifelong model editing with discrete key-value adaptors. In *Thirty-seventh Conference on Neural Information Processing Systems*.

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.

Zeyu Huang, Yikang Shen, Xiaofeng Zhang, Jie Zhou, Wenge Rong, and Zhang Xiong. 2023. Transformer-patcher: One mistake worth one neuron. In *The Eleventh International Conference on Learning Representations*.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, and et al. 2023. Mistral 7b. *Preprint*, arXiv:2310.06825.

Mandar Joshi, Eunsol Choi, Daniel Weld, and Luke Zettlemoyer. 2017. TriviaQA: A large scale distantly supervised challenge dataset for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1601–1611, Vancouver, Canada. Association for Computational Linguistics.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, and et al. 2019. Natural questions: A benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:452–466.

Omer Levy, Minjoon Seo, Eunsol Choi, and Luke Zettlemoyer. 2017. Zero-shot relation extraction via reading comprehension. In *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017)*, pages 333–342, Vancouver, Canada. Association for Computational Linguistics.

Dawei Li, Bohan Jiang, Liangjie Huang, Alimohammad Beigi, Chengshuai Zhao, Zhen Tan, Amrita Bhattacharjee, Yuxuan Jiang, Canyu Chen, Tianhao Wu, et al. 2024a. From generation to judgment: Opportunities and challenges of llm-as-a-judge. *arXiv preprint arXiv:2411.16594*.

Xiaopeng Li, Shasha Li, Shezheng Song, Jing Yang, Jun Ma, and Jie Yu. 2024b. Pmet: precise model editing in a transformer. In *Proceedings of the Thirty-Eighth AAAI Conference on Artificial Intelligence and Thirty-Sixth Conference on Innovative Applications of Artificial Intelligence and Fourteenth Symposium on Educational Advances in Artificial Intelligence*, AAAI'24/IAAI'24/EAAI'24. AAAI Press.

Zhoubo Li, Ningyu Zhang, Yunzhi Yao, Mengru Wang, Xi Chen, and Huajun Chen. 2024c. Unveiling the pitfalls of knowledge editing for large language models. In *The Twelfth International Conference on Learning Representations*.

Jun-Yu Ma, Hong Wang, Hao-Xiang Xu, Zhen-Hua Ling, and Jia-Chen Gu. 2024. Perturbation-restrained sequential model editing. *Preprint*, arXiv:2405.16821.

Kevin Meng, David Bau, Alex J Andonian, and Yonatan Belinkov. 2022. Locating and editing factual associations in GPT. In *Advances in Neural Information Processing Systems*.

Kevin Meng, Arnab Sen Sharma, Alex J Andonian, Yonatan Belinkov, and David Bau. 2023. Mass-editing memory in a transformer. In *The Eleventh International Conference on Learning Representations*.

Meta. 2024. Introducing meta llama 3: The most capable openly available llm to date. `https://ai.meta.com/blog/meta-llama-3/`.

Eric Mitchell, Charles Lin, Antoine Bosselut, Chelsea Finn, and Christopher D Manning. 2022. Fast model editing at scale. In *International Conference on Learning Representations*.

Tsung-Hsuan Pan, Chung-Chi Chen, Hen-Hsen Huang, and Hsin-Hsi Chen. 2024. "why" has the least side effect on model editing. *Preprint*, arXiv:2409.18679.

Chenmien Tan, Ge Zhang, and Jie Fu. 2024. Massive editing for large language models via meta learning. In *International Conference on Learning Representations*.

Hugo Touvron, Louis Martin, Kevin Stone, and et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *Preprint*, arXiv:2307.09288.

Haoyu Wang, Tianci Liu, Ruirui Li, Monica Xiao Cheng, Tuo Zhao, and Jing Gao. 2024a. RoseLoRA: Row and column-wise sparse low-rank adaptation of pre-trained language model for knowledge editing and fine-tuning. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 996–1008, Miami, Florida, USA. Association for Computational Linguistics.

