Fisher-Driven Adaptive Locating for Knowledge Editing in Large Language Models

Chenghao Xu[1], **Jiexi Yan**[2], **Guangtao Lyu**[3], **Qi Liu**[3], **Muli Yang**[4], **Cheng Deng**[1*]
[1] College of Information Science and Engineering, Hohai University, Changzhou, China,
[2] School of Computer Science and Technology, Xidian University, Xi'an, China,
[3] School of Electronic Engineering, Xidian University, Xi'an, China,
[4] Institute for Infocomm Research (I^{2}R), A*STAR, Singapore
chx@hhu.edu.cn, {qiliu,guangtaolyu}@stu.xidian.edu.cn,
{jxyan1995,muliyang.xd,chdeng.xd}@gmail.com

Abstract

Large language models (LLMs) store extensive factual knowledge acquired during pretraining, yet this knowledge is inherently static and may become inaccurate or outdated, leading to knowledge hallucinations. Knowledge editing offers an efficient alternative to full retraining by enabling targeted factual updates while preserving overall model behavior. Existing locate-then-edit methods, however, rely on fixed layer selection strategies, treating the locating stage as a static design choice and failing to account for the hierarchical and instance-dependent nature of knowledge representation in LLMs. In this paper, we propose FiDAL, a Fisher-driven adaptation-aware locating strategy that dynamically identifies which model components should be edited for a given knowledge update. FiDAL formulates localization as a weight-level decision problem and leverages Fisher Information to select layers that are both influential and sensitive to factual modifications. A lightweight probing stage with low-rank modulation enables efficient localization with minimal overhead. Experiments on standard benchmarks demonstrate that FiDAL consistently improves editing effectiveness and knowledge preservation across multiple editing methods.

1 Introduction

Large language models have demonstrated strong capabilities in storing and retrieving extensive factual knowledge acquired during pretraining on large-scale web corpora (Zhao et al., 2023; Brown et al., 2020; Radford et al., 2019; Bi et al., 2025b; Xu et al., 2024). However, this knowledge is inherently static and may contain inaccuracies or become outdated due to the uncurated and time-sensitive nature of the training data (De Cao et al., 2021; Mitchell et al., 2021; Bi et al., 2025a; Lyu et al., 2026a). As a result, such models often exhibit knowledge hallucinations, producing confident but incorrect or obsolete information. For example, when asked about the most recent Olympic host city, a model may incorrectly respond with "Tokyo" despite the 2024 Games having been held in Paris (Meng et al., 2022a,b). Although full retraining or fine-tuning could address these errors, such approaches are computationally expensive and risk catastrophic forgetting or overfitting existing knowledge. These limitations have motivated growing interest in knowledge editing, which aims to precisely update specific factual errors in pretrained models without requiring exhaustive retraining. Among the proposed approaches, the locate-then-edit framework has emerged as a principled and scalable solution. Introduced by the ROME method (Meng et al., 2022a), this framework follows a two-stage process. First, it locates a minimal set of model parameters that are causally responsible for a target fact using techniques such as causal tracing, for example by identifying layers where the subject "Olympics" activates outdated location information. Second, it edits these parameters to align the model output with verified updates, such as replacing "Tokyo" with "Paris." By enabling targeted modifications to knowledge representations while preserving overall model behavior, locate-then-edit methods provide an efficient means of maintaining factual accuracy as real-world knowledge evolves.

Current knowledge editing methods (Deng et al., 2025; Fang et al., 2024; Jiang et al., 2024; Liu et al., 2025; Xu et al., 2026; Liu et al., 2026) predominantly emphasize the editing stage while treating the locating step as a fixed and inflexible procedure. Most approaches, including widely used locate-then-edit frameworks, constrain modifications to predetermined model layers, with layer selection handled as a static hyperparameter rather than an adaptive process. Even methods that introduce

*Corresponding author

Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 20902–20913

neuron-level filtering for more refined localization remain restricted to these pre-specified layers. This rigidity stands in contrast to well-established findings on hierarchical knowledge representation in transformer architectures, where shallow layers primarily capture syntactic information, intermediate layers encode factual or world knowledge, and deeper layers represent more abstract semantics (Chuang et al., 2023). Importantly, the effectiveness of factual knowledge editing varies substantially across layers, as demonstrated by successful interventions that target specific feedforward networks within language models. Consequently, enforcing edits within fixed layers, irrespective of the type or subject of the knowledge being updated, fails to exploit the model's inherent functional hierarchy (Wu et al., 2025; Liang et al., 2025). This limitation reduces editing precision and hinders generalization across diverse factual updates, highlighting a fundamental mismatch between existing methodologies and the dynamic, layered nature of knowledge representation in LLMs.

In this paper, we propose a *Fisher-driven adaptation-aware locating strategy* (FiDAL) that dynamically determines which components of a model should be edited. Motivated by the observation that knowledge in LLMs is hierarchically distributed across layers, our method performs knowledge preservation and modification directly at the weight level. Specifically, we formulate the locating process as a decision-making problem that evaluates the importance of individual weights in the linear layers of feed-forward networks (FFNs) when editing a specific knowledge instance. To this end, we leverage the Fisher Information (Rissanen, 2002) as a principled criterion for identifying layers that are both influential and critical for knowledge editing via weight modulation. The underlying intuition is that parameters with higher Fisher Information contribute more significantly to the likelihood of a given knowledge instance and are therefore more sensitive to factual modifications.

During the probing stage, only two low-rank modulation parameters are trainable, and the Fisher Information is computed exclusively with respect to these parameters. Consequently, the probing process is highly lightweight and introduces minimal computational overhead. The outcome of this stage is a set of decisions indicating whether kernel modulation should be applied or whether each individual linear layer should remain frozen. Based on these decisions, the main knowledge editing procedure is subsequently performed. Our proposed method demonstrates superior performance compared to existing baseline approaches across multiple evaluation metrics on standard benchmarks.

2 Related Work

Knowledge editing techniques can be broadly classified into two families: parameter-editing methods, which explicitly modify a model's weights, and parameter-preserving methods, which retain the original parameters and instead introduce auxiliary components. Parameter-editing approaches encode new information by fine-tuning a small subset of weights. Representative examples include meta-learning formulations that leverage hypernetworks to produce targeted parameter updates (Jiang et al., 2024), with computational efficiency often improved via low-rank approximations to gradient updates (Mitchell et al., 2021). In contrast, locate-then-edit frameworks first identify knowledge-relevant components through causal attribution (Meng et al., 2022a) and then revise them using closed-form procedures such as least-squares optimization (Zheng et al., 2023). To alleviate catastrophic forgetting, several methods further constrain edits by restricting updates to selected neurons (Jiang et al., 2024) or by projecting modifications into a null space, thereby improving suitability for lifelong learning scenarios (Fang et al., 2024).

Parameter-preserving methods, by comparison, maintain the integrity of the original weights by allocating additional mechanisms (Hartvigsen et al., 2023). Some exploit in-context learning to incorporate new knowledge without any parameter changes (Zheng et al., 2023; Lyu et al., 2025; Xu et al., 2025; Bi et al., 2024), whereas others rely on external memory retrieval (Mitchell et al., 2022) or dynamically expand the model (e.g., by introducing new neurons) to represent edited content (Huang et al., 2023; Dong et al., 2022; Bi et al., 2025c; Lyu et al., 2026b). Additional advances replace internal hidden states with entries from a discrete codebook or improve memory-based fusion via learned, parameterized modules (Wang et al., 2024).

Recent work increasingly focuses on editing unstructured knowledge, i.e., information expressed in free-form text rather than structured triples (Deng et al., 2025). In this direction, Wu et al. (2024) critiques the limitations of prior bench-

marks and proposes AKEW to specifically evaluate unstructured knowledge editing. Extending the locate-then-edit paradigm, UnKE (Deng et al., 2025) updates all parameters within a single layer to better accommodate unstructured content and is assessed on the newly introduced UnKEBench. Meanwhile, DEM (Huang et al., 2024) proposes a dynamic perception module that localizes commonsense knowledge representations, enabling precise updates from textual descriptions. Broadening the scope of model editing beyond factual assertions, AnyEdit supports modifications to heterogeneous textual forms, thereby enabling more general and versatile knowledge manipulation (Jiang et al., 2025).

However, most existing methods constrain edits to predetermined layers, regardless of the type or subject matter of the knowledge being modified, and thus fail to exploit the model's intrinsic functional hierarchy. This rigidity undermines editing precision and limits generalization across diverse factual updates. To address this limitation, we propose an adaptation-aware locating strategy that dynamically identifies appropriate layers for knowledge editing.

3 Method

In this section, we first formulaically introduce the knowledge editing task and highlight the limitations of existing methods, followed by a detailed description of the proposed FiDAL.

