GeoEdit: Geometric Knowledge Editing for Large Language Models

Yujie Feng[1], **Liming Zhan**[1], **Zexin Lu**[1], **Yongxin Xu**[2], **Xu Chu**[2]
Yasha Wang[2], **Jiannong Cao**[1], **Philip S. Yu**[3], **Xiao-Ming Wu**[1]
[1]The Hong Kong Polytechnic University
[2]Peking University [3]University of Illinois at Chicago
yujie.feng@connect.polyu.hk, xiao-ming.wu@polyu.edu.hk

Abstract

Regular updates are essential for maintaining up-to-date knowledge in large language models (LLMs). However, existing training-based model editing methods often struggle to effectively incorporate new knowledge while preserving unrelated general knowledge. To address this challenge, we propose a novel framework called Geometric Knowledge Editing (GeoEdit). GeoEdit utilizes the geometric relationships of parameter updates from fine-tuning to differentiate between neurons associated with new knowledge updates and those related to general knowledge perturbations. By employing a direction-aware knowledge identification method, we avoid updating neurons with directions approximately orthogonal to existing knowledge, thus preserving the model's generalization ability. For the remaining neurons, we integrate both old and new knowledge for aligned directions and apply a "forget-then-learn" editing strategy for opposite directions. Additionally, we introduce an importance-guided task vector fusion technique that filters out redundant information and provides adaptive neuron-level weighting, further enhancing model editing performance. Extensive experiments on two publicly available datasets demonstrate the superiority of GeoEdit over existing state-of-the-art methods.

1 Introduction

Large language models (LLMs) have demonstrated the ability to store vast amounts of knowledge during pre-training and retrieve it during inference (Yao et al., 2023; Wang et al., 2024d). However, much of the knowledge in the real world is constantly evolving. For instance, the answer to the question "Who is the President of the United States?" was "Joe Biden" in 2024, but it is now "Donald Trump". As a result, some knowledge that was once correct in LLMs can become obsolete or inaccurate (Li and Chu, 2024; Huang et al., 2024).

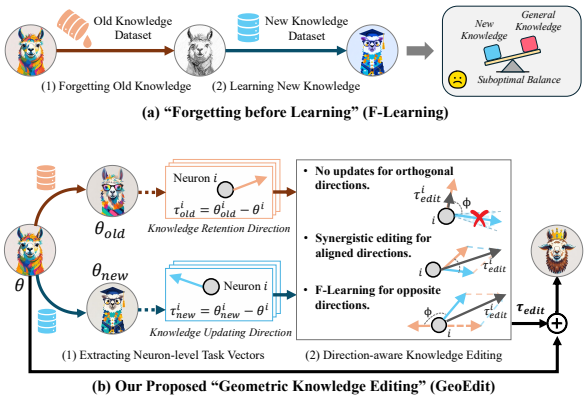

Figure 1: Conceptual illustration of F-Learning (Ni et al., 2024) and our proposed GeoEdit.

To address this issue, model editing methods have been developed to update the target new knowledge while preserving unrelated general knowledge within the model (Hong and Lipani, 2024; Ma et al., 2024; Wang et al., 2024e). Specifically, current model editing methods typically follow the "locate-and-edit" paradigm (Wang et al., 2023, 2024a; Li et al., 2024). The core idea is first to locate influential weights in LLMs and then edit them by introducing a perturbation. Although effective, these approaches incur significant computational overhead to identify the important neurons and parameters (Meng et al., 2022). Some methods also require sampling additional data (e.g., from Wikipedia) to mitigate the impact on the general knowledge within LLMs during editing (Meng et al., 2023; Fang et al., 2024), introducing extra costs and potential biases.

In contrast, fine-tuning with updated knowledge, as demonstrated in recent studies (Zhao et al., 2024; Wang and Li, 2024; Liu et al., 2024), offers a more straightforward solution through the use of parameter-efficient fine-tuning (PEFT) techniques. These methods employ various strategies to edit the model without the need to differentiate the importance of individual parameters (Feng et al., 2025).

Among these, the pioneering work F-Learning (Ni et al., 2024) introduces a new learning framework called "Forgetting before Learning," as illustrated in Figure 1(a). This approach is based on the empirical observation that new knowledge can be difficult to learn when it conflicts with existing knowledge. By first forgetting outdated knowledge, the learning of new knowledge becomes easier. However, this approach has a critical limitation: it struggles to balance the integration of new knowledge with the preservation of existing general knowledge. Specifically, F-Learning assumes that all updates between old and new knowledge are inherently conflicting, which oversimplifies the complexity of knowledge integration. Furthermore, the unconstrained forgetting process can significantly impact the model's generalization ability to out-of-scope samples, considerably reducing the performance of the Locality metric (see Section 3).

To address these limitations, we propose Geometric Knowledge Editing (**GeoEdit**), a novel fine-tuning-based model editing framework that enhances editing precision while strongly preserving model generalization without the need for additional unrelated data. The core insight of GeoEdit is to distinguish between neurons associated with new knowledge updates and those linked to general knowledge perturbations by analyzing the geometric relationships of parameter updates caused by fine-tuning. By masking the updates of ***general-knowledge-related neurons***, we prevent negative impacts on the model's generalization ability. At the same time, we optimize the update strategy for ***new-knowledge-related neurons***, further enhancing the effectiveness of model editing.

Specifically, we first fine-tune the current model separately on the old and new knowledge datasets. This allows us to derive neuron-level task vectors, τ_{old} and τ_{new}, using task arithmetic (Ilharco et al., 2022), which capture the directions of knowledge retention and updating w.r.t. each neuron, as shown in Figure 1(b). We then introduce a direction-aware knowledge identification method that computes the angle ϕ between these two directions to classify neurons, followed by customized editing strategies: (i) ***Orthogonal Knowledge Editing*** (for approximately orthogonal directions): Neurons with updates orthogonal to old knowledge are classified as general-knowledge-related neurons. These updates are considered detrimental to the model's generalization ability, so we refrain from updating these neurons. The remaining neurons are treated as new-knowledge-related neurons, which are updated with two different strategies: (ii) ***Synergistic Knowledge Editing*** (for aligned directions): When there is slight conflict between old and new knowledge, we can leverage their similarities to simultaneously integrate both. (iii) ***Conflicting Knowledge Editing*** (for opposite directions): For updates with significant conflict, we apply the F-Learning strategy, where old knowledge is first forgotten before integrating new information.

Additionally, to mitigate angular bias in high-dimensional space, GeoEdit employs a combined dimensionality reduction approach to more effectively extract angular information, ensuring the reliability of the edits. To optimize vector fusion, we introduce an importance-guided task vector fusion technique, which applies fine-grained weights to the vectors and suppresses noise from redundant parameters, further enhancing the effectiveness of model editing. Extensive experiments demonstrate that GeoEdit achieves the best performance among all fine-tuning-based methods and show its significant potential for complementing locate-and-edit methods, further enhancing performance.

Our main contributions are summarized as:

- We propose a novel geometric knowledge editing **framework** (GeoEdit) for updating LLMs.
- We develop new direction-aware knowledge identification and importance-guided task vector fusion **techniques**.
- Extensive **evaluation** on two widely-used datasets shows that GeoEdit overcomes the limitations of F-Learning, improving the Locality metric by 7.4% while maintaining the best performance in the Reliability and Generality metrics.