Peng Wang, Zexi Li, Ningyu Zhang, Ziwen Xu, Yunzhi Yao, Yong Jiang, Pengjun Xie, Fei Huang, and Huajun Chen. 2024b. WISE: Rethinking the knowledge memory for lifelong model editing of large language models. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

Song Wang, Yaochen Zhu, Haochen Liu, Zaiyi Zheng, Chen Chen, and Jundong Li. 2024c. Knowledge editing for large language models: A survey. *ACM Computing Surveys*, 57(3):1–37.

Jason Wei, Nguyen Karina, Hyung Won Chung, Yunxin Joy Jiao, Spencer Papay, Amelia Glaese, John Schulman, and William Fedus. 2024. Measuring short-form factuality in large language models. *Preprint*, arXiv:2411.04368.

Xiaobao Wu, Liangming Pan, William Yang Wang, and Anh Tuan Luu. 2024. AKEW: Assessing knowledge editing in the wild. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 15118–15133, Miami, Florida, USA. Association for Computational Linguistics.

Xinwei Wu, Junzhuo Li, Minghui Xu, Weilong Dong, Shuangzhi Wu, Chao Bian, and Deyi Xiong. 2023. DEPN: Detecting and editing privacy neurons in pretrained language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 2875–2886, Singapore. Association for Computational Linguistics.

Wanli Yang, Fei Sun, Xinyu Ma, Xun Liu, Dawei Yin, and Xueqi Cheng. 2024a. The butterfly effect of model editing: Few edits can trigger large language models collapse. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 5419–5437, Bangkok, Thailand. Association for Computational Linguistics.

Wanli Yang, Fei Sun, Jiajun Tan, Xinyu Ma, Du Su, Dawei Yin, and Huawei Shen. 2024b. The fall of ROME: Understanding the collapse of LLMs in model editing. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 4079–4087, Miami, Florida, USA. Association for Computational Linguistics.

Yunzhi Yao, Peng Wang, Bozhong Tian, Siyuan Cheng, Zhoubo Li, Shumin Deng, Huajun 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.

Lang Yu, Qin Chen, Jie Zhou, and Liang He. 2024. Melo: enhancing model editing with neuron-indexed dynamic lora. In *Proceedings of the Thirty-Eighth AAAI Conference on Artificial Intelligence and Thirty-Sixth Conference on Innovative Applications of Artificial Intelligence and Fourteenth Symposium on Educational Advances in Artificial Intelligence*, AAAI'24/IAAI'24/EAAI'24. AAAI Press.

Ningyu Zhang, Yunzhi Yao, Bozhong Tian, and et al. 2024. A comprehensive study of knowledge editing for large language models. *Preprint*, arXiv:2401.01286.

Chen Zhu, Ankit Singh Rawat, Manzil Zaheer, Srinadh Bhojanapalli, Daliang Li, Felix Yu, and Sanjiv Kumar. 2020. Modifying memories in transformer models. *Preprint*, arXiv:2012.00363.

Category	Example	Count
Art & Culture	Who wrote the song the glory of love?	5277
History & Politics	Who wrote the first declaration of human rights?	4070
People & Biographies	Which award did Reza Aslan receive in 2014?	2188
Geography & Environment	Which is the largest saltwater lake in India?	1954
Science & Technology	Which year was the actinide concept proposed?	1829
Sports & Leisure	In what year did Kristin Otto retire from swimming?	1807
Health & Medicine	Where are the cones in the eye located?	771
Society & Humanities	Which is the ring finger for male in India?	573
Economics & Business	When is the world consumer right day celebrated?	463
Others	What kind of beer is St. Pauli Girl?	317

Table 10: Statistics and examples of QAEdit, encompassing ten categories of knowledge. The underlined content represents the subjects identified by GPT-4.

A Appendix

A.1 Detailed Introduction of QA Datasets

Natural Questions (NQ) (Kwiatkowski et al., 2019) is a comprehensive question-answering (QA) dataset that contains real questions posed by users to the Google search, paired with high-quality, human-verified answers. The dataset consists of over 300,000 question-answer pairs, with each question derived from user queries on Google Search. These questions cover a wide variety of topics, ranging from fact-based inquiries to more complex, open-ended questions. The golden answers are sourced from Wikipedia pages, ensuring their accuracy and relevance. We adopt the test set of NQ, which contains 3610 samples, to construct our QAEdit benchmark.

