Mitigating Negative Interference in Multilingual Sequential Knowledge Editing through Null-Space Constraints

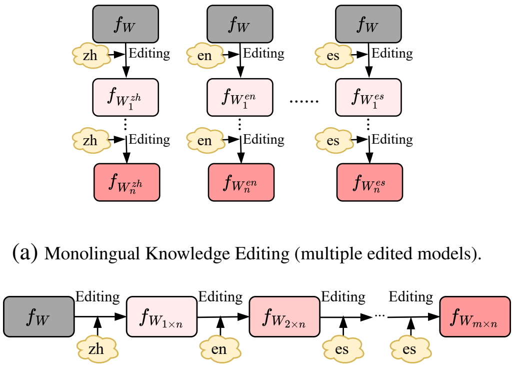
Wei Sun, Tingyu Qu, Mingxiao Li[*], Jesse Davis and Marie-Francine Moens
Department of Computer Science, KU Leuven
Celestijnenlaan 200A 3001 Heverlee, Belgium
{sun.wei, tingyu.qu, mingxiao.li, jesse.davis, sien.moens}@kuleuven.be

Abstract

Efficiently updating multilingual knowledge in large language models (LLMs), while preserving consistent factual representations across languages, remains a long-standing and unresolved challenge. While deploying separate editing systems for each language might seem viable, this approach incurs substantial costs due to the need to manage multiple models. A more efficient solution involves integrating knowledge updates across all languages into a unified model. However, performing sequential edits across languages often leads to destructive parameter interference, significantly degrading multilingual generalization and the accuracy of injected knowledge. To address this challenge, we propose LangEdit, a novel null-space constrained framework designed to precisely isolate language-specific knowledge updates. The core innovation of LangEdit lies in its ability to project parameter updates for each language onto the orthogonal complement of previous updated subspaces. This approach mathematically guarantees update independence while preserving multilingual generalization capabilities. We conduct a comprehensive evaluation across three model architectures, six languages, and four downstream tasks, demonstrating that LangEdit effectively mitigates parameter interference and outperforms existing state-of-the-art editing methods. Our results highlight its potential for enabling efficient and accurate multilingual knowledge updates in LLMs. The code is available at https://github.com/VRCMF/LangEdit.git.

1 Introduction

Modern large language models (LLMs) exhibit strong capabilities in encoding and retrieving factual knowledge. Knowledge editing has emerged as an efficient approach to updating knowledge within LLMs, reducing hallucinations without the need for resource-intensive retraining (Gu et al., 2024).



(a) Monolingual Knowledge Editing (multiple edited models).

(b) Multilingual Knowledge Editing (only one edited model).

Figure 1: Illustration of Monolingual (a) and Multilingual (b) Sequential Knowledge Editing. f_W denote the pre-edited model. After performing sequential edits, f_W^* represents the set of monolingual models, where each model has undergone n edits, and its superscript indicates the specific language of editing. For multilingual scenarios, $f_{W_{m\times n}}$ represents a single multilingual model trained on m languages, with each language containing n samples.

However, efficiently updating knowledge in multilingual scenarios remains a significant challenge, as models must maintain factual consistency across multiple languages. Although existing monolingual knowledge editing methods (Meng et al., 2023; Gu et al., 2024; Fang et al., 2025; Ma et al., 2025) have shown promising results, they lack effective solutions for managing multiple monolingual models concurrently, as illustrated in Figure 1a.

To address this challenge, we introduce the task of multilingual sequential knowledge editing, which involves updating knowledge across multiple languages in a sequential manner, as depicted in Figure 1b. This task holds particular significance for applications including knowledge-informed multilingual information retrieval (Zhang et al., 2022; Wang et al., 2024c) and factual updates of multilingual LLMs (Singhal et al., 2024). A criti-

[*] Corresponding author.

Findings of the Association for Computational Linguistics: ACL 2025, pages 8796–8810
July 27 - August 1, 2025 ©2025 Association for Computational Linguistics

cal challenge arises from the fact that editing knowledge in one language can negatively impact the model's performance in other languages. We term this phenomenon *negative interference*, describing how knowledge editing in a multilingual setting degrades performance on previously updated languages and undermines the multilingual generalization capabilities of LLMs, as demonstrated in our experiments.

We propose LangEdit, a multilingual knowledge editing framework designed to mitigate negative interference. LangEdit constrains parameter updates for each language to the null space (Greub, 2012) of previous language updated subspaces. For example, when editing Chinese facts after English updates, LangEdit projects Chinese updates onto the null-space of the parameter updates for English, mathematically ensuring minimal negative interference. This approach creates protective "language safeguards" that prevent parameter conflicts. Compared to existing editing models, LangEdit yields substantial improvements, achieving up to 5.65 percentage points increase in multilingual generalization tasks and 2.20 percentage points improvement in editing accuracy, thereby demonstrating its dual capacity for precise knowledge editing and effective multilingual knowledge retention.

We conduct extensive experiments spanning three model architectures, six languages, four multilingual generalization evaluation tasks, and two multilingual knowledge editing datasets. Across these settings, LangEdit consistently outperforms strong sequential editing baselines, establishing state-of-the-art performance. Our contributions are:

- We introduce the task of multilingual sequential knowledge editing, which involves sequentially updating a multilingual LLM with knowledge in multiple languages.

- We develop LangEdit, a null-space projection framework that provably perform language-specific knowledge updates.

- We show LangEdit's consistent gains across diverse languages, model scales and downstream tasks commonly used for evaluating the multilingual generalization ability.