3.1 Preliminary

Locate then Edit Paradigm. The locate-then-edit paradigm targets the modification of triplet-structured factual knowledge represented as (s, r, o), where s denotes the subject, r the relation, and o the object. A typical knowledge update transforms an existing fact (s, r, o_{old}) into a new one (s, r, o_{new}), for example changing (Olympics, will be held in, Paris) to (Olympics, will be held in, Los Angles). For notational convenience, a knowledge instance is expressed as (x, y), where $x = (s, r)$ represents the input query and $y = o$ denotes the corresponding object.

This paradigm assumes that factual knowledge is primarily stored in the feedforward network (FFN) layers of transformer-based language models and treats these layers as linear associative memory modules. Knowledge editing is conducted through a two-stage process. First, a single transformer layer is identified and designated as the editing layer. Second, the knowledge update is performed by modifying model parameters or internal representations exclusively within this fixed layer.

The selection of the editing layer is often motivated by the early decoding phenomenon observed in transformer architectures. Previous studies indicate that shallow layers mainly encode key vectors capturing entity-specific information and contextual semantics, whereas deeper layers function as decoders that transform these keys into value vectors corresponding to target outputs. Based on this observation, the L-th layer is chosen as a functional boundary that partitions the model into two components: a key generator, comprising layers up to L, and a value generator, comprising layers above L. Consequently, the L-th layer is selected for editing, and once chosen, it remains fixed throughout the entire editing process.

Computing Key-Value. For a given knowledge pair (x, y), the corresponding key vector $\boldsymbol{k}$ is extracted from the hidden state of the n-th token associated with x at layer L, denoted as $\boldsymbol{k} = h^L_{x,n}$. To edit the knowledge to a new pair $(\tilde{x}, \tilde{y})$, the method computes an updated key-value pair $(\boldsymbol{k}^*, \boldsymbol{v}^*)$. Specifically, the updated key vector is defined as $\boldsymbol{k}^* = \boldsymbol{k} + \delta_n$, where δ_n is a residual vector optimized via gradient descent. This optimization is formulated as

$$\boldsymbol{k}^* = \boldsymbol{k} + \arg\min_{\delta_n} \left(-\log \mathbb{P}_{f^L_{\theta_L}(\boldsymbol{k}+\delta_n)}(\tilde{y}|\tilde{x}) \right), \quad (1)$$

where $f^L_{\theta_L}(\boldsymbol{k} + \delta_n)$ denotes the transformation of the L-th layer when the original key $\boldsymbol{k}$ is replaced by $\boldsymbol{k} + \delta_n$. During this optimization, the parameters of the value generator are held fixed. Minimizing this objective ensures that, given the optimized key vector $\boldsymbol{k}^*$, the value generator can decode the desired target output $\tilde{y}$.

A defining characteristic of this paradigm is that all localization and editing operations are confined to the fixed layer L, irrespective of the semantic type or complexity of the knowledge being edited. While effective in certain settings, this rigid design limits the flexibility and generality of existing locate-then-edit methods and motivates us to develop a more adaptive knowledge editing mechanism.

3.2 Fisher-Driven Adaptive Locating

LLMs are composed of multiple stacked layers, each containing an attention mechanism and a feed-

forward network. Most existing knowledge editing methods modify a fixed linear submodule within the FFN of a predetermined layer, resulting in a rigid editing strategy.

Prior studies demonstrate that knowledge is hierarchically distributed across layers, with shallow layers encoding syntactic features, intermediate layers capturing factual knowledge, and deeper layers representing abstract semantics. The effectiveness of knowledge editing varies substantially across layers, with different feedforward networks proving optimal for different knowledge types. Consequently, constraining edits to a fixed layer fails to exploit this functional hierarchy, limiting editing precision and generalization across diverse knowledge updates.

To address this issue, we propose an adaptation-aware locating strategy that dynamically determines which components of the model should be edited. Recognizing that knowledge in LLMs is hierarchically distributed across layers, our approach performs knowledge preservation and modification at the weight level. Specifically, we formulate the locating process as a decision problem that assesses the importance of individual weights across linear layers within the FFNs when editing a particular piece of knowledge.

Fisher-Driven Adaptive Sensitivity Probing. To enable dynamic and knowledge-dependent localization, we propose leveraging Fisher Information as a principled criterion for identifying layers that are most influential and critical for knowledge editing. Fisher Information characterizes the curvature of the loss landscape with respect to model parameters, thereby indicating which weights most strongly affect the model's predictions for a given fact. The underlying intuition is that parameters with higher Fisher Information contribute more substantially to the likelihood of a specific knowledge instance and are consequently more sensitive to factual modifications.

Parameters associated with higher aggregated Fisher scores are therefore regarded as more important for encoding the target knowledge and are selected for editing. This adaptive locating strategy enables knowledge edits to better align with the model's inherent hierarchical representation, thereby improving editing precision and enhancing generalization across diverse types of factual updates.

Weight Modulation. In the probing stage, we perform weight modulation as a parameter-efficient approach to assess the importance of different linear layers within the FFNs across various layers of LLMs. Specifically, for each layer in the large language model, we represent the core information of the feedforward network by the weight matrix of its i-th linear transformation, denoted as $\boldsymbol{W}_i \in \mathbb{R}^{m\times n}$. We then modulate $\boldsymbol{W}_i$ by element-wise multiplying it with a modulation matrix ΔW_i added to an all-ones matrix $\mathbb{I}$, formulated as

$$\tilde{\boldsymbol{W}}_i = \boldsymbol{W}_i \odot (\mathbb{I} + \Delta W_i), \tag{2}$$

where $\odot$ denotes Hadamard multiplication. The inclusion of $\mathbb{I}$ enables the modulation matrix to be learned in a residual manner. Consequently, the modulation parameters are optimized as perturbations around the pretrained weights, which facilitates the preservation of existing knowledge. Moreover, the original pretrained weights can remain unchanged if they are already optimal for the target knowledge.

In this manner, this approach can lead to parameter explosion, as linear layers in large language models contain a substantial number of parameters. To mitigate this issue, rather than directly learning the full modulation matrix, we learn a low-rank approximation of it. More specifically, rather than directly learning the full modulation matrix $\Delta W_i \in \mathbb{R}^{m\times n}$, we learn two proxy vectors, $\Delta^i_{\mathrm{H}} \in \mathbb{R}^m$ and $\Delta^i_{\mathrm{W}} \in \mathbb{R}^n$. The modulation matrix is then constructed via the outer product of these vectors, *i.e.*, $\Delta W_i = \Delta^i_{\mathrm{H}} \otimes \Delta^i_{\mathrm{W}}$. Since the trainable parameters are limited to the proxy vectors, this formulation substantially reduces the number of trainable parameters.

Importance Measurement. In the probing stage, we employ a parameter-efficient technique to assess the importance of individual linear layers. Specifically, we apply weight modulation to all linear layers and identify which modulated weight matrices are most critical for the editing task. To quantify the importance of each modulated weight matrix, we compute the Fisher Information (FI) with respect to the corresponding modulation parameters. Specifically, for a modulated LLM with parameters Θ, Fisher information $\mathcal{F}$ can be computed as:

$$\mathcal{F}(\Theta) = -\frac{\partial^2}{\partial \Theta^2}\mathcal{L}(x, y|\Theta), \tag{3}$$

where $\mathcal{L}(x, y|\Theta)$ denotes the cross-entropy loss. Then, the FI of a modulation matrix $\mathcal{F}(\Delta W_i)$ is

computed by averaging the FI values of the parameters within the matrix. Since we adopt a low-rank formulation to construct the modulation matrix, $\mathcal{F}(\Delta W_i)$ can be efficiently estimated using the FI values of the corresponding proxy vectors. In particular, under the low-rank formulation based on the outer product, we estimate the Fisher Information of the modulation matrix by taking the unweighted average of the FI values associated with the parameters of Δ^i_{H} and Δ^i_{W}. This averaging is proportional to their respective occurrence frequencies in the construction of ΔW_i, yielding an estimate of $\mathcal{F}(\Delta W_i)$ as follows:

$$\hat{\mathcal{F}}(\Delta W_i) = \sum_{j=1}^{M} \left(\mathcal{F}(\Delta^i_{\mathrm{H},j}) + \frac{1}{N} \sum_{k=1}^{N} \mathcal{F}(\Delta^i_{\mathrm{W},k}) \right), \tag{4}$$

where $\Delta^i_{,j}$ and $\Delta^i_{\mathrm{W},k}$ denote the j-th element of Δ^i_{H} and the k-th element of Δ^i_{W}, respectively. More details can be seen in the supplementary material.

Editing within Adaptive Locating. Based on the estimated FI scores $\hat{\mathcal{F}}(\Delta W_i)$, the weight matrices are ranked in descending order, and the top K weight matrices are selected as the most important ones, denoted as $\mathcal{W}_K$. To incorporate new knowledge $(\boldsymbol{k}^*, \boldsymbol{v}^*)$ into the key–value memory, we solve a constrained least-squares problem over the selected set of weight matrices $\mathcal{W}_K$ as follows:

$$\begin{aligned} \min_{\boldsymbol{W}_i \in \mathcal{W}_K} & \|\boldsymbol{W}_i \boldsymbol{K} - \boldsymbol{V}\|_2^2, \\ \text{s.t.} \quad & \boldsymbol{W}_i \boldsymbol{k}^* = \boldsymbol{v}^*. \end{aligned} \tag{5}$$

The final parameter updates can be computed using existing editing methods, including closed-form solutions (e.g., ROME, MEMIT, and AlphaEdit) and gradient-based optimization approaches (e.g., UnKE).