2 Related Work

Knowledge editing has gained significant attention due to the increasing need to update the knowledge in LLMs (Shengyuan et al., 2023; Wang et al., 2024d; Bi et al., 2024c,b). Existing methods can be classified into two main approaches:

Locate and edit methods usually locate influential parameters and then edit them by introducing a perturbation (Zhang et al., 2024; Jiang et al., 2024; Xu et al., 2024). Classic methods like ROME (Meng et al., 2022) and MEMIT (Meng et al., 2023) use causal reasoning to identify key neuron activations and adjust specific weights. Additionally, Yu et al. (2023) employs gradient-based

attribution to identify important weights. More recent approaches, such as AlphaEdit (Fang et al., 2024), improve the method by projecting perturbations onto the null space of preserved knowledge, demonstrating strong performance.

Fine-tuning is an intuitive and straightforward way to update the model's knowledge (Feng et al., 2023; Gangadhar and Stratos, 2024; Zheng et al., 2024). Recently, a series of PEFT methods, such as Prefix-Tuning (Li and Liang, 2021) and LoRA (Hu et al., 2022), have made knowledge editing based on fine-tuning more feasible. Zhang et al. (2023) enhance update efficiency and adaptability by performing incremental parameter updates of varying magnitudes, which are determined by calculating the importance of the weight matrix.

However, both of these methods struggle to balance new knowledge updates with preserving unrelated knowledge (Gupta et al., 2024; Feng et al., 2024a; Chen et al., 2024). For instance, locate-and-edit methods typically require large additional datasets to capture general knowledge and avoid disruption during editing (Wang et al., 2024c; Hsueh et al., 2024; Bi et al., 2024a; Zhang et al., 2025). Furthermore, locate-and-edit methods primarily focus on editing the MLP layers of the model, while fine-tuning methods offer the flexibility to adjust different regions of the model.

Thus, our paper focuses on improving fine-tuning methods. By distinguishing between general knowledge and updated knowledge based on the angular divergence between the updated directions of old and new knowledge, our GeoEdit avoids updating general knowledge, ensuring model generalization while applying tailored strategies to enhance the effectiveness of knowledge updates. Additionally, our method can fine-tune parameters in regions different from those targeted by locate-and-edit methods, allowing for potential complementarity that further enhances performance.

3 Problem Statement

Model editing, also referred to as knowledge updating, involves modifying the behavior of an initial target model on specific edit examples without compromising its performance on unrelated examples. More precisely, given an initial model f_θ and a set of input-output knowledge pairs $D_{old} = \{(x_1, y_1), (x_2, y_2), \ldots, (x_k, y_k)\}$, the task is to update the model parameters to obtain a new model f_{θ_e} and a corresponding set of new input-output pairs $D_{new} = \{(x_1, y_1^{\text{new}}), (x_2, y_2^{\text{new}}), \ldots, (x_k, y_k^{\text{new}})\}$, where k denotes the number of knowledge pairs to be updated. The objective of the post-edit model f_{θ_e} is to meet three essential properties: reliability, generality, and locality (Wang et al., 2024b).

Reliability Reliability measures the accuracy of the updated model on the new knowledge. Specifically, the output for "Who is the President of the US?" should be updated from "Joe Biden" to "Donald Trump." This can be formalized as follows:

$$\mathbb{E}_{x_e, y_e \sim D_{new}} \mathbb{1} \left\{ \text{argmax}_y f_{\theta_e}(y|x_e) = y_e \right\}. \quad (1)$$

Generality Generality means that the new model f_{θ_e} should also update rephrased in-scope examples $I(x_e, y_e)$. Such as the answer to "Who holds the position of the President of the US?" should also be changed from "Joe Biden" to "Donald Trump". This is evaluated by the average accuracy of f_{θ^*} on examples from the equivalence neighborhood, as expressed by:

$$\mathbb{E}_{x'_e, y'_e \sim I(x_e, y_e)} \mathbb{1} \left\{ \text{argmax}_y f_{\theta_e}(y|x'_e) = y'_e \right\}. \quad (2)$$

Locality A good edit should modify relevant knowledge without affecting other irrelevant out-of-scope examples $O(x_e, y_e)$. For example, the question, "Who said: this is a battle for the soul of the nation?" should remain unchanged as "Joe Biden". Locality (or specificity) is defined as:

$$\mathbb{E}_{x'_e, y'_e \sim O(x_e, y_e)} \mathbb{1} \left\{ \text{argmax}_y f_{\theta_e}(y|x'_e) = f_\theta(y|x'_e) \right\}. \quad (3)$$

4 Proposed Method: GeoEdit

In this section, we present our method for knowledge editing in LLMs. As illustrated in Figure 2, GeoEdit follows a three-step process:

4.1 Extracting Neuron-level Task Vectors

Supervised fine-tuning (SFT) on a dataset injects new knowledge into the LLMs, reflected in model parameter changes. For an initial model f_θ with parameters θ, fine-tuning on dataset D produces updated parameters. The difference between the updated and original parameters is referred to as the *task vector* (Ilharco et al., 2022), calculated as:

$$\tau = \text{FT}\{\theta, D\} - \theta \quad (4)$$

where τ is the corresponding task vector, and FT is the fine-tuning operation. Unlike F-Learning,

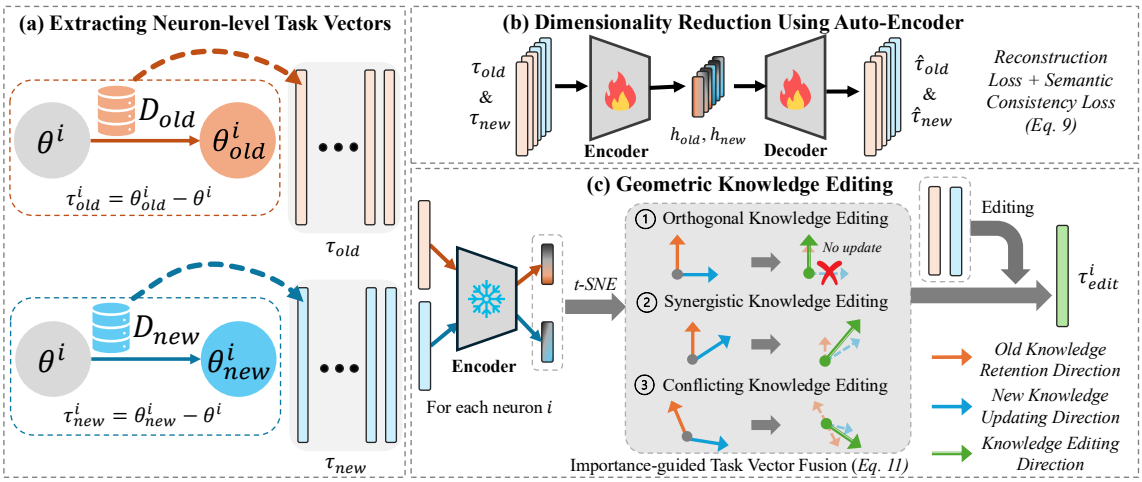

Figure 2: **Overview of GeoEdit. Step (a)**: Neuron-level task vectors τ_{old} and τ_{new} are extracted for both the old and new knowledge datasets using parametric arithmetic. **Step (b)**: An auto-encoder is trained to project a low-dimensional representation of the task vectors, eliminating the angular bias issue in high-dimensional space. **Step (c)**: The latent task vectors, h_{new} and h_{old}, are reduced to two dimensions using t-SNE to compute the angular relationships, which are used to classify neurons based on the angle. Finally, after applying different editing strategies, we obtain the edited vector τ_{edit}, which is added to the initial model to generate the edited model f_{θ_e}.

we fine-tune the initial model f_θ separately on the old and new knowledge datasets to isolate their respective adaptations, then compute task vectors:

$$\tau_{old} = \mathrm{FT}\{\theta, D_{old}\} - \theta \tag{5}$$

$$\tau_{new} = \mathrm{FT}\{\theta, D_{new}\} - \theta \tag{6}$$

where D_{old} and D_{new} are datasets encoding outdated and updated knowledge respectively.