2 Preliminaries

2.1 Multilingual Knowledge Editing in LLMs

An autoregressive LLM iteratively generates the next token of a sentence based on the previously generated tokens. Let h^l denote the hidden state of the next token at layer l. The hidden state is computed as follows (Meng et al., 2022; Fang et al., 2025): :

$$\begin{aligned} \mathbf{h}^l &= \mathbf{h}^{l-1} + \mathbf{a}^l + \mathbf{m}^l, \\ \mathbf{m}^l &= \mathbf{W}^l_{\text{out}} \sigma(\mathbf{W}^l_{\text{in}} \gamma(\mathbf{h}^{l-1} + \mathbf{a}^l)), \end{aligned} \tag{1}$$

where $\mathbf{a}^l$ represents the output of the attention block in layer l and $\mathbf{m}^l$ corresponds to the output of the multilayer perceptron (MLP) in layer l. $\mathbf{W}^l_{\text{out}}$ and $\mathbf{W}^l_{\text{in}}$ are weight matrices of the l_{th} MLP. σ is the activation function and γ represents layer normalization. Following prior works (Meng et al., 2022, 2023; Fang et al., 2025), we express the attention block and MLP in parallel.

The MLP layers can be interpreted as linear associative memory (Geva et al., 2021). Specifically, the knowledge stored in the model can be formalized as triplets (s, r, o), where s represents the subject, r the relation, and o the object. For example, the triplet (s = Space Needle, r = is in, o = the center of Seattle) encodes a factual relationship. In this framework, the output representation of $\sigma(\mathbf{W}^l_{\text{in}} \gamma(\mathbf{h}^{l-1} + \mathbf{a}^l))$ corresponds to the key (subject and relation) of the knowledge, while $\mathbf{m}^l$ represents the value (object) of the knowledge. In the multilingual sequential knowledge editing, our objective is to optimize $\mathbf{W}^l_{out}$ within the neural network f. For simplicity of notation, in what follows we leave out these sub- and superscripts of $\mathbf{W}^l_{\text{out}}$ and write $\mathbf{W}$.

Let the stacked keys of new input knowledge in language j at time step t be $\mathbf{K}_t = [\mathbf{k}_{t,1} \mid \mathbf{k}_{t,2} \mid \ldots \mid \mathbf{k}_{t,n_t}] \in \mathbb{R}^{d_0 \times n_t}$, where n_t denotes the number of keys and d_0 is the dimension of the intermediate layer. Note that the language index j is omitted in the following equations for simplicity. Similarly, let the corresponding values be $\mathbf{V}_t = [\mathbf{v}_{t,1} \mid \mathbf{v}_{t,2} \mid \ldots \mid \mathbf{v}_{t,n_t}] \in \mathbb{R}^{d_1 \times n_t}$, where d_1 is the dimension of the output layer. Each input data $\mathbf{L}_t$ is represented as a set of key-value pairs $\{\mathbf{K}_t, \mathbf{V}_t\}$ with a size of n_t. The network is trained sequentially on a stream of multilingual data $\{\mathbf{L}_1, \mathbf{L}_2, \cdots, \mathbf{L}_T\}$, where each data $\mathbf{L}_t$ corresponds to the time step t that introduces knowledge of a language j. The initial parameters for training on $\mathbf{L}_t$ are initialized as $\mathbf{W}_{t-1}$, which are the optimal parameters obtained after training on the previous data $\mathbf{L}_{t-1}$.

2.2 Null-space Projection

Assume that the model f is trained on data $\mathbf{L}_t$ (for $t > 1$), with the corresponding weight update denoted as $\boldsymbol{\Delta}\mathbf{W}_t$. Let $\bar{\mathbf{K}}_{t-1} = [\mathbf{K}_1; \cdots; \mathbf{K}_{t-1}]$ and $\bar{\mathbf{V}}_{t-1} = [\mathbf{V}_1; \cdots; \mathbf{V}_{t-1}]$ represent the concatenated keys and values from all previous update steps, respectively. To mitigate the caused negative interference, the updates are constrained to lie within the null space of the previously injected knowledge representations. Specifically, the weight updates $\boldsymbol{\Delta}\mathbf{W}_t$ are projected into the null space of $\bar{\mathbf{K}}_{t-1}$ and compute:

$$(\boldsymbol{\Delta}\mathbf{W}_t + \mathbf{W}_{t-1})\bar{\mathbf{K}}_{t-1} = \bar{\mathbf{V}}_{t-1}, \tag{2}$$

As the dimensions of $\bar{\mathbf{K}}_{t-1}$ and $\bar{\mathbf{V}}_{t-1}$ grow with the injection of additional knowledge, computing the null space becomes computationally expensive. To address this, we replace the concatenated key and value matrices with their corresponding uncentered covariance matrices, inspired by techniques used in continual learning for computer vision (Wang et al., 2021): $\bar{\mathcal{K}}_{t-1} \triangleq \frac{1}{\bar{n}_{t-1}}(\bar{\mathbf{K}}_{t-1})^\top \bar{\mathbf{K}}_{t-1}$, and $\bar{\mathcal{V}}_{t-1} \triangleq \frac{1}{\bar{n}_{t-1}}(\bar{\mathbf{V}}_{t-1})^\top \bar{\mathbf{V}}_{t-1}$, where $\bar{n}_{t-1}$ is the number of rows in $\bar{\mathbf{K}}_{t-1}$. Consequently, Equation 2 is reformulated as:

$$(\boldsymbol{\Delta}\mathbf{W}_t + \mathbf{W}_{t-1})\bar{\mathcal{K}}_{t-1} = \bar{\mathcal{V}}_{t-1}. \tag{3}$$

It is straightforward to verify that the null space of $\bar{\mathbf{K}}_{t-1}$ is equivalent to the null space of the uncentered feature covariance matrix $\bar{\mathcal{K}}_{t-1}$.

