FABLE: Fine-grained Fact Anchoring for Unstructured Model Editing

Peng Wang[1,2], Biyu Zhou[1]*, Xuehai Tang[1], Jizhong Han[1], Songlin Hu[1,2]*

[1]Institute of Information Engineering, Chinese Academy of Sciences
[2]School of Cyber Security, University of Chinese Academy of Sciences
Correspondence: {wangpeng2022, zhoubiyu, tangxuehai, hanjizhong, husonglin}@iie.ac.cn

Abstract

Unstructured model editing aims to update models with real-world text, yet existing methods often memorize text holistically without reliable fine-grained fact access. To address this, we propose **FABLE**, a hierarchical framework that decouples fine-grained fact injection from holistic text generation. FABLE follows a two-stage, fact-first strategy: discrete facts are anchored in shallow layers, followed by minimal updates to deeper layers to produce coherent text. This decoupling resolves the mismatch between holistic recall and fine-grained fact access, reflecting the unidirectional Transformer flow in which surface-form generation amplifies rather than corrects underlying fact representations. We also introduce **UnFine**, a diagnostic benchmark with fine-grained question–answer pairs and fact-level metrics for systematic evaluation. Experiments show that FABLE substantially improves fine-grained question answering while maintaining state-of-the-art holistic editing performance. Our code is publicly available at https://github.com/caskcsg/FABLE.

1 Introduction

Large language models (LLMs) have transformed natural language processing(Brown et al., 2020; Huang et al., 2022; Liu et al., 2024), yet static pre-training limits their capacity to absorb dynamically evolving factual knowledge(Cao et al., 2021; Mitchell et al., 2022a). Fully retraining models to incorporate new information is computationally prohibitive and impractical in real-world settings(Gupta et al., 2023; Yao et al., 2023). This has motivated growing interest in model editing, which aims to update specific knowledge by modifying a small subset of parameters while preserving overall model behavior(Wang et al., 2025b). Among existing approaches, locate-then-edit methods such as

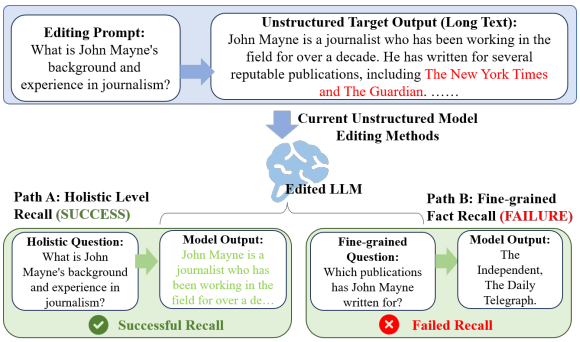

Figure 1: Limitation of existing unstructured editing methods (holistic recall vs. fine-grained fact access)

ROME(Meng et al., 2022) and MEMIT(Meng et al., 2023) have shown notable success by identifying and modifying the causally relevant parameters to inject structured triple knowledge in the form of <subject, relation, object>.

Nevertheless, in real-world scenarios, approximately 80% of knowledge is expressed as unstructured form (Bavota, 2016), which introduces two key challenges for model editing: entity localization in open text and limited editing capacity for long content. Recent work on unstructured model editing, such as UnKE(Deng et al., 2025) and AnyEdit(Jiang et al., 2025), tackles these challenges by expanding the scope of editable parameters and anchoring optimization to context-preserving key tokens, thereby enabling holistic memorization and recall of unstructured text.

Despite these advances, we identify a fundamental limitation: existing methods enable **holistic recall** of edited text, but fail to support **fine-grained fact access**, preventing reliable retrieval of atomic facts(Min et al., 2023). As shown in Figure 1, a model edited by UnKE can restate the entire text for the question "What is John Mayne's background and experience in journalism?", suggesting successful memorization at the textual level. However, when the query shifts to specific details within the

*Corresponding authors.

Findings of the Association for Computational Linguistics: ACL 2026, pages 14549–14567

text, the model fails to produce accurate answers. We attribute this limitation to the model primarily learning a high-level mapping from the question to surface-form representations for holistic recall (observable in layers associated with surface form generation), without consistently encoding the underlying atomic facts into lower-level knowledge storage. In real-world applications, downstream tasks often require fine-grained, targeted knowledge retrieval (e.g., querying specific facts in a report or particular imagery in a poem), rather than mere reproduction of edited text.

Therefore, we propose **FABLE**, a novel framework for unstructured model editing. Unlike existing methods that primarily optimize surface-form recall at the holistic text level, FABLE adopts a two-stage hierarchical strategy that decouples fine-grained facts from holistic textual surface forms and anchors them into distinct parameter layers. Motivated by the "early decoding" phenomenon in Transformers, FABLE adopts a fact-first, generation-later design, embedding fine-grained facts into shallow layers before surface-form realization. Editing proceeds in two stages: discrete facts are first injected into shallow fine-grained key generators via synthesized question–answer pairs, from our **UnFine** benchmark, then minimal, localized adjustments are applied to deeper surface-form key generators to ensure global coherence and fluent generation. This hierarchically decoupled design enables the model to achieve both reliable holistic surface-form recall and accurate fine-grained factual access to the edited content.

In summary, the main contributions are:

(1) We identify a mismatch in unstructured editing between holistic surface-form memorization and fine-grained fact recall.

(2) We introduce UnFine, a diagnostic benchmark for fine-grained fact recall, together with tailored metrics (e.g., HR, C_{LCS}).

(3) We propose FABLE, a hierarchical editing framework that decouples fine-grained fact anchoring from surface-form generation.

2 Benchmark Construction

Existing benchmarks evaluate if a model outputs A given Q, but not whether it captures the fine-grained fact within A. To close this gap, we introduce UnFine, which extends existing benchmarks by incorporating fine-grained factual question answering as a core evaluation dimension.

2.1 Datasets

UnFine is built upon three widely-used datasets in the field of unstructured model editing: UnKEBench(Deng et al., 2025), AKEW (CounterFact)(Wu et al., 2024), and AKEW (MQuAKE)(Wu et al., 2024). Among them, UnKEBench is the first benchmark specifically designed for evaluating unstructured model editing capabilities in LLMs, comprising complex unstructured long-text question-answer pairs. AKEW is the first large-scale unstructured model editing benchmark constructed from real-world scenarios; its subsets based on CounterFact(Meng et al., 2022) and MQuAKE(Zhong et al., 2023) are referred to as AKEW (CounterFact) and AKEW (MQuAKE), respectively.

We have made improvements in the following two aspects: **(1) Generation of Fine-Grained QA Pairs**: Noting that while UnKEBench includes natural language-based fine-grained QA pairs, AKEW (CounterFact) and AKEW (MQuAKE) lack such data. Following the methodology of UnKEBench, we augment the latter two with corresponding fine-grained QA pairs. **(2) Key Knowledge Phrase Extraction**: However, due to the nature of natural language, answers in these QA pairs contain not only factual knowledge but also certain linguistic styles. To more precisely evaluate whether the model has acquired the key knowledge (mitigating interference from language style), we further extract key knowledge phrases from each fine-grained answer. The improved datasets are renamed **UnFine-UnKE**, **UnFine-CF**, and **UnFine-MQ**, respectively. Detailed dataset construction procedures are provided in Appendix A.1.

2.2 Evaluation Metrics

We use a two-level evaluation framework to assess both holistic and fine-grained consistency between model outputs and target edits.

(1) Holistic Knowledge QA Evaluation. Following prior works(Deng et al., 2025; Jiang et al., 2025), we adopt lexical similarity (ROUGE-L(Lin, 2004)) and semantic similarity (BERT-Score(Zhang et al., 2020)) to evaluate the editing success rate.

(2) Fine-Grained Knowledge QA Evaluation. Beyond sentence-level similarity, we introduce the following two fact-level metrics to assess the model's ability to recall fine-grained fact.

Hit Rate (HR): It evaluates precise fact recall by checking, via exact string matching, whether the model output contains all key knowledge phrases extracted from the gold-standard answer.

Longest Common Subsequence Coverage (C_{LCS}): This metric quantifies the completeness of content coverage. It is calculated as the ratio of the length of the Longest Common Subsequence (LCS) between the model output and the fine-grained gold answer to the total word count of the fine-grained gold answer. It effectively measures the output's ability to capture key information points. For example, for a answer "**North** Island **of** New **Zealand**" and a model output "**North of Zealand**", the LCS is "North", "of", "Zealand" (length 3), yielding a coverage of $C_{\text{LCS}} = \frac{3}{5} = 0.6$. Formal definitions and detailed formulations for all evaluation metrics are provided in Appendix A.2.

3 FABLE

3.1 Hierarchical Key-Value Storage Architecture

Existing research often views Transformer-based LLMs as continuous key-value memory networks (Deng et al., 2025). In this view, shallow layers are seen as a key generator, responsible for compressing input information to produce knowledge representations (keys); while the middle to deep layers act as a value generator, responsible for decoding the keys and injecting them into the residual stream to form value vectors. This block-level key-value partitioning retains the attention mechanism and non-linear transformations, making it more suitable for representing unstructured knowledge compared to traditional MLP-level key-value pairs(Meng et al., 2022, 2023).

However, the prevailing view treats the key generator as a monolithic operation, which can lead to overfitting surface forms while failing to reliably encode fine-grained facts. For unstructured text, generating an effective "key" should instead be decomposed into two levels: (1) **fine-grained fact anchoring**, which extracts and stabilizes discrete factual units, and (2) **holistic surface-form construction**, which organizes these units into a coherent, fluent narrative. This decomposition leverages the natural stratification of representations in Transformers, where lower layers excel at capturing local, fine-grained features and higher layers integrate them into global semantic representations.