During the probing stage, only the modulation parameters Δ^i_{H} and Δ^i_{W} are trainable, and the FI is computed exclusively with respect to these parameters. Consequently, the probing procedure is computationally lightweight and can be performed with minimal overhead. The output of the probing stage consists of layer-wise decisions on whether to apply editing or to freeze individual linear layers. Based on these decisions, the main editing phase is subsequently conducted. The overall proposed framework is summarized in Algorithm 1.

Algorithm 1 Fisher-Driven Adaptive Locating

Require: Pretrained model parameters Θ, , number of iterations for probing T, thredshold K, learning rate η.

Probing Stage:

Freeze all weight matrices $\{\boldsymbol{W}_i\}_{i=1}^{L}$ in LLMs
Randomly initialize a modulation matrix ΔW_i for each weight matrix $\boldsymbol{W}_i$
Let $t = 1$
while $t \leq T$ **do**
    Perform weight modulation for all weight matrices using Eq.(2) to obtain correponding modulated weights $\{\tilde{\boldsymbol{W}}_i\}_{i=1}^{L}$
    $\Delta W_i \leftarrow \Delta W_i - \eta \nabla_{\Delta W_i} \mathcal{L}(x, y; \tilde{\boldsymbol{W}}_i)$
    $t = t + 1$
end while
Measure importance of each weight $\boldsymbol{W}_i$ by computing FI for the corresponding ΔW_i using Eq.(4)
Select the top K important weight matrices $\mathcal{W}_K$ according to the FI values

Editing Stage:

Perform knowleding editing within $\mathcal{W}_K$ by solving Eq.(5)

4 Experiments

In this section, we evaluate our proposed FiDAL and analyze its essential characteristics.

4.1 Experimental Setup

LLMs and Baseline Methods. To comprehensively assess the performance of our proposed model, we evaluate it on three large language models (LLMs) with diverse architectural designs: GPT-J (Wang and Komatsuzaki, 2021), Qwen2.5-7B-Instruct (Yang et al., 2025), and LLaMA3-8B-Instruct (Dubey et al., 2024). For comparative analysis, we benchmark our approach against a range of representative knowledge editing methods, including Fine-Tuning (FT), LoRA (Wu et al., 2023), Knowledge Neurons (KN) (Dai et al., 2021), ROME (Meng et al., 2022a), MEMIT (Meng et al., 2022b), PMET (Li et al., 2024), UnKE (Deng et al., 2025), AlphaEdit (Fang et al., 2024), and NSE (Jiang et al., 2024).

Datasets. We conduct experiments on three datasets that cover both knowledge insertion and knowledge modification scenarios: WikiData$_{Counterfact}$ (Cohen et al., 2024), WikiData$_{Recent}$ (Cohen et al., 2024), and

LLMs	Method	Edit Succ.	Portability ↑			Locality ↑		Fluency
		(↑)	SAA	LGA	RA	RSA	FA	(↑)
GPT-J	FT	$64.2_{\pm 1.6}$	$47.3_{\pm 2.0}$	$7.1_{\pm 1.9}$	$21.3_{\pm 2.9}$	$4.4_{\pm 0.6}$	$6.4_{\pm 1.3}$	$304.1_{\pm 7.6}$
	LoRA	$\mathbf{100.0}_{\pm 0.0}$	$75.2_{\pm 1.9}$	$\mathbf{22.2}_{\pm 3.1}$	$\mathbf{40.3}_{\pm 2.8}$	$25.7_{\pm 1.6}$	$51.4_{\pm 2.8}$	$595.8_{\pm 4.1}$
	KN	$18.1_{\pm 2.4}$	$17.9_{\pm 2.4}$	$10.8_{\pm 2.6}$	$18.5_{\pm 2.2}$	$80.2^{*}_{\pm 1.3}$	$80.6^{*}_{\pm 1.5}$	$580.0_{\pm 3.8}$
	ROME	$99.2_{\pm 0.5}$	$74.1_{\pm 2.2}$	$16.1_{\pm 2.6}$	$29.2_{\pm 2.4}$	$37.4_{\pm 1.3}$	$33.1_{\pm 2.6}$	$\underline{600.0}_{\pm 3.6}$
	MEMIT	$99.5_{\pm 0.5}$	$56.5_{\pm 2.5}$	$16.7_{\pm 2.6}$	$25.9_{\pm 2.1}$	$53.2_{\pm 1.4}$	$40.7_{\pm 2.8}$	$591.6_{\pm 4.3}$
	PMET	$95.3_{\pm 0.9}$	$54.1_{\pm 2.6}$	$16.6_{\pm 2.6}$	$25.3_{\pm 2.1}$	$47.6_{\pm 1.5}$	$36.8_{\pm 2.8}$	$\mathbf{600.3}_{\pm 3.6}$
	UnKE	$99.6_{\pm 0.9}$	$86.9_{\pm 1.7}$	$17.1_{\pm 2.3}$	$36.1_{\pm 2.5}$	$51.7_{\pm 1.3}$	$40.1_{\pm 2.1}$	$591.2_{\pm 6.8}$
	AlphaEdit	$98.7_{\pm 0.5}$	$88.1_{\pm 2.5}$	$17.3_{\pm 2.1}$	$35.5_{\pm 2.5}$	$79.2_{\pm 1.4}$	$55.7_{\pm 1.6}$	$592.7_{\pm 4.8}$
	NSE	$99.1_{\pm 0.3}$	$\underline{90.3}_{\pm 1.9}$	$17.4_{\pm 2.0}$	$36.8_{\pm 2.7}$	$73.6_{\pm 1.5}$	$51.8_{\pm 2.2}$	$584.8_{\pm 5.3}$
	Ours ($K=1$)	$\underline{99.7}_{\pm 0.2}$	$90.2_{\pm 1.7}$	$18.2_{\pm 2.3}$	$37.7_{\pm 3.1}$	$\mathbf{80.8}_{\pm 1.8}$	$\mathbf{62.2}_{\pm 2.5}$	$599.2_{\pm 6.6}$
	Ours ($K=2$)	$\underline{99.7}_{\pm 0.1}$	$\mathbf{90.5}_{\pm 1.8}$	$\underline{19.0}_{\pm 3.1}$	$\underline{38.2}_{\pm 3.5}$	$\underline{80.3}_{\pm 1.6}$	$\underline{61.7}_{\pm 2.2}$	$593.5_{\pm 7.2}$
Qwen2.5-7B	FT	$49.0_{\pm 1.5}$	$46.3_{\pm 2.1}$	$15.3_{\pm 1.4}$	$29.3_{\pm 1.5}$	$21.6_{\pm 1.5}$	$30.1_{\pm 3.5}$	$493.8_{\pm 8.5}$
	LoRA	$\mathbf{100.0}_{\pm 0.0}$	$\mathbf{91.5}_{\pm 1.0}$	$\mathbf{31.4}_{\pm 3.5}$	$\mathbf{46.1}_{\pm 2.1}$	$71.2_{\pm 1.2}$	$50.1_{\pm 2.5}$	$564.2_{\pm 5.3}$
	KN	$20.5_{\pm 2.9}$	$22.1_{\pm 2.6}$	$18.9_{\pm 2.4}$	$26.4_{\pm 1.9}$	$79.2^{*}_{\pm 1.6}$	$71.2^{*}_{\pm 4.6}$	$568.6_{\pm 6.2}$
	ROME	$98.9_{\pm 0.4}$	$73.7_{\pm 2.0}$	$19.2_{\pm 3.5}$	$36.2_{\pm 2.7}$	$49.1_{\pm 1.5}$	$38.6_{\pm 2.4}$	$579.8_{\pm 3.9}$
	MEMIT	$98.2_{\pm 0.8}$	$78.3_{\pm 2.2}$	$26.7_{\pm 2.7}$	$39.3_{\pm 2.5}$	$47.2_{\pm 1.3}$	$42.7_{\pm 2.4}$	$575.8_{\pm 4.2}$
	PMET	$96.2_{\pm 1.1}$	$58.6_{\pm 2.5}$	$28.1_{\pm 3.2}$	$32.5_{\pm 2.6}$	$60.1_{\pm 1.9}$	$51.2_{\pm 2.9}$	$577.4_{\pm 5.2}$
	UnKE	$99.4_{\pm 0.8}$	$76.8_{\pm 1.8}$	$27.1_{\pm 2.4}$	$35.3_{\pm 2.9}$	$55.3_{\pm 1.8}$	$48.6_{\pm 2.1}$	$588.4_{\pm 5.9}$
	AlphaEdit	$98.9_{\pm 0.4}$	$75.2_{\pm 2.4}$	$26.9_{\pm 2.7}$	$36.7_{\pm 2.7}$	$\underline{72.9}_{\pm 3.2}$	$\underline{60.8}_{\pm 1.7}$	$576.1_{\pm 3.8}$
	NSE	$99.0_{\pm 0.6}$	$81.3_{\pm 2.8}$	$27.7_{\pm 1.8}$	$37.5_{\pm 2.3}$	$61.8_{\pm 1.2}$	$53.7_{\pm 2.3}$	$561.2_{\pm 5.4}$
	Ours ($K=1$)	$99.4_{\pm 0.1}$	$82.1_{\pm 1.7}$	$28.4_{\pm 2.4}$	$40.1_{\pm 3.1}$	$\mathbf{73.5}_{\pm 2.1}$	$\mathbf{61.1}_{\pm 2.4}$	$\mathbf{591.7}_{\pm 5.6}$
	Ours ($K=2$)	$\underline{99.5}_{\pm 0.1}$	$\underline{82.7}_{\pm 2.1}$	$\underline{28.9}_{\pm 2.6}$	$\underline{40.4}_{\pm 2.5}$	$72.2_{\pm 2.5}$	$58.3_{\pm 2.9}$	$\underline{589.4}_{\pm 6.3}$
Llama3-8B	FT	$47.2_{\pm 1.9}$	$48.6_{\pm 1.6}$	$8.4_{\pm 1.6}$	$25.6_{\pm 2.0}$	$27.1_{\pm 1.8}$	$12.0_{\pm 1.7}$	$379.2_{\pm 11.7}$
	LoRA	$\mathbf{100.0}_{\pm 0.0}$	$78.1_{\pm 1.5}$	$\underline{24.4}_{\pm 3.6}$	$\mathbf{44.2}_{\pm 3.1}$	$15.6_{\pm 1.0}$	$24.9_{\pm 2.4}$	$471.8_{\pm 10.1}$
	KN	$16.8_{\pm 2.0}$	$18.2_{\pm 2.0}$	$14.6_{\pm 2.4}$	$19.8_{\pm 2.0}$	$83.7^{*}_{\pm 1.0}$	$88.2^{*}_{\pm 2.2}$	$591.2_{\pm 5.9}$
	ROME	$99.2_{\pm 0.2}$	$74.1_{\pm 2.2}$	$16.1_{\pm 2.6}$	$29.2_{\pm 2.4}$	$37.1_{\pm 1.7}$	$33.2_{\pm 2.3}$	$590.1_{\pm 3.6}$
	MEMIT	$99.2_{\pm 0.4}$	$73.5_{\pm 2.7}$	$\mathbf{24.5}_{\pm 2.2}$	$32.1_{\pm 2.3}$	$41.2_{\pm 1.5}$	$41.0_{\pm 2.5}$	$568.9_{\pm 6.9}$
	PMET	$97.2_{\pm 0.8}$	$55.9_{\pm 2.7}$	$24.1_{\pm 2.4}$	$34.3_{\pm 2.1}$	$44.7_{\pm 2.1}$	$33.8_{\pm 2.1}$	$\mathbf{598.6}_{\pm 3.2}$
	UnKE	$99.7_{\pm 0.8}$	$85.8_{\pm 2.3}$	$17.1_{\pm 2.3}$	$35.1_{\pm 2.5}$	$48.6_{\pm 1.7}$	$44.1_{\pm 2.0}$	$589.7_{\pm 5.7}$
	AlphaEdit	$98.9_{\pm 0.4}$	$82.4_{\pm 2.1}$	$18.8_{\pm 2.7}$	$36.9_{\pm 3.0}$	$60.8_{\pm 1.9}$	$53.9_{\pm 1.5}$	$575.4_{\pm 4.4}$
	NSE	$99.2_{\pm 0.5}$	$87.2_{\pm 2.0}$	$21.4_{\pm 2.0}$	$29.1_{\pm 2.2}$	$\underline{73.6}_{\pm 1.5}$	$51.8_{\pm 2.2}$	$584.8_{\pm 5.3}$
	Ours ($K=1$)	$\underline{99.8}_{\pm 0.1}$	$\underline{89.3}_{\pm 1.7}$	$22.4_{\pm 2.2}$	$37.8_{\pm 2.2}$	$\mathbf{74.1}_{\pm 1.6}$	$\mathbf{59.3}_{\pm 1.8}$	$\underline{595.6}_{\pm 3.9}$
	Ours ($K=2$)	$99.7_{\pm 0.3}$	$\mathbf{90.3}_{\pm 1.8}$	$23.6_{\pm 2.6}$	$\underline{39.4}_{\pm 2.5}$	$73.5_{\pm 1.8}$	$\underline{58.8}_{\pm 2.1}$	$592.4_{\pm 4.3}$

