Orthogonal Representation Editing: Decoupling Semantic Entanglement in Batch Knowledge Editing of LLMs

Wenhao Yu[1,2], Zhicong Lu[3], Bo Lv[4], Fangyin Ma[1], Kaiwen Wei[5], Shihao Yang[1], Nayu Liu[1]†

[1]School of Computer Science and Technology, Tianjin University
[2]Kexin Technology, [3]University of Chinese Academy of Sciences
[4]Tencent Hunyuan, [5]College of Computer Science, Chongqing University
nyliu@tju.edu.cn

Abstract

Knowledge editing aims to efficiently update factual information in Large Language Models (LLMs) without full retraining. However, existing methods still suffer from performance degradation in batch knowledge editing. We identify that semantic representation entanglement, such as overlapping concepts and shared syntactic patterns, accumulates interference in the representation space and reduces editing precision. To bridge this gap, in this paper, we propose Orthogonal Representation Editing (ORE), which performs edits in the hidden representation space of LLMs by constructing a general semantic subspace and enforcing orthogonal constraints on edit vectors, effectively decoupling semantic entanglement. Furthermore, we introduce a gated non-linear representation head to enable adaptive learning of editing locations and precise control over knowledge injection. Extensive experiments show that ORE outperforms existing methods and achieves superior performance in cross-lingual knowledge editing scenarios. We release our code at https://github.com/YVVH/ORE.

1 Introduction

Large Language Models (LLMs) have demonstrated strong capabilities in question answering and reasoning (Mann et al., 2020; Brown et al., 2020). Despite these advances, their parameters remain inherently static, limiting their ability to accommodate the continual evolution of real-world knowledge. Knowledge editing has therefore emerged as a crucial research direction for efficiently updating specific facts in pretrained models without retraining from scratch (Yao et al., 2023; Wang et al., 2024b; Gupta et al., 2024).

Existing knowledge editing methods can be broadly categorized into two paradigms. Parameter-preserving methods (Huang et al., 2023; Hernandez et al., 2023; Hartvigsen et al., 2023; Scialanga et al., 2025; Liu et al., 2025; Bi et al., 2025) maintain a frozen backbone and introduce auxiliary modules to update knowledge, while parameter-modifying methods (Meng et al., 2022; Fang et al., 2025; Jiang et al., 2025a) directly locate and update a subset of model parameters to inject new facts. Notably, MEMIT (Meng et al., 2023) extends parameter-modifying approaches to batch knowledge editing, enabling the simultaneous injection of thousands of knowledge entries by distributing update residuals across multiple layers.

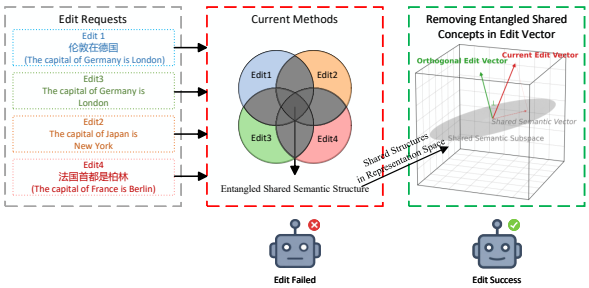

Figure 1: Edits with different factual targets activate overlapping regions in representation space, finally leading to interference.

Despite their empirical success, existing knowledge editing methods exhibit performance degradation in such batch editing settings (Meng et al., 2023). To explore the reasons, we observe that edits targeting different facts are not isolated in the representation space but instead occupy overlapping regions. For example, as shown in Fig. 1, the edits "The capital of Japan is Tokyo" and "The capital of France is Berlin" follow the same template "[Country's capital] is [City]" and invoke the shared concept of "capital," leading to interference within a common region of the representation space.

We attribute this degradation to **semantic representation entanglement**. In practice, many edit requests share overlapping concepts or general syntactic templates, which induces interference among their corresponding representations in the hidden

† Corresponding author.

Findings of the Association for Computational Linguistics: ACL 2026, pages 37967–37979

space. Existing methods struggle to disentangle fact-specific information from such general semantic structures; as the editing size grows, accumulated interference substantially degrades editing precision. This issue is further exacerbated in cross-lingual settings (Beniwal et al., 2024; Sun et al., 2025), where shared multilingual semantic spaces allow edits to propagate along semantic directions into non-target languages, resulting in unintended cross-lingual interference. Empirical evidence supporting these observations is presented in **Section 2**.

Motivated by these observations, we propose **Orthogonal Representation Editing (ORE)**, a representation-based knowledge editing framework guided by geometric constraints (Hernandez et al., 2023; Cai and Cao, 2024; Xu et al., 2025). ORE operates directly in the hidden representation space and aims to decouple editing directions, thereby mitigating interference induced by shared semantic patterns and enabling more reliable knowledge updates (Wang et al., 2025). Specifically, ORE leverages a set of irrelevant but structurally similar samples to estimate a *general semantic subspace*. Each edit vector is then orthogonalized by subtracting its projection onto this subspace, ensuring that knowledge updates occur along directions independent of shared semantics. To realize orthogonal editing in practice, ORE builds upon representation fine-tuning (ReFT) (Wu et al., 2024). In addition, to move beyond linear interventions and manual positional priors, ORE introduces a gated non-linear representation head that adaptively determines *when* and *where* to intervene, enabling precise knowledge injection with minimal impact on general capabilities. Extensive experiments demonstrate that ORE consistently outperforms existing methods and remains robust in cross-lingual knowledge editing scenarios. In summary, the main contributions of this paper are as follows:

- We identify **semantic representation entanglement** as a fundamental limitation of batch knowledge editing, where interference accumulates in the representation space and degrades editing performance.
- We propose Orthogonal Representation Editing (ORE), which constructs a general semantic subspace and performs orthogonal, gated interventions in the representation space to decouple shared semantic entanglement.
- Extensive experiments demonstrate that ORE achieves strong and consistent performance across multiple benchmarks and remains effective in challenging cross-lingual knowledge editing scenarios.

2 Observation of General Semantic Representation Entanglement

To empirically verify the hypothesis that general semantic structure entanglement leads to performance degradation in batch editing, we designed a controlled experiment to compare the performance differences of existing methods on random data versus data with high general semantic entanglement.