To formalize this, we first represent an N-layer model as $f_\theta = f^1_{\theta_1} \circ f^2_{\theta_2} \circ \cdots \circ f^N_{\theta_N}$, where θ denotes the model parameters, $\circ$ denotes layer composition, and $f^l_{\theta_l}$ represents the l-th layer and its parameters θ_l. We formalize f_θ as a combination of two core modules: an Unstructured Knowledge Key Generator(G_K) and a Value Generator(G_V). The key generator can be further decomposed into two consecutive stages:

$$f_\theta = \underbrace{(\mathcal{F}_{\text{fine}} \circ \mathcal{F}_{\text{hol}})}_{G_K} \circ \underbrace{\mathcal{V}.}_{G_V} \tag{1}$$

Here, the fine-grained key generator $\mathcal{F}_{\text{fine}}$ encodes layers 1 to L_f. The holistic key generator $\mathcal{F}_{\text{hol}}$ encodes layers L_{f+1} to L_h. Finally, the value generator $\mathcal{V}$ maps layers L_{h+1} to N into the token space. Here, L_f and L_h denote the split layer between fine-grained and holistic key generators, and the boundary separating key and value generators, respectively.

Given a question $q = [q_1, \ldots, q_n]$ (where n is the token count of q), its representation at layer l is $h^l_q = [h^l_{q,1}, \ldots, h^l_{q,n}]$, and its forward propagation process is:

$$h^l_q = f^l_{\theta_l}(h^{l-1}_q), \tag{2}$$

where $h^1_q = f^1_{\theta_1}(h^0_q) = f^1_{\theta_1}(q)$. For convenience in subsequent discussion, we denote the representation corresponding to the i-th token at layer l as: $h^l_{q,i} = h^l_q[i] = f^l_{\theta_l}(h^{l-1}_q)[i]$. Since the vector corresponding to the last token typically highly aggregates information from the entire q, we further define the fine-grained fact key k_{fine}, the holistic semantic key k_{hol}, and the value v as:

$$k_{\text{fine}} = \mathcal{F}_{\text{fine}}(q)[n]. \tag{3}$$

$$k_{\text{hol}} = \mathcal{F}_{\text{hol}}([h^{L_f}_{q,1}, \ldots, h^{L_f}_{q,n-1}, k_{\text{fine}}])[n]. \tag{4}$$

$$v = \mathcal{V}([h^{L_h}_{q,1}, \ldots, h^{L_h}_{q,n-1}, k_{\text{hol}}])[n]. \tag{5}$$

Finally, the model decodes based on v to obtain the output answer $a = [a_1, \ldots, a_m]$ (where m is the token count of a).

3.2 Stage One: Fine-grained Fact Anchoring

This stage aims to inject fine-grained facts into the model parameters, i.e., to update the parameters of $\mathcal{F}_{\text{fine}}$ so that the model can accurately output the target answer a^*_f when presented with a fine-grained question q_f. The specific implementation is divided into two steps: first, compute the target fine-grained fact key k^*_{fine} that can guide the model to generate a^*_f; then, adjust the parameters of $\mathcal{F}_{\text{fine}}$ so that it stably produces k^*_{fine} when input with q_f.

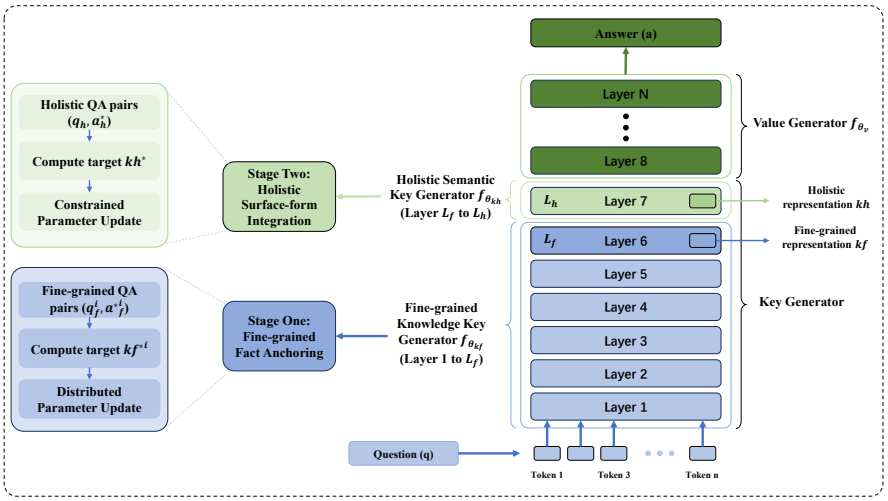

Figure 2: FABLE decomposes the key generator in a Transformer-based LLM into a two-stage hierarchical process: (1) fine-grained fact anchoring, which first encodes and stabilizes discrete factual knowledge; and (2) holistic surface-form integration, which then organizes these facts into a coherent narrative. The architecture consists of a fine-grained key generator, a holistic key generator, and a value generator.

(1) Computing the Target Fine-Grained Fact Key We search for the optimal direction in the residual stream at layer L_f that can trigger the target fact. By optimizing the residual vector δ_f to minimize the negative log probability of the target answer a_f^*, we obtain δ_f:

$$\delta_f = \arg\min_{\delta_f} -\log P_{f_\theta(k_{\text{fine}} \mapsto k_{\text{fine}}+\delta_f)}(a_f^* \mid q_f) \tag{6}$$

$$k_{\text{fine}}^* = k_{\text{fine}} + \delta_f, \tag{7}$$

where $f_\theta(k_{\text{fine}} \mapsto k_{\text{fine}} + \delta_f)$ denotes replacing k_{fine} with $k_{\text{fine}} + \delta_f$ during forward propagation.

(2) Distributed Parameter Update Since $\mathcal{F}_{\text{fine}}$ involves multi-layer continuous transformations, to ensure consistency of the bottom-up residual flow, we distribute the parameter update across its layers ($l \in [1, L_f]$). We define a per-layer optimization objective for each layer, making it share a portion of the total offset δ_f. Specifically, for layer l, the objective is to make its output at the last token position $h^{*l}_{q_f,n} = f^l_{\theta_l^*}(h^{l-1}_{q_f})[n]$ approximate $h^l_{q_f,n} + \frac{\delta_f}{L_f - l + 1}$, where θ_l^* represents the updated parameters for layer l. In particular, when $l = L_f$, we have: $h^{L_f}_{q_f,n} + \frac{\delta_f}{L_f - L_f + 1} = k_{\text{fine}} + \delta_f = k_{\text{fine}}^*$.

To preserve the model's behavior on the input prefix, the representations of the first $n-1$ tokens are kept unchanged. The layer-l optimization objective is:

$$\theta_l^* = \arg\min_{\theta_l^*} \Big(\Big\| h^{*l}_{q_f,n} - h^l_{q_f,n} - \frac{\delta_f}{L_f - l + 1} \Big\|^2 + \sum_{j=1}^{n-1} \Big\| h^{*l}_{q_f,j} - h^l_{q_f,j} \Big\|^2 \Big). \tag{8}$$

Since unstructured text often contain more than a piece of fine-grained fact, we consider u different fine-grained questions $\{q_f^i\}_{i=1}^u$, each q_f^i contains n tokens (for the method of extracting multi-aspect fine-grained QA pairs (q_f^i, a^{*i}_f) from unstructured text, see Appendix B). Additionally, to preserve the model's predictions on irrelevant samples as much as possible while editing the target knowledge, we introduce v irrelevant samples $D = \{d^i\}_{i=1}^v$, each d^i contains w tokens. Therefore, the complete optimization objective for layer l is:

$$\theta_l^* = \arg\min_{\theta_l^*} \Big(\underbrace{\sum_{i=1}^{u} \Big\| h^{*l}_{q_f^i,n} - h^l_{q_f^i,n} - \frac{\delta_f^i}{L_f - l + 1} \Big\|^2}_{\text{Edit Efficacy}} + \underbrace{\sum_{i=1}^{u}\sum_{j=1}^{n-1} \Big\| h^{*l}_{q_f^i,j} - h^l_{q_f^i,j} \Big\|^2}_{\text{Prefix Consistency}} + \underbrace{\sum_{k=1}^{v}\sum_{j=1}^{w} \Big\| h^{*l}_{d^k,j} - h^l_{d^k,j} \Big\|^2}_{\text{Locality Preservation}} \Big). \tag{9}$$

By optimizing Eq. 9 layer-by-layer, we obtain the updated $\mathcal{F}^*_{\text{fine}}$ with $\mathcal{F}^*_{\text{fine}} = f^1_{\theta_1^*} \circ f^2_{\theta_2^*} \circ \cdots \circ f^{L_f}_{\theta^*_{L_f}}$.