3 Method

This section presents **LangEdit**, a novel model designed for multilingual sequential knowledge editing. Our approach utilizes feature covariance matrices, which are incrementally updated as new language-specific data arrives. Specifically, given a network trained on prior multilingual data, LangEdit updates the model parameters corresponding to language j at the current time step t. The key innovation lies in projecting the candidate parameter updates into the approximate null space of the feature covariance matrix of the previously learned knowledge representation, ensuring that language-specific knowledge updates remain decoupled and do not interfere with each other.

After training on data $\mathbf{L}_{t-1}$, we compute the uncentered covariance matrix as follows: $\mathcal{K}_{t-1} = \frac{1}{n_{t-1}}(\mathbf{K}_{t-1})^\top \mathbf{K}_{t-1}$, where n_{t-1} denotes the number of data points in $\mathbf{L}_{t-1}$. If the $\mathbf{L}_t$ is in language j, the uncentered feature covariance matrix is then updated recursively:

$$\bar{\mathcal{K}}_{t-1} = \frac{\bar{n}_{t-2}}{\bar{n}_{t-1}}\bar{\mathcal{K}}_{t-2} + \frac{n_{t-1}}{\bar{n}_{t-1}}\mathcal{K}_{t-1}, \tag{4}$$

where $\bar{n}_{t-1} = \bar{n}_{t-2} + n_{t-1}$ represents the cumulative number of data points observed up to step $t-1$. To preserve previously injected knowledge, we compute the approximate null space of $\bar{\mathcal{K}}_{t-1}$ (Equation 3).

When training the network with $\mathbf{W}_{t-1}$ as initialization on data $\mathbf{L}_t$, the parameter update $\Delta\mathbf{W}_t$ is obtained by optimizing the following objective:

$$\Delta\mathbf{W}_t = \arg\min_{\boldsymbol{\Delta}\tilde{\mathbf{W}}_\mathbf{t}} \left(\left\| (\boldsymbol{\Delta}\tilde{\mathbf{W}}_\mathbf{t} + \mathbf{W}_{t-1})\mathbf{K}_t - \mathbf{V}_t \right\|^2 + \left\| (\boldsymbol{\Delta}\tilde{\mathbf{W}}_\mathbf{t} + \mathbf{W}_{t-1})\bar{\mathbf{K}}_{t-1} - \bar{\mathbf{V}}_{t-1} \right\|^2 \right), \tag{5}$$

where $\boldsymbol{\Delta}\tilde{\mathbf{W}}_\mathbf{t}$ is the optimization variable.

To ensure that parameter updates reside in the null space of the uncentered covariance matrix of previous data, we adopt the methodology described in (Wang et al., 2021; Fang et al., 2025). Specifically, we construct an approximate null space by performing Singular Value Decomposition (SVD) of $\bar{\mathcal{K}}_{t-1}$:

$$\mathbf{U}_{t-1}, \boldsymbol{\Sigma}_{t-1}, \mathbf{V}_{t-1} = \mathrm{SVD}(\bar{\mathcal{K}}_{t-1}), \tag{6}$$

We retain only eigenvectors in $\mathbf{U}_{t-1}$ corresponding to zero eigenvalues, yielding the matrix $\mathbf{U}'_{t-1}$. The null space projection matrix is then defined as:

$$\mathbf{P}_{t-1} = \mathbf{U}'_{t-1}(\mathbf{U}'_{t-1})^\top \tag{7}$$

The parameter update $\boldsymbol{\Delta}\tilde{\mathbf{W}}_\mathbf{t}$ is projected onto the null space of $\bar{\mathcal{K}}_{t-1}$ using $\mathbf{P}_{t-1}$, the right hand side of Equation 5 becomes:

$$\arg\min_{\boldsymbol{\Delta}\tilde{\mathbf{W}}_\mathbf{t}} \left\| (\mathbf{P}_{t-1}\boldsymbol{\Delta}\tilde{\mathbf{W}}_\mathbf{t} + \mathbf{W}_{t-1})\mathbf{K}_t - \mathbf{V}_t \right\|^2, \tag{8}$$

because $(\mathbf{P}_{t-1}\boldsymbol{\Delta}\tilde{\mathbf{W}}_\mathbf{t} + \mathbf{W}_{t-1})\bar{\mathbf{K}}_{t-1} = \bar{\mathbf{V}}_{t-1}$, where $\mathbf{P}_{t-1}$ is the null space projection matrix. To stabilize convergence, a regularization term