2.1 Data Settings

Entangled Samples: We utilized Gemini 3 to construct 200 cross-lingual samples with identical syntactic structures and highly correlated semantics based on the theme of "Capital," including 100 French samples and 100 Chinese samples. All samples follow the pattern "The capital of [Country] is [City]." Under this setting, different subject entities activate a general semantic structure within the model's representation space. The entangled samples are divided into three groups: the French group (100 French samples), the Chinese group (100 Chinese samples), and the Cross-lingual group (50 French and 50 Chinese samples).
Random Samples: We randomly sampled 100 entries from the ZsRE (Levy et al., 2017) dataset and translated them into French and Chinese languages. These samples cover diverse relation types and distinct semantic categories, possess strong semantic independence, and serve as a control group.

2.2 Observations

We employed MEMIT (Meng et al., 2023) and AlphaEdit (Fang et al., 2025), currently representative editing methods as baselines to edit the aforementioned two groups of data on the LLaMA-3-8B model and recorded the Editing Success. As shown in Figure 2, MEMIT demonstrated robust performance on the random group, achieving a high success rate of 86%. However, on the entangled samples, MEMIT's editing accuracy declined sharply: the success rate for the French group was only 71%, for the Chinese group it was 64%, and for the Cross-lingual group, it further dropped to 56%. Similar observations also occurred in AlphaEdit, which indicate that performance degrades when facing samples with general concept entanglement.

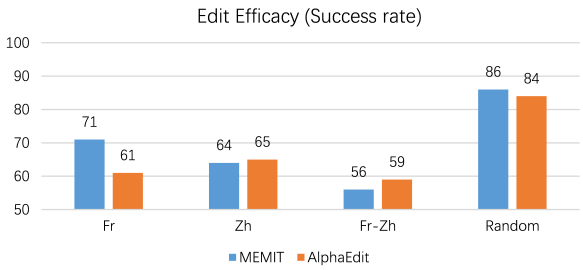

Figure 2: Editing efficacy of MEMIT and AlphaEdit on entangled (Fr, Zh, Fr-Zh) and random samples. Performance drops on entangled samples, especially in cross-lingual (Fr-Zh) settings.

This result supports our hypothesis: in batch editing, shared general semantic structures cause update vectors to conflict and accumulate noise within the representation subspace, thereby leading to a decline in the performance of existing methods.

3 Methodology

3.1 Overview of ORE

Motivated by the empirical observations in Section 2, we propose ORE, a representation editing framework based on geometric constraints. As shown in Figure 3, ORE comprises: (1) Representation Subspace Orthogonalization, which aims to explicitly construct a general semantic subspace and strip away general semantic noise from edit vectors via orthogonal projection, thereby mitigating the entanglement between knowledge items from a geometric perspective. (2) Gated Non-linear Representation Head, which is designed to adapt representation fine-tuning for vector editing. It utilizes a non-linear bottleneck and a dynamic gating mechanism to perform fine-grained intervention and adaptive injection on representation vectors.

3.2 Representation Subspace Orthogonalization

To address general semantic entanglement in batch knowledge editing, we explicitly model shared semantics as a subspace in the representation space and constrain edits to lie outside this subspace.

3.2.1 Construction of General Semantic Subspace

To extract the general semantic structure involved in batch editing, we construct a sample set D_{gs} consisting of N samples that share similar structures with the target edits while being factually unrelated. Concretely, these samples are randomly drawn from the ZsRE (Levy et al., 2017) and CounterFact (Meng et al., 2022) datasets, excluding all instances used for training or evaluation, following the same prompt pattern as the target edits, but involving different subject–object pairs, ensuring that no target facts are included.

We extract their hidden states at the target layer l on the backbone model, denoted as $H_{gs} = \{h_1^{(l)}, h_2^{(l)}, ..., h_N^{(l)}\}$. To capture the dominant directions corresponding to these shared semantics, we perform Principal Component Analysis (PCA) (Abdi and Williams, 2010) on H_{gs} and select the top k principal components to form an orthogonal basis matrix $U_{gs} \in \mathbb{R}^{d\times k}$. The subspace spanned by U_{gs} is defined as the General Semantic Subspace S_{gs}.

3.2.2 Orthogonal Constraint on Edit Vectors

To operationalize subspace orthogonalization, we compute two representations of the source input: the source representation produced by the original frozen model, denoted as $h_{src,orig}^{(l)}$, and the source representation produced by the current edited model, denoted as $h_{src}^{(l)}$. Moreover, $h_{pred}^{(l)}$ and $h_{alt}^{(l)}$ denote the representations at layer l corresponding to the model's prediction statement and the desired alternative statement, respectively. Specifically, these representations are obtained by using the declarative statement of the target knowledge as the input prompt and extracting the hidden states at the last token position. For instance, given an edit request where the prompt is "What university did Watts Humphrey take part in?", if the model originally predicts "Trinity College" but the target is "University of Michigan", we construct the declarative statements "Watts Humphrey attended Trinity College." for $h_{pred}^{(l)}$ and "Watts Humphrey attended the University of Michigan." for $h_{alt}^{(l)}$. We then feed these full sentences into the model and extract the vectors from their respective final token positions (Meng et al., 2022). The edit vector $\Delta^{(l)}$ is computed as:

$$\Delta^{(l)} = h_{alt}^{(l)} - h_{pred}^{(l)}, \tag{1}$$

which captures the direction required to modify the model's behavior from the current prediction toward the target fact. We then remove the projection of $\Delta^{(l)}$ onto the general semantic subspace S_{gs} to obtain an orthogonalized edit direction:

$$\Delta_{orth}^{(l)} = \Delta^{(l)} - U_{gs}U_{gs}^{\top}\Delta^{(l)}, \tag{2}$$