Table 1: Editing Performance comparison on WikiData$_{Counterfact}$. * indicates invalid results. Locality results obtained under conditions of low Edit Success are deemed invalid, as locality trivially reaches 100% when the edit is not successfully applied. The best results are indicated as **Bold**, and the second best are indicated as Underline.

ZsRE (Levy et al., 2017).

Evaluation Metrics. To ensure a thorough and systematic evaluation of knowledge editing methods, we follow established evaluation protocols and adopt four metrics: Edit Success, Portability, Locality, and Fluency. The Portability metric is further divided into Subject Aliasing Accuracy (SAA), Logical Generalization Accuracy (LGA), and Reasoning Accuracy (RA). Specifically, SAA measures the model's ability to generalize edited knowledge to alternative expressions of the subject, such as aliases or synonyms. LGA evaluates whether semantically related facts that should be affected by the edit are updated accordingly, while RA assesses the model's reasoning capability based on the modified knowledge. The Locality metric consists of Forgetfulness Accuracy (FA) and Relation Specificity Accuracy (RSA). FA examines whether the model forgets only the targeted knowledge while preserving other facts in one-to-many relations, whereas RSA evaluates whether unrelated attributes of the edited subject remain unchanged after the knowledge update.

4.2 Experimental Results

We present a quantitative evaluation of knowledge editing on the WikiData$_{Counterfact}$, WikiData$_{Recent}$, and ZsRE benchmarks, reporting the experimental results in Tables 1, 2, and 3. With η set to 0.0001, our method achieves superior performance in Edit Success, Portability, and Locality compared to existing approaches. We attribute the suboptimal performance of prior methods to the rigidity inherent in fixed layer-locating strategies. To overcome this limitation and enable more effective adaptation, our approach leverages a Fisher-driven adaptive locating strategy. Collectively, these results indicate that our method