Algorithm 1 LangEdit: multilingual sequential knowledge editing

```
Require: Initialized weight $\mathbf{W}_0$, Data sequence $\{\mathbf{L}_1, \ldots, \mathbf{L}_T\}$, Initial covariance $\bar{\mathcal{K}}_0 = \mathcal{K}_0 = \frac{1}{n_0}\mathbf{K}_0^\top \mathbf{K}_0$, Data sizes $\{n_t\}_{t=1}^T$
for $t = 1$ to $T$ do
    Extract input keys: $\mathbf{K}_t$
    Extract corresponding values: $\mathbf{V}_t$
    Obtain previous keys: $\mathbf{K}_{t-1}$
    Compute covariance: $\mathcal{K}_{t-1} \leftarrow \frac{1}{n_{t-1}}(\mathbf{K}_{t-1})^\top \mathbf{K}_{t-1}$,
    if $t > 1$, then update running covariance: $\bar{\mathcal{K}}_{t-1} \leftarrow \frac{\bar{n}_{t-2}}{\bar{n}_{t-1}}\bar{\mathcal{K}}_{t-2} + \frac{n_{t-1}}{\bar{n}_{t-1}}\mathcal{K}_{t-1}$, $\bar{n}_{t-1} = \bar{n}_{t-2} + n_{t-1}$
    Perform SVD: $(\mathbf{U}_{t-1}, \mathbf{\Sigma}_{t-1}, \mathbf{V}_{t-1}) \leftarrow \mathrm{SVD}(\bar{\mathcal{K}}_{t-1})$
    Compute projection matrix: $\mathbf{P}_{t-1} \leftarrow \mathbf{U}'_{t-1}\mathbf{U}'^{\top}_{t-1}$ (Keep eigenvectors in $\mathbf{U}_{t-1}$ whose eigenvalues are zero, to obtain $\mathbf{U}'_{t-1}$).
    Solve optimization:
        $\Delta \mathbf{W}_t = \arg\min_{\mathbf{\Delta}\tilde{\mathbf{W}}_\mathbf{t}} \Big( \|\mathbf{P}_{t-1}\mathbf{\Delta}\tilde{\mathbf{W}}_\mathbf{t}\|^2 + \|(\mathbf{P}_{t-1}\mathbf{\Delta}\tilde{\mathbf{W}}_\mathbf{t} + \mathbf{W}_{t-1})\mathbf{K}_t - \mathbf{V}_t\|^2 \Big)$
    Update model: $\mathbf{W}_t \leftarrow \mathbf{W}_{t-1} + \Delta \mathbf{W}_t$
end for
Ensure: Updated model $\mathbf{W}_T$
```

$\mathbf{P}_{t-1}\mathbf{\Delta}\tilde{\mathbf{W}}_\mathbf{t}$ is added, leading to the final optimization objective:

$$\Delta \mathbf{W}_t = \arg\min_{\mathbf{\Delta}\tilde{\mathbf{W}}_\mathbf{t}} \left(\left\| \mathbf{P}_{t-1}\mathbf{\Delta}\tilde{\mathbf{W}}_\mathbf{t} \right\|^2 + \left\| (\mathbf{P}_{t-1}\mathbf{\Delta}\tilde{\mathbf{W}}_\mathbf{t} + \mathbf{W}_{t-1})\mathbf{K}_t - \mathbf{V}_t \right\|^2 \right), \tag{9}$$

The closed solution for the final objective is:

$$\Delta \mathbf{W}_t = \mathbf{R}_t \mathbf{K}_t^\top \mathbf{P}_{t-1}(\mathbf{K}_t \mathbf{K}_t^\top \mathbf{P}_{t-1} + \mathbf{I})^{-1}. \tag{10}$$

where $\mathbf{R}_t = (\mathbf{V}_t - \mathbf{W}_{t-1}\mathbf{K}_t)$. A critical step in this process is computing the stacked keys $\mathbf{K}_0$, which represent the old knowledge stored in the LLM. Following the approach in (Meng et al., 2022), $\mathbf{K}_0 \in \mathbb{R}^{d_0 \times 100{,}000}$ is computed using randomly sampled triplets from Wikipedia. This procedure is consistently applied across all models and baselines. The LangEdit method is summarized in Algorithm 1.

4 Experiments

4.1 Experimental Setup

4.1.1 Datasets

Since multilingual sequential knowledge editing is a novel task, we constructed a benchmark using two multilingual datasets(Wang et al., 2024a,b). We evaluated all models and baselines on these datasets. **bzsre:** The Bi-ZsRE(Bilingual Zero-Shot Relation Extraction) dataset (Wang et al., 2024a) is designed to assess the impact of knowledge editing in multilingual LLMs. It comprises question-answer pairs in two languages: English and Chinese. From this dataset, we randomly selected 800 samples per language to create a bilingual sequential knowledge editing dataset (bzsre). The total number of edits is 1600. **mzsre:** The Multilingual Zero-Shot Relation Extraction (M-ZsRE) dataset (Wang et al., 2024b) contains question-answer pairs in twelve languages. Due to constraints of the downstream task, we focused on six languages: English, German, Dutch, Spanish, French, and Chinese. For each language, we extracted 400 samples to construct a multilingual sequential knowledge editing dataset (mzsre). The total number of edits is 2400. We put the description for MLaKE dataset (Wei et al., 2025) and experimental results on this dataset to the Appendix A.6.

To evaluate the multilingual generalization capabilities of the edited LLMs, we employed four tasks from the Cross-lingual TRansfer Evaluation of Multilingual Encoders (XTREME) benchmark (Hu et al., 2020): **XNLI** A cross-lingual natural language inference (NLI) benchmark extending MultiNLI (Gururangan et al., 2018) to 15 languages. It evaluates multilingual generalization in NLI tasks through textual entailment classification. **PAWS-X** (Yang et al., 2019) A cross-lingual paraphrase identification dataset featuring adversarial examples generated via word-order perturbations. It tests model robustness against structural ambiguities in multilingual contexts. **MLQA** (Lewis et al., 2020) A multilingual question answering benchmark derived from Wikipedia articles. It measures extractive question-answering (QA) performance across 7 languages. **Wikiann** (Pan et al., 2017) A multilingual named entity recognition resource with consistent PER/ORG/LOC annotations, providing standardized evaluation for multilingual named-entity recognition (NER) model adaptation.