3.3 Stage Two: Holistic Surface-form Integration

After embedding fine-grained fact into $\mathcal{F}^*_{\text{fine}}$, we further adjust the parameters of $\mathcal{F}_{\text{hol}}$ to perform semantic integration for a holistic question q_h, in order to generate fluent unstructured narrative a^*_h. According to prior work(Deng et al., 2025), updating only the single layer L_h parameters can effectively enable the model to narrate unstructured knowledge, so we maintain this setting. Similarly, the implementation of this stage is also divided into two steps. First, analogous to Eq. 6, we can obtain the residual vector δ_h and the target semantic key k^*_{hol} via optimization. Then, analogous to Eq. 9, we can formulate the optimization objective for $\mathcal{F}^*_{\text{hol}}$, but with three key differences: (1) Only update the single layer L_h; (2) The QA pair is a single (q_h, a^*_h) rather than a batch; (3) To prevent blindly updating the semantic layer and overwriting the fine-grained fact signals passed from the first stage, a fine-grained preservation constraint is introduced. Therefore, the optimization objective is:

$$\theta^*_{L_h} = \arg\min_{\theta^*_{L_h}} \Bigg(\underbrace{\left\| h^{*L_h}_{q_h,n} - h^{L_h}_{q_h,n} - k^*_{\text{hol}} \right\|^2}_{\text{Edit Efficacy}} + \underbrace{\sum_{j=1}^{n-1} \left\| h^{*L_h}_{q_h,j} - h^{L_h}_{q_h,j} \right\|^2}_{\text{Prefix Consistency}} + \underbrace{\sum_{k=1}^{v} \sum_{j=1}^{w} \left\| h^{*L_h}_{d^k,j} - h^{L_h}_{d^k,j} \right\|^2}_{\text{Locality Preservation}} + \underbrace{\sum_{i=1}^{u} \sum_{j=1}^{n} \left\| h^{*L_h}_{q^i_f,j} - h^{L_h}_{q^i_f,j} \right\|^2}_{\text{Fine-grained Preservation}} \Bigg). \tag{10}$$

3.4 Implementation Details

In the concrete implementation, to balance editing effectiveness and model stability, we designed the parameter update scope and data scale specifically. First, for updating the fine-grained fact injection stage $\mathcal{F}_{\text{fine}}$, we do not update all layers 1 to L_f. Because the initial layers of the model primarily handle basic functions like syntactic understanding. Therefore, we choose to update only the middle layers 4, 5, and 6. Second, for updating the holistic semantic integration stage $\mathcal{F}_{\text{hol}}$, following prior work, we set L_h as 7 and update only its parameters. Regarding data, the number of QA pairs used for fine-grained fact injection is set to 5 times the number of seed QA pairs S extracted from the unstructured text in the editing sample, i.e., $5 \times S$ (details on seed QA pairs are in Appendix B). Meanwhile, to preserve the model's general capabilities as much as possible during editing, following related research settings, for each editing sample, we randomly sample 20 samples from the Alpaca instruction-tuning dataset[1] to serve as the irrelevant sample set D. For ablation studies on the choice of layers and number of QA pairs, see Section 4.5.

4 Experiments

4.1 Experimental Setup

Base LLMs. This study selects two categories of representative LLMs in this field: **Llama3-8B-Instruct**[2] and **Qwen2.5-7B-Instruct**(Yang et al., 2024) as the base models.

Baseline Methods. For a comprehensive comparison, we select the following three representative categories of knowledge editing methods as baselines: (1) the fine-tuning-based method **FT-L**; (2) the classic "locate-then-edit" methods for structured editing scenarios, including **ROME**(Meng et al., 2022) and **MEMIT**(Meng et al., 2023); (3) the "locate-then-edit" methods optimized for unstructured editing scenarios, including **AnyEdit**(Jiang et al., 2025) and **UnKE**(Deng et al., 2025). Details for each method are provided in Appendix C.1.

Benchmark. We evaluate on **UnFine**, our unstructured knowledge editing benchmark built on prior work for comprehensive assessment. Details are in Section 2.

Implementation Details. Following the experimental setup of works(Deng et al., 2025; Jiang et al., 2025), this experiment sets the editing batch size to 1. More detailed experimental configurations and tools used can be found in Appendix C.2.

4.2 Editing Performance Results

Table 1 reports the unstructured knowledge editing performance of different editing methods across various models and datasets. The experimental results show that FABLE significantly outperforms all baseline methods in both holistic generation quality and fine-grained knowledge recall.

[1] https://github.com/tatsu-lab/stanford_alpaca
[2] https://llama.meta.com/llama3

Table 1: Unstructured model editing performance with different models, methods, and datasets. "Pre-edited" denotes the original model before editing; "–" indicates that the corresponding field is absent from the dataset. Best results are **boldfaced**; second-best are underlined.

Model	Method	Holistic		Fine-grained			
		Bert-Score↑	Rouge-L↑	Bert-Score↑	Rouge-L↑	HR↑	C_{LCS}↑
UnFine-UnKE							
Llama3-8B	**Pre-edited**	71.83±0.11	22.69±0.10	22.26±0.08	12.81±0.09	5.00±0.10	17.50±0.12
	FT-L	33.29±0.22	13.35±0.12	26.10±0.10	15.49±0.11	7.60±0.12	21.83±0.14
	ROME	79.51±0.13	40.00±0.23	24.76±0.09	17.79±0.12	9.53±0.14	24.70±0.15
	MEMIT	81.78±0.20	49.64±0.29	25.60±0.09	20.54±0.13	12.51±0.16	28.30±0.17
	UnKE	98.20±0.06	93.39±0.16	27.93±0.09	27.15±0.16	20.55±0.20	37.13±0.19
	AnyEdit	96.06±0.07	90.34±0.15	28.61±0.09	29.22±0.15	23.69±0.20	40.51±0.18
	FABLE	**99.36±0.04**	**97.78±0.10**	**65.63±0.16**	**53.79±0.19**	**53.31±0.23**	**61.78±0.19**
Qwen2.5-7B	**Pre-edited**	72.96±0.10	23.74±0.10	23.75±0.09	13.21±0.10	3.93±0.09	17.74±0.11
	FT-L	28.11±0.24	14.19±0.10	23.34±0.10	14.23±0.10	4.44±0.10	21.39±0.12
	ROME	78.53±0.13	39.19±0.18	25.59±0.09	18.21±0.12	8.52±0.13	24.26±0.14
	MEMIT	81.44±0.16	49.92±0.24	25.33±0.09	19.65±0.13	11.13±0.15	26.53±0.16
	UnKE	96.86±0.07	90.24±0.17	26.89±0.09	22.47±0.13	12.86±0.16	30.10±0.16
	AnyEdit	97.02±0.07	91.95±0.17	27.75±0.09	25.47±0.15	17.63±0.17	34.74±0.17
	FABLE	**98.86±0.04**	**97.35±0.09**	**50.87±0.16**	**38.31±0.18**	**31.35±0.21**	**44.58±0.18**
UnFine-CF							
Llama3-8B	**Pre-edited**	71.42±0.12	15.53±0.09	26.37±0.09	26.56±0.17	20.22±0.19	32.43±0.19
	FT-L	29.71±0.20	10.19±0.09	30.12±0.10	25.80±0.17	20.27±0.19	31.86±0.19
	ROME	80.38±0.15	40.33±0.23	27.89±0.08	30.40±0.17	24.89±0.21	38.01±0.19
	MEMIT	83.74±0.18	54.10±0.30	28.64±0.08	32.88±0.18	29.25±0.21	41.48±0.20
	UnKE	**99.28±0.04**	97.02±0.12	29.22±0.08	35.89±0.19	32.64±0.22	45.62±0.20
	AnyEdit	98.19±0.05	94.97±0.11	30.41±0.08	39.05±0.19	38.94±0.22	51.40±0.19
	FABLE	**99.28±0.05**	**98.35±0.09**	**71.89±0.13**	**64.59±0.19**	**66.62±0.22**	**73.27±0.18**
Qwen2.5-7B	**Pre-edited**	70.47±0.12	18.77±0.09	26.18±0.09	24.12±0.17	16.83±0.18	28.56±0.18
	FT-L	25.35±0.25	14.21±0.12	28.01±0.10	23.57±0.16	15.45±0.17	29.41±0.18
	ROME	77.56±0.17	40.01±0.22	27.35±0.09	28.38±0.17	22.31±0.19	34.35±0.19
	MEMIT	81.08±0.16	51.43±0.26	27.43±0.08	30.13±0.18	24.36±0.20	36.96±0.20
	UnKE	97.25±0.07	89.76±0.18	26.97±0.08	28.26±0.17	21.18±0.19	33.86±0.18
	AnyEdit	97.13±0.07	90.78±0.18	29.97±0.08	34.59±0.19	32.27±0.22	45.38±0.20
	FABLE	**99.23±0.04**	**97.61±0.09**	**56.23±0.15**	**45.62±0.20**	**41.89±0.22**	**52.22±0.20**
UnFine-MQ							
Llama3-8B	**Pre-edited**	69.67±0.13	20.60±0.09	27.74±0.09	30.35±0.17	25.88±0.21	37.71±0.19
	FT-L	22.65±0.18	7.40±0.09	28.04±0.10	24.71±0.18	18.16±0.19	29.65±0.20
	ROME	74.04±0.17	36.10±0.20	28.01±0.08	31.37±0.17	27.80±0.20	40.47±0.19
	MEMIT	78.86±0.19	50.91±0.29	29.22±0.08	34.61±0.18	31.94±0.23	44.09±0.21
	UnKE	98.23±0.06	95.18±0.15	30.27±0.09	39.08±0.20	37.36±0.24	49.46±0.22
	AnyEdit	97.53±0.07	93.77±0.12	29.96±0.08	38.42±0.18	36.62±0.22	49.00±0.20
	FABLE	**98.71±0.07**	**96.59±0.13**	**66.99±0.14**	**62.55±0.21**	**65.86±0.23**	**71.31±0.20**
Qwen2.5-7B	**Pre-edited**	69.08±0.12	20.98±0.08	28.28±0.09	28.81±0.16	22.94±0.19	34.91±0.18
	FT-L	23.03±0.22	12.51±0.10	27.27±0.09	21.15±0.14	14.29±0.16	27.95±0.16
	ROME	75.39±0.16	40.29±0.19	28.66±0.08	31.73±0.17	26.47±0.21	39.39±0.19
	MEMIT	79.44±0.16	51.71±0.25	28.54±0.08	33.15±0.18	26.75±0.20	40.83±0.19
	UnKE	94.88±0.11	85.35±0.21	29.31±0.09	34.57±0.18	29.81±0.21	42.89±0.19
	AnyEdit	96.45±0.07	90.30±0.17	29.48±0.09	35.66±0.18	31.59±0.22	44.43±0.21
	FABLE	**98.84±0.05**	**96.98±0.10**	**53.16±0.15**	**44.01±0.20**	**42.13±0.25**	**51.10±0.22**

(1) Holistic unstructured text generation performance is further improved. In terms of the semantic similarity metric Bert-Score and the lexical overlap metric Rouge-L, FABLE achieves comprehensive superiority. Compared to the second-best baseline AnyEdit, FABLE improves Bert-Score and Rouge-L by an average of 1.98% and 5.42%, respectively. Notably, after editing the Llama3-8B model on the UnFine-UnKE dataset, the improvement in Rouge-L reaches 7.44%.