LLMs	Method	Edit Succ.	Portability ↑			Locality ↑		Fluency
		(↑)	SAA	LGA	RA	RSA	FA	(↑)
GPT-J	FT	$70.8_{\pm 1.5}$	$50.1_{\pm 2.7}$	$18.3_{\pm 1.9}$	$35.3_{\pm 2.0}$	$7.1_{\pm 0.3}$	$8.4_{\pm 1.8}$	$351.8_{\pm 6.2}$
	LoRA	$\mathbf{100.0}_{\pm 0.0}$	$81.6_{\pm 1.2}$	$\mathbf{35.2}_{\pm 2.8}$	$\mathbf{48.6}_{\pm 2.8}$	$28.3_{\pm 1.7}$	$22.4_{\pm 3.1}$	$591.8_{\pm 3.8}$
	KN	$28.3_{\pm 3.1}$	$22.6_{\pm 3.2}$	$23.3_{\pm 3.1}$	$35.1_{\pm 1.6}$	$86.6^{*}_{\pm 1.3}$	$81.4^{*}_{\pm 1.2}$	$579.6_{\pm 3.4}$
	ROME	$99.5_{\pm 0.2}$	$84.6_{\pm 2.0}$	$28.3_{\pm 2.8}$	$36.9_{\pm 1.7}$	$37.3_{\pm 1.3}$	$51.0_{\pm 2.2}$	$\mathbf{596.8}_{\pm 2.8}$
	MEMIT	$99.6_{\pm 0.2}$	$68.9_{\pm 3.2}$	$27.2_{\pm 2.6}$	$32.4_{\pm 1.9}$	$49.6_{\pm 1.0}$	$52.7_{\pm 1.9}$	$585.1_{\pm 3.2}$
	PMET	$99.0_{\pm 0.4}$	$63.6_{\pm 3.6}$	$25.4_{\pm 2.8}$	$31.2_{\pm 2.0}$	$46.3_{\pm 1.0}$	$49.5_{\pm 2.4}$	$584.2_{\pm 3.0}$
	UnKE	$99.5_{\pm 0.3}$	$87.6_{\pm 1.8}$	$26.4_{\pm 2.4}$	$34.8_{\pm 1.8}$	$56.8_{\pm 0.7}$	$48.9_{\pm 2.3}$	$589.4_{\pm 3.8}$
	AlphaEdit	$99.1_{\pm 0.4}$	$67.4_{\pm 2.5}$	$28.9_{\pm 1.8}$	$38.6_{\pm 2.5}$	$68.9_{\pm 1.2}$	$54.2_{\pm 2.5}$	$578.3_{\pm 3.9}$
	NSE	$\underline{99.8}_{\pm 0.5}$	$\mathbf{94.2}_{\pm 1.2}$	$25.6_{\pm 2.3}$	$40.1_{\pm 1.7}$	$\mathbf{73.6}_{\pm 1.5}$	$51.8_{\pm 2.2}$	$565.4_{\pm 5.3}$
	Ours ($K=1$)	$\underline{99.8}_{\pm 0.4}$	$91.8_{\pm 1.5}$	$\underline{29.5}_{\pm 2.2}$	$\underline{43.1}_{\pm 1.8}$	$\underline{69.8}_{\pm 2.1}$	$\mathbf{56.8}_{\pm 2.1}$	$590.3_{\pm 3.2}$
	Ours ($K=2$)	$\underline{99.8}_{\pm 0.2}$	$\underline{92.5}_{\pm 1.8}$	$29.1_{\pm 2.5}$	$\underline{43.1}_{\pm 3.2}$	$69.4_{\pm 1.8}$	$\underline{56.4}_{\pm 2.1}$	$\underline{594.1}_{\pm 3.9}$
Qwen2.5-7B	FT	$53.4_{\pm 1.7}$	$48.8_{\pm 1.9}$	$13.7_{\pm 1.8}$	$32.1_{\pm 1.7}$	$25.7_{\pm 1.1}$	$21.5_{\pm 2.4}$	$415.9_{\pm 6.7}$
	LoRA	$\mathbf{100.0}_{\pm 0.0}$	$89.1_{\pm 0.7}$	$32.8_{\pm 3.1}$	$41.6_{\pm 2.1}$	$65.1_{\pm 1.0}$	$41.6_{\pm 1.8}$	$571.1_{\pm 6.8}$
	KN	$21.6_{\pm 2.3}$	$26.2_{\pm 2.4}$	$21.5_{\pm 1.8}$	$25.7_{\pm 2.3}$	$81.3^{*}_{\pm 1.4}$	$67.3^{*}_{\pm 3.8}$	$566.3_{\pm 4.2}$
	ROME	$98.8_{\pm 0.6}$	$81.4_{\pm 1.6}$	$36.7_{\pm 2.2}$	$44.1_{\pm 1.8}$	$50.3_{\pm 1.4}$	$58.4_{\pm 1.7}$	$\underline{573.8}_{\pm 2.8}$
	MEMIT	$99.2_{\pm 0.3}$	$86.2_{\pm 1.4}$	$39.3_{\pm 3.4}$	$43.1_{\pm 1.6}$	$51.2_{\pm 1.0}$	$58.3_{\pm 1.8}$	$567.8_{\pm 3.4}$
	PMET	$97.6_{\pm 0.3}$	$69.3_{\pm 1.8}$	$36.6_{\pm 2.7}$	$46.8_{\pm 1.9}$	$61.3_{\pm 1.7}$	$65.3_{\pm 2.1}$	$571.8_{\pm 2.6}$
	UnKE	$99.5_{\pm 1.0}$	$87.1_{\pm 2.4}$	$42.1_{\pm 2.1}$	$47.8_{\pm 2.2}$	$54.2_{\pm 1.7}$	$59.4_{\pm 1.9}$	$566.5_{\pm 3.1}$
	AlphaEdit	$99.1_{\pm 0.5}$	$84.2_{\pm 1.6}$	$38.3_{\pm 1.8}$	$45.3_{\pm 2.3}$	$72.8_{\pm 2.4}$	$67.8_{\pm 1.9}$	$573.3_{\pm 3.5}$
	NSE	$99.6_{\pm 0.8}$	$91.2_{\pm 2.1}$	$37.5_{\pm 1.6}$	$47.2_{\pm 1.5}$	$78.6_{\pm 1.6}$	$64.2_{\pm 2.1}$	$561.2_{\pm 2.3}$
	Ours ($K=1$)	$99.7_{\pm 0.3}$	$\mathbf{92.5}_{\pm 1.8}$	$\mathbf{43.8}_{\pm 1.5}$	$\mathbf{48.2}_{\pm 2.1}$	$\mathbf{81.3}_{\pm 1.4}$	$\underline{69.5}_{\pm 2.4}$	$\mathbf{576.0}_{\pm 3.2}$
	Ours ($K=2$)	$\underline{99.8}_{\pm 0.2}$	$\underline{92.1}_{\pm 2.0}$	$\underline{42.7}_{\pm 1.7}$	$\underline{47.9}_{\pm 1.8}$	$\underline{80.8}_{\pm 1.8}$	$\mathbf{74.2}_{\pm 2.4}$	$571.2_{\pm 3.8}$
Llama3-8B	FT	$51.3_{\pm 0.4}$	$49.2_{\pm 2.6}$	$10.5_{\pm 1.2}$	$25.6_{\pm 2.0}$	$31.0_{\pm 1.4}$	$18.2_{\pm 2.7}$	$431.9_{\pm 8.8}$
	LoRA	$\mathbf{100.0}_{\pm 0.0}$	$83.1_{\pm 2.0}$	$25.6_{\pm 3.1}$	$\underline{46.3}_{\pm 3.6}$	$16.2_{\pm 1.5}$	$21.3_{\pm 2.4}$	$491.8_{\pm 12.5}$
	KN	$20.3_{\pm 2.4}$	$17.3_{\pm 2.5}$	$17.3_{\pm 2.7}$	$20.3_{\pm 2.4}$	$81.1^{*}_{\pm 3.8}$	$86.5^{*}_{\pm 1.9}$	$589.7_{\pm 6.4}$
	ROME	$98.4_{\pm 0.4}$	$82.6_{\pm 1.7}$	$34.8_{\pm 2.9}$	$\mathbf{46.7}_{\pm 1.4}$	$49.3_{\pm 1.5}$	$52.1_{\pm 2.2}$	$581.3_{\pm 2.8}$
	MEMIT	$99.1_{\pm 0.4}$	$81.1_{\pm 2.5}$	$34.7_{\pm 2.5}$	$45.2_{\pm 1.5}$	$48.3_{\pm 2.1}$	$55.1_{\pm 1.8}$	$584.7_{\pm 2.6}$
	PMET	$98.1_{\pm 0.8}$	$58.7_{\pm 2.1}$	$37.3_{\pm 2.4}$	$42.5_{\pm 1.9}$	$65.8_{\pm 1.7}$	$64.0_{\pm 1.4}$	$591.8_{\pm 2.5}$
	UnKE	$\underline{99.8}_{\pm 0.8}$	$87.6_{\pm 1.5}$	$33.8_{\pm 3.5}$	$44.8_{\pm 1.9}$	$51.3_{\pm 1.4}$	$57.2_{\pm 2.3}$	$593.8_{\pm 3.3}$
	AlphaEdit	$98.9_{\pm 0.3}$	$82.3_{\pm 2.2}$	$35.7_{\pm 2.5}$	$43.9_{\pm 2.1}$	$\underline{75.3}_{\pm 1.8}$	$\mathbf{65.1}_{\pm 1.6}$	$584.4_{\pm 3.5}$
	NSE	$99.3_{\pm 0.6}$	$87.4_{\pm 2.3}$	$36.2_{\pm 3.0}$	$42.7_{\pm 1.9}$	$67.3_{\pm 1.7}$	$61.7_{\pm 2.0}$	$585.3_{\pm 2.9}$
	Ours ($K=1$)	$\underline{99.8}_{\pm 0.1}$	$\mathbf{90.1}_{\pm 1.6}$	$\mathbf{38.8}_{\pm 2.9}$	$45.8_{\pm 1.6}$	$\mathbf{81.2}_{\pm 1.9}$	$\underline{64.3}_{\pm 1.8}$	$\underline{596.4}_{\pm 2.9}$
	Ours ($K=2$)	$\underline{99.8}_{\pm 0.2}$	$\underline{89.1}_{\pm 2.0}$	$\underline{37.8}_{\pm 2.1}$	$45.1_{\pm 2.0}$	$80.6_{\pm 1.6}$	$62.9_{\pm 2.1}$	$\mathbf{597.5}_{\pm 2.4}$

Table 2: Editing Performance comparison on WikiData$_{Recent}$. * indicates invalid results. Locality results obtained under conditions of low Edit Success are deemed invalid, as locality trivially reaches 100% when the edit is not successfully applied. The best results are indicated as **Bold**, and the second best are indicated as Underline.

consistently outperforms competing baselines, thereby demonstrating its effectiveness.

4.3 Ablation Study

To more thoroughly evaluate the effectiveness of our method and to better demonstrate its advantages, we conduct a series of ablation studies on the WikiData$_{\text{Counterfact}}$ dataset using Qwen2.5-7B.