TriviaQA (Joshi et al., 2017) is a large-scale QA dataset designed specifically for evaluating models on trivia-style question answering. It contains over 650,000 question-answer pairs sourced from trivia websites and is curated by trivia enthusiasts. These questions are often fact-based and test the model's ability to retrieve information from large text corpora. We utilize 11,313 samples from the TriviaQA test set to construct QAEdit.

SimpleQA (Wei et al., 2024) is a challenging QA benchmark specifically designed to test fact-seeking question-answering models. It contains 4326 question-answer pairs curated by OpenAI, with an emphasis on short-form factuality. The questions in SimpleQA are concise, direct, and designed to probe factual knowledge. Unlike more general-purpose QA datasets, SimpleQA emphasizes clarity and the ability of models to provide precise, factually accurate answers. We employ all samples from SimpleQA for QAEdit construction.

A.2 Construction and Statistics of QAEdit

In this section, we describe the detailed construction procedures and statistics of QAEdit.

While aforementioned QA benchmarks provide questions and answers as *edit prompts* and *targets*, they lack *subjects* for editing, as well as *rephrased prompts* and *locality QA pairs* to evaluate generalization and locality. To supplement the missing fields, our construction procedures encompass the following steps: i) We employ GPT-4 (gpt-4-1106-preview) to extract the subjects directly from the edit prompts. To improve the accuracy of extraction, we prompt the model with 5-shot examples to utilize its in-context learning capability, which can be seen in Figure 7. ii) We utilize GPT-4 to paraphrase the edit prompts to obtain rephrased prompts. Considering that paraphrasing questions is easy for GPT-4, the specific instruction is straightforward and is presented in Figure 8. Furthermore, we manually reviewed some of the rephrased results and found them to be highly effective. iii) Moreover, for each sample of QAEdit, we randomly select a QA pair from the locality sets of ZsRE (Levy et al., 2017) as locality prompt and corresponding answer to assess locality.

As a result, our QAEdit benchmark encompasses ten categories of knowledge, covering mainstream topics with significant real-world impact. The statistical information and examples of each category are presented in Table 10.

A.3 Prompt of LLM-as-a-Judge

In light of the significant advancements in LLM-as-a-Judge (Li et al., 2024a), we employ GPT-4o-mini to perform binary judgments based on the provided questions, target answers, and generated responses. Following previous work (Wei et al., 2024), our complete prompt is presented in Figure 9.

A.4 Detailed Experimental Setup

A.4.1 Editing Methods

FT-M (Zhang et al., 2024) is an enhanced version of FT-L (Zhu et al., 2020; Meng et al., 2022). FT-L introduces an l_∞-norm constraint into the fine-tuning objective to explicitly restrict the parameter changes between the original and edited models, thereby mitigating side effects on unrelated knowledge. However, FT-L deviates from the original fine-tuning objective by using only the last token's prediction to maximize the probability of all tokens in the target sequence. To address this issue, FT-M improves upon FT-L by applying the cross-entropy loss to the target answer while masking the original text, which aligns more closely with the traditional fine-tuning objective and enhances performance.

MEND (Mitchell et al., 2022) employs a hypernetwork to learn low-rank decompositions of standard

Method	ZsRE		COUNTERFACT		QAEdit	
	Edit.	Real.	Edit.	Real.	Edit.	Real.
Llama-2-7b-chat						
FT-M	0.979	0.875	0.672	0.592	0.963	0.848
MEND	0.990	0.922	0.581	0.649	0.981	0.891
ROME	0.995	0.946	0.972	0.939	0.991	0.929
MEMIT	0.989	0.920	0.953	0.905	0.980	0.881
GRACE	1.000	1.000	1.000	1.000	1.000	1.000
WISE	1.000	0.999	0.830	0.958	1.000	0.999
Mistral-7b						
FT-M	0.994	0.937	0.823	0.760	0.980	0.943
MEND	0.994	0.903	0.618	0.665	0.970	0.889
ROME	0.870	0.839	0.964	0.908	0.990	0.959
MEMIT	0.994	0.950	0.946	0.884	0.982	0.935
GRACE	1.000	1.000	1.000	1.000	1.000	1.000
WISE	1.000	1.000	0.840	0.967	0.999	1.000
Llama-3-8b						
FT-M	0.953	0.597	0.243	0.138	0.917	0.610
ROME	0.994	0.923	0.931	0.845	0.982	0.920
MEMIT	0.988	0.889	0.918	0.828	0.967	0.881
GRACE	1.000	1.000	1.000	1.000	1.000	1.000
WISE	0.993	0.873	0.847	0.931	0.994	0.881