4.1.2 Metrics

Following prior works (Meng et al., 2022, 2023; Fang et al., 2025), we define each editing metric given a LLM f, a knowledge fact prompt (s_i, r_i), the target output of the edited model o_i, and the output of the original model o_i^c as follows:

Efficacy: Efficacy quantifies the model's ability to produce the target object o_i when prompted with (s_i, r_i). It is computed as the average top-1 accuracy over all edited samples:

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

Generality: Generality evaluates the performance of the model on equivalent prompts of (s_i, r_i), such as rephrased statements $N((s_i, r_i))$. This is evaluated by the average top-1 accuracy on these $N((s_i, r_i))$:

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

Specificity: Specificity ensures that the editing does not affect samples $O(s_i, r_i)$ which are unrelated to the edit cases. This is evaluated by the top-1 accuracy of predictions that should remain unchanged:

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

To evaluate multilingual generalization in edited LLMs, we report **F1** Scores across four NLP tasks in the XTREME benchmark. **F1 (avg.)** denotes the average F1 Score across these tasks.

4.1.3 Implementation Details

We perform multilingual sequential knowledge editing with 100 samples per edit time step t. We conducted all experiments on a single A100 GPU (80GB) and repeated three times with different seeds to ensure reliability. Results are reported as the mean and standard deviation. We evaluate the computational cost of different editing methods in the Appendix A.4. Following (Meng et al., 2023; Fang et al., 2025), we configure each model as follows: (1) For the GPT-J-6B model, we target critical layers [3, 4, 5, 6, 7, 8] for editing inspired by monolingual editing. When computing the hidden representations of a critical layer, we perform 25 optimization steps with a learning rate of 0.5. (2) For the Llama3-8B model, we target critical layers [4, 5, 6, 7, 8] for editing. When computing the hidden representations of a critical layer, we perform 25 steps with a learning rate of 0.1. (3) For the Qwen2.5-7B model, we target critical layers [4, 5, 6, 7, 8] for editing. When computing the hidden representations of a critical layer, we perform 25 steps with a learning rate of 0.1. The selection of critical layers is guided by the causal tracing technique, with detailed results provided in Appendix A.1. For the package versions, we report them in the Appendix A.3.

4.1.4 Baselines

This work establishes the first benchmark for multilingual sequential knowledge editing. As no existing methods specifically address this emerging challenge, we establish baseline performance by adapting state-of-the-art approaches from monolingual sequential editing research. We faithfully re-implemented these models based on their original codebases.

FT (Fine-Tuning) updates a subset of the model parameters via gradient descent using the training examples from the multilingual knowledge editing dataset. While effective for adapting models to new distributions, FT risks catastrophic forgetting of pre-trained knowledge and demands substantial computational resources.

ROME (Rank-One Model Editing) (Meng et al., 2022) localizes factual associations to middle-layer feed-forward modules and updates their weights through rank-one adjustments. By identifying critical neuron activations tied to specific knowledge, ROME demonstrates that direct manipulation of feed-forward layers can edit factual predictions without requiring full retraining.

MEMIT (Mass-Editing Memory in Transformer) (Meng et al., 2023) extends ROME by propagating edits across multiple layers. It identifies critical neuron activations in parallelized weight matrices during knowledge editing, enabling the insertion of thousands of new associations.

PRUNE (Perturbation Restraint on Upper bouNd for Editing) (Ma et al., 2025) mitigates performance degradation during sequential editing by optimizing the condition number (Smith, 1967) of edited weights, which limits catastrophic interference with unrelated knowledge as the number of edits accumulates.

RECT (RElative Change in weighT) (Gu et al., 2024) mitigates the side effects of editing on general reasoning abilities by controlling weight updates. Specifically, the top-k% parameters that