While prior research typically captures task vectors at the model level (Ilharco et al., 2022), we propose extracting them at the neuron level for finer control. Let $\theta = \{\theta^1, \theta^2, \ldots, \theta^N\}$ represent the N neurons in the LLM, where the i-th neuron is represented by $\theta^i \in \mathbb{R}^{d_n}$ with d_n dimensional parameters. The neuron-level task vectors are given by $\tau_{new} = \{\tau^1_{new}, \tau^2_{new}, \ldots, \tau^N_{new}\}$, where τ^i_{new} corresponds to the new knowledge task vector for the i-th neuron [1]. This approach enables more granular analysis of parameter changes and selective editing of knowledge-specific neurons, enhancing model editing precision.

After obtaining the task vectors for both old and new knowledge, we focus on the directional characteristics, which are more crucial than magnitudes for knowledge editing. We define the direction of τ_{old} as the knowledge retention direction and τ_{new} as the knowledge updating direction. By analyzing the angle between these directions, we can distinguish general-knowledge-related neurons to avoid harming generalization and new-knowledge-related neurons to enhance editing effectiveness.

4.2 Angular Relationship Extraction through Dimensionality Reduction

Due to the tendency of high-dimensional vectors to become nearly orthogonal, it is necessary to reduce the dimensionality of the original vectors in order to better capture the underlying angular relationships. However, experiments have shown that directly applying PCA or t-SNE for dimensionality reduction on τ yields suboptimal results. Therefore, we propose an alternative approach where an auto-encoder (AE) is first used to encode the high-dimensional vectors. This effectively filters out irrelevant information and extracts meaningful features. Subsequently, applying t-SNE to the encoded vectors allows for a more accurate representation of the true angular relationships.

Thus, we train a semantic encoder and decoder, both implemented using multi-layer perceptrons (MLPs). Specifically, the semantic encoder, denoted as SemEnc($\cdot$), maps the high-dimensional task vectors τ_{old} and τ_{new} into the latent space as:

$$h^i = \mathrm{SemEnc}(\tau^i) \tag{7}$$

where $h^i \in \mathbb{R}^{d_{latent}}$ is the latent task vector, and d_{latent} denotes its dimensionality. The decoder,

[1] We define a "neuron" as the linear transformation corresponding to a single column in matrix $W \in \mathbb{R}^{in \times out}$, where W consists of out neurons.

Dec(·), then generates $\hat{\tau}^i$ from h^i as follows:

$$\hat{\tau}^i = \text{Dec}(h^i) \tag{8}$$

where $\hat{\tau}^i$ is the reconstructed task vector. The auto-encoder is optimized using both a reconstruction loss and a semantic consistency loss:

$$\begin{aligned} \mathcal{L}_{AE} = \text{MSE}\left(\tau^i, \hat{\tau}^i\right) + \\ \lambda \cdot \text{KL}\left(f_{\theta+\tau^i}(x) \| f_{\theta+\hat{\tau}^i}(x)\right) \end{aligned} \tag{9}$$

where MSE(·) is mean square error loss function, and KL(·) is the Kullback-Leibler divergence.

4.3 Geometric Knowledge Editing

After training the AE, we project τ_{old} and τ_{new} into the latent space and then apply t-SNE to further project them into a 2D space to compute the angular relationships. GeoEdit then edit the original task vectors to obtain the edited task vector τ_{edit}. This vector is subsequently added to the initial model, resulting in the final edited model f_{θ_e}.

Direction-aware Knowledge Identification For neuron i, we first use the encoder to reduce the dimensionality of τ^i_{old} and τ^i_{new} yielding the latent task vectors h^i_{old} and h^i_{new}. We then apply t-SNE to obtain the 2D vectors $\hat{h}^i_{old}$ and $\hat{h}^i_{new}$. Next, we compute the angular divergence ϕ as:

$$\phi = \arccos \frac{\hat{h}^i_{old} \cdot \hat{h}^i_{new}}{\left|\hat{h}^i_{old}\right| \cdot \left|\hat{h}^i_{new}\right|} \tag{10}$$

Neurons with angles near orthogonality (within the range of ϕ_1 to ϕ_2) are classified as general-knowledge-related, while the remaining neurons are classified as new-knowledge-related.

Importance-guided Task Vector Fusion We then apply customized editing strategies based on the classification of neurons as follows:

$$\tau^i_{edit} = \begin{cases} \alpha^i \tau^i_{old} + \beta^i \tau^i_{new}, & \text{if } \phi \in (0°, \phi_1) \\ 0, & \text{if } \phi \in [\phi_1, \phi_2] \\ -\alpha^i \tau^i_{old} + \beta^i \tau^i_{new}, & \text{if } \phi \in (\phi_2, 180°) \end{cases} \tag{11}$$

where $\alpha^i, \beta^i \in [0, 1]$ are the fusion weights, automatically assigned based on the neuron's importance to both new and old knowledge, removing the need for manual adjustment.

To calculate the fusion weights, we measure the importance of each neuron by analyzing the gradient trajectory of its parameters during fine-tuning. The importance is determined by the collective contribution of its trainable parameters:

$$\mathcal{I}(\theta^i) = \frac{1}{d_n} \sum_{j=1}^{d_n} s(w_j) \tag{12}$$

where w_j represents the trainable parameters and d_n is the total number of parameters in neuron θ^i. The function $\mathcal{I}(\theta^i)$ reflects the importance of the neuron, with higher values indicating greater significance. The function $s(\cdot)$ computes the importance of individual parameters based on the magnitude of the gradient-weight product (Zhang et al., 2023):

$$s\left(w\right) = \left|w \nabla_w \mathcal{L}\right| \tag{13}$$

Due to stochastic sampling and training dynamics, the metric in Eq. (13) may vary, reducing reliability (Feng et al., 2024b). To address this, we apply an exponential moving average to smooth the trajectory gradients across training iterations.

We normalize the importance scores $\mathcal{I}_{\text{old}} = \{\mathcal{I}^1_{\text{old}}, \ldots, \mathcal{I}^N_{\text{old}}\}$ and $\mathcal{I}_{\text{new}} = \{\mathcal{I}^1_{\text{new}}, \ldots, \mathcal{I}^N_{\text{new}}\}$ independently to the range [0, 1]. This yields the final fusion weights $\alpha = \{\alpha^1, \ldots, \alpha^N\}$ and $\beta = \{\beta^1, \ldots, \beta^N\}$, which are then applied to the corresponding task vectors τ_{old} and τ_{new} for editing.

By applying Eq. (11), our GeoEdit effectively addresses the challenges in model editing:

- **Preserving general knowledge** (Case 2): We mask updates to general-knowledge-related neurons to avoid negatively impacting the model's generalization ability.
- **Improving knowledge editing** (Case 1 & 3): For acute angles, we leverage the similarity between old and new knowledge for efficient integration. For obtuse angles, significant conflict triggers a "forget-then-learn" strategy, optimizing the updates for new-knowledge-related neurons.