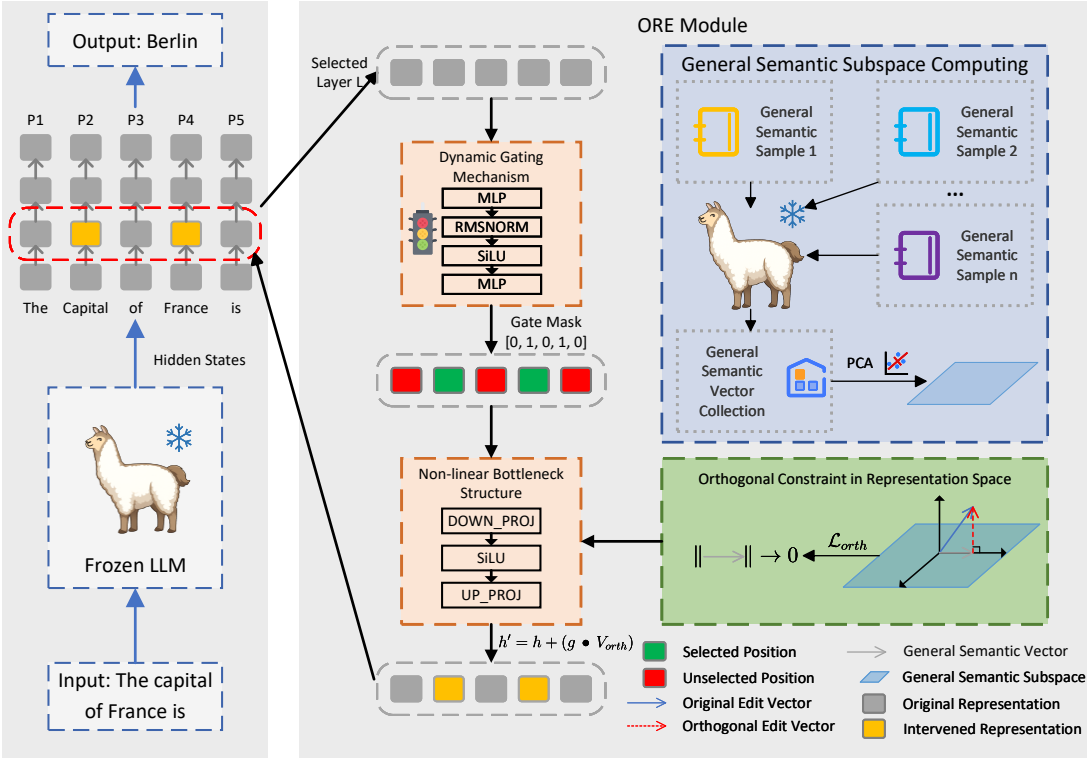

Figure 3: Overview of the proposed ORE framework. It edits a frozen LLM by applying gated, non-linear representation interventions at selected layers, orthogonalizing edit vectors against a general semantic subspace, and updating hidden states with the resulting orthogonalized edits.

which is subsequently used to define the target representation as

$$h_{tgt}^{(l)} = h_{src,orig}^{(l)} + \Delta_{orth}^{(l)}. \tag{3}$$

The training objective then encourages $h_{src}^{(l)}$ to align with $h_{tgt}^{(l)}$.

3.3 Non-linear Gated Representation Fine-tuning

To enable vector-level intervention within the model and implement Representation Subspace Orthogonalization, we utilize ReFT (Wu et al., 2024) to execute this process. ReFT in its standard form was not originally designed for knowledge editing, and thus provides limited support for fine-grained and adaptive intervention. Therefore, ORE introduces a non-linear bottleneck structure to finely adjust semantic expressiveness and incorporates a dynamic gating mechanism to enable adaptive selection of intervention positions.

3.3.1 Non-linear Bottleneck Structure

To enhance the ability of edit vectors to capture fine-grained semantic features, we designed a non-linear bottleneck structure based on low-rank projection (Hu et al., 2022). Specifically, for the input hidden state $h^{(l)}$, it is first mapped to a low-dimensional manifold via a bias-free dimensionality reduction matrix $W_{down} \in \mathbb{R}^{r \times d}$, then passed through a SiLU non-linear activation function, and finally mapped back to the original representation space via a bias-free dimensionality expansion matrix $W_{up} \in \mathbb{R}^{d \times r}$, formalized as follows:

$$v_{edit} = W_{up} \cdot \text{SiLU}(W_{down} \cdot h^{(l)}), \tag{4}$$

where d is the hidden layer dimension, r is the bottleneck rank, and $r \ll d$.

3.3.2 Dynamic Gating Mechanism

To overcome the flexibility limitations imposed by manually specifying edit positions and to minimize interference with irrelevant knowledge, we introduce a dynamic gating network parallel to the bottleneck structure. This module aims to adaptively generate a binary gating coefficient $g \in \{0, 1\}$ based on the current hidden state, thereby achieving precise and sparse intervention at specific knowledge positions. Specifically, the input state $h^{(l)}$ is first projected to an intermediate dimension via a linear layer, followed by normalization, and then passed through a SiLU activation function to obtain gating features. A final linear projection followed

by a sigmoid function σ then produces a soft gating score s in the range of $(0, 1)$. The calculation process is defined as follows:

$$s = \sigma(W_{g2} \cdot \text{SiLU}(\text{Norm}(W_{g1} h^{(l)}))), \quad (5)$$

To obtain the final discrete gating coefficient g, we introduce an indicator function $I(\cdot)$ to apply threshold truncation to s. Setting the threshold as τ, g is defined as:

$$g = I(s > \tau) = \begin{cases} 1, & \text{if } s > \tau \\ 0, & \text{if } s \leq \tau \end{cases}. \quad (6)$$

We define the intervention function $\Phi(h^{(l)})$ as:

$$\Phi(h^{(l)}) = h^{(l)} + g \cdot v_{edit}. \quad (7)$$

To ensure gradient continuity through the non-differentiable indicator $I(\cdot)$, we adopt the straight-through estimator (STE) during training.

3.4 Loss Functions

ORE comprises the following supervision mechanisms to facilitate representation fine-tuning:

Orthogonal Projection Loss: To constrain the model's output vectors to be orthogonal to the general semantic subspace, we treat the sum of the original representation and the orthogonalized edit vector as the target direction. We use cosine similarity to constrain the current model's output representation $h_{src}^{(l)}$ to approach this target:

$$\mathcal{L}_{orth} = 1 - \cos(h_{src}^{(l)}, h_{src,orig}^{(l)} + \Delta_{orth}^{(l)}). \quad (8)$$

Gating Supervision Loss: To ensure that the dynamic gating network can precisely localize the key positions where knowledge is stored while remaining silent in non-key regions and on irrelevant samples, we employ Binary Cross-Entropy (BCE) loss for explicit supervision. Concretely, we utilize the syntactic structure information of the input text to construct a target gating mask $m \in \{0, 1\}^T$, where T is the sequence length. The supervision signals are divided into two scenarios: First, we set the mask values corresponding to the subject (Meng et al., 2022, 2023) token positions to 1 and the remaining positions to 0. This guides s to approach 1 in the subject region, thereby activating the indicator function to output $g = 1$. For non-subject regions in edit samples, as well as all irrelevant samples used to preserve general capabilities, we set the mask values entirely to 0. This guides the gating network to output low scores in these regions, thereby closing the intervention channel.

Based on the above definitions, the gating loss $\mathcal{L}_{gate}$ is defined as the BCE loss between the predicted soft gating score s and the target mask m:

$$\mathcal{L}_{gate} = -\frac{1}{T} \sum_{t=1}^{T} [m_t \log(s_t) + (1 - m_t) \log(1 - s_t)]. \quad (9)$$

where s_t is the soft gating score.