(2) Fine-grained knowledge recall performance is significantly enhanced. On sentence-level evaluation metrics (Bert-Score and Rouge-L), FABLE outperforms the second-best baseline AnyEdit by an average of 31.43% and 17.74%, respectively. On the fact-level metrics we further propose (HR and C_{LCS}), the corresponding average improvements are 20.07% and 14.80%.

These results show that FABLE enhances both holistic text storage and fine-grained knowledge recall. Case studies are in Appendix D.5.

4.3 Analysis of the Performance Enhancement

Based on the findings in Section 4.2—that FABLE, compared to baseline methods like UnKE and AnyEdit, not only improves fine-grained knowledge recall but also further enhances the overall performance of generating unstructured text—we

Table 2: The ablation experiment results of Llama3-8B-Instruct on the UnFine-UnKE dataset, where "Layers" indicates the updated layers of $\mathcal{F}_{\text{fine}}$, "Number" denotes the quantity of fine-grained question-answer pairs injected in the first stage, "Method" refers to the ablation on the method itself, and "Augmentation" represents the ablation on data augmentation.

Ablation Dimensions	Configuration	Holistic		Fine-grained			
		Bert-Score↑	Rouge-L↑	Bert-Score↑	Rouge-L↑	HR↑	C_{LCS}↑
Base	**L=4,5,6; N=5 × S**	**99.36**$_{\pm 0.04}$	**97.78**$_{\pm 0.10}$	**65.63**$_{\pm 0.16}$	**53.79**$_{\pm 0.19}$	**53.31**$_{\pm 0.23}$	**61.78**$_{\pm 0.19}$
Ablation on Layers							
Layers	**L=3**	99.51$_{\pm 0.04}$	98.60$_{\pm 0.09}$	63.18$_{\pm 0.16}$	50.36$_{\pm 0.19}$	48.33$_{\pm 0.23}$	57.83$_{\pm 0.19}$
	L=4	99.70$_{\pm 0.02}$	98.79$_{\pm 0.08}$	65.52$_{\pm 0.15}$	52.93$_{\pm 0.19}$	52.07$_{\pm 0.23}$	61.09$_{\pm 0.19}$
	L=5	99.68$_{\pm 0.02}$	98.78$_{\pm 0.07}$	64.93$_{\pm 0.15}$	52.97$_{\pm 0.19}$	51.88$_{\pm 0.24}$	60.96$_{\pm 0.19}$
	L=6	99.00$_{\pm 0.05}$	96.66$_{\pm 0.12}$	65.72$_{\pm 0.15}$	53.96$_{\pm 0.19}$	53.67$_{\pm 0.23}$	61.90$_{\pm 0.19}$
	L=4,5	99.67$_{\pm 0.03}$	98.87$_{\pm 0.07}$	65.92$_{\pm 0.15}$	53.94$_{\pm 0.19}$	53.17$_{\pm 0.23}$	62.04$_{\pm 0.18}$
	L=5,6	99.24$_{\pm 0.03}$	96.80$_{\pm 0.12}$	64.38$_{\pm 0.15}$	52.72$_{\pm 0.19}$	51.86$_{\pm 0.23}$	60.41$_{\pm 0.19}$
Ablation on Number							
Number	**N=1 × S**	99.65$_{\pm 0.02}$	98.75$_{\pm 0.07}$	57.97$_{\pm 0.16}$	49.16$_{\pm 0.19}$	46.90$_{\pm 0.23}$	57.19$_{\pm 0.19}$
	N=10 × S	97.11$_{\pm 0.09}$	92.23$_{\pm 0.19}$	66.42$_{\pm 0.15}$	53.69$_{\pm 0.20}$	53.79$_{\pm 0.23}$	61.75$_{\pm 0.19}$
Ablation on Method							
Method	**w/o Stage1**	98.63$_{\pm 0.05}$	94.92$_{\pm 0.15}$	23.29$_{\pm 0.08}$	14.98$_{\pm 0.11}$	7.56$_{\pm 0.13}$	20.42$_{\pm 0.14}$
	w/o Stage2	45.98$_{\pm 0.20}$	9.13$_{\pm 0.08}$	72.00$_{\pm 0.14}$	55.36$_{\pm 0.19}$	55.04$_{\pm 0.24}$	62.47$_{\pm 0.19}$
Ablation on Augmentation							
Augmentation	**UnKE+Augmentation**	97.17$_{\pm 0.10}$	92.12$_{\pm 0.21}$	68.54$_{\pm 0.15}$	54.26$_{\pm 0.20}$	54.11$_{\pm 0.23}$	61.77$_{\pm 0.19}$
	AnyEdit+Augmentation	61.57$_{\pm 0.23}$	17.71$_{\pm 0.18}$	61.50$_{\pm 0.15}$	44.63$_{\pm 0.19}$	41.31$_{\pm 0.22}$	51.00$_{\pm 0.19}$

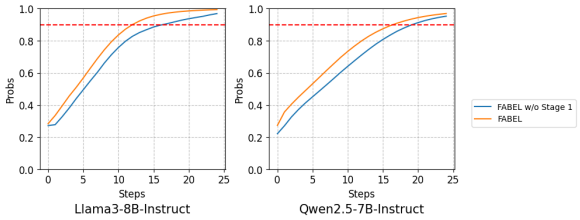

Figure 3: Performance comparison during optimization between FABEL (with Stage 1) and its ablated variant (without Stage 1) on the UnFine-UnKE. The horizontal axis shows the number of optimization steps, and the vertical axis shows the average output probability of the target unstructured text.

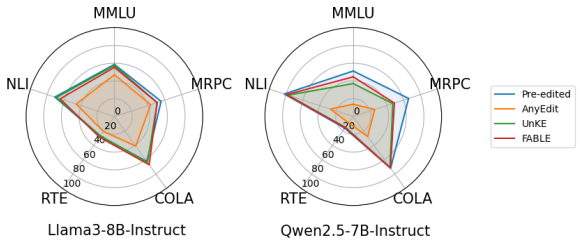

Figure 4: F1-scores of each task for unstructured editing methods on different models after editing on the UnFine-UnKE dataset

delve into the underlying reasons in this section. In stage two (semantic integration), FABLE optimizes the residual vector δ_h through gradient descent to search for the optimal target semantic key k_h^*, thereby maximizing the probability of the model generating the target unstructured text. To reveal this process, we take the model edited on the UnFine-UnKE dataset as an example and plot the curve of the model's average output probability as a function of optimization steps (Figure 3). The figure compares the optimization trajectories of the full two-stage method (FABEL) and the method using only stage two (FABEL w/o Stage1), from which two key observations can be made:

(1) The introduction of stage one significantly improves the initial probability. As shown in the figure, FABLE (orange) starts higher than FABLE w/o Stage One (blue) on both Llama3-8B-Instruct and Qwen2.5-7B-Instruct. Therefore, higher initial probability facilitates editing and improves success (Jiang et al., 2025), explaining the enhanced final text generation of FABLE.

(2) Stage one effectively reduces the difficulty of editing and accelerates convergence. The red dashed line marks a 0.9 probability threshold. FABLE with Stage One reaches it faster—e.g., in approximately 10 steps on Llama3-8B-Instruct versus in approximately 15 without Stage One—showing that prior atomic fact injection simplifies subsequent semantic integration.

These results show that FABLE's hierarchical key-value architecture and two-stage editing improve both unstructured knowledge modeling and editing efficiency. Additional datasets are in Appendix D.1.

4.4 General Capability Evaluation

Unstructured editing methods have demonstrated promising editing performance in Section 4.2. However, their impact on the general capabilities of the models remains unclear. To address this, we selected several unstructured editing methods with strong editing performance—UnKE, AnyEdit, and FABLE—to further examine their effects on

general capabilities. Previous work (Deng et al., 2025) employed a relatively narrow evaluation of the general capabilities of edited models, testing solely on the MMLU dataset (Hendrycks et al., 2021). To assess the impact of editing methods on model general capabilities more comprehensively, we introduced an additional 5 datasets involving language understanding and logical reasoning for a multi-faceted evaluation; specific details are provided in Appendix C.4.

Taking the models edited on the UnFine-UnKE dataset as an example, the results of their general capability assessment are shown in Figure 4. The key findings are summarized as follows: **(1) Baseline methods exhibit varying degrees of degradation in general capabilities**: For models edited using the AnyEdit method, performance significantly declined across multiple tasks. While the UnKE method outperformed AnyEdit, it still showed a noticeable drop in performance on the Qwen2.5-7B-Instruct model. This indicates that baseline methods still have shortcomings in preserving the stability of the original knowledge unrelated to the edits. **(2) FABLE demonstrates an outstanding ability to preserve general capabilities**: On Llama3-8B-Instruct, FABLE matches UnKE in maintaining pre-editing performance. On Qwen2.5-7B-Instruct, although all methods decline slightly, FABLE best preserves general capabilities. This shows it updates unstructured knowledge accurately while minimizing interference. Additional results are in Appendix D.2.

4.5 Ablation Study

Taking the Llama3-8B-Instruct model as an example, we conduct editing tasks on the UnFine-UnKE dataset and perform ablation experiments from multiple dimensions. The results are shown in Table 2. Specifically, we analyze the following four aspects: **(1) Layer Selection**: Updating the 4th, 5th, and 6th layers in $\mathcal{F}_{\text{fine}}$ simultaneously achieves the best trade-off between overall performance and fine-grained performance, validating the effectiveness of this combination. **(2) Data Volume**: The number of fine-grained question-answer pairs has an optimal value. Insufficient quantity (N=1×S) leads to inadequate fine-grained performance, while excessive quantity (N=10×S) significantly impairs overall performance (Rouge-L drops to 92.23) when fine-grained metrics saturate. Experiments indicate that N=5×S is the ideal setting. **(3) Method Necessity**: Removing the first stage (w/o Stage1) or second stage (w/o Stage2) results in declines in both overall and fine-grained performance, demonstrating that fine-grained knowledge injection in the first stage and holistic semantic integration in the second stage serve as the critical foundation. **(4) Data Augmentation Comparison**: To investigate whether performance improvement is merely due to data augmentation, we augment baseline methods with the same fine-grained question-answer pairs. For UnKE, results show that while fine-grained metrics improve slightly, overall performance is significantly impaired (Rouge-L: 92.12), and the combined performance is lower than that of the proposed method. Consistent with the UnKE, augmenting AnyEdit with our data leads to a decline in its holistic performance and fails to achieve the balanced effectiveness exhibited by FABLE. This indicates that performance gains primarily stem from the FABLE method itself, rather than simple data augmentation.