5 Experiments and Analysis

Datasets We use two widely recognized datasets: ZsRE (Levy et al., 2017) and CounterFact (Meng et al., 2022). We adopt the experimental setup from Yao et al. (2023), using the eval and edit subsets consisting of 19,085 and 10,000 examples, respectively. The datasets are partitioned into old and new knowledge categories, as in F-Learning (Ni et al., 2024). For example, in ZsRE, old knowledge is modified to new knowledge, such as the change from "Los Angeles" to "New Orleans." Further details and additional examples are provided in Appendix A.

Dataset	Paradigm	Method	LLAMA2-7B			LLAMA-7B		
			Reliability	Generality	Locality	Reliability	Generality	Locality
ZsRE		Original model	43.70	43.17	/	43.29	42.85	/
	Locate & edit	MEND	29.77	25.86	71.54	30.99	27.12	69.83
		ROME	43.67	42.66	**93.14**	43.45	42.94	**98.60**
		MEMIT	83.57	79.06	70.52	78.30	77.43	69.44
		RECT	84.08	77.80	69.03	78.78	76.20	67.97
		AlphaEdit	**87.91**	**81.52**	**77.14**	**87.09**	**80.41**	**76.53**
	Fine-tuning	LoRA	43.10	42.20	70.83	46.93	45.87	75.86
		FT-c	49.02	46.96	67.37	47.33	45.51	68.14
		Full-FT	81.02	74.67	70.51	70.52	66.69	65.26
		F-Learning	84.65	81.51	70.92	83.06	79.50	70.09
		GeoEdit	**85.21**	**82.43**	**75.71**	**84.81**	**79.86**	**75.15**
		GeoEdit*	**88.13**	**82.07**	**79.75**	**87.76**	**80.70**	**77.98**
CounterFact		Original model	18.47	16.95	/	21.61	17.88	/
	Locate & edit	MEND	14.77	14.67	90.93	17.51	16.27	89.64
		ROME	18.41	17.20	**93.60**	21.83	19.08	**92.27**
		MEMIT	61.94	37.45	21.90	56.94	31.48	25.70
		RECT	62.90	39.86	20.03	57.82	33.51	23.48
		AlphaEdit	**71.79**	**48.36**	**36.07**	**60.01**	**38.19**	**41.70**
	Fine-tuning	LoRA	30.56	23.24	40.08	27.54	21.21	39.75
		FT-c	29.23	19.32	19.70	26.97	17.90	20.09
		Full-FT	65.99	44.08	28.34	32.13	31.95	32.51
		F-Learning	**69.53**	45.56	28.41	56.39	39.75	31.87
		GeoEdit	68.34	**46.53**	**37.73**	**55.88**	**40.60**	**42.33**
		GeoEdit*	**72.20**	**48.57**	**38.71**	**60.89**	**41.37**	**43.99**

Table 1: Results on three metrics for the two datasets using LLAMA2-7B and LLAMA-7B. The best-performing method for each paradigm is highlighted in bold. AlphaEdit and our GeoEdit each achieve the best performance within their respective paradigms. Notably, the optimal performance is attained by GeoEdit*, which results from applying GeoEdit to the non-located parameters in AlphaEdit, effectively combining the strengths of both methods.

Baselines We evaluate GeoEdit against two types of methods: fine-tuning-based approaches, including full fine-tuning (**Full-FT**), **LoRA** (Hu et al., 2021), **FT-c** (Zhu et al., 2020), and **F-Learning** (Ni et al., 2024); and locate-and-edit-based methods, including **MEND** (Mitchell et al., 2022), **ROME** (Meng et al., 2022), **MEMIT** (Meng et al., 2023), **RECT** (Gu et al., 2024) and **AlphaEdit** (Fang et al., 2024). Detailed descriptions are provided in the Appendix B.

Training Details Following the setup of F-Learning, we first fine-tune the base model on the old knowledge for three epochs, resulting in the **original model**, which serves as the baseline for our experiments. In GeoEdit and F-Learning, we use LoRA to enhance the efficiency of fine-tuning. The encoder and decoder consists of 2-layer MLPs with dimensions $[4096 \to 2048, 2048 \to 512]$ and $[512 \to 2048, 2048 \to 4096]$, respectively, where d_{latent} is set to 512. We set λ in Eq. (9) to 0.5, ϕ_1 and ϕ_2 in Eq. (11) to 85° and 95°, respectively. Further details on the experimental setup can be found in Appendix C.

5.1 Experimental Results

The overall results are presented in Table 1. Firstly, ROME maintains high Reliability and Generality across both datasets while achieving excellent Locality (greater than 90). Since the injection of new knowledge typically impacts Locality, this suggests that ROME performs minimal knowledge updating, likely due to its limited parameter edits. In contrast, F-Learning shows a significant drop in Locality due to the lack of constraints during the forgetting phase, negatively impacting generalization. Our GeoEdit method outperforms fine-tuning-based methods, improving locality by 7.4% over F-Learning. Additionally, by classifying different knowledge editing strategies for new-knowledge-related neurons, our method further improves Reliability and Generality.

AlphaEdit achieves the best performance among locate-and-edit-based methods and outperforms GeoEdit in most cases. This is because AlphaEdit requires the use of an additional 100,000 Wikipedia entries to enhance general knowledge encoding and editing accuracy. In contrast, GeoEdit achieves its

Method	Reliability	Generality	Locality
GeoEdit	**85.21**	**82.43**	**75.71**
- Synergistic	84.37	81.13	75.94
- Orthogonal	85.29	82.86	71.70
- Conflict	82.57	79.40	75.45
+ MW	84.91	82.04	73.73

Table 2: Ablation study. "- Synergistic", "- Orthogonal", and "- Conflict" refer to removing the synergistic, orthogonal, and conflict knowledge editing strategies, respectively. "+ MW" denotes replacing the importance-guided fusion with a manually set weighting approach.

h_{latent}	Reliability	Generality	Locality
128	44.94	43.79	75.68
256	46.75	46.54	77.33
512	46.11	47.19	78.42
1024	25.39	24.49	94.81
2048	23.03	24.14	96.14

Table 3: Ablation study on latent dimension h_{latent}.

performance without relying on any external data. Furthermore, due to the flexibility of fine-tuning methods, we can effectively combine GeoEdit with AlphaEdit (which edits the parameters of the MLP layer, while GeoEdit targets the parameters of the attention layer), creating a complementary approach that further enhances performance.

5.2 Ablation Study

We conduct ablation studies to evaluate the effectiveness of the techniques in GeoEdit. The results on the ZsRE dataset with LLaMA2-7B are shown in Table 2. Additional analysis of hyperparameter sensitivity is provided in Appendix D.

Effect of Geometric Editing Strategies. In GeoEdit, knowledge updates are categorized into synergistic, orthogonal, and conflict editing strategies, based on the angle between knowledge retention and updating directions. To evaluate their impact, we disable each strategy and replace it with vanilla fine-tuning on new knowledge. For example, "- Orthogonal" means setting $h_{edit} = h_{new}$ instead of $h_{edit} = 0$. As shown in Table 2, removing any strategy results in performance degradation. Excluding orthogonal editing significantly reduces locality, from 75.7% to 71.7%, while removing conflict editing lowers the reliability metric from 85.2% to 82.6%. These findings underscore the importance of each editing strategy.

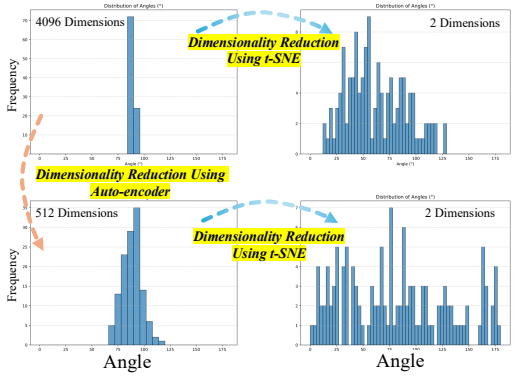

Figure 3: Distribution of the angles ϕ between task vectors before and after dimensionality reduction.