In addition, cross-entropy and KL-divergence losses are employed to enforce the desired output behavior while preserving locality. Specifically, the KL-divergence loss constrains the output distribution of the edited model to remain close to that of the original frozen model on prompts unrelated to the target edits. The overall training objective is formulated as:

$$\mathcal{L}_{total} = \lambda_1 \mathcal{L}_{ce} + \lambda_2 \mathcal{L}_{kl} + \lambda_3 \mathcal{L}_{gate} + \lambda_4 \mathcal{L}_{orth}, \quad (10)$$

where λ_1, λ_2, λ_3, and λ_4 are hyperparameters that balance the contributions of each loss term.

4 Experiments

4.1 Experiment Settings

Datasets. We evaluate ORE on three widely used knowledge editing benchmarks: ZsRE (Levy et al., 2017), CounterFact (Meng et al., 2022), and Bi-ZsRE (Wang et al., 2024a). ZsRE is a fact-based question-answering dataset commonly used to assess precise factual updates. CounterFact is a large-scale and challenging benchmark for counterfactual knowledge editing, featuring diverse relations, entities, paraphrased prompts, and semantically related but factually independent neighborhood samples. Bi-ZsRE is a cross-lingual extension of ZsRE with parallel Chinese–English question-answer pairs, which we use to evaluate ORE in cross-lingual editing scenarios.

Implementation Details and Metrics. All experiments are conducted on LLaMA-3-8B and Qwen-2.5-7B models. Please refer to the **Appendix C** for hyperparameters and more implementation details. Following prior works (Meng et al., 2023; Fang et al., 2025), we evaluate models using three standard metrics: **Efficacy**, **Generality**, and **Specificity**. Efficacy measures whether the edited fact is correctly produced for the original prompt; Generality evaluates the robustness of the edited knowledge under paraphrased prompts; and Specificity

Model	Method	ZsRE				CounterFact			
		Eff.↑	Gen.↑	Spe.↑	Avg.↑	Eff.↑	Gen.↑	Spe.↑	Avg.↑
LLaMA-3-8B	FT	26.50	25.93	15.16	22.53	**99.75**	88.65	39.62	76.01
	ROME	42.58	39.84	30.70	37.71	9.60	11.45	88.00	36.35
	MEMIT	87.13	84.18	**32.11**	67.81	94.55	69.55	88.31	84.14
	PRUNE	67.37	62.48	27.51	52.45	98.05	**95.18**	74.33	89.19
	RECT	78.49	74.88	32.00	61.79	75.25	50.88	**89.24**	71.79
	NSE	45.69	44.95	31.47	40.70	85.70	53.95	88.35	76.00
	AlphaEdit	87.37	83.93	31.95	67.75	98.85	92.90	67.28	86.34
	ReFT	47.65	46.48	22.82	38.98	83.50	52.32	40.26	58.69
	ORE (Ours)	**94.20**	**88.98**	29.90	**71.03**	98.70	92.21	83.10	**91.34**
Qwen2.5-7B	FT	36.96	35.87	31.65	34.83	99.75	72.32	40.22	70.76
	ROME	36.52	35.42	38.34	36.76	14.00	16.75	**86.04**	38.93
	MEMIT	95.53	90.96	**41.72**	76.07	99.50	92.45	83.61	91.85
	PRUNE	69.92	65.54	27.28	54.25	99.55	**98.15**	72.67	90.12
	RECT	91.25	83.86	39.86	71.66	98.10	84.92	84.41	89.14
	NSE	49.65	48.78	40.81	46.41	57.45	51.35	85.18	64.66
	AlphaEdit	96.49	91.47	39.04	75.67	**99.80**	95.80	82.88	92.83
	ReFT	49.81	47.25	37.08	44.71	77.25	48.95	48.40	58.20
	ORE (Ours)	**99.85**	**94.21**	35.37	**76.48**	99.18	95.50	84.73	**93.14**

Table 1: Comparison of 2000 edits on ZsRE and CounterFact.

assesses whether the editing operation avoids affecting factually unrelated knowledge.

Baselines. We compare ORE with the following baselines. **FT** (Zhu et al., 2020) directly fine-tunes selected model parameters on edit samples using cross-entropy loss. **ROME** (Meng et al., 2022) identifies knowledge-bearing neurons via causal tracing and injects rank-one updates into MLP layers, while **MEMIT** (Meng et al., 2023) extends ROME to support batch editing by distributing update residuals across multiple layers. **PRUNE** (Ma et al., 2025) constrains parameter perturbations by limiting the condition number of the edit matrix to reduce interference during sequential edits, and **RECT** (Gu et al., 2024) improves robustness through consistency regularization to prevent overfitting. **AlphaEdit** (Fang et al., 2025) applies null-space projection to inject updates while preserving existing knowledge. **NSE** (Jiang et al., 2025b) edits models by selectively updating activation-critical neurons based on target hidden states computed from frozen parameters. **ReFT** (Wu et al., 2024) learns lightweight intervention functions that manipulate hidden representations in low-rank subspaces of a frozen model at inference time.

4.2 Performance Analysis

Tables 1 and 2 report comparisons between ORE and representative knowledge editing methods on ZsRE and CounterFact. From the results, we observe that: (1) Across datasets and model backbones, ORE consistently achieves the competitive performance in terms of editing success and generalization, demonstrating its effectiveness in large-scale batch editing settings. This advantage mainly stems from alleviating general semantic entanglement, which encourages editing directions to focus on semantic dimensions directly relevant to the target facts. ORE is comparatively less dominant in specificity, as its edits are regulated by soft constraints imposed in the representation space rather than hard parameter-level isolation, making it inherently weaker than parameter-editing methods in preserving unrelated knowledge. (2) Compared with conventional representation fine-tuning, ORE exhibits improved stability and overall performance, validating the effectiveness of subspace orthogonalization combined with the non-linear bottleneck and dynamic gating design. For completeness, we also report experimental results on sequential knowledge editing in **Appendix E**.