5 Related Work

Model editing has become a rapidly growing research area, as it enables LLMs to update internal knowledge without full retraining (Cao et al., 2021; Mitchell et al., 2022a; Tan et al., 2024; Zhang et al., 2024; Mitchell et al., 2022b; Dong et al., 2022; Huang et al., 2023; Hartvigsen et al., 2023; Yu et al., 2024; Wang et al., 2024; Gu et al., 2024; Ma et al., 2025; Fang et al., 2025; Wang et al., 2025a). Locate-then-edit is a dominant paradigm, enabling precise updates to parameters associated with triple-based knowledge. Representative methods include KN (Dai et al., 2022), ROME (Meng et al., 2022), and MEMIT (Meng et al., 2023). Recent efforts extend editing to unstructured knowledge, evaluated on benchmarks like AKEW (Wu et al., 2024) and UnKEBench (Deng et al., 2025). Methods such as UnKE (Deng et al., 2025), AnyEdit (Jiang et al., 2025), and μKE (Su et al., 2025) adapt locate-then-edit for noisy or lengthy content, while DEM (Huang et al., 2024) focuses on efficient parameter localization.

6 Conclusion

In this paper, we address the limitation that existing unstructured model editing methods focus on holistic text recall but lack reliable fine-grained fact access. We propose FABLE, a hierarchical framework that first anchors fine-grained facts in shallow layers and then updates deeper layers for coherent

surface-form generation. We also introduce UnFine for evaluating fine-grained fact recall. Experiments demonstrate that FABLE improves factual accuracy while maintaining overall text quality.

Statements

LLM Usage. LLMs were used only for sentence polishing and grammar correction, not for content creation or scientific claims.

Ethics statements. All code and datasets used in this work are publicly available. While LLMs offer substantial benefits, they can be misused to generate harmful content, misinformation, or offensive material. Our research on unstructured knowledge editing is conducted ethically and solely for scientific purposes, aiming to improve LLM reliability, interpretability, and factual accuracy. The methods are not intended for malicious use or privacy violations, and we strongly encourage rigorous validation and oversight to ensure responsible application.

Limitation

While our method significantly enhances fine-grained fact recall while maintaining state-of-the-art holistic editing performance, we observe a slight degradation in the generalization of surface-form construction. This may stem from the constrained preservation of fine-grained knowledge in Stage Two, highlighting an inherent trade-off between achieving coherent holistic generation and ensuring precise fine-grained fact access. Future work will explore strategies to further balance holistic fluency and fine-grained factual accuracy.

The present experiments and evaluations are conducted exclusively under a single-edit scenario (batch size = 1). While this setup suffices to demonstrate the feasibility of unstructured-knowledge updates, it omits the more realistic and challenging regimes of batched and sequential editing. Specifically, two limitations are evident: (1) the study does not address batch editing, where multiple independent knowledge-update requests are processed simultaneously—a common real-world situation that heightens the risk of internal knowledge conflicts and parameter interference; and (2) it does not examine sequential editing, in which the model undergoes a series of potentially interrelated updates over time, demanding long-term consistency and robust mitigation of catastrophic forgetting. Future work should extend along both of these dimensions.

Acknowledgments

This work is supported by the National Natural Science Foundation of China (No.U24A20335).

References

Gabriele Bavota. 2016. Mining unstructured data in software repositories: Current and future trends. In *Leaders of Tomorrow Symposium: Future of Software Engineering, FOSE@SANER 2016, Osaka, Japan, March 14, 2016*, pages 1–12. IEEE Computer Society.

Luisa Bentivogli, Bernardo Magnini, Ido Dagan, Hoa Trang Dang, and Danilo Giampiccolo. 2009. The fifth PASCAL recognizing textual entailment challenge. In *TAC*. NIST.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, and 12 others. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Nicola De Cao, Wilker Aziz, and Ivan Titov. 2021. Editing factual knowledge in language models. In *EMNLP (1)*, pages 6491–6506. Association for Computational Linguistics.

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 (Volume 1: Long Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022*, pages 8493–8502. Association for Computational Linguistics.

Jingcheng Deng, Zihao Wei, Liang Pang, Hanxing Ding, Huawei Shen, and Xueqi Cheng. 2025. Everything is editable: Extend knowledge editing to unstructured data in large language models. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net.

William B. Dolan and Chris Brockett. 2005. Automatically constructing a corpus of sentential paraphrases. In *IWP@IJCNLP*. Asian Federation of Natural Language Processing.

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, Abu Dhabi, United Arab Emirates, December 7-11, 2022, pages 5937–5947. 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, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net.

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, EMNLP 2024, Miami, FL, USA, November 12-16, 2024*, pages 16801–16819. Association for Computational Linguistics.

Anshita Gupta, Debanjan Mondal, Akshay Krishna Sheshadri, Wenlong Zhao, Xiang Li, Sarah Wiegreffe, and Niket Tandon. 2023. Editing common sense in transformers. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*, pages 8214–8232. Association for Computational Linguistics.

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 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021. Measuring massive multitask language understanding. In *ICLR*. OpenReview.net.

Xiusheng Huang, Yequan Wang, Jun Zhao, and Kang Liu. 2024. Commonsense knowledge editing based on free-text in llms. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, EMNLP 2024, Miami, FL, USA, November 12-16, 2024*, pages 14870–14880. Association for Computational Linguistics.

Xiusheng Huang, Hang Yang, Yubo Chen, Jun Zhao, Kang Liu, Weijian Sun, and Zuyu Zhao. 2022. Document-level relation extraction via pair-aware and entity-enhanced representation learning. In *Proceedings of the 29th International Conference on Computational Linguistics, COLING 2022, Gyeongju, Republic of Korea, October 12-17, 2022*, pages 2418–2428. International Committee on 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, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.

Aaron Hurst, Adam Lerer, Adam P. Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, Aleksander Madry, Alex Baker-Whitcomb, Alex Beutel, Alex Borzunov, Alex Carney, Alex Chow, Alex Kirillov, Alex Nichol, Alex Paino, and 79 others. 2024. Gpt-4o system card. *CoRR*, abs/2410.21276.

Houcheng Jiang, Junfeng Fang, Ningyu Zhang, Mingyang Wan, Guojun Ma, Xiang Wang, Xiangnan He, and Tat-Seng Chua. 2025. Anyedit: Edit any knowledge encoded in language models. In *Forty-second International Conference on Machine Learning, ICML 2025, Vancouver, BC, Canada, July 13-19, 2025*. OpenReview.net.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, and 1 others. 2024. Deepseek-v3 technical report. *arXiv preprint arXiv:2412.19437*.

Jun-Yu Ma, Hong Wang, Hao-Xiang Xu, Zhen-Hua Ling, and Jia-Chen Gu. 2025. Perturbation-restrained sequential model editing. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. 2022. Locating and editing factual associations in gpt. *Advances in Neural Information Processing Systems*, 35:17359–17372.

Kevin Meng, Arnab Sen Sharma, Alex J. Andonian, Yonatan Belinkov, and David Bau. 2023. Mass-editing memory in a transformer. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.

Sewon Min, Kalpesh Krishna, Xinxi Lyu, Mike Lewis, Wen-tau Yih, Pang Wei Koh, Mohit Iyyer, Luke Zettlemoyer, and Hannaneh Hajishirzi. 2023. Factscore: Fine-grained atomic evaluation of factual precision in long form text generation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*, pages 12076–12100. Association for Computational Linguistics.

Eric Mitchell, Charles Lin, Antoine Bosselut, Chelsea Finn, and Christopher D. Manning. 2022a. Fast model editing at scale. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net.

Eric Mitchell, Charles Lin, Antoine Bosselut, Christopher D. Manning, and Chelsea Finn. 2022b. Memory-based model editing at scale. In *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA*, volume 162 of *Proceedings of Machine Learning Research*, pages 15817–15831. PMLR.

Zian Su, Ziyang Huang, Kaiyuan Zhang, and Xiangyu Zhang. 2025. μke: Matryoshka unstructured knowledge editing of large language models. *CoRR*, abs/2504.01196.

Chenmien Tan, Ge Zhang, and Jie Fu. 2024. Massive editing for large language models via meta learning. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net.

Peng Wang, Zexi Li, Ningyu Zhang, Ziwen Xu, Yunzhi Yao, Yong Jiang, Pengjun Xie, Fei Huang, and Huajun Chen. 2024. WISE: rethinking the knowledge memory for lifelong model editing of large language models. In *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*.

Peng Wang, Ningyu Zhang, Bozhong Tian, Zekun Xi, Yunzhi Yao, Ziwen Xu, Mengru Wang, Shengyu Mao, Xiaohan Wang, Siyuan Cheng, Kangwei Liu, Yuansheng Ni, Guozhou Zheng, and Huajun Chen. 2023. Easyedit: An easy-to-use knowledge editing framework for large language models. *CoRR*, abs/2308.07269.

Peng Wang, Biyu Zhou, Xuehai Tang, Jizhong Han, and Songlin Hu. 2025a. Lyaplock: Bounded knowledge preservation in sequential large language model editing. *CoRR*, abs/2505.15702.