Effect of Importance-guided Task Vector Fusion. We replace the importance-based weights α and β in Eq. (11) with manually set values ("+ MW"), applying the same weight to all neurons instead of assigning neuron-specific weights as in GeoEdit. Through grid search, we set $\alpha = 0.3$ and $\beta = 1$.

The performance decline in Table 2 highlights the effectiveness of our importance-guided fusion. This approach provides two key benefits: it offers neuron-level adaptive weights for greater precision and ensures that parameter updates are influenced by both the task vector's magnitude and each neuron's importance. The smaller weights "masks" significant changes for less important neurons, minimizing their impact on the model's generalizability.

Effect of Latent Space Dimension. We investigate the effect of the latent dimension h_{latent} in the auto-encoder on model performance. As shown in Table 3, both larger latent dimensions (greater than 1024) and very small latent dimensions (less than 256) lead to performance collapse. This is because the large training loss of the auto-encoder results in poor t-SNE dimensionality reduction, ultimately affecting the accuracy of angular calculations. Our experiments demonstrate that a latent dimension of 512 strikes an optimal balance, effectively removing noise and extracting key features for calculating the angular distribution, which ensures effective model editing and strong generalization.

5.3 Visualization

We present two key visualizations to demonstrate the effectiveness of our approach:

Angle Distribution Between Old Knowledge Retention and New Knowledge Updating Directions. We visualize the angle distribution ϕ be-

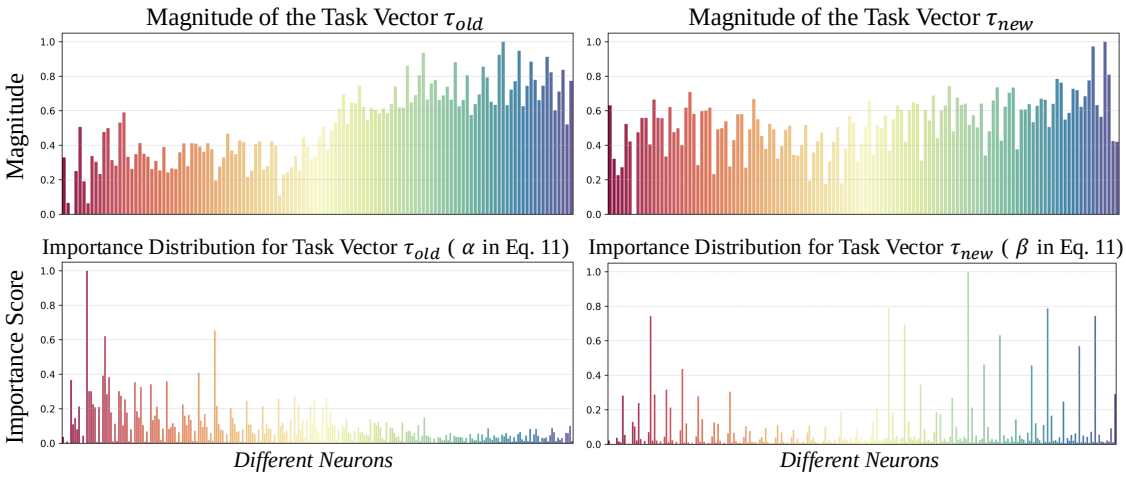

Figure 4: Visualization of the magnitudes of task vectors τ_{old} and τ_{new} along with the importance-guided fusion weights. All results are normalized to the range of 0 to 1.

tween task vectors using different dimensionality reduction methods, as shown in Figure 3. In high-dimensional space, angles are primarily concentrated around 90 degrees, indicating near orthogonality. Although directly applying t-SNE alleviates this issue to some extent, the angular distribution remains insufficiently dispersed. By first using an AE for denoising and key feature extraction, followed by t-SNE, we achieve a more uniform distribution that spans the full range from 0 to 180 degrees. This allows us to reveal various types of conflicts between old and new knowledge. This motivates the development of editing strategies based on angles, enabling us to distinguish between updates that correspond to learning new knowledge and those that modify general knowledge. The results of the ablation study on different dimensionality reduction methods are provided in Appendix D.1.

Visualization of Task Vector Magnitudes and Importance-guided Fusion Weights. Figure 4 illustrates that while the magnitudes of the task vectors τ_{old} and τ_{new} are generally large, only a subset of the parameters are truly important, highlighting redundancy in the task vectors. Our importance-guided fusion mechanism effectively filters out this redundancy, enhancing the model editing process and minimizing its impact on generalization.

5.4 Editing Time Analysis

Table 4 shows the average time to edit 1000 samples. We find that the editing time of fine-tuning methods is comparable to that of location-based methods, thanks to the use of PEFT techniques and the avoidance of the complex location-based process. Among fine-tuning approaches, F-Learning, which follows a two-stage process of forgetting before learning, takes approximately twice as long as Full-FT. In contrast, our method enables the parallel acquisition of old and new knowledge, resulting in training times comparable to Full-FT. Thus, our method requires less time than F-Learning, delivering substantial performance improvements.

Method	Average time per 1000 edits	
	zsRE	COUNTERFACT
FT-c	653.2(s)	579.3(s)
ROME	2184.2(s)	1810.4(s)
MEMIT	862.2(s)	847.7(s)
Full-FT	810.2(s)	792.4(s)
F-Learning	1670.4(s)	1603.8(s)
GeoEdit	**1028.0(s)**	**1010.6(s)**

Table 4: Editing time for two datasets on LLAMA2-7B.

6 Conclusion

In this paper, we introduce Geometric Knowledge Editing (GeoEdit), a novel framework that utilizes the geometric relationships between parameter updates to improve model editing. By applying a direction-aware knowledge identification technique, GeoEdit classifies neurons into two categories: general-knowledge-related neurons, whose parameter updates are masked to prevent negative impacts on model generalization, and new-knowledge-related neurons, where an importance-guided task vector fusion technique is applied to enhance editing. Extensive experiments demonstrate the effectiveness of GeoEdit for model editing.

Limitations

We acknowledge two limitations in this work.

First, GeoEdit requires access to old knowledge datasets to extract the task vector τ_{old}. In some cases, however, such datasets may not be available, meaning we only know the updated results. A potential solution is to input the task to be edited directly into the initial model and use the output as the old knowledge. However, this introduces additional inference costs, especially in our mass-editing settings. Furthermore, for open-ended questions, selecting the appropriate output as the reference is another challenge. We plan to explore ways to extend GeoEdit to address these issues and improve its adaptability.

Second, the core of GeoEdit relies on using the angle between parameter updates to differentiate between disturbances to general knowledge and the learning of new knowledge. While this approach offers valuable insights, it still results in some loss of model generalization, suggesting that the angle alone cannot fully decouple new knowledge learning from general knowledge disturbance. To address this, we aim to consider multiple geometric variables, such as task vector projections and magnitude, to further refine GeoEdit and enhance performance in the future.

References

Baolong Bi, Shenghua Liu, Lingrui Mei, Yiwei Wang, Pengliang Ji, and Xueqi Cheng. 2024a. Decoding by contrasting knowledge: Enhancing llms' confidence on edited facts. *ACL 2025*.