4.3 Analysis of Anti-Interference Capability

To validate ORE's ability to eliminate common semantic entanglement from a geometric perspective, we conducted experiments on the shared structure dataset "The capital of [Country] is [City]" constructed in Section 2. Beyond standard Efficacy, we measured the cosine similarity between each method's edit vectors and the common semantic subspace S_{gs}, where higher similarity indicates greater interference from shared seman-

Model	Method	ZsRE				CounterFact			
		Eff.↑	Gen.↑	Spe.↑	Avg.↑	Eff.↑	Gen.↑	Spe.↑	Avg.↑
LLaMA-3-8B	FT	37.20	36.52	**39.94**	37.89	**98.95**	91.94	38.73	76.54
	ROME	41.45	39.35	30.64	37.15	8.32	10.47	88.25	35.68
	MEMIT	87.46	84.12	31.67	67.75	96.4	76.77	86.57	86.58
	PRUNE	41.73	39.50	20.94	34.06	95.29	87.75	67.48	83.51
	RECT	68.08	64.50	30.91	54.50	67.18	45.69	**88.91**	67.26
	NSE	45.21	44.47	30.58	40.09	85.22	52.33	87.49	75.01
	AlphaEdit	86.35	82.73	31.10	66.73	94.07	75.15	84.69	84.64
	ReFT	44.57	43.38	21.28	36.41	80.86	55.37	40.83	59.02
	ORE (Ours)	**93.24**	**88.22**	29.35	**70.27**	97.78	**96.81**	76.63	**90.41**
Qwen2.5-7B	FT	33.30	31.26	26.72	30.43	**99.98**	75.60	38.83	71.47
	ROME	35.41	34.44	38.29	36.05	12.72	15.31	**85.98**	38.00
	MEMIT	94.52	90.29	38.36	74.39	99.42	92.39	81.20	91.00
	PRUNE	69.10	65.92	**42.00**	59.01	98.36	**96.87**	64.57	86.60
	RECT	91.01	83.99	40.71	71.90	97.72	80.04	83.33	87.03
	NSE	49.15	48.32	40.73	46.07	56.72	50.14	84.01	63.62
	AlphaEdit	95.23	87.75	38.97	73.98	99.76	94.73	79.16	91.22
	ReFT	49.95	47.51	35.30	44.25	77.30	51.99	44.91	58.07
	ORE (Ours)	**99.83**	**93.85**	30.64	**74.77**	99.10	93.56	83.64	**92.10**

Table 2: Comparison of 5000 edits on ZsRE and CounterFact.

Method	Eff.↑	Gen.↑	Spe.↑	Avg.↑
ORE	95.54	92.18	29.03	72.25
-$\mathcal{L}_{orth}$	93.34	88.77	29.18	70.43
-SiLU	93.00	90.52	27.46	70.33
-Gate	95.13	85.78	28.64	69.85

Table 3: Ablation study of ORE with 1,000 edits on ZsRE.

tics. As shown in Figure 4, MEMIT's post-edit vectors remain highly aligned with S_{gs}, suggesting that its updates fail to avoid shared semantic regions. In contrast, ORE, via representation orthogonal projection, achieves 92% Efficacy while reducing cosine similarity to only one-sixth of MEMIT's, demonstrating its effectiveness in mitigating representation entanglement and enhancing anti-interference capability in batch editing.

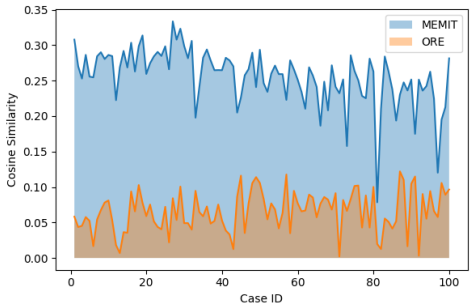

Figure 4: Cosine similarity between edit representations and the general semantic subspace across sequential edits. MEMIT (blue) consistently shows higher similarity, indicating stronger entanglement with shared semantics, while ORE (orange) maintains substantially lower similarity throughout the editing process.

4.4 Ablation Study

We conduct an ablation study on ZsRE under a batch editing setting with 1,000 samples to evaluate the contributions of ORE's key components. As shown in Table 3, removing the representation orthogonalization loss (-$\mathcal{L}_{orth}$) decreases both efficacy and generality, indicating that orthogonalization is crucial for decoupling shared semantic patterns and mitigating cumulative interference. Eliminating the non-linear activation in the gated representation head (-SiLU) leads to a drop in overall performance. Disabling the dynamic gating mechanism (-Gate) reduces generality and specificity, highlighting the importance of adaptive gating for accurately localizing knowledge-carrying positions while avoiding unnecessary perturbations in irrelevant regions. Figure 5 further shows the cosine similarity between edit vectors and the general semantic subspace after removing the representation orthogonalization, illustrating that the orthogonal constraint effectively decouples the edited facts from shared semantic patterns.

4.5 Cross-Lingual Editing Scenarios

Furthermore, we evaluated the performance of ORE in cross-lingual editing scenarios. Table 4 shows the comparative results with an editing batch size of 1600. The experimental results demonstrate

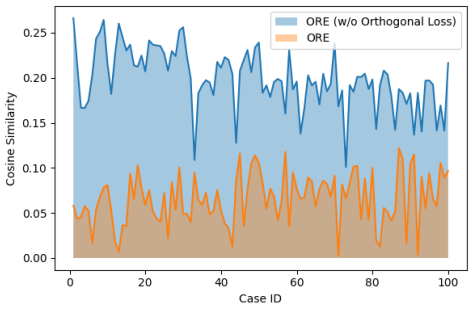

Figure 5: Ablation study of Representation Subspace Orthogonalization Loss.

Model	Method	Bi-ZsRE			
		Eff.↑	Gen.↑	Spe.↑	Avg.↑
LLaMA-3-8B	FT	48.52	45.06	22.47	38.68
	ROME	42.08	40.80	30.18	37.69
	MEMIT	76.93	74.55	**31.40**	60.96
	PRUNE	53.09	50.59	23.12	42.27
	RECT	67.78	65.06	31.39	54.74
	NSE	49.46	48.51	30.76	42.91
	AlphaEdit	76.14	73.30	31.23	60.22
	ReFT	53.67	52.29	24.49	43.48
	ORE (Ours)	**85.10**	**81.84**	29.14	**65.36**
Qwen2.5-7B	FT	47.25	42.39	33.32	40.99
	ROME	37.02	36.66	39.72	37.80
	MEMIT	79.45	76.52	40.77	65.58
	PRUNE	67.39	64.91	**43.07**	58.46
	RECT	76.36	72.23	41.00	63.20
	NSE	52.54	51.27	38.79	47.53
	AlphaEdit	82.28	78.49	39.81	66.86
	ReFT	52.89	50.37	35.27	46.18
	ORE (Ours)	**86.49**	**83.45**	31.04	**66.99**

Table 4: Comparison of 1600 edits on Bi-ZsRE.

that ORE outperforms existing methods on the Bi-ZsRE benchmark, achieving an average score of 65.36%. Notably, this represents an improvement of 4.40 percentage points compared to the SOTA method MEMIT (60.96%). This indicates that ORE maintains high knowledge editing accuracy and generalization capabilities even in challenging mixed-language scenarios, successfully mitigating the interference typically caused by general semantic spaces across languages.