Performance Comparison with Different K Settings. To more comprehensively evaluate the effectiveness of our method, we compare its performance under different values of K with the results reported in Table 4. During the editing stage, we employ the gradient-based optimization strategy introduced in UnKE to solve the constrained least-squares problem formulated in Eq. (5). Overall, the performance remains relatively stable as K varies. However, as K increases, metrics related to knowledge preservation exhibit a gradual decline. This degradation can be attributed to the increasing number of fine-tuned layers and parameters, which weakens the preservation of existing knowledge. Moreover, larger values of K incur higher computational overhead. To balance computational cost and editing performance, we therefore choose $K = 1$ and $K = 2$ in our experiments.

Performance Comparison with Different Number of Iterations in Probing Stage T. To more comprehensively evaluate the effectiveness of our method, we compare its performance under different values of T, with the results presented in Figure 1. During the editing phase, we employ the gradient-based optimization strategy proposed in UnKE to solve the constrained least-squares problem defined in Eq. (5). When $T = 0$, our adaptive locating strategy is disabled, and the model reduces

LLMs	Method	Edit Succ.↑	Portability ↑			Locality ↑		Fluency ↑
			SAA	LGA	RA	RSA	FA	
GPT-J	ROME	99.6$_{\pm 0.2}$	40.0$_{\pm 4.2}$	46.4$_{\pm 3.1}$	50.2$_{\pm 1.7}$	47.1$_{\pm 1.5}$	-	573.7$_{\pm 5.0}$
	MEMIT	99.3$_{\pm 0.3}$	19.9$_{\pm 4.9}$	45.9$_{\pm 3.0}$	46.5$_{\pm 1.8}$	70.0$_{\pm 1.0}$	-	581.7$_{\pm 4.5}$
	PMET	96.6$_{\pm 0.8}$	16.5$_{\pm 5.2}$	43.6$_{\pm 3.3}$	48.7$_{\pm 1.7}$	65.3$_{\pm 1.3}$	-	586.9$_{\pm 3.4}$
	UnKE	99.6$_{\pm 0.7}$	45.6$_{\pm 4.6}$	46.2$_{\pm 2.8}$	49.3$_{\pm 2.0}$	50.5$_{\pm 1.4}$	-	582.3$_{\pm 5.1}$
	Alph	98.9$_{\pm 0.2}$	42.6$_{\pm 4.1}$	45.8$_{\pm 2.5}$	48.8$_{\pm 1.6}$	81.3$_{\pm 1.8}$	-	548.2$_{\pm 4.8}$
	NSE	99.3$_{\pm 0.4}$	45.7$_{\pm 4.8}$	47.5$_{\pm 3.5}$	49.1$_{\pm 1.9}$	73.6$_{\pm 1.5}$	-	584.8$_{\pm 5.3}$
	Ours ($K = 1$)	**99.7**$_{\pm 0.4}$	**49.2**$_{\pm 1.4}$	**52.1**$_{\pm 2.9}$	51.9$_{\pm 1.5}$	88.2$_{\pm 1.5}$	-	582.0$_{\pm 4.1}$
	Ours ($K = 2$)	**99.7**$_{\pm 0.5}$	48.9$_{\pm 1.4}$	51.9$_{\pm 2.5}$	**53.0**$_{\pm 1.8}$	**89.8**$_{\pm 1.3}$	-	**589.3**$_{\pm 3.1}$
Qwen2.5-7B	ROME	97.1$_{\pm 0.4}$	33.2$_{\pm 3.5}$	46.3$_{\pm 3.1}$	52.4$_{\pm 1.5}$	50.7$_{\pm 1.5}$	-	562.0$_{\pm 3.4}$
	MEMIT	94.8$_{\pm 1.2}$	32.7$_{\pm 3.8}$	43.9$_{\pm 3.9}$	53.8$_{\pm 1.6}$	47.9$_{\pm 1.8}$	-	539.7$_{\pm 4.0}$
	PMET	91.7$_{\pm 2.0}$	26.8$_{\pm 4.0}$	46.7$_{\pm 3.3}$	57.2$_{\pm 1.5}$	68.1$_{\pm 1.3}$	-	562.5$_{\pm 3.4}$
	UnKE	99.5$_{\pm 0.6}$	48.9$_{\pm 3.6}$	55.7$_{\pm 1.5}$	51.9$_{\pm 1.4}$	60.3$_{\pm 1.0}$	-	580.8$_{\pm 4.7}$
	Alph	99.1$_{\pm 0.4}$	46.7$_{\pm 3.5}$	52.7$_{\pm 3.1}$	51.7$_{\pm 1.5}$	81.7$_{\pm 1.6}$	-	571.0$_{\pm 4.3}$
	NSE	99.2$_{\pm 0.5}$	47.1$_{\pm 4.1}$	50.2$_{\pm 3.0}$	50.4$_{\pm 1.7}$	83.8$_{\pm 1.4}$	-	575.5$_{\pm 5.6}$
	Ours ($K = 1$)	**99.6**$_{\pm 0.5}$	**54.2**$_{\pm 2.9}$	55.5$_{\pm 3.1}$	56.6$_{\pm 1.5}$	**85.3**$_{\pm 1.5}$	-	578.3$_{\pm 3.8}$
	Ours ($K = 2$)	**99.6**$_{\pm 0.6}$	51.0$_{\pm 3.8}$	**56.0**$_{\pm 3.4}$	**57.5**$_{\pm 1.6}$	84.8$_{\pm 1.3}$	-	**581.1**$_{\pm 3.5}$
Llama3-8B	ROME	98.2$_{\pm 0.6}$	43.2$_{\pm 3.7}$	46.1$_{\pm 2.3}$	**57.6**$_{\pm 1.8}$	49.5$_{\pm 1.6}$	-	547.6$_{\pm 6.6}$
	MEMIT	96.5$_{\pm 0.6}$	46.7$_{\pm 3.5}$	48.4$_{\pm 2.9}$	50.1$_{\pm 2.7}$	52.3$_{\pm 1.7}$	-	511.4$_{\pm 6.2}$
	PMET	97.5$_{\pm 0.7}$	27.2$_{\pm 3.6}$	45.2$_{\pm 2.5}$	54.2$_{\pm 1.7}$	67.5$_{\pm 1.8}$	-	568.3$_{\pm 3.7}$
	UnKE	99.7$_{\pm 0.4}$	49.6$_{\pm 3.1}$	53.8$_{\pm 3.8}$	52.1$_{\pm 2.3}$	68.1$_{\pm 1.5}$	-	571.3$_{\pm 5.5}$
	Alph	99.1$_{\pm 0.1}$	45.8$_{\pm 3.8}$	51.3$_{\pm 3.2}$	52.7$_{\pm 1.8}$	81.2$_{\pm 1.8}$	-	549.3$_{\pm 4.1}$
	NSE	99.4$_{\pm 0.4}$	52.8$_{\pm 4.2}$	53.9$_{\pm 3.3}$	55.6$_{\pm 2.2}$	79.5$_{\pm 1.4}$	-	557.3$_{\pm 2.3}$
	Ours ($K = 1$)	99.8$_{\pm 0.6}$	57.7$_{\pm 3.8}$	**55.1**$_{\pm 2.6}$	55.4$_{\pm 2.0}$	**89.3**$_{\pm 2.0}$	-	570.8$_{\pm 4.7}$
	Ours ($K = 2$)	**99.9**$_{\pm 0.4}$	**59.6**$_{\pm 4.0}$	54.8$_{\pm 3.2}$	55.9$_{\pm 2.0}$	86.9$_{\pm 1.8}$	-	**579.3**$_{\pm 4.3}$

Table 3: Editing Performance comparison on ZsRE. The best results are indicated as **Bold**, and the second ones are indicated as Underline.

K	ES	SAA	RA	FA	Fluency
1	99.4	82.1	40.1	61.1	591.7
2	99.5	82.7	40.4	58.3	589.4
3	99.5	82.9	41.0	57.6	590.6
4	99.6	83.2	41.2	57.0	592.0
5	99.6	83.5	41.6	56.1	589.4

Table 4: Performance of the proposed method on the WikiData$_{\text{Counterfact}}$ dataset with Qwen2.5-7B under different values of K.

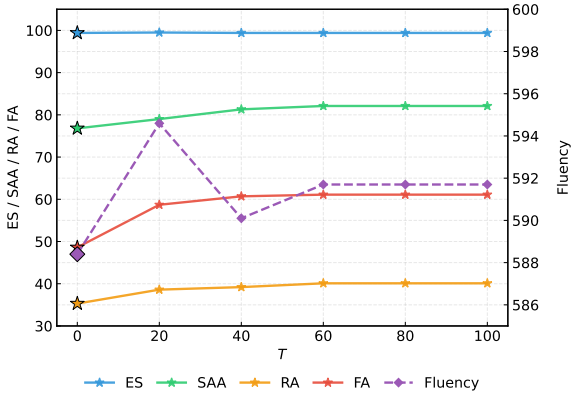

Figure 1: Performance of the proposed method on the WikiData$_{\text{Counterfact}}$ dataset with Qwen2.5-7B under different values of T.

to the original UnKE baseline. As shown in Figure 1, incorporating the proposed strategy consistently improves performance over UnKE. Moreover, performance gradually improves as the number of probing-stage iterations increases and begins to plateau after $T = 40$. However, larger values of T introduce higher computational overhead. To balance editing performance and computational cost, we therefore set $T = 60$ in all experiments.