Baolong Bi, Shenghua Liu, Yiwei Wang, Lingrui Mei, Hongcheng Gao, Junfeng Fang, and Xueqi Cheng. 2024b. Struedit: Structured outputs enable the fast and accurate knowledge editing for large language models. *arXiv preprint arXiv:2409.10132*.

Baolong Bi, Shenghua Liu, Yiwei Wang, Lingrui Mei, Hongcheng Gao, Yilong Xu, and Xueqi Cheng. 2024c. Adaptive token biaser: Knowledge editing via biasing key entities. *EMNLP 2024*.

Shengyuan Chen, Qinggang Zhang, Junnan Dong, Wen Hua, Qing Li, and Xiao Huang. 2024. Entity alignment with noisy annotations from large language models. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

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

Yujie Feng, Xu Chu, Yongxin Xu, Zexin Lu, Bo Liu, Philip S Yu, and Xiao-Ming Wu. 2024a. Kif: Knowledge identification and fusion for language model continual learning. *arXiv preprint arXiv:2408.05200*.

Yujie Feng, Xu Chu, Yongxin Xu, Guangyuan Shi, Bo Liu, and Xiao-Ming Wu. 2024b. Tasl: Continual dialog state tracking via task skill localization and consolidation. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1266–1279.

Yujie Feng, Bo Liu, Xiaoyu Dong, Zexin Lu, Li-Ming Zhan, Xiao-Ming Wu, and Albert Lam. 2024c. Continual dialogue state tracking via reason-of-select distillation. In *Findings of the Association for Computational Linguistics ACL 2024*, pages 7075–7087.

Yujie Feng, Zexin Lu, Bo Liu, Liming Zhan, and Xiao-Ming Wu. 2023. Towards llm-driven dialogue state tracking. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 739–755.

Yujie Feng, Xujia Wang, Zexin Lu, Shenghong Fu, Guangyuan Shi, Yongxin Xu, Yasha Wang, Philip S Yu, Xu Chu, and Xiao-Ming Wu. 2025. Recurrent knowledge identification and fusion for language model continual learning. *arXiv preprint arXiv:2502.17510*.

Govind Krishnan Gangadhar and Karl Stratos. 2024. Model editing by standard fine-tuning. In *Findings of the Association for Computational Linguistics ACL 2024*, pages 5907–5913.

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

Akshat Gupta, Dev Sajnani, and Gopala Anumanchipalli. 2024. A unified framework for model editing. *arXiv preprint arXiv:2403.14236*.

Yihuai Hong and Aldo Lipani. 2024. Interpretability-based tailored knowledge editing in transformers. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 3847–3858.

Cheng-Hsun Hsueh, Paul Kuo-Ming Huang, Tzu-Han Lin, Che-Wei Liao, Hung-Chieh Fang, Chao-Wei Huang, and Yun-Nung Chen. 2024. Editing the mind of giants: An in-depth exploration of pitfalls of knowledge editing in large language models. *arXiv preprint arXiv:2406.01436*.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. 2022. Lora: Low-rank adaptation of large language models. *ICLR*, 1(2):3.

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

Gabriel Ilharco, Marco Tulio Ribeiro, Mitchell Wortsman, Suchin Gururangan, Ludwig Schmidt, Hannaneh Hajishirzi, and Ali Farhadi. 2022. Editing models with task arithmetic. *arXiv preprint arXiv:2212.04089*.

Yuxin Jiang, Yufei Wang, Chuhan Wu, Wanjun Zhong, Xingshan Zeng, Jiahui Gao, Liangyou Li, Xin Jiang, Lifeng Shang, Ruiming Tang, et al. 2024. Learning to edit: Aligning llms with knowledge editing. *arXiv preprint arXiv:2402.11905*.

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

Qi Li and Xiaowen Chu. 2024. Can we continually edit language models? on the knowledge attenuation in sequential model editing. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 5438–5455.

Shuaiyi Li, Yang Deng, Deng Cai, Hongyuan Lu, Liang Chen, and Wai Lam. 2024. Consecutive batch model editing with hook layers. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 13817–13833.

Xiang Lisa Li and Percy Liang. 2021. Prefix-tuning: Optimizing continuous prompts for generation. *arXiv preprint arXiv:2101.00190*.

Bo Liu, Liming Zhan, Zexin Lu, Yujie Feng, Lei Xue, and Xiao-Ming Wu. 2023. How good are large language models at out-of-distribution detection? *arXiv preprint arXiv:2308.10261*.

Jiateng Liu, Pengfei Yu, Yuji Zhang, Sha Li, Zixuan Zhang, Ruhi Sarikaya, Kevin Small, and Heng Ji. 2024. Evedit: Event-based knowledge editing for deterministic knowledge propagation. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 4907–4926.

Zexin Lu, Keyang Ding, Yuji Zhang, Jing Li, Baolin Peng, and Lemao Liu. 2021a. Engage the public: Poll question generation for social media posts. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 29–40.

Zexin Lu, Jing Li, Yingyi Zhang, and Haisong Zhang. 2021b. Getting your conversation on track: Estimation of residual life for conversations. In *2021 IEEE Spoken Language Technology Workshop (SLT)*, pages 1036–1043. IEEE.

Xinbei Ma, Tianjie Ju, Jiyang Qiu, Zhuosheng Zhang, Hai Zhao, Lifeng Liu, and Yulong Wang. 2024. On the robustness of editing large language models. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 16197–16216.

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*.

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

Shiwen Ni, Dingwei Chen, Chengming Li, Xiping Hu, Ruifeng Xu, and Min Yang. 2024. Forgetting before learning: Utilizing parametric arithmetic for knowledge updating in large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5716–5731, Bangkok, Thailand. Association for Computational Linguistics.

Chen Shengyuan, Yunfeng Cai, Huang Fang, Xiao Huang, and Mingming Sun. 2023. Differentiable neuro-symbolic reasoning on large-scale knowledge graphs. *Advances in Neural Information Processing Systems*, 36.

Guangyuan Shi, Zexin Lu, Xiaoyu Dong, Wenlong Zhang, Xuanyu Zhang, Yujie Feng, and Xiao-Ming Wu. 2024. Understanding layer significance in llm alignment. *arXiv preprint arXiv:2410.17875*.

Haoyu Wang, Tianci Liu, Ruirui Li, Monica Cheng, Tuo Zhao, and Jing Gao. 2024a. Roselora: Row and column-wise sparse low-rank adaptation of pre-trained language model for knowledge editing and fine-tuning. *arXiv preprint arXiv:2406.10777*.

Huazheng Wang, Haifeng Sun, Jingyu Wang, Qi Qi, Zixuan Xia, Menghao Zhang, and Jianxin Liao. 2024b. Sss: Editing factual knowledge in language models towards semantic sparse space. In *Findings of the Association for Computational Linguistics ACL 2024*, pages 5559–5570.

Jiaan Wang, Yunlong Liang, Zengkui Sun, Yuxuan Cao, Jiarong Xu, and Fandong Meng. 2023. Cross-lingual knowledge editing in large language models. *arXiv preprint arXiv:2309.08952*.

Mengru Wang, Ningyu Zhang, Ziwen Xu, Zekun Xi, Shumin Deng, Yunzhi Yao, Qishen Zhang, Linyi Yang, Jindong Wang, and Huajun Chen. 2024c. Detoxifying large language models via knowledge editing. *arXiv preprint arXiv:2403.14472*.

Renzhi Wang and Piji Li. 2024. Lemoe: Advanced mixture of experts adaptor for lifelong model editing of large language models. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 2551–2575.