5 Related Work

Existing knowledge editing methods can be broadly categorized into two groups: parameter-modifying and parameter-preserving methods.

Parameter-modifying methods inject new knowledge by locating and updating specific model weights. ROME (Meng et al., 2022) introduced a "Locate-Then-Edit" paradigm with rank-one updates to MLP layers for single facts, and MEMIT (Meng et al., 2023) extended this to batch editing by distributing update residuals across layers. To mitigate interference and preserve general capabilities, RECT (Gu et al., 2024) adds regularization, PRUNE (Ma et al., 2025) controls the condition number of edit weights, AlphaEdit (Fang et al., 2025) projects updates onto the null space of retained knowledge, LangEdit (Sun et al., 2025) and KDE (Xu et al., 2025) project updates onto dynamic or orthogonal subspaces to reduce cross-lingual or lifelong interference, and AdaEdit (Li and Chu, 2025) addresses sequential decline via disentangled FFN representations and SVD-based sparsification. While KDE and LangEdit also introduce orthogonality constraints, they enforce orthogonality on parameter updates, whereas ORE operates directly in the representation space to decouple semantic entanglement.

Parameter-preserving methods keep the backbone frozen and introduce external modules or representation interventions. T-Patcher (Huang et al., 2023) adds task-specific parameters, while SERAC (Mitchell et al., 2022) and GRACE (Hartvigsen et al., 2023) use memory-based modules for non-intrusive updates. Recently, ReFT (Wu et al., 2024) demonstrates that low-rank interventions on hidden states at inference time are sufficient to steer model behavior, and BaFT (Liu et al., 2025) proposed an input-based basis vector weighting mechanism, achieving a better trade-off between editing and locality through non-linear, fine-grained control of the representation subspace. In comparison, the proposed ORE follows the ReFT paradigm and improves upon it to adapt to knowledge editing scenarios. Addressing the issue of general semantic entanglement in batch editing, we introduce explicit geometric orthogonal constraints on top of representation intervention, achieving great performance in batch and cross-lingual scenarios.

6 Conclusion

In this work, we identify that general semantic structure entanglement negatively impacts batch knowledge editing in LLMs. To address this issue, we propose Orthogonal Representation Editing (ORE), a parameter-preserving knowledge editing framework that operates directly on representation vectors by constructing a general semantic subspace and enforcing orthogonal editing directions to decouple shared semantics, together with a gated non-linear representation tuning mechanism

for precise and localized representation intervention. Extensive experiments demonstrate that ORE consistently outperforms existing methods in both editing accuracy and generalization, including challenging cross-lingual editing scenarios.

Limitations

Despite ORE demonstrating superior performance in batch editing and cross-lingual scenarios, several limitations remain that merit further exploration in future work: (1) Our current experiments are primarily conducted on models with 7B to 8B parameters. Larger-scale models typically possess higher-dimensional representation spaces and more complex entanglement patterns. Consequently, the geometric constraint mechanism proposed in ORE requires further experimental verification on these larger-scale architectures. (2) Current experiments are mainly concentrated on standard and restricted editing settings. Future work should extend the evaluation scope to a broader range of downstream application scenarios to verify the feasibility and stability of the method in complex and dynamic real-world deployments. (3) Our current work primarily focuses on improving the accuracy of the model in answering questions, with less emphasis on capabilities such as logical reasoning following the editing process. Future research should further investigate the impact of knowledge editing on the model's comprehensive reasoning abilities.

7 Acknowledgments

The work is supported by the National Natural Science Foundation of China (Grant 62406223)

References

Hervé Abdi and Lynne J Williams. 2010. Principal component analysis. *Wiley interdisciplinary reviews: computational statistics*, 2(4):433–459.

Himanshu Beniwal, D Kowsik, and Mayank Singh. 2024. Cross-lingual editing in multilingual language models. In *Findings of the Association for Computational Linguistics: EACL 2024*, pages 2078–2128.

Baolong Bi, Shenghua Liu, Lingrui Mei, Yiwei Wang, Junfeng Fang, Pengliang Ji, and Xueqi Cheng. 2025. Decoding by contrasting knowledge: Enhancing large language model confidence on edited facts. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 17198–17208, Vienna, Austria. Association for Computational Linguistics.

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.

Yuchen Cai and Ding Cao. 2024. O-edit: Orthogonal subspace editing for language model sequential editing. *arXiv preprint arXiv:2410.11469*.

Junfeng Fang, Houcheng Jiang, Kun Wang, Yunshan Ma, Jie Shi, Xiang Wang, Xiangnan He, and Tat-Seng Chua. 2025. Alphaedit: Null-space constrained model editing for language models. In *The Thirteenth International Conference on Learning Representations*.

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

Akshat Gupta, Anurag Rao, and Gopala Anumanchipalli. 2024. Model editing at scale leads to gradual and catastrophic forgetting. *arXiv preprint arXiv:2401.07453*.

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

Evan Hernandez, Belinda Z Li, and Jacob Andreas. 2023. Inspecting and editing knowledge representations in language models. *arXiv preprint arXiv:2304.00740*.

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

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. 2025a. Anyedit: Edit any knowledge encoded in language models. *arXiv preprint arXiv:2502.05628*.

Houcheng Jiang, Junfeng Fang, Tianyu Zhang, Baolong Bi, An Zhang, Ruipeng Wang, Tao Liang, and Xiang Wang. 2025b. Neuron-level sequential editing for large language models. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 16678–16702, Vienna, Austria. Association for Computational Linguistics.

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. 2025. AdaEdit: Advancing continuous knowledge editing for large language models. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4127–4149, Vienna, Austria. Association for Computational Linguistics.

Tianci Liu, Ruirui Li, Yunzhe Qi, Hui Liu, Xianfeng Tang, Tianqi Zheng, Qingyu Yin, Monica Xiao Cheng, Jun Huan, Haoyu Wang, and Jing Gao. 2025. Unlocking efficient, scalable, and continual knowledge editing with basis-level representation fine-tuning. In *The Thirteenth International Conference on Learning Representations*.

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

Ben Mann, Nick Ryder, Melanie Subbiah, J Kaplan, P Dhariwal, A Neelakantan, P Shyam, G Sastry, A Askell, S Agarwal, and 1 others. 2020. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*, 1(3):3.