Performance Comparison with Different Editing Methods. Since our method introduces only an additional layer-locating step, it is readily compatible with a wide range of existing model editing methods. To verify the effectiveness of the proposed adaptive locating strategy, we evaluate our approach integrated with several representative editing methods, with the results reported in Table 5. To ensure compatibility across different methods, all ablation studies are conducted under the setting $K = 1$. As shown in Table 4, incorporating our adaptive locating strategy consistently improves the performance of the original editing methods, demonstrating both the effectiveness and the general compatibility of our approach.

Method	ES	SAA	RA	FA	Fluency
MEMIT	98.2	78.3	39.3	42.7	575.8
+FiDAL	98.9	83.0	43.2	52.8	581.6
Unke	99.4	76.8	35.3	48.6	588.4
+FiDAL	99.4	82.1	40.1	61.1	591.7
AlphaEdit	98.9	75.2	36.7	60.8	576.1
+FiDAL	99.2	81.4	38.8	63.4	583.5

Table 5: Performance of the proposed method on the $\text{WikiData}_{\text{Counterfact}}$ dataset with Qwen2.5-7B with different editing methods.

Figure 2: Comparison of the runtime and memory consumption of each method.

Comparison and Analysis of the Runtime and Memory Consumption. We report the runtime of each knowledge editing method in Figure 2. Compared to fixed-layer locating strategies, our adaptive locating approach incurs additional computational overhead. However, this overhead is relatively small in comparison to the computational cost of the knowledge editing phase itself. Moreover, knowledge editing is not inherently time-sensitive, and moderate increases in computation time are generally acceptable in practical settings. Nonetheless, further reducing the computational footprint of our method remains an important direction for future work.

5 Conclusion

We presented FiDAL, a Fisher-driven adaptation-aware locating strategy for knowledge editing in large language models. By revisiting the locating stage and treating it as an adaptive, instance-dependent decision process, FiDAL dynamically identifies which model components should be edited for a given factual update. Leveraging Fisher Information as a principled measure of parameter sensitivity, our approach enables precise and lightweight localization while remaining fully compatible with existing locate-then-edit methods. Experimental results on standard benchmarks demonstrate consistent improvements in editing effectiveness and knowledge preservation across multiple editing frameworks.

Limitations

Our approach focuses exclusively on weight-level modulation within the FFNs. While FFNs have been shown to play a central role in factual knowledge storage, other components of transformer architectures, such as attention layers or embeddings, may also contribute to knowledge representation in certain cases. Extending FiDAL to these components is left for future work. Moreover, although the probing stage is lightweight due to low-rank parameterization, it introduces additional computation compared to methods with fixed locating strategies.

Acknowledgments

Our work is supported in part by the National Natural Science Foundation of China (U25B2048, 62132016).

References

Baolong Bi, Shenghua Liu, Lingrui Mei, Yiwei Wang, Pengliang Ji, and Xueqi Cheng. 2024. Decoding by contrasting knowledge: Enhancing llms' confidence on edited facts. *arXiv preprint arXiv:2405.11613*.

Jinhe Bi, Yifan Wang, Danqi Yan, Xun Xiao, Artur Hecker, Volker Tresp, and Yunpu Ma. 2025a. Prism: Self-pruning intrinsic selection method for training-free multimodal data selection. *Preprint*, arXiv:2502.12119.

Jinhe Bi, Yujun Wang, Haokun Chen, Xun Xiao, Artur Hecker, Volker Tresp, and Yunpu Ma. 2025b. Llava steering: Visual instruction tuning with 500x fewer parameters through modality linear representation-steering. In *ACL*.

Jinhe Bi, Danqi Yan, Yifan Wang, Wenke Huang, Haokun Chen, Guancheng Wan, Mang Ye, Xun Xiao, Hinrich Schuetze, Volker Tresp, and 1 others. 2025c. Cot-kinetics: A theoretical modeling assessing lrm reasoning process. *arXiv preprint arXiv:2505.13408*.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, and 1 others. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Yung-Sung Chuang, Yujia Xie, Hongyin Luo, Yoon Kim, James R Glass, and Pengcheng He. 2023. Dola:

Decoding by contrasting layers improves factuality in large language models. In *The Twelfth International Conference on Learning Representations*.

Roi Cohen, Eden Biran, Ori Yoran, Amir Globerson, and Mor Geva. 2024. Evaluating the ripple effects of knowledge editing in language models. *Transactions of the Association for Computational Linguistics*, 12:283–298.

Damai Dai, Li Dong, Yaru Hao, Zhifang Sui, Baobao Chang, and Furu Wei. 2021. Knowledge neurons in pretrained transformers. *arXiv preprint arXiv:2104.08696*.

Nicola De Cao, Wilker Aziz, and Ivan Titov. 2021. Editing factual knowledge in language models. *arXiv preprint arXiv:2104.08164*.

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 *ICLR*.

Qingxiu Dong, Damai Dai, Yifan Song, Jingjing Xu, Zhifang Sui, and Lei Li. 2022. Calibrating factual knowledge in pretrained language models. *arXiv preprint arXiv:2210.03329*.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, and 1 others. 2024. The llama 3 herd of models. *arXiv e-prints*, pages arXiv–2407.

Junfeng Fang, Houcheng Jiang, Kun Wang, Yunshan Ma, Shi Jie, Xiang Wang, Xiangnan He, and Tat-Seng Chua. 2024. Alphaedit: Null-space constrained knowledge editing for language models. *arXiv preprint arXiv:2410.02355*.

Tom Hartvigsen, Swami Sankaranarayanan, Hamid Palangi, Yoon Kim, and Marzyeh Ghassemi. 2023. Aging with grace: Lifelong model editing with discrete key-value adaptors. *Advances in Neural Information Processing Systems*, 36:47934–47959.

Xiusheng Huang, Yequan Wang, Jun Zhao, and Kang Liu. 2024. Commonsense knowledge editing based on free-text in llms. *arXiv preprint arXiv:2410.23844*.

Zeyu Huang, Yikang Shen, Xiaofeng Zhang, Jie Zhou, Wenge Rong, and Zhang Xiong. 2023. Transformer-patcher: One mistake worth one neuron. *arXiv preprint arXiv:2301.09785*.

Houcheng Jiang, Junfeng Fang, Ningyu Zhang, Guojun Ma, Mingyang Wan, Xiang Wang, Xiangnan He, and Tat-seng Chua. 2025. Anyedit: Edit any knowledge encoded in language models. *arXiv preprint arXiv:2502.05628*.

Houcheng Jiang, Junfeng Fang, Tianyu Zhang, An Zhang, Ruipeng Wang, Tao Liang, and Xiang Wang. 2024. Neuron-level sequential editing for large language models. *arXiv preprint arXiv:2410.04045*.

Omer Levy, Minjoon Seo, Eunsol Choi, and Luke Zettlemoyer. 2017. Zero-shot relation extraction via reading comprehension. *arXiv preprint arXiv:1706.04115*.

Xiaopeng Li, Shasha Li, Shezheng Song, Jing Yang, Jun Ma, and Jie Yu. 2024. Pmet: Precise model editing in a transformer. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 18564–18572.

Tian Liang, Yuetian Du, Jing Huang, Ming Kong, Luyuan Chen, Yadong Li, Siye Chen, and Qiang Zhu. 2025. Mole: Decoding by mixture of layer experts alleviates hallucination in large vision-language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 18684–18692.

Wei Liu, Haomei Xu, Bingqing Liu, Zhiying Deng, Haozhao Wang, Jun Wang, Ruixuan Li, Yee Whye Teh, and Wee Sun Lee. 2025. Is model editing built on sand? revealing its illusory success and fragile foundation.

Wei Liu, Haomei Xu, Hongkai Liu, Zhiying Deng, Ruixuan Li, Heng Huang, Yee Whye Teh, and Wee Sun Lee. 2026. Are we evaluating the edit locality of llm model editing properly? *arXiv preprint arXiv:2601.17343*.

Guangtao Lyu, Xinyi Cheng, Qi Liu, Chenghao Xu, Jiexi Yan, Muli Yang, Fen Fang, and Cheng Deng. 2026a. Towards interpretable hallucination analysis and mitigation in lvlms via contrastive neuron steering. *arXiv preprint arXiv:2602.00621*.

Guangtao Lyu, Xinyi Cheng, Chenghao Xu, Qi Liu, Muli Yang, Fen Fang, Huilin Chen, Jiexi Yan, Xu Yang, and Cheng Deng. 2025. Revealing perception and generation dynamics in lvlms: Mitigating hallucinations via validated dominance correction. *arXiv preprint arXiv:2512.18813*.