Table 11: Locality of single-edit experiments under editing evaluation (**Edit.**) and real-world evaluation (**Real.**) across various methods, LLMs, and benchmarks.

fine-tuning gradients. By disentangling gradients into learnable rank-one matrices, it achieves explicit control over parameter updates while maintaining tractable editing in LLMs.

ROME (Meng et al., 2022) identifies knowledge-critical layers in Transformer MLP modules through causal tracing analysis. It implements precise knowledge updates via rank-one matrix modification on the identified layer, guided by causal mediation effects in model outputs.

MEMIT (Meng et al., 2023) extends ROME by developing cross-layer propagation analysis and coordinated parameter updates across multiple MLP layers, enabling efficient batch editing of large-scale knowledge.

GRACE (Hartvigsen et al., 2023) is a lifelong editing method that performs local corrections on streaming errors of deployed models. The approach writes new mappings into a pretrained model's latent space, creating a discrete local codebook of edits without modifying model weights, allowing for sequential editing operations.

WISE (Wang et al., 2024b) addresses the similar challenge of sequential editing like GRACE. It employs a dual memory architecture comprising a main memory for pretrained knowledge and a side memory for edited content. The system utilizes a router to direct queries between these memories.

A.4.2 Edited LLMs

Llama-2-7b-chat (Touvron et al., 2023) is a model designed for conversational scenarios with 7 billion parameters. It excels in generating human-like responses in real-time, offering smooth and context-aware dialogue generation.

```
Please answer the question:
Q: Who got the first Nobel Prize in physics?
A:
```

Figure 6: The context-guided prompt for QA tasks.

Mistral-7b (Jiang et al., 2023) is a superior pre-trained base model with 7 billion parameters, outperforming Llama-2-13b on all examined benchmarks, offering strong performance while being resource-efficient. Specifically, we employ the version of Mistral-7B-v0.1.

Llama-3-8b (Meta, 2024) is a cutting-edge 8-billion-parameter model designed for diverse AI applications. It combines advanced techniques with scalability, ensuring high-quality generation for complex tasks like multi-turn dialogues, creative writing, and complex reasoning tasks.

A.4.3 Editing Datasets

ZsRE (Levy et al., 2017) is a popular dataset for Question Answering (QA), where each entry consists of a counterfactual statement derived from a factual Wikipedia page that needs to be edited.

COUNTERFACT (Meng et al., 2022) is a challenging dataset curated for model editing. It contains 21,919 nonfactual statements, initially assigned low probabilities by models, and designed to encourage substantial and meaningful modifications to the original factual statements.

A.5 Locality Results of Single Editing

The locality results of single editing experiments are presented in Table 11. The results show that for almost all baselines, their locality results are very high across two evaluation frameworks, indicating that a single edit generally has little impact on the model's general capabilities.

A.6 Detailed Practical Prompt

In Section 6.1, we prefix the target question with a common QA task instruction (Gao et al., 2024) as the input prompt, as shown in Figure 6. We aim to utilize this context-guided prompt to represent and simulate various contexts that might occur in practical applications.