Song Wang, Yaochen Zhu, Haochen Liu, Zaiyi Zheng, Chen Chen, and Jundong Li. 2025b. Knowledge editing for large language models: A survey. *ACM Comput. Surv.*, 57(3):59:1–59:37.

Alex Warstadt, Amanpreet Singh, and Samuel R. Bowman. 2019. Neural network acceptability judgments. *Trans. Assoc. Comput. Linguistics*, 7:625–641.

Adina Williams, Nikita Nangia, and Samuel R. Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. In *NAACL-HLT*, pages 1112–1122. Association for Computational Linguistics.

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, EMNLP 2024, Miami, FL, USA, November 12-16, 2024*, pages 15118–15133. Association for Computational Linguistics.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, and 1 others. 2024. Qwen2. 5 technical report. *arXiv preprint arXiv:2412.15115*.

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, EMNLP 2023, Singapore, December 6-10, 2023*, pages 10222–10240. Association for Computational Linguistics.

Lang Yu, Qin Chen, Jie Zhou, and Liang He. 2024. MELO: enhancing model editing with neuron-indexed dynamic lora. In *Thirty-Eighth AAAI Conference on Artificial Intelligence, AAAI 2024, Thirty-Sixth Conference on Innovative Applications of Artificial Intelligence, IAAI 2024, Fourteenth Symposium on Educational Advances in Artificial Intelligence, EAAI 2014, February 20-27, 2024, Vancouver, Canada*, pages 19449–19457. AAAI Press.

Ningyu Zhang, Bozhong Tian, Siyuan Cheng, Xiaozhuan Liang, Yi Hu, Kouying Xue, Yanjie Gou, Xi Chen, and Huajun Chen. 2024. Instructedit: Instruction-based knowledge editing for large language models. In *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence, IJCAI 2024, Jeju, South Korea, August 3-9, 2024*, pages 6633–6641. ijcai.org.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. Bertscore: Evaluating text generation with BERT. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Zexuan Zhong, Zhengxuan Wu, Christopher D. Manning, Christopher Potts, and Danqi Chen. 2023. Mquake: Assessing knowledge editing in language models via multi-hop questions. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*, pages 15686–15702. Association for Computational Linguistics.

A Benchmark

A.1 Datasets Construction

Generation of Fine-Grained QA Pairs We employ the GPT-4o(Hurst et al., 2024) model to generate five corresponding fine-grained question-answer (QA) pairs for each target output A in AKEW (CounterFact) and AKEW (MQuAKE). The generation process adheres to the following principles:

- **Diverse Perspectives:** Questions are formulated from different angles using varied interrogatives (e.g., what, when, how) to ensure coverage of distinct aspects of the target output, with non-repetitive answer content.

- **Entity Consistency:** All entities mentioned in the answers must originate from the original target output A.
- **Length Control:** Each answer is constrained to a maximum of 15 tokens.
- **Clarity of Expression:** Both questions and answers must be concise, grammatically correct, and avoid meta-references to the text's own structure.

To ensure quality, we verify via string matching that all answer content is derived from the target output A. The specific prompt template is provided in Table 3.

Key Knowledge Phrase Extraction For each dataset, we similarly utilize GPT-4o in a few-shot manner, instructing the model to perform the following operations for each answer:

- **Atomize:** Decompose the answer into its minimal factual units.
- **Concise:** Remove modifiers, fillers, and redundant connecting words.
- **Stay-source:** Ensure the extracted phrases are strictly sub-sequences of the original answer text.
- **Split:** If an answer contains multiple independent facts, split them into separate, parallel phrases.
- **Objectify:** Retain only objective factual statements, filtering out subjective or evaluative language.

The extraction results are also validated via string matching against the original answers to ensure faithfulness. The corresponding prompt template is shown in Table 4.

A.2 Evaluation Metrics

The evaluation in this study employs the following primary metrics:

Lexical Similarity. We use ROUGE-L to measure the n-gram overlap between the model-generated text and the target answer. This metric assesses the surface-level accuracy of the generated content.

Semantic Similarity. To complement the limitations of lexical metrics, we compute the semantic similarity Bert-Score using the all-MiniLM-L6-v2 encoder. This evaluates whether the model has genuinely understood the textual meaning rather than merely repeating surface-level patterns.

Hit Rate (HR). This metric measures the model's ability to accurately recall key knowledge. Formally, for a fine-grained gold answer A^{fine}, let the set of extracted key phrases be $KP = \{kp_1, kp_2, \ldots, kp_m\}$, and the model output be O^{fine}. An indicator function $\mathbb{I}(kp_i, O)$ is defined for each key phrase:

$$\mathbb{I}(kp_i, O) = \begin{cases} 1, & \text{if } kp_i \text{ is a substring of } O^{fine}, \\ 0, & \text{otherwise.} \end{cases} \tag{11}$$

The Hit Rate is then the average of these indicators over all key phrases:

$$HR = \frac{1}{m} \sum_{i=1}^{m} \mathbb{I}(kp_i, O^{fine}). \tag{12}$$

Longest Common Subsequence Coverage (C_{LCS}). This metric quantifies the completeness of the model output in covering the content of the answer. Let the fine-grained gold answer A^{fine} be tokenized into a word sequence $A^{fine} = [a_1^{fine}, a_2^{fine}, \ldots, a_n^{fine}]$ and the model output O^{fine} into $O^{fine} = [o_1^{fine}, o_2^{fine}, \ldots, o_l^{fine}]$. Denoting the length (in words) of the longest common subsequence between A^{fine} and O^{fine} as $\text{LCS}(A^{fine}, O^{fine})$, the coverage C_{LCS} is defined as:

$$C_{\text{LCS}} = \frac{\text{LCS}(A^{fine}, O^{fine})}{n} \tag{13}$$

where n is the total number of words in A^{fine}. The LCS length is computed using a standard dynamic programming algorithm. Let $\text{LCS}(i, j)$ represent the length of the LCS between the prefixes $A^{fine}[1:i]$ and $O^{fine}[1:j]$. Let $\text{LCS}(i, 0) = 0$ and $\text{LCS}(0, j) = 0$, For $\forall i, j \geq 0$, If $a_i^{fine} = o_j^{fine}$, $\text{LCS}(i, j) = \text{LCS}(i-1, j-1) + 1$, else $\text{LCS}(i, j) = \max\{\text{LCS}(i-1, j), \text{LCS}(i, j-1)\}$. Finally, $\text{LCS}(A^{fine}, O^{fine}) = \text{LCS}(n, l)$.

B Multi-Aspect Fine-Grained Knowledge Extraction for Unstructured Text

In target unstructured texts, there often exist various types of fine-grained knowledge. To ex-

Table 3: Prompt Template for Generating Fine-Grained QA Pairs

Prompt Template for Generating Fine-Grained QA Pairs

You are asked to generate some short question-answer pairs based on the specified <Text>. These question-answer pairs mainly ask questions about the knowledge entities in the <Text>, and the answers should be the knowledge entities being asked.

Rules:
- Diversify Perspectives: Formulate questions from multiple angles to minimize overlapping answers. Use diverse question words (What, When, Where, Why, How, etc.); avoid "Who" or "Whose". Avoid yes/no questions unless they are highly informative.
- Entity Inclusion: Ensure all entities mentioned in the answers are present in the <Text>.
- Length Constraint: Each answer must not exceed 15 tokens.
- Fluency and Clarity: Ensure that both questions and answers are clear, grammatical, and concise. Do not include meta-phrases such as "in the sentence", "according to the sentence", "mentioned in the text", etc. inside the questions themselves; the question must read as if it were asked in real life, not about the text.
- The <Output> formats the pairs as a JSON object with "questions" and "answers" lists.

```
{
   "questions": []
   "answers": []
}
```

Now generate 5 question-answer pairs for the following <Text>.
<Text>: {text}
<Output>:

tract this knowledge as comprehensively and non-overlap as possible from different perspectives, we designed the following pipeline:

First, the target text is segmented into several sub-sentences using the NLTK toolkit.

Next, a word-count threshold is set to filter out sub-sentences shorter than the threshold, thereby reducing noise caused by insufficient information.

Then, for each retained sub-sentence, multiple high-quality natural-language questions are generated from various angles based on GPT-4o (details of the templates are shown in Table 5). The question generation follows these principles: (1) Answerability: the question should be directly answerable based on the sub-sentence, with the answer either explicitly stated or reasonably inferable from it; (2) Clarity: the question must be grammatically correct, clearly expressed, highly consistent with the sub-sentence content, and unambiguous; (3) Non-binary: yes/no questions are avoided unless their answers carry high informational value, and open-ended interrogatives are encouraged; (4) Diversity: varied interrogative words are used to prevent repetitive questioning patterns.

Subsequently, for each generated question, GPT-4o is called once more to produce a corresponding answer based on the target text, forming an initial question-answer pair. The question-answer pairs corresponding to one sub-sentence constitute a cluster. We first filter out those pairs whose answers are primarily derived from that sub-sentence, and then perform merging and deduplication: if the answer of one pair completely contains the content of another answer, they are regarded as more complete expressions of the same questioning perspective; the former is retained and the latter is removed. After this step, each sub-sentence ultimately retains a set of the most diverse and information-complete fine-grained question-answer pairs.

Finally, to increase the number of retained questions, each kept question is used as a seed question, and GPT-4o is employed to generate multiple semantically similar, fine-grained questions that point to the same answer, thereby further expanding the set of fine-grained knowledge questions.

C Experimental Setup

C.1 Baseline Methods

Here, we outline the baseline knowledge editing methods employed for comparison in this study:

FT-L. The FT-L method adjusts designated layers of the LLM via autoregressive loss minimization to incorporate new knowledge.

ROME. ROME is a model editing technique that localizes and updates factual associations in autoregressive transformers by identifying critical mid-layer MLP modules as key-value memories. It computes a subject-specific key from hidden states and optimizes a value vector to represent new knowledge, then applies a rank-one weight update to the MLP projections, effectively inserting facts while maintaining generalization and specificity.