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

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

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.

David Rolnick, Arun Ahuja, Jonathan Schwarz, Timothy Lillicrap, and Gregory Wayne. 2019. Experience replay for continual learning. *Advances in neural information processing systems*, 32.

Marco Scialanga, Thibault Laugel, Vincent Grari, and Marcin Detyniecki. 2025. SAKE: Steering activations for knowledge editing. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15966–15978, Vienna, Austria. Association for Computational Linguistics.

Wei Sun, Tingyu Qu, Mingxiao Li, Jesse Davis, and Marie-Francine Moens. 2025. Mitigating negative interference in multilingual knowledge editing through null-space constraints. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 8796–8810, Vienna, Austria. Association for Computational Linguistics.

Changyue Wang, Weihang Su, Qingyao Ai, Yujia Zhou, and Yiqun Liu. 2025. Decoupling reasoning and knowledge injection for in-context knowledge editing. *arXiv preprint arXiv:2506.00536*.

Jiaan Wang, Yunlong Liang, Zengkui Sun, Yuxuan Cao, Jiarong Xu, and Fandong Meng. 2024a. Cross-lingual knowledge editing in large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 11676–11686.

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

Zhengxuan Wu, Aryaman Arora, Zheng Wang, Atticus Geiger, Dan Jurafsky, Christopher D Manning, and Christopher Potts. 2024. ReFT: Representation fine-tuning for language models. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

Haoyu Xu, Pengxiang Lan, Enneng Yang, Guibing Guo, Jianzhe Zhao, Linying Jiang, and Xingwei Wang. 2025. Knowledge decoupling via orthogonal projection for lifelong editing of large language models. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 13194–13213, Vienna, Austria. Association for Computational Linguistics.

Yunzhi Yao, Peng Wang, Bozhong Tian, Siyuan Cheng, Zhoubo Li, Shumin Deng, Huajun Chen, and Ningyu Zhang. 2023. Editing large language models: Problems, methods, and opportunities. *arXiv preprint arXiv:2305.13172*.

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

To comprehensively evaluate the performance of ORE, we conducted experiments on three widely used knowledge editing benchmark datasets:

ZsRE: A standard fact-based dataset in question-answering format. Each sample contains a natural language question and its corresponding target answer (Levy et al., 2017).

CounterFact: A large-scale and highly challenging benchmark for counterfactual knowledge editing. This dataset covers diverse relation types and entities, and equips each editing target with semantically equivalent paraphrase prompts as well

as semantically related but factually independent neighborhood samples (Meng et al., 2022).

Bi-ZsRE: A cross-lingual extension of ZsRE, containing parallel Chinese-English question-answer pairs. Given that ORE aims to address the problem of general concept entanglement, we introduce this dataset to specifically evaluate the model's performance in cross-lingual scenarios, examining whether it can effectively strip away linguistic noise and achieve precise cross-lingual knowledge synchronization within the semantic space shared by multilingual models (Wang et al., 2024a).

We follow the experimental settings of (Fang et al., 2025) for ZsRE and CounterFact, and those of (Sun et al., 2025) for Bi-ZsRE.

B Metrics

Following the standards of previous knowledge editing works (Meng et al., 2022, 2023; Fang et al., 2025; Sun et al., 2025), Let the given edit sample be denoted as (s_i, r_i, o_i^*), the prompt as (s_i, r_i), the paraphrase prompts as $N(s_i, r_i)$, and the neighborhood prompts as $O(s_i, r_i)$. The metrics are defined as follows:

Efficacy: Measures whether the target knowledge has been successfully injected into the model.

$$\mathbb{E}_i \left\{ o_i = \arg\max_o \mathbb{P}_f(o \mid (s_i, r_i)) \right\} \tag{11}$$

Generality: Measures the robustness of the edited knowledge to semantic variations.

$$\mathbb{E}_i \left\{ o_i = \arg\max_o \mathbb{P}_f(o \mid N((s_i, r_i))) \right\} \tag{12}$$

Specificity: Measures the locality of the editing operation, evaluating whether the model avoids corrupting irrelevant knowledge.

$$\mathbb{E}_i \left\{ o_i^c = \arg\max_o \mathbb{P}_f(o \mid O((s_i, r_i))) \right\} \tag{13}$$

C Implementation Details

All experiments are conducted on LLaMA-3-8B and Qwen-2.5-7B models. For the LLaMA-3-8B model, we apply interventions at layers [9, 18, 24, 28]; For the Qwen2.5-7B model, we apply interventions at layers [9, 18, 24, 26]. The general semantic subspace is constructed from 2000 structurally similar but factually unrelated samples. PCA is applied to their representations, and the top 4 principal components are retained to define the subspace. All experiments were conducted on a single Ascend 910B NPU (64GB).

For all experiments, the loss weights are set to $\lambda_1 = 1.0$, $\lambda_2 = 2.0$, $\lambda_3 = 1.0$, and $\lambda_4 = 3.0$. The projection dimension of the non-linear bottleneck is fixed to 128. Models are trained for 30 epochs with a batch size of 1 using the AdamW optimizer. We adopt a cosine annealing learning rate schedule, with the learning rate decayed from 5×10^{-4} to 2×10^{-6} over the course of training.

For baseline methods, we follow the hyperparameter settings reported in prior work. Specifically, the hyperparameters of FT, ROME, MEMIT, PRUNE, RECT, AlphaEdit, and NSE are adopted from (Fang et al., 2025), while ReFT follows (Liu et al., 2025).

All experiments are repeated three times, and the reported results are averaged over the three runs.

D Baselines

We compare ORE against the following existing knowledge editing methods:

FT (Fine-Tuning): The standard fine-tuning approach. We directly update the parameters of specific layers in the model using the cross-entropy loss on the edit samples. FT serves as a fundamental baseline for measuring editing performance (Zhu et al., 2020).

ROME (Rank-One Model Editing): A classic "Locate-Then-Edit" method. ROME employs causal tracing to locate the critical neurons responsible for storing knowledge and uses a rank-one approximation to inject specific key-value pairs into the MLP layers (Meng et al., 2022).

MEMIT (Mass-Editing Memory in a Transformer): As an extension of ROME, MEMIT achieves the batch injection of thousands of knowledge entries by distributing the update residuals across the MLP modules of multiple layers (Meng et al., 2023).

PRUNE (Perturbation Restraint on Upper bouNd for Editing) : An editing framework constrains the range of parameter perturbation by limiting the condition number of the edit matrix, thereby reducing damage to irrelevant knowledge during sequential editing processes (Ma et al., 2025).