Model	Methods	mzsre			XTREME‡	bzsre			XTREME†
		Efficacy↑	**Generality↑**	**Specificity↑**	**F1 (avg.)↑**	**Efficacy↑**	**Generality↑**	**Specificity↑**	**F1 (avg.)↑**
	Pre-edited	$31.15_{\pm 0.13}$	$31.01_{\pm 0.22}$	$31.93_{\pm 0.17}$	$69.3_{\pm 0.21}$	$31.55_{\pm 0.11}$	$31.17_{\pm 0.15}$	$30.5_{\pm 0.19}$	$69.38_{\pm 0.25}$
Llama3-8B	FT	$30.76_{\pm 0.12}$	$29.48_{\pm 0.19}$	$9.53_{\pm 0.11}$	$10.57_{\pm 0.03}$	$31.41_{\pm 0.15}$	$29.97_{\pm 0.17}$	$15.29_{\pm 0.21}$	$44.38_{\pm 0.33}$
	ROME	$0.39_{\pm 0.16}$	$0.35_{\pm 0.18}$	$0.32_{\pm 0.22}$	$4.51_{\pm 0.01}$	$2.54_{\pm 0.42}$	$2.46_{\pm 0.44}$	$0.39_{\pm 0.49}$	$4.66_{\pm 0.19}$
	MEMIT	$1.45_{\pm 1.03}$	$1.46_{\pm 0.99}$	$0.67_{\pm 0.03}$	$4.54_{\pm 0.00}$	$4.58_{\pm 0.27}$	$4.03_{\pm 0.09}$	$2.84_{\pm 0.42}$	$15.64_{\pm 2.41}$
	PRUNE	$0.43_{\pm 0.23}$	$0.36_{\pm 0.02}$	$0.02_{\pm 0.01}$	$4.48_{\pm 0.00}$	$4.92_{\pm 1.53}$	$4.22_{\pm 1.23}$	$1.90_{\pm 0.80}$	$9.32_{\pm 2.14}$
	RECT	$2.91_{\pm 0.38}$	$2.79_{\pm 0.36}$	$0.64_{\pm 0.18}$	$7.71_{\pm 2.49}$	$41.01_{\pm 2.47}$	$38.58_{\pm 1.95}$	$20.80_{\pm 0.76}$	$55.97_{\pm 3.44}$
	AlphaEdit	$\underline{80.34}_{\pm 0.58}$	$\underline{75.84}_{\pm 0.73}$	$\underline{30.91}_{\pm 0.49}$	$\underline{60.59}_{\pm 0.38}$	$\underline{71.88}_{\pm 0.81}$	$\underline{66.55}_{\pm 0.33}$	$\underline{30.47}_{\pm 0.20}$	$\underline{71.25}_{\pm 0.70}$
	LangEdit	$\mathbf{82.54}_{\pm 0.14}$*	$\mathbf{77.53}_{\pm 0.43}$*	$\mathbf{31.90}_{\pm 0.14}$*	$\mathbf{66.24}_{\pm 0.28}$*	$\mathbf{73.18}_{\pm 0.35}$*	$\mathbf{66.95}_{\pm 0.17}$*	$\mathbf{31.11}_{\pm 0.18}$*	$\mathbf{73.14}_{\pm 0.83}$*
	Pre-edited	$33.52_{\pm 0.09}$	$33.11_{\pm 0.13}$	$38.76_{\pm 0.08}$	$71.9_{\pm 0.15}$	$33.77_{\pm 0.07}$	$33.24_{\pm 0.20}$	$38.81_{\pm 0.15}$	$73.91_{\pm 0.14}$
Qwen2.5-7B	FT	$32.46_{\pm 0.13}$	$30.28_{\pm 0.47}$	$28.76_{\pm 0.20}$	$45.20_{\pm 0.40}$	$35.6_{\pm 0.15}$	$33.06_{\pm 0.23}$	$33.48_{\pm 0.17}$	$59.51_{\pm 0.43}$
	ROME	$12.44_{\pm 0.47}$	$11.35_{\pm 0.73}$	$2.25_{\pm 0.71}$	$4.70_{\pm 0.15}$	$16.36_{\pm 0.77}$	$15.27_{\pm 0.53}$	$1.60_{\pm 0.29}$	$4.70_{\pm 0.17}$
	MEMIT	$1.36_{\pm 0.36}$	$1.24_{\pm 0.29}$	$0.13_{\pm 0.05}$	$4.55_{\pm 0.01}$	$75.75_{\pm 0.06}$	$70.03_{\pm 0.02}$	$40.04_{\pm 0.47}$	$73.43_{\pm 0.86}$
	PRUNE	$24.99_{\pm 2.43}$	$24.45_{\pm 2.34}$	$18.02_{\pm 1.69}$	$40.57_{\pm 1.61}$	$37.24_{\pm 2.20}$	$35.95_{\pm 1.59}$	$27.91_{\pm 0.17}$	$60.35_{\pm 0.89}$
	RECT	$79.98_{\pm 1.03}$	$74.50_{\pm 0.88}$	$42.69_{\pm 0.17}$	$72.43_{\pm 0.73}$	$75.73_{\pm 0.20}$	$68.80_{\pm 0.55}$	$\mathbf{41.52}_{\pm 1.11}$	$\underline{73.69}_{\pm 0.82}$
	AlphaEdit	$\underline{93.50}_{\pm 0.18}$	$\mathbf{87.18}_{\pm 0.50}$	$\underline{42.58}_{\pm 0.29}$	$\underline{73.01}_{\pm 0.74}$	$\underline{82.41}_{\pm 1.86}$	$\underline{73.57}_{\pm 0.53}$	$40.08_{\pm 0.48}$	$73.56_{\pm 0.62}$
	LangEdit	$\mathbf{93.90}_{\pm 0.04}$*	$\underline{87.02}_{\pm 0.41}$	$\mathbf{42.64}_{\pm 0.32}$	$\mathbf{74.06}_{\pm 1.33}$*	$\mathbf{83.47}_{\pm 0.91}$*	$\mathbf{74.32}_{\pm 0.19}$*	$\underline{40.55}_{\pm 0.41}$	$\mathbf{75.70}_{\pm 0.36}$*
	Pre-edited	$24.05_{\pm 0.12}$	$23.71_{\pm 0.23}$	$26.07_{\pm 0.13}$	$37.5_{\pm 0.20}$	$14.51_{\pm 0.10}$	$13.92_{\pm 0.17}$	$15.08_{\pm 0.09}$	$33.66_{\pm 0.23}$
GPT-J-6B	FT	$23.58_{\pm 0.10}$	$21.16_{\pm 0.13}$	$1.64_{\pm 0.07}$	$4.67_{\pm 0.09}$	$20.86_{\pm 0.10}$	$19.55_{\pm 0.13}$	$4.04_{\pm 0.17}$	$5.06_{\pm 0.11}$
	ROME	$19.66_{\pm 0.10}$	$18.37_{\pm 0.38}$	$1.36_{\pm 0.10}$	$4.93_{\pm 0.15}$	$14.41_{\pm 0.37}$	$13.08_{\pm 0.30}$	$0.78_{\pm 0.15}$	$6.52_{\pm 1.78}$
	MEMIT	$48.25_{\pm 4.25}$	$46.07_{\pm 4.14}$	$22.63_{\pm 1.08}$	$36.85_{\pm 0.76}$	$44.98_{\pm 0.22}$	$41.75_{\pm 0.36}$	$14.47_{\pm 0.16}$	$31.53_{\pm 0.76}$
	PRUNE	$3.10_{\pm 0.83}$	$2.93_{\pm 0.86}$	$2.40_{\pm 0.62}$	$8.26_{\pm 1.88}$	$2.41_{\pm 0.38}$	$2.42_{\pm 0.44}$	$2.37_{\pm 0.39}$	$5.77_{\pm 3.44}$
	RECT	$71.10_{\pm 1.78}$	$67.05_{\pm 1.99}$	26.26_{36}	$\underline{37.17}_{\pm 0.95}$	$48.30_{\pm 0.30}$	$43.33_{\pm 0.56}$	$14.48_{\pm 0.17}$	$32.67_{\pm 1.29}$
	AlphaEdit	$\underline{83.59}_{\pm 0.26}$	$\underline{78.34}_{\pm 0.05}$	$\underline{26.55}_{\pm 0.33}$	$36.74_{\pm 1.19}$	$\underline{54.36}_{\pm 0.11}$	$\underline{47.52}_{\pm 0.15}$	$\underline{15.13}_{\pm 0.20}$	$\underline{33.28}_{\pm 1.21}$
	LangEdit	$\mathbf{84.27}_{\pm 0.27}$*	$\mathbf{79.74}_{\pm 0.36}$*	$\mathbf{27.23}_{\pm 0.04}$*	$\mathbf{38.59}_{\pm 1.35}$*	$\mathbf{54.86}_{\pm 0.21}$*	$\mathbf{48.40}_{\pm 0.44}$*	$\mathbf{15.31}_{\pm 0.05}$	$\mathbf{35.75}_{\pm 0.94}$*