MEMIT. MEMIT, building upon ROME, is a scalable multi-layer editing algorithm. It efficiently

Table 4: Prompt Template for Key Knowledge Phrase Extraction

Prompt Template for Key Knowledge Phrase Extraction

You are an assistant designed to extract minimal factual answers from any given <Answer>.

Rules:
- Break down the <Answer> into the smallest possible factual units. Each unit should represent a standalone piece of information.
- Remove any stylistic flourishes, adjectives, adverbs, or unnecessary connectors (e.g., "over" in "over a decade" is retained only if it's part of the factual unit, as shown in the example).
- All tokens of the extracted answer must strictly reside within the <Answer>.
- If the <Answer> contains multiple distinct facts separated by conjunctions like "and" or commas, split them into separate items in the list.
- Focus solely on factual content; ignore any subjective or embellished language.

Example:
<Question>: How long has George Rankin been involved in politics?
<Answer>: Over a decade.
<Output>:
{
  "answers": ["over a decade"]
}

<Question>: What positions has George Rankin held in politics?
<Answer>: City council member and state representative.
<Output>:
{
"answers": ["city council member", "state representative"]
}

<Question>: Where is George Rankin frequently quoted?
<Answer>: Local and national news outlets.
<Output>:
{
  "answers": ["local news outlets", "national news outlets"]
}

<Question>: What did John Mayne discuss in his interview with The Huffington Post?
<Answer>: His passion for journalism and his commitment to reporting on important issues.
<Output>:
{
  "answers": ["passion for journalism", "commitment to reporting on important issues"]
}

Now generate the minimal factual answers for the following input.
<Question>: {question}
<Answer>: {answer}
<Output>:

integrates large-scale factual updates into LLMs by computing explicit parameter adjustments, thereby modifying memories while preserving the model's overall integrity.

AnyEdit. AnyEdit is an autoregressive editing paradigm designed to overcome the limitations of single-token editing methods in large language models. It breaks down long-form knowledge into sequential chunks and iteratively edits the key token in each chunk, leveraging the Chain Rule of Mutual Information to ensure consistent and accurate generation of diverse formats.

UnKE. UnKE is a method designed to edit unstructured knowledge in large language models by extending previous approaches in two key dimensions: it replaces local layer key-value storage with non-local block key-value storage to better represent distributed knowledge across layers, and employs cause-driven optimization that edits the last token directly to preserve context without needing term localization, thereby handling complex, long-form unstructured data effectively.

C.2 Implementation Details

All experiments were conducted on a single NVIDIA A100 (80 GB) GPU. To control variables and ensure a fair comparison, all methods in this study were implemented based on the widely-used model editing toolkit EasyEdit(Wang et al., 2023), with its default optimal hyperparameters adopted.

Table 5: Prompt Template for Multi-Aspect Fine-Grained Knowledge Extraction

Prompt Template for Multi-Aspect Fine-Grained Knowledge Extraction

You are a question generation expert. Your task is to generate **non-overlapping** natural language questions based on the given sentence.

Rules:
- Each question must be answerable directly using only the information in the sentence. The answer to each question must appear in the sentence (verbatim or semantically contained)
- The question should be clear, grammatical, and relevant.
- Avoid yes/no questions unless they are highly informative.
- Use diverse question words (What, When, Where, Why, How, etc.); avoid "Who" or "Whose".
- Avoid pronouns; use specific names, places, etc.
- Do not include meta-phrases such as “in the sentence”, “according to the sentence”, “mentioned in the text”, etc. inside the questions themselves; the question must read as if it were asked in real life, not about the text.
- Do not repeat the same information in two different questions; ensure every answer is mutually exclusive in content.

Example:
Paragraph: "Marie Curie discovered radium in 1898 with her husband Pierre Curie. This breakthrough came after years of painstaking work in a leaky, unheated shed, where the Curies processed tons of pitchblende residue to isolate minute quantities of the glowing new element. Marie herself coined the name "radium" from the Latin word for "ray", captivated by its mysterious, persistent luminescence.
Sentence: "Marie Curie discovered radium in 1898 with her husband Pierre Curie."
Output:
{
"questions": ["Who discovered radium in 1898?", "What element did Marie Curie discover?", "When did Marie Curie and Pierre Curie discover radium?"]
}

Now generate questions for the following sentence (no fixed count; stop when angles are exhausted):

Paragraph: {paragraph}
Sentence: "{sentence}"
Output:

C.3 Time Cost

Table 6 further compares the editing efficiency of different methods. It can be observed that while FT-L is the fastest, it sacrifices editing effectiveness (see Table 1), revealing an inherent efficiency–effectiveness trade-off in this task. The relatively higher time cost of FABLE is a direct consequence of its design objective: to achieve more reliable unstructured knowledge editing with minimal side effects, the method employs a refined multi-step optimization process. Although this design increases the per-edit computational cost, it yields significant advantages in terms of editing performance and the preservation of general capabilities. Improving time efficiency will be a clear direction for future optimization.

C.4 Details of General Capability Evaluation Datasets

To assess the impact of knowledge editing on the model’s general capabilities, we selected five datasets covering different linguistic understanding and reasoning tasks. For each editing sample, we uniformly sampled 5 data instances per task category (25 instances in total) to form an evaluation subset, and used the F1-score to measure the model’s post-editing performance. Each dataset is described in detail below:

Table 6: Time cost of different methods across various models and datasets.

Model	Method	UnFine-UnKE(/s)	UnFine-CF(/s)	UnFine-MQ(/s)
LLaMA3-8B	FT-L	3.76	3.44	3.55
	ROME	17.74	15.02	16.78
	MEMIT	54.50	46.43	39.28
	UnKE	33.79	29.84	30.54
	AnyEdit	20.99	15.55	15.94
	FABLE	356.71	340.18	329.63
Qwen2.5-7B	FT-L	2.90	2.74	2.73
	ROME	72.44	60.37	61.53
	MEMIT	54.95	45.96	46.78
	UnKE	33.25	32.04	31.65
	AnyEdit	22.83	21.64	21.73
	FABLE	330.04	322.12	309.77

CoLA. CoLA (Warstadt et al., 2019) evaluates grammatical acceptability via binary classification of single-sentence judgments.

MMLU. MMLU (Hendrycks et al., 2021) measures multi-task accuracy across diverse domains, focusing specifically on zero-shot and few-shot learning scenarios in text models.

NLI. NLI (Williams et al., 2018) assesses language understanding by requiring models to identify logical relationships—such as entailment, contradiction, or neutrality—between pairs of sentences.

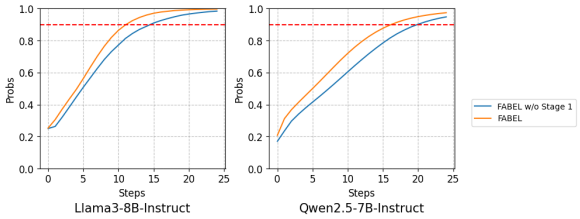

Figure 5: Performance comparison during optimization between FABEL (with Stage 1) and its ablated variant (without Stage 1) on the UnFine-UnKE. The horizontal axis shows the number of optimization steps, and the vertical axis shows the average output probability of the target unstructured text.

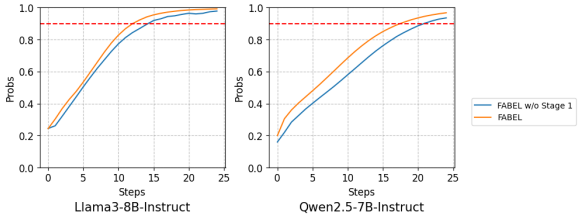

Figure 6: Performance comparison during optimization between FABEL (with Stage 1) and its ablated variant (without Stage 1) on the UnFine-UnKE. The horizontal axis shows the number of optimization steps, and the vertical axis shows the average output probability of the target unstructured text.

MRPC. MRPC (Dolan and Brockett, 2005) evaluates semantic equivalence detection, where models determine whether two sentences convey the same meaning.

RTE. RTE (Bentivogli et al., 2009) examines whether a premise sentence logically supports a given hypothesis.

D More Experimental Results

D.1 More Analysis of the Performance Enhancement Results

Figure 5 and Figure 6 illustrate the optimization process of the proposed method on the UnFine-CF and UnFine-MQ datasets, respectively. As shown in the figures, the complete method with stage one (injection of atomic facts), i.e., FABLE, demonstrates both higher initial probability and faster convergence rate across the two different scenarios, consistent with the observations on the UnFine-UnKE dataset. This further validates the effectiveness and robustness of FABLE across diverse tasks and settings, highlighting its strong generalizability.

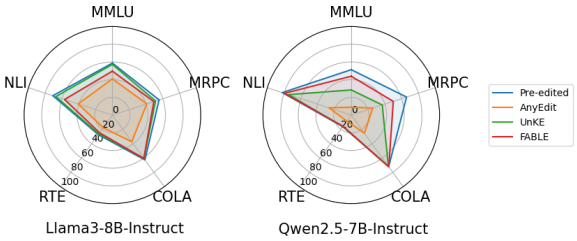

Figure 7: F1-scores of each task for unstructured editing methods on different models after editing on the UnFine-CF dataset

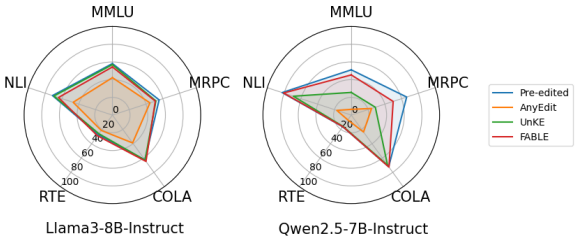

Figure 8: F1-scores of each task for unstructured editing methods on different models after editing on the UnFine-MQ dataset

D.2 More General Capability Evaluation Results

To further evaluate the performance of editing methods under the influence of different editing datasets, we additionally present the general capabilities of models after editing on the UnFine-CF and UnFine-MQ datasets, with the results shown in Figure 7 and Figure 8, respectively. Overall, FABLE still demonstrates relatively more stable retention of general capabilities across the evaluated tasks. However, its specific performance is influenced by factors such as model architectures and types of editing datasets, leading to variations under different experimental settings.