RECT (RElative Change in weighT) : A method focusing on editing consistency and robustness by introducing an additional regularization term dur-

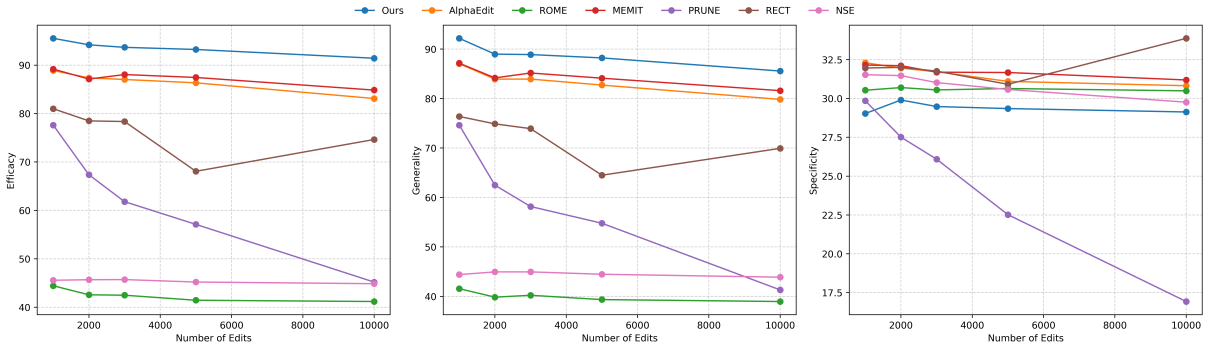

Figure 6: Performance Comparison between ORE and Existing Methods.

Method	Eff.↑	Gen.↑	Spe.↑	Avg.↑
FT	30.48	30.22	15.49	25.40
ROME	2.01	1.80	0.69	1.50
MEMIT	34.62	31.28	18.49	28.13
PRUNE	24.77	23.87	20.69	23.11
RECT	86.05	80.54	31.67	66.09
AlphaEdit	94.47	**91.13**	**32.55**	**72.72**
NSE	62.29	47.13	32.32	47.25
ReFT	19.47	18.77	13.88	17.37
ORE (Ours)	**94.71**	90.86	29.24	71.60

Table 5: Sequential Editing on the ZsRE benchmark using LLaMA-3-8B.

ing optimization to prevent weight updates from overfitting to the edit samples (Gu et al., 2024).

AlphaEdit: The latest SOTA method among parameter-modifying approaches. AlphaEdit proposes a Null-Space Projection mechanism, achieving efficient updates without disrupting the original knowledge (Fang et al., 2025).

NSE (Neuron-level Sequential Editing): NSE utilizes original parameters to calculate target hidden states to prevent model collapse. It also employs an activation-based neuron filtering strategy to update only critical neurons, thereby alleviating catastrophic forgetting (Jiang et al., 2025b).

E Experimental Results on Sequential Editing

Although ORE is primarily designed to address general semantic entanglement in batch knowledge editing, we additionally report results on sequential editing as a complementary evaluation. All the experiment settings are followed (Fang et al., 2025) and (Sun et al., 2025).

To mitigate catastrophic forgetting under sequential edits, we incorporate an experience replay mechanism following prior continual learning practice (Rolnick et al., 2019). Specifically, we maintain a replay buffer consisting of previously edited requests. During training for the current edits, we uniformly sample a small subset from the buffer, where the replay size is 1. The overall objective at each optimization step is augmented by an additional replay loss computed on the sampled historical edits: $\mathcal{L}' = \mathcal{L} + \mathcal{L}_{replay}$.

Method	Eff.↑	Gen.↑	Spe.↑	Avg.↑
FT	83.33	67.79	46.63	65.92
ROME	64.40	61.42	49.44	58.42
MEMIT	65.65	64.65	51.56	60.62
PRUNE	68.25	64.75	49.82	60.94
RECT	66.05	63.62	61.41	63.69
AlphaEdit	**98.90**	**94.22**	67.88	87.00
NSE	96.14	78.42	**87.66**	**87.41**
ReFT	42.03	42.46	56.50	47.00
ORE (Ours)	86.76	84.20	85.26	85.41

Table 6: Sequential Editing on the CounterFact benchmark using LLaMA-3-8B.

Table 5, 6 and 7 report the performance of different knowledge editing methods under sequential editing settings on ZsRE, CounterFact, and Bi-ZsRE benchmarks using LLaMA3-8B as the backbone model. From the results, we observe that: (1) ORE achieves competitive overall performance across all three benchmarks and outperforms all the baselines on the Bi-ZsRE dataset, indicating its effectiveness in complex knowledge editing scenarios. (2) Interestingly, ORE exhibits higher Specificity in sequential editing compared to batch editing. This behavior can be attributed to the dynamic gating mechanism. In sequential settings, each editing step focuses on a smaller batch, making the

knowledge-carrying token positions more clearly identifiable. This allows the gating network to activate intervention in a highly selective and sparse manner, while remaining silent on irrelevant tokens and non-target contexts.

Method	Eff.↑	Gen.↑	Spe.↑	Avg.↑
FT	31.41	29.97	15.29	25.56
ROME	2.54	2.46	0.39	1.80
MEMIT	4.58	4.03	2.84	3.82
PRUNE	4.92	4.22	1.90	3.68
RECT	41.01	38.58	20.80	33.46
AlphaEdit	71.88	66.55	30.47	56.30
LangEdit	73.18	66.95	**31.11**	57.08
NSE	49.43	48.06	30.58	42.69
ReFT	26.07	25.54	15.22	22.28
ORE (Ours)	**80.53**	**76.31**	26.81	**61.22**

Table 7: Sequential Editing on the Bi-ZsRE benchmark using LLaMA-3-8B.

F Impact of Batch Editing Scale on Editing Performance

Figure 6 illustrates the trends of Efficacy, Generality, and Specificity as the batch editing scale increases for different baselines. As the batch size further expands, the overall performance of most baselines exhibits a clear degradation, reflecting the fact that large-scale editing inevitably introduces general semantic entanglement, which weakens the model's ability to precisely control target facts. In contrast, ORE demonstrates a notably more stable performance trend and is able to maintain over 90% Efficacy and Generality even under 10,000 edits. These results validate the effectiveness of ORE in mitigating general semantic entanglement through representation-space orthogonal constraints, enabling more reliable performance scaling in large-scale batch knowledge editing scenarios.