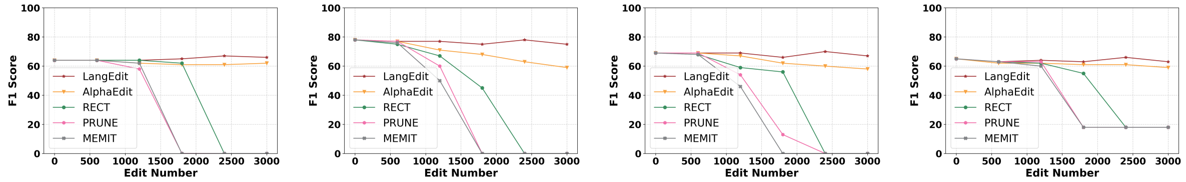
Table 1: We assess the performance of various model editing methods using three LLMs (GPT-J-6B, Llama3-8B, and Qwen2.5-7B) on the mzsre and bzsre datasets. The best results are highlighted in **bold**, while the second-best results are underlined. Statistical significance (*) is determined using a paired t-test with p=0.05. XTREME‡ represents average F1 Scores on XTREME tasks after training on the mzsre dataset; XTREME† denotes the average F1 Scores after training on the bzsre dataset. All baselines are adapted for multilingual sequential knowledge editing.

change the most according to relative changes in parameters are considered as the principal editing information and their obtained values are kept, while the remaining parameters are kept unchanged.

AlphaEdit (Fang et al., 2025) is designed for monolingual knowledge editing and decouples knowledge updates from preservation objectives by null-space projection. Parameter perturbations in AlphaEdit are projected onto a static null space of key matrices, whereas LangEdit leverages a dynamic null space, where the projection of the key matrices varies at each update step t. The difference between the original AlphaEdit and our method is two-fold. The first difference is that the original AlphaEdit and our method leverage the null-space projection to resolve a different task. The original AlphaEdit focuses on monolingual sequential knowledge editing and our work solves the task of multilingual sequential knowledge editing. The second difference is the usage of null space projection. Parameter perturbations in AlphaEdit are projected onto a static null space of key matrices, while LangEdit leverages a dynamic null space, where the projection of the key matrices is different for any update step.

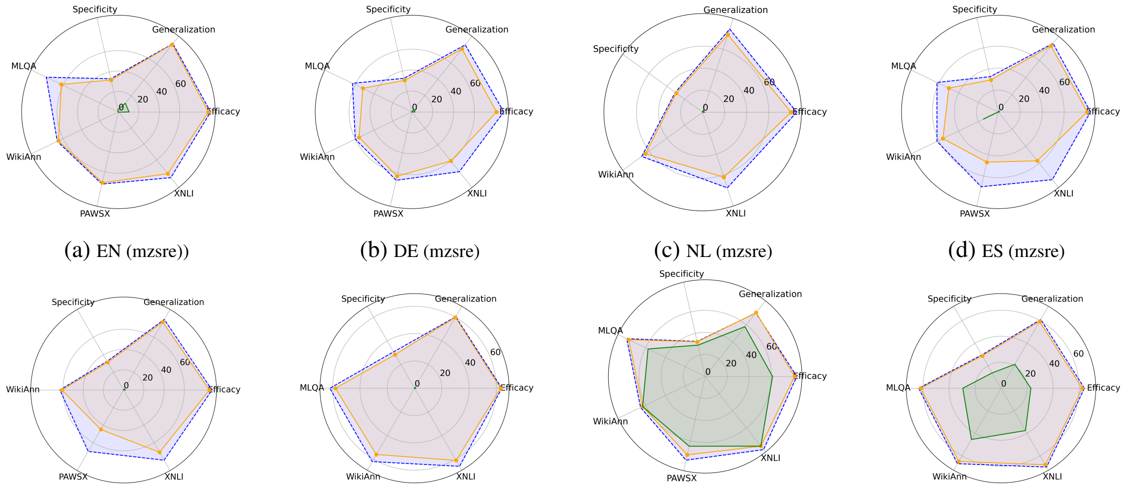
Baseline models (e.g., MEMIT, AlphaEdit) are designed to support batch editing (i.e., 100 samples per time-step). As ROME does not support batch-editing, we run ROME 100 times iteratively to adapt the multilingual sequential knowledge editing task.

Moreover, we propose several baseline methods derived from our model architecture, with comprehensive comparison results provided in Appendix A.2. We also evaluate the pre-edited model — the original LLMs without any editing.