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

Xiaohan Wang, Shengyu Mao, Ningyu Zhang, Shumin Deng, Yunzhi Yao, Yue Shen, Lei Liang, Jinjie Gu, and Huajun Chen. 2024e. Editing conceptual knowledge for large language models. *arXiv preprint arXiv:2403.06259*.

Yongxin Xu, Ruizhe Zhang, Xinke Jiang, Yujie Feng, Yuzhen Xiao, Xinyu Ma, Runchuan Zhu, Xu Chu, Junfeng Zhao, and Yasha Wang. 2024. Parenting: Optimizing knowledge selection of retrieval-augmented language models with parameter decoupling and tailored tuning. *arXiv preprint arXiv:2410.10360*.

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. *arXiv preprint arXiv:2305.13172*.

Charles Yu, Sullam Jeoung, Anish Kasi, Pengfei Yu, and Heng Ji. 2023. Unlearning bias in language models by partitioning gradients. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 6032–6048.

Ningyu Zhang, Yunzhi Yao, Bozhong Tian, Peng Wang, Shumin Deng, Mengru Wang, Zekun Xi, Shengyu Mao, Jintian Zhang, Yuansheng Ni, et al. 2024. A comprehensive study of knowledge editing for large language models. *arXiv preprint arXiv:2401.01286*.

Qinggang Zhang, Shengyuan Chen, Yuanchen Bei, Zheng Yuan, Huachi Zhou, Zijin Hong, Junnan Dong, Hao Chen, Yi Chang, and Xiao Huang. 2025. A survey of graph retrieval-augmented generation for customized large language models. *arXiv preprint arXiv:2501.13958*.

Qingru Zhang, Minshuo Chen, Alexander Bukharin, Nikos Karampatziakis, Pengcheng He, Yu Cheng, Weizhu Chen, and Tuo Zhao. 2023. Adalora: Adaptive budget allocation for parameter-efficient fine-tuning. *arXiv preprint arXiv:2303.10512*.

Zihao Zhao, Yuchen Yang, Yijiang Li, and Yinzhi Cao. 2024. Ripplecot: Amplifying ripple effect of knowledge editing in language models via chain-of-thought in-context learning. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 6337–6347.

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*.

Jiamu Zheng, Jinghuai Zhang, Tianyu Du, Xuhong Zhang, Jianwei Yin, and Tao Lin. 2024. Collabedit: Towards non-destructive collaborative knowledge editing. *arXiv preprint arXiv:2410.09508*.

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

A Datasets and Examples

We follow the F-Learning approach (Ni et al., 2024), which divides datasets into old and new knowledge. Below, we provide an overview of the datasets used, with detailed descriptions available in the original F-Learning paper. We use two well-known datasets: ZsRE (Levy et al., 2017) and COUNTERFACT (Meng et al., 2022). ZsRE is a Question Answering (QA) dataset that incorporates question rephrasings via back-translation (Lu et al., 2021a; Feng et al., 2024c), while COUNTERFACT is a more challenging counterfactual dataset. We use the eval and edit sets, containing 19,085 and 10,000 examples, respectively. Here's an example from the ZsRE dataset:

{**"subject":** "Watts Humphrey", **"src":** "What university did Watts Humphrey attend?", **"pred":** **"Trinity College"**, **"rephrase":** "What university did Watts Humphrey take part in?", **"alt":** **"University of Michigan"**, **"answers":** ["Illinois Institute of Technology"], **"loc":** "nq question: who played desmond doss father in hacksaw ridge", **"loc-ans":** "Hugo Weaving", **"cond":** "Trinity College » University of Michigan || What university did Watts Humphrey attend?"}

In this example, old knowledge ("Trinity College") is replaced with new knowledge ("University of Michigan") for the same question. The "rephrase" field evaluates the model's generalization, while "loc" assesses the locality of the model's output. The datasets are divided into old and new knowledge, with the same format maintained for effective supervised fine-tuning. Below are examples of old and new knowledge in an instruction-based format:

Old knowledge:

{**"instruction"**: "What university did Watts Humphrey attend?", **"input"**: "", **"output"**: "Trinity College" }

λ	Reliability	Generality	Locality
0	46.07	46.81	76.65
0.3	46.21	47.59	77.74
0.5	46.11	47.19	78.42
0.7	46.25	47.70	77.41
0.9	46.40	46.87	77.06

Table 5: Performance comparisons of GeoEdit equipped with different λ.

ϕ_1	ϕ_2	Reliability	Generality	Locality
87°	93°	48.07	49.13	74.81
75°	105°	44.57	44.81	83.63
80°	100°	45.86	46.55	80.26
85°	95°	46.11	47.19	78.42

Table 6: Performance comparisons of GeoEdit equipped with different ϕ.

New knowledge:

{**"instruction"**: "What university did Watts Humphrey attend?", **"input"**: "", **"output"**: "University of Michigan" }

It's important to note that old knowledge represents correct real-world facts, while new knowledge is deliberately incorrect, ensuring that the original model has not previously learned it. This setup avoids ambiguity in determining whether the new knowledge was already part of the model's prior knowledge (Shi et al., 2024).

B Baseline Details

We evaluate our GeoEdit method against a range of fine-tuning and locate-and-edit-based approaches.

For fine-tuning methods, we first compare our approach with full fine-tuning (**Full-FT**) and **LoRA** (Hu et al., 2021). LoRA (Low-Rank Adaptation) introduces small, trainable matrices into each layer of the model, enabling efficient adaptation while keeping most of the pre-trained parameters frozen. We also evaluate **FT-c** (Lu et al., 2021b), a fine-tuning method that applies an L_∞ constraint to help retain irrelevant knowledge. Additionally, we compare with the **F-Learning** method (Ni et al., 2024), which first forgets outdated knowledge to facilitate the incorporation of new information.

For locate-and-edit-based methods, we start by evaluating **MEND** (Mitchell et al., 2022), which learns a hypernetwork to generate weight updates by decomposing fine-tuning gradients. We also experiment with **ROME** (Meng et al., 2022), a method that updates specific factual associations through causal intervention. Additionally, we compare with **MEMIT** (Liu et al., 2023), a method designed for directly updating large-scale memories. Finally, we include **RECT** (Gu et al., 2024), which regularizes edit updates by imposing constraints on the complexity of the weight changes.

C Implementation Details

Here we will introduce more completion details and settings of experiments. First, we used LLAMA2-7B and LLAMA-7B as the base models, and then we trained the base model on the old knowledge for 3 epochs by full fine-tuning to simulate an original model that has fully learned old knowledge for our experiments. This makes the forgetting operation more reasonable and effective, and at the same time tries to avoid the problem of being unable to determine whether the new knowledge output by the LLM is learned from the data or commanded by itself as mentioned above.

We use LoRA to enhance the efficiency of fine-tuning, the hyperparameters were set as follows: $r = 8$, $\alpha = 32$, dropout = 0.05, with the targeting modules being [q_proj, k_proj, v_proj, o_proj, up_proj, down_proj]. The encoder and decoder consists of 2-layer MLPs with dimensions $[4096 \rightarrow 2048, 2048 \rightarrow 512]$ and $[512 \rightarrow 2048, 2048 \rightarrow 4096]$, respectively, where d_{latent} is set to 512. We set λ in Eq. (9) to 0.5, ϕ_1 and ϕ_2 in Eq. (11) to 85° and 95°, respectively. During testing, we use a greedy decoding strategy to ensure the uniqueness of the model's output. All experiments were conducted on a setup using 4 × A100-80G GPUs.

It is worth noting that we used the same hyperparameters across different datasets and backbones, demonstrating the generalizability of our method without requiring extensive hyperparameter tuning for each specific setting.