A.7 Generalization of Sequential Editing

The generalization results of sequential editing experiments are presented in Table 13. Compare to

Method	Llama-2-7b-chat				Mistral-7b				Llama-3-8b			
	Reliability		Locality		Reliability		Locality		Reliability		Locality	
	Edit.	Real.	Edit.	Real.	Edit.	Real.	Edit.	Real.	Edit.	Real.	Edit.	Real.
ZsRE												
FT-M	0.935	0.517	0.583	0.036	0.925	0.465	0.813	0.187	0.879	0.013	0.117	0.001
MEND	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	–	–	–	–
ROME	0.000	0.000	0.002	0.000	0.044	0.004	0.012	0.001	0.087	0.020	0.018	0.000
MEMIT	0.035	0.000	0.014	0.000	0.035	0.000	0.016	0.000	0.052	0.000	0.022	0.000
GRACE	0.317	0.025	1.000	1.000	0.351	0.031	1.000	1.000	0.264	0.033	1.000	1.000
WISE	0.756	0.215	1.000	1.000	0.742	0.017	0.998	0.970	0.514	0.098	1.000	1.000
COUNTERFACT												
FT-M	0.931	0.592	0.225	0.041	0.827	0.538	0.222	0.049	0.782	0.080	0.029	0.003
MEND	0.000	0.000	0.000	0.006	0.000	0.000	0.000	0.000	–	–	–	–
ROME	0.370	0.094	0.093	0.000	0.265	0.131	0.009	0.005	0.484	0.022	0.034	0.000
MEMIT	0.000	0.000	0.056	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
GRACE	0.153	0.017	0.996	1.000	0.148	0.006	0.996	1.000	0.012	0.006	0.996	1.000
WISE	0.797	0.296	0.340	0.522	0.595	0.119	0.196	0.081	0.158	0.027	0.621	0.912

Table 12: Results of sequential editing on ZsRE and COUNTERFACT under editing evaluation (**Edit.**) and real-world evaluation (**Real.**) across various editing methods and LLMs.

Method	ZsRE		COUNTERFACT		QAEdit	
	Edit.	Real.	Edit.	Real.	Edit.	Real.
Llama-2-7b-chat						
FT-M	0.906	0.480	0.723	0.394	0.932	0.461
MEND	0.000	0.000	0.000	0.000	0.000	0.000
ROME	0.000	0.000	0.241	0.066	0.076	0.007
MEMIT	0.035	0.000	0.000	0.000	0.057	0.002
GRACE	0.312	0.027	0.119	0.005	0.371	0.044
WISE	0.705	0.195	0.364	0.102	0.732	0.173
Mistral-7b						
FT-M	0.859	0.404	0.493	0.266	0.856	0.381
MEND	0.000	0.000	0.000	0.000	0.000	0.000
ROME	0.037	0.005	0.244	0.122	0.049	0.000
MEMIT	0.035	0.000	0.000	0.000	0.058	0.002
GRACE	0.340	0.031	0.118	0.004	0.410	0.062
WISE	0.697	0.015	0.326	0.043	0.699	0.065
Llama-3-8b						
FT-M	0.827	0.021	0.532	0.029	0.850	0.271
ROME	0.079	0.017	0.430	0.019	0.020	0.000
MEMIT	0.052	0.000	0.000	0.000	0.000	0.000
GRACE	0.257	0.032	0.008	0.005	0.358	0.078
WISE	0.482	0.089	0.046	0.006	0.503	0.057

Table 13: Generalization results of sequential editing experiments under editing evaluation (**Edit.**) and real-world evaluation (**Real.**) across various editing methods, LLMs, and benchmarks.

Table 8, the results indicate that current editing methods exhibit worse generalization than reliability when dealing with sequential editing requests. All methods except FT-M and WISE demonstrate near-zero generalization ability under real-world evaluation, which further proves that existing editing methods cannot effectively fulfill the practical needs of continuous editing.

A.8 Sequential Editing on Other Datasets

The results of sequential editing on ZsRE and COUNTERFACT are presented in Table 12. These two datasets exhibit trends similar to those observed in QAEdit, including the poor practical effectiveness of existing editing methods, the inadequacy of simplified editing evaluations, and the dilemma of achieving editing success and preserving unrelated knowledge.