D.3 Conflicting Unstructured Knowledge Updates

The "conflicting unstructured knowledge updates" means the model must learn two different descriptions about the same subject simultaneously—is indeed a crucial test for evaluating the robustness and practicality of an editing method. We have given this serious consideration and conducted preliminary explorations. We manually constructed a small-scale Conflicting Editing DataSet (100 samples) based on the existing UnFine-UnKE dataset. Each sample contains two different unstructured descriptions about the same subject, simulating real-world knowledge conflicts. We conducted ex-

periments using the Llama3-8B-Instruct model and the FABLE method. The preliminary results are shown in the Table 7.

The experimental results show that in the conflicting scenario, all performance metrics of FABLE experience a slight decline compared to the standard non-conflicting scenario. This is expected, indicating that contradictory knowledge indeed poses an additional challenge to the model's parameter updates and knowledge retention. However, it is noteworthy that the magnitude of the performance drop is relatively limited. Even under the conflicting setting, FABLE maintains strong capabilities in both holistic text generation (Rouge-L: 93.04) and fine-grained fact recall (HR: 49.62). This preliminary finding suggests that the hierarchical fact-anchoring strategy employed by FABLE provides a certain buffer against the conflicts arising from direct parameter overwriting, demonstrating potential robustness when facing contradictory knowledge updates.

D.4 Representation Comparison

In unstructured editing methods (e.g., UnKE, AnyEdit), the standard optimization target is typically the hidden state of the last token of the complete input sequence, not a specific token representing the "subject word" within the fact. This has become an empirical setup adopted by many works in this field. As an advancement within the unstructured editing framework, our method follows this setup in its first stage. This aims to maintain comparability with baseline methods while exploring a superior hierarchical decoupling design. We acknowledge that in earlier structured editing approaches, the last token of the subject word (not the last token of the sequence) is often used as the clean fact representation. While we recognize this might lead to contextual mixing, it also makes the representation more suitable for the subsequent unstructured text generation objective.

In order to empirically evaluate the advantages and disadvantages of using the "last sequence token" versus the "last subject word token" as the fact representation, the experimental results are presented as Table 8. The experiments above take editing the Llama3-8B-Instruct model on the UnFine-UnKE dataset as an example.

The experimental results indicate that all metrics for FABLE (Last Subject Word Token) are lower than those of our proposed FABLE (Last Token). The reasons may lie in the following: (1) Advantage in Holistic Metrics: In unstructured editing tasks, using the Last Token is more advantageous. Employing the Last Token in Stage 1 can incorporate richer contextual information, thereby better facilitating semantic integration in Stage 2; (2) Advantage in Fine-grained Metrics: When multiple fine-grained facts share the same Last Subject Word Token, it may trigger representational competition or conflict. The design using the Last Token helps mitigate this issue. This experiment is designed to clearly reveal the impact of different representation choices on the final editing effectiveness, thereby strengthening the persuasiveness of our design decision.

D.5 Case Study

We present generation examples of the Llama3-8B-Instruct and Qwen2.5-7B-Instruct models after being edited by different methods, as shown in Table 9 and Table 10. Here, <Holistic> represents the model's generation result when the editing prompt is used as input, while <Fine-grained> represents the result for a fine-grained question. The text highlighted in red corresponds to the ground truth answer for that fine-grained question (which is contained within the Unstructured Target Output). Through comparative analysis, the following conclusions can be drawn:

(1) Structured editing methods show limited performance in both holistic and fine-grained generation. Methods represented by ROME and MEMIT fail to effectively store the unstructured target output into the model parameters, leading to factual hallucinations during generation.

(2) Unstructured editing methods achieve good holistic generation but suffer from insufficient fine-grained knowledge recall. Methods represented by UnKE and AnyEdit, while capable of storing and reproducing the holistic target output reasonably well, perform poorly in recalling the internal fine-grained knowledge.

(3) FABLE demonstrates excellent performance in both holistic and fine-grained generation. Our method not only accurately stores the unstructured target output but also effectively recalls the fine-grained knowledge contained within it, achieving stable and reliable performance in both types of generation tasks.

Table 7: The experiment results of Llama3-8B-Instruct on the UnFine-UnKE dataset and Conflicting Editing DataSet.

Mehtod	Holistic		Fine-grained			
	Bert-Score↑	Rouge-L↑	Bert-Score↑	Rouge-L↑	HR↑	C_{LCS}↑
FABLE (Non-conflicting)	99.36±0.04	97.78±0.10	65.63±0.16	53.79±0.19	53.31±0.23	61.78±0.19
FABLE (Conflicting)	98.91±0.03	93.04±0.15	60.33±0.09	50.48±0.11	49.62±0.14	56.39±0.08

Table 8: The representation comparison results of Llama3-8B-Instruct on the UnFine-UnKE dataset.

Mehtod	Holistic		Fine-grained			
	Bert-Score↑	Rouge-L↑	Bert-Score↑	Rouge-L↑	HR↑	C_{LCS}↑
FABLE (Last Token)	99.36±0.04	97.78±0.10	65.63±0.16	53.79±0.19	53.31±0.23	61.78±0.19
FABLE (Last Subject Word Token)	98.49±0.06	94.82±0.16	58.44±0.16	43.33±0.20	42.43±0.23	51.67±0.20

Table 9: Model Editing Case Study on Llama3-8B-Instruct

Model Editing Case Study on LLAMA3	
Editing Prompt	What are Kathy Saltzman's notable achievements in the field of composition and what makes her a leading figure in contemporary classical music?
Unstructured Target Output	Kathy Saltzman is a highly accomplished composer . . . . . some of the most prestigious musical groups in the world, including the New York Philharmonic and the London Symphony Orchestra. . . . . in contemporary classical music.
Fine-grained Question	Which renowned musical groups have performed Kathy Saltzman's work?
Generation Output	
FT-L	**<Holistic>**: and Saltzman is a composer and composer-in-residence . . . . . companies and orchestras, including **<Fine-grained>**: and Saltzman is a composer and composer, and the work of the composer . . . . .
ROME	**<Holistic>**: Kathy Saltzman is not a well-known composer, and a . . . . . assist you with that as well. **<Fine-grained>**: I apologize, but I couldn't find any information on a renowned musical group performing . . . . .
MEMIT	**<Holistic>**: Kathy Saltzman is a renowned composer and has written . . . . . numerous awards and accolades. **<Fine-grained>**: Kathy Saltzman is a composer and songwriter who has worked with a variety of musical . . . . .
UnKE	**<Holistic>**: Kathy Saltzman is a highly accomplished composer . . . . . some of the most prestigious musical groups in the world, including the New York Philharmonic and the London Symphony Orchestra. . . . . in contemporary classical music. **<Fine-grained>**: To answer your question, Kathy Saltzman is a talented musician . . . . .
AnyEdit	**<Holistic>**: Kathy Saltzman is a highly accomplished composer . . . . . some of the most prestigious musical groups in the world, including the New York Philharmonic and the London Symphony Orchestra. . . . . dedicated advocate for the arts, and **<Fine-grained>**: Kathy Saltzman is a talented musician, and her work has been performed . . . . .
FABLE	**<Holistic>**: Kathy Saltzman is a highly accomplished composer . . . . . some of the most prestigious musical groups in the world, including the New York Philharmonic and the London Symphony Orchestra. . . . . in contemporary classical music. **<Fine-grained>**: Some of the most prestigious musical groups in the world, including the New York Philharmonic and the London Symphony Orchestra.

Table 10: Model Editing Case Study on Qwen2.5-7B-Instruct

Model Editing Case Study on LLAMA3	
Editing Prompt	What are Kathy Saltzman's notable achievements in the field of composition and what makes her a leading figure in contemporary classical music?
Unstructured Target Output	Kathy Saltzman is a highly accomplished composer . . . . . some of the most prestigious musical groups in the world, including the New York Philharmonic and the London Symphony Orchestra. . . . . in contemporary classical music.
Fine-grained Question	Which renowned musical groups have performed Kathy Saltzman's work?
Generation Output	
FT-L	**<Holistic>**: zman. It is highly likely that there is a name or a musical . . . . . the work of the New Philhar **<Fine-grained>**: zman and her work. After a more in-depth and widely recognized music and . . . . .
ROME	**<Holistic>**: Kathy Saltzman is a highly accomplished composer, having . . . . . talent and dedication as a composer. **<Fine-grained>**: I do not have specific information about which musical groups have performed Kathy Saltzman's work. . . . .
MEMIT	**<Holistic>**: Kathy is a highly accomplished composer, having written . . . . . influence composers around the world. **<Fine-grained>**: . . . . . including the New York Philharmonic, the London Symphony Orchestra . . . . .
UnKE	**<Holistic>**: Kathy Saltzman is a highly accomplished composer . . . . . some of the most prestigious musical groups in the world, including the New York Philharmonic and the London Symphony Orchestra. . . . . in contemporary classical music. **<Fine-grained>**: . . . . . including the Los Angeles Philharmonic and the Boston Pops Orchestra . . . . .
AnyEdit	**<Holistic>**: Kathy Saltzman is a highly accomplished composer . . . . . some of the most prestigious musical groups in the world, including the New York Philharmonic and the London Symphony Orchestra. . . . . dedicated advocate for the arts, and **<Fine-grained>**: Kathy Saltzman is a highly talented composer, having written numerous pieces . . . . .
FABLE	**<Holistic>**: Kathy Saltzman is a highly accomplished composer . . . . . some of the most prestigious musical groups in the world, including the New York Philharmonic and the London Symphony Orchestra. . . . . in contemporary classical music. **<Fine-grained>**: The New York Philharmonic and the Los Angeles Philharmonic.