Guangtao Lyu, Qi Liu, Chenghao Xu, Jiexi Yan, Muli Yang, Xueting Li, Fen Fang, and Cheng Deng. 2026b. Revealing and enhancing core visual regions: Harnessing internal attention dynamics for hallucination mitigation in lvlms. *arXiv preprint arXiv:2602.15556*.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. 2022a. Locating and editing factual associations in gpt. *Advances in neural information processing systems*, 35:17359–17372.

Kevin Meng, Arnab Sen Sharma, Alex Andonian, Yonatan Belinkov, and David Bau. 2022b. Mass-editing memory in a transformer. *arXiv preprint arXiv:2210.07229*.

Eric Mitchell, Charles Lin, Antoine Bosselut, Chelsea Finn, and Christopher D Manning. 2021. Fast model editing at scale. *arXiv preprint arXiv:2110.11309*.

Eric Mitchell, Charles Lin, Antoine Bosselut, Christopher D Manning, and Chelsea Finn. 2022. Memory-based model editing at scale. In *International Conference on Machine Learning*, pages 15817–15831. PMLR.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, and 1 others. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Jorma J Rissanen. 2002. Fisher information and stochastic complexity. *IEEE transactions on information theory*, 42(1):40–47.

Ben Wang and Aran Komatsuzaki. 2021. Gpt-j-6b: A 6 billion parameter autoregressive language model.

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. *Advances in Neural Information Processing Systems*, 37:53764–53797.

Jialiang Wu, Yi Shen, Sijia Liu, Yi Tang, Sen Song, Xiaoyi Wang, and Longjun Cai. 2025. Improve decoding factuality by token-wise cross layer entropy of large language models. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pages 3912–3921.

Suhang Wu, Minlong Peng, Yue Chen, Jinsong Su, and Mingming Sun. 2023. Eva-kellm: A new benchmark for evaluating knowledge editing of llms. *arXiv preprint arXiv:2308.09954*.

Xiaobao Wu, Liangming Pan, William Yang Wang, and Anh Tuan Luu. 2024. Akew: Assessing knowledge editing in the wild. *arXiv preprint arXiv:2402.18909*.

Chenghao Xu, Guangtao Lyu, Jiexi Yan, Muli Yang, and Cheng Deng. 2024. Llm knows body language, too: Translating speech voices into human gestures. In *ACL*, pages 5004–5013.

Chenghao Xu, Guangtao Lyu, Jiexi Yan, Muli Yang, and Cheng Deng. 2025. Smooth and flexible camera movement synthesis via temporal masked generative modeling. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*.

Chenghao Xu, Jiexi Yan, Muli Yang, Fen Fang, Huilin Chen, and Cheng Deng. 2026. Editing is a bargaining game: Balanced knowledge editing in large language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pages 34097–34105.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, and 1 others. 2025. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.

Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, and 1 others. 2023. A survey of large language models. *arXiv preprint arXiv:2303.18223*, 1(2).

Ce Zheng, Lei Li, Qingxiu Dong, Yuxuan Fan, Zhiyong Wu, Jingjing Xu, and Baobao Chang. 2023. Can we edit factual knowledge by in-context learning? *arXiv preprint arXiv:2305.12740*.

A Fisher Information Approximation Using Proxy Vectors

As discussed in the main paper, we adopt a low-rank approximation of the modulation matrix using the outer product of two proxy vectors. To compute the FI of the modulation matrix, we begin by analyzing the FI of each individual element. For an element defined as $\Delta^i_{jk} = \Delta^i_{\mathrm{H},j}\Delta^i_{\mathrm{W},k}$, the corresponding expression can be derived through a straightforward application of the chain rule of differentiation, as shown below:

$$\frac{\partial \mathcal{L}}{\partial \Delta^i_{jk}} = \frac{1}{2\Delta^i_{\mathrm{W},k}} \frac{\partial \mathcal{L}}{\partial \Delta^i_{\mathrm{H},j}} + \frac{1}{2\Delta^i_{\mathrm{H},j}} \frac{\partial \mathcal{L}}{\partial \Delta^i_{\mathrm{W},k}}, \quad (6)$$

We estimate the FI using the squared gradients. Consequently, the following relationship can be derived between the Fisher Information values of these variables:

$$\mathcal{F}(\Delta^i_{jk}) = \frac{1}{4{\Delta^i_{\mathrm{W},k}}^2}\mathcal{F}(\Delta^i_{\mathrm{H},j}) + \frac{1}{4{\Delta^i_{\mathrm{H},j}}^2}\mathcal{F}(\Delta^i_{\mathrm{W},k}) + \frac{1}{2\Delta^i_{\mathrm{H},j}\Delta^i_{\mathrm{W},k}} \frac{\partial \mathcal{L}}{\partial \Delta^i_{\mathrm{H},j}} \frac{\partial \mathcal{L}}{\partial \Delta^i_{\mathrm{W},k}}. \quad (7)$$

Then, the FI of the modulation matrix $\Delta W_i = (\Delta^i_{jk}) \in \mathbb{R}^{m\times n}$, can be calculated as:

$$\begin{aligned}
\mathcal{F}(\Delta W_i) &= \sum_{j=1}^{M}\sum_{k=1}^{N} \mathcal{F}(\Delta^i_{jk}) \\
&= \sum_{j=1}^{M}\sum_{k=1}^{N} \left(\frac{1}{4{\Delta^i_{\mathrm{W},k}}^2}\mathcal{F}(\Delta^i_{\mathrm{H},j}) \right. \\
&\quad + \frac{1}{4{\Delta^i_{\mathrm{H},j}}^2}\mathcal{F}(\Delta^i_{\mathrm{W},k}) \\
&\quad \left. + \frac{1}{2\Delta^i_{\mathrm{H},j}\Delta^i_{\mathrm{W},k}} \frac{\partial \mathcal{L}}{\partial \Delta^i_{\mathrm{H},j}} \frac{\partial \mathcal{L}}{\partial \Delta^i_{\mathrm{W},k}} \right) \\
&= \sum_{j=1}^{M} \left(\mathcal{F}(\Delta^i_{\mathrm{H},j}) \sum_{k=1}^{N} \frac{1}{4{\Delta^i_{\mathrm{W},k}}^2} \right. \\
&\quad + \frac{1}{4{\Delta^i_{\mathrm{H},j}}^2} \sum_{k=1}^{N} \mathcal{F}(\mathcal{F}(\Delta^i_{\mathrm{W},k}) \\
&\quad \left. + \frac{1}{2\Delta^i_{\mathrm{H},j}\Delta^i_{\mathrm{W},k}} \sum_{k=1}^{N} \frac{1}{\Delta^i_{\mathrm{W},k}} \frac{\partial \mathcal{L}}{\partial \Delta^i_{\mathrm{W},k}} \right).
\end{aligned} \quad (8)$$

We empirically observe that discarding (i) the cross-term and (ii) the coefficient terms in the weight importance computation yields comparable performance in the final model. This observation suggests that these components have a negligible impact on overall performance and can therefore be omitted. As a result, the estimation procedure can be simplified and made more computationally efficient. Accordingly, we adopt the following simplified estimator of $\mathcal{F}(\Delta W_i)$ in our method:

$$\hat{\mathcal{F}}(\Delta W_i) = \sum_{j=1}^{M} \left(\mathcal{F}(\Delta^i_{\mathrm{H},j}) + \frac{1}{N}\sum_{k=1}^{N} \mathcal{F}(\Delta^i_{\mathrm{W},k}) \right). \quad (9)$$

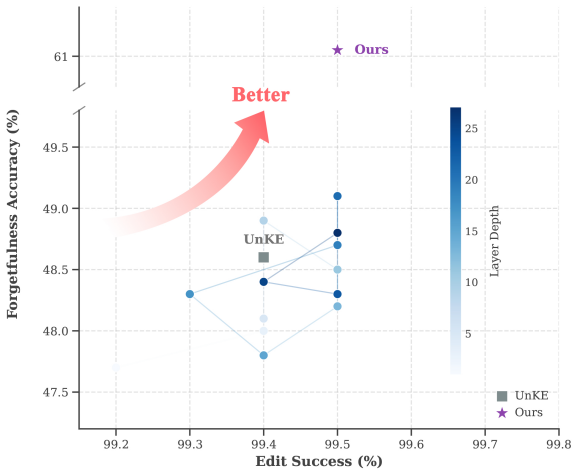

Figure 3: Ablation study for location.

B More Experimental Results

Performance Comparison with Different Locating Strategies. To more comprehensively evaluate the effectiveness of our method, we compare the performance with different locating strategies and report the results reported in Figure 3. During the editing phase, we adopt the gradient-based optimization strategy proposed in UnKE to solve the constrained least-squares problem defined in Eq. (5). Prior to editing, we compare our adaptive locating strategy with fixed-layer locating approaches applied to different layers. Overall, editing at any single fixed layer consistently underperforms our editing within adaptive locating, demonstrating the effectiveness of our proposed method.