D Additional Results

D.1 Comparison of Different Angle Extraction Methods

Our GeoEdit framework allows using various dimensionality reduction strategies to extract angle information between task vectors. It's crucial to emphasize that these strategies are simply options or alternatives. The core value of our framework lies in its innovative approach to geometric editing and the proven effectiveness of this method. For example, one could directly apply PCA or t-SNE

to the original high-dimensional vectors. However, our empirical results show that the best angle information is achieved by first applying the autoencoder for denoising, followed by using t-SNE for angle calculation. The related ablation study results are shown in the Table 7.

D.2 Evaluating Fluency and Consistency Scores

In Table 8, we provide the Fluency and Consistency scores for LLAMA-7B and LLAMA2-7B, calculated using the formulas in ROME. Our GeoEdit method consistently outperforms F-Learning in both LoRA-based and full fine-tuning settings, with an average improvement of 33.2 in Fluency and 2.2 in Consistency scores.

D.3 Results on Different Backbone Models

We have conducted additional experiments using different backbone models, including GPT2-XL (1.5B), Qwen 2.5 (7B), and LLaMA3 (8B). The results in Table 9 show that GeoEdit consistently outperforms RECT and F-Learning across the five key metrics for each backbone model, demonstrating its generalizability across different LLM architectures.

D.4 The Effect of GeoEdit on Model Generalization

To assess the impact of GeoEdit on generalization, we evaluated mathematical reasoning ability using GSM8K and MATH, as well as broader knowledge retention using MMLU and NLI. The results, summarized in the Table 10, indicate that AlphaEdit, which utilizes additional Wikipedia data, experiences less degradation on MMLU and NLI. Conversely, GeoEdit shows less decline on GSM8K and MATH, demonstrating its effective retention of general knowledge.

D.5 Analyzing the Importance of Task Vectors in Different MLP Layers for Knowledge Editing

We conducted experiments to analyze the importance of different layers for knowledge editing. For example, in LLaMA-2-7B (which has 32 layers), we separately edited the parameters in the lower (1-11), middle (12-22), and upper (23-32) layers of the MLP using LoRA. The results on the ZsRE dataset are shown in Table 11.

These results indicate that editing only the top layers yields the poorest performance, suggesting that most of the model's knowledge is stored in the mid-early MLP layers. This is consistent with findings from ROME, and the important parameters are more concentrated in the lower layers, as shown in Figure 4 of the paper.

D.6 Comparison with In-Context-Learning-Based Methods

We evaluated IKE (Zheng et al., 2023) on both benchmarks using LLaMA2-7B, with 32 examples and demonstrations selected based on cosine similarity. The results are shown in Table 12.

The results show that IKE, which does not modify model parameters, minimizes unintended side effects, achieving the highest locality scores, but its reliability and generality are weaker compared to F-Leanring and GeoEdit. While in-context learning methods appear efficient, they face challenges such as the need for large demonstration corpora and sensitivity to factors like demonstration count, selection method, and prompt formatting. These issues can result in performance variability, making them less stable in practice. In contrast, an edited model, whether fine-tuning-based or locate-and-edit, tends to provide more consistent performance and is generally easier to use.

D.7 Sensitivity Analysis for Hyperparameters

The proposed framework incorporates two key hyperparameters: λ, which balances the autoencoder loss in Eq. (9), and ϕ, which defines the thresholds for dividing different editing strategies. Our analysis aims to assess the impact of varying these hyperparameters on the performance of our method, with tests conducted on the ZsRE dataset using LLaMA2-7B backbone model (LoRA fine-tuning).

As shown in Table 5, we determine that the optimal setting for λ is 0.5. Regarding the selection of the threshold for dividing editing strategies, the article sets ϕ_1 and ϕ_2 to $85°$ and $95°$, respectively. Table 6 below shows the model's performance with varying thresholds for ϕ. It can be seen that as the range between ϕ_1 and ϕ_2 increases, meaning more updates are masked, this better prevents interference with the model's general knowledge but limits the learning of new knowledge. This results in an increase in locality but a decrease in reliability. Conversely, narrowing the range of ϕ_1 and ϕ_2 enhances the model's ability to update, but it also impacts its generalization ability. Therefore, we choose the range of $85°$ to $95°$ as the optimal balance for masking, achieving the best trade-off

Method	Different Angle Calculation Methods	Reliability	Generality	Locality
F-Learning	-	46.9	46.2	72.5
GeoEdit	PCA	30.2	27.4	87.7
	t-SNE	45.9	46.7	75.5
	Auto-Encoder + t-SNE (ours)	46.1	47.2	78.4

Table 7: Comparison of different angle extraction methods.

Method	LLAMA2-7B		LLAMA-7B	
	Fluency (↑)	**Consistency** (↑)	**Fluency** (↑)	**Consistency** (↑)
Original model	624.69	26.45	622.94	25.21
LoRA	509.56	18.55	509.01	17.68
Full-FT	251.14	12.33	254.42	11.75
ROME	434.11	8.81	431.86	10.39
RECT	530.82	24.47	532.29	23.32
F-Learning	557.63	26.61	556.02	25.76
AlphaEdit	581.58	**30.51**	581.68	28.52
GeoEdit	**585.98**	29.81	**584.29**	**28.90**

Table 8: Comparison of different methods for LLAMA2-7B and LLAMA-7B fluency and consistency.

between learning new knowledge and preserving general knowledge.

Method	Backbone	Reliability (↑)	Generality (↑)	Locality (↑)	Fluency (↑)	Consistency (↑)
RECT		63.35	41.55	25.99	529.66	26.67
F-Learning	GPT2-XL	64.51	42.56	30.29	544.73	32.34
GeoEdit		**66.19**	**44.43**	**39.55**	**575.11**	**34.92**
RECT	-	71.80	47.10	29.46	590.31	27.97
F-Learning	Qwen 2.5	**75.06**	48.66	35.31	585.06	27.46
GeoEdit		74.37	**50.15**	**45.41**	**611.22**	**30.94**
RECT		66.74	43.78	27.38	558.38	26.03
F-Learning	LLaMA3	70.69	47.73	33.20	575.16	29.97
GeoEdit		**71.42**	**48.63**	**41.43**	**602.38**	**32.15**

Table 9: Comparison of different methods with varying backbones for various metrics.

Dataset	Original Model (LLaMA2-7B)	F-Learning	AlphaEdit	GeoEdit
GSM8K	3.14	0.80	1.55	**1.85**
MATH	4.32	0.92	2.17	**3.01**
MMLU	27.74	17.26	**21.61**	19.72
NLI	67.37	32.60	**46.68**	42.94

Table 10: Comparison of different methods on various datasets. The best performing method for each dataset is highlighted in bold.

Method	Reliability	Generality	Locality
GeoEdit	54.68	53.45	81.67
GeoEdit (Lower Layer)	49.12	48.46	75.21
GeoEdit (Middle Layer)	47.42	46.18	76.96
GeoEdit (Upper Layer)	46.13	45.31	74.50

Table 11: Comparison of GeoEdit across different layers.

Method	Benchmark	Reliability	Generality	Locality
F-Learning		84.65	81.51	70.92
IKE	ZsRE	77.66	76.50	**98.30**
GeoEdit		**85.21**	**82.43**	75.71
F-Learning		**69.53**	45.56	28.41
IKE	Counterfact	60.42	41.71	**97.24**
GeoEdit		68.34	**46.53**	37.73

Table 12: Comparison of F-Learning, IKE, and GeoEdit across different benchmarks. The best performance for each metric is highlighted in bold.