Edit Num	BS 1	BS 2	BS 4	BS 8	BS 16
100	0.000	0.000	0.000	0.000	0.000
200	0.000	0.000	0.000	0.000	0.000
400	0.000	0.000	0.000	0.000	0.000
800	0.000	0.000	0.000	0.000	0.000
1000	0.000	0.000	0.000	0.000	0.000

Table 14: The reliability for sequentially editing Llama-3-8b using MEND, illustrating the impact of different batch sizes (BS) across varying numbers of edits.

A.9 Mini-Batch Sequential Editing for MEND

As shown in Table 14, unlike FT-M and MEMIT, which can maintain a certain level of editing performance under specific batch sizes (as depicted in Figure 4), MEND is completely unusable in sequential editing scenarios, regardless of the batch size. This poor effectiveness can be attributed to the limitation of the meta-learning paradigm, wherein the hypernetwork of MEND for parameter updates is specifically trained on the original model state. Consequently, the predicted parameter modifications are optimized solely for the original model and fail to effectively adapt to the evolving states of the sequentially edited model. This limitation fundamentally constrains MEND's efficacy in sequential editing scenarios.

Prompt for Subject Extraction

```
Please identify the subject in the provided prompt and respond solely with the subject, ensuring
↪  the subject is directly drawn from the prompt itself (including the need for exact match in
↪  case, both uppercase and lowercase).

Here are some examples for guidance:
```
{'prompt': 'Who published Journal of Clinical Microbiology?', 'subject': 'Journal of Clinical
↪ Microbiology'}
{'prompt': 'Who was mainly responsible for the design of Abney Park Chapel?', 'subject': 'Abney
↪ Park Chapel'}
{'prompt': 'Who was behind the creation of IAC Building?', 'subject': 'IAC Building'}
{'prompt': "Who is Li Jiancheng's sister?", 'subject': 'Li Jiancheng'}
{'prompt': "Who is the Haitz's law named after?", 'subject': "Haitz's law"}
```

Based on the examples, for 'prompt': 'Who got the first Nobel Prize in physics?', the 'subject'
↪  is:
```

Figure 7: Complete prompt used for directly extracting subject from edit prompt for QAEdit.

Prompt for Question Paraphrasing

```
Role and Goal: Serves as a data engineer, use your knowledge to rewrite the following question in
↪  a different way, ensuring it conveys the same meaning and maintains a neutral tone but with
↪  different wording. Avoid using phrases such as 'Could you tell me'. Instead, directly
↪  rephrase it into a structured question.

Please rephrase the following question: Who got the first Nobel Prize in physics?
```

Figure 8: Complete prompt for paraphrasing edit question into rephrased question for generalization evaluation.

Prompt for LLM-as-a-Judge

```
Your job is to look at a question, a gold target, and a predicted answer, and then assign a grade
↪  of either ["CORRECT", "INCORRECT"].

The following are examples of CORRECT predicted answers.
```
Question: What are the names of Barack Obama's children?
Gold target: Malia Obama and Sasha Obama
Predicted answer 1: sasha and malia obama
Predicted answer 2: Malia and Sasha Obama are the names of Barack Obama's children.
```
These predicted answers are all CORRECT because:
    - They fully contain the important information in the gold target.
    - They do not contain any information that contradicts the gold target.

The following are examples of INCORRECT predicted answers.
```
Question: What are the names of Barack Obama's children?
Gold target: Malia and Sasha
Predicted answer 1: Malia.
Predicted answer 2: Malia, Sasha, and Susan.
Predicted answer 3: Malia and Sasha, Malia and Sasha, Malia and Sasha, Malia and Sasha (repeated
↪ answer)
```
These predicted answers are all INCORRECT because:
    - A factual statement in the answer contradicts the gold target or contain repeated answer.


Here is a sample. Simply reply with either CORRECT or INCORRECT.

```
Question: {question}
Gold target: {target}
Predicted answer: {predicted_answer}
```

According to the gold target, please grade the predicted answer of this question as one of:
A: CORRECT
B: INCORRECT

Just return the letters "A" or "B", with no text around it.
```

Figure 9: The complete prompt used to employ a LLM as a judge for providing binary assessments (correct or incorrect) based on a given question, gold target answer, and predicted answer.