(a) MLQA (b) Wikiann (c) PAWSX (d) XNLI

Figure 2: F1 Scores of the edited Llama3-8B on XTREME benchmark evaluating multilingual generalization.



(a) EN (mzsre)) (b) DE (mzsre) (c) NL (mzsre) (d) ES (mzsre)

(e) FR (mzsre) (f) ZH (mzsre) (g) EN (bzsre) (h) ZH (bzsre)

Figure 3: Radar chart of editing performance and multilingual downstream task performance when editing Llama3-8B with different languages. We use ┄┄, —— and —— to represent LangEdit, AlphaEdit and RECT, respectively.

4.2 Experimental Results

Table 1 demonstrates that LangEdit significantly outperforms the state-of-the-art model AlphaEdit in terms of Efficacy Score across three model architectures (Llama3-8B, Qwen2.5-7B, GPT-J-6B) and two knowledge editing datasets (mzsre, bzsre). When using Llama3-8B as the backbone, LangEdit achieves average improvements of +2.20 and +1.30 percentage points [1] on the mzsre and bzsre datasets, respectively. Similar enhancements are observed with Qwen2.5-7B (+0.40 on mzsre, +1.06 on bzsre) and GPT-J-6B (+0.68 on mzsre, +0.60 on bzsre). These results underscore the robustness of our method across diverse architectures and its effectiveness in knowledge injection for LLMs.

Furthermore, Table 1 reveals that LangEdit surpasses the best baseline (AlphaEdit in a multilingual setting) by an average margin of +2.85 F1 Score and +2.17 F1 Score on XTREME benchmark when evaluated across three backbone architectures (GPT-J-6B, Qwen2.5-7B, Llama3-8B) after training on mzsre and bzsre datasets, respectively. The averaged F1 Score reflects the performance across four tasks designed to assess multilingual generalization. Notably, the improvement is most pronounced for Llama3-8B, which exhibits a +5.65 F1 Score gain on the XTREME benchmark after training on the mzsre dataset. This highlights LangEdit's ability to preserve and enhance multilingual generalization across diverse model architectures during sequential knowledge editing.

Our findings indicate that multilingual knowledge editing significantly improves multilingual generalization capabilities. For GPT-J-6B and Qwen2.5-7B, LangEdit consistently outperforms their pre-edit counterparts on both datasets. While Llama3-8B shows a 3.06 F1 Score decrease on XTREME benchmark after training on the mzsre dataset, it achieves a 3.76 F1 Score improvement on the XTREME benchmark after training on the bzsre dataset. We hypothesize that this discrepancy arises because the injected multilingual knowledge benefits languages structurally aligned with the editing corpus. This observation aligns with prior work (Chua et al., 2024), which suggests that even minimal multilingual exposure can enhance generalization ability.

[1] All increases or decreases are given in percentage points.

Figure 2 illustrates editing performance and multilingual generalization performance trends under varying edits. A key pattern is: LangEdit consistently outperforms baselines across all edit scales in tasks evaluating multilingual generalization.

4.3 Per-Language Evaluation of Editing and Generalization

To evaluate editing efficiency and multilingual generalization capabilities, we conducted comprehensive experiments using Llama3-8B as the backbone architecture across six languages.

Figure 3 presents radar charts illustrating the language-specific performance characteristics of LangEdit, AlphaEdit, and RECT. Our analysis reveals two key findings: (1) Our model consistently outperforms baselines across all evaluated languages, achieving state-of-the-art results in multilingual sequential knowledge editing. Specifically, it demonstrates improvements in editing accuracy (average +1.13) and F1 Scores on downstream tasks (average +3.32). (2) We observe a notable disparity in performance gains between English and Spanish knowledge updates. While English updates show modest improvements (editing: +0.61, F1 Scores on downstream: average +3.39), Spanish updates yield significantly higher gains (editing: average +1.29, F1 Scores on downstream: average +9.00). We attribute this discrepancy to the pretraining data imbalance in Llama3-8B, where the model's strong English generalization capacity leaves limited room for improvement through knowledge editing. This finding aligns with recent studies on multilingual capacity scaling (Conneau et al., 2020; Fernandes et al., 2023), suggesting that editing effectiveness is inversely correlated with the pretraining data volume of the target language.

We further conduct in-depth analysis of knowledge sharing to explore whether a fact update in one language is accessible to other languages. The results is illustrated in the Appendix A.5.

4.4 Negative Interference in Multilingual Knowledge Editing

To analyze negative interference, we evaluate Llama3-8B edited with knowledge in six languages using two editing datasets. Editing efficacy is measured through the Efficacy Score, while the multilingual generalization ability is assessed by the averaged F1 Score. We formalize the magnitude of negative interference as the performance gap between the original AlphaEdit model - AlphaEdit (mono), designed for monolingual knowledge updates and a model designed for multilingual knowledge editing (AlphaEdit (multi) and LangEdit).

Our experiments reveal significant negative interference in AlphaEdit (multi). As shown in Figure 4, AlphaEdit (multi) exhibits substantial performance degradation in both Efficacy Score (Δ = +0.10 $\sim$ +3.27) and F1 Score (Δ = +3.20 $\sim$ +15.19) across all language pairs compared to the AlphaEdit (mono). In contrast, our model demonstrates remarkable robustness to negative interference: (1) It reduces the F1 Score disparity to (Δ = +0.20 $\sim$ +9.29) across languages, significantly lower than the (Δ = +3.20 $\sim$ +15.19) obtained by AlphaEdit. (2) It surpasses AlphaEdit (mono) in Efficacy Score for English (+1.10), German (+3.01), Dutch (+2.58), and French (+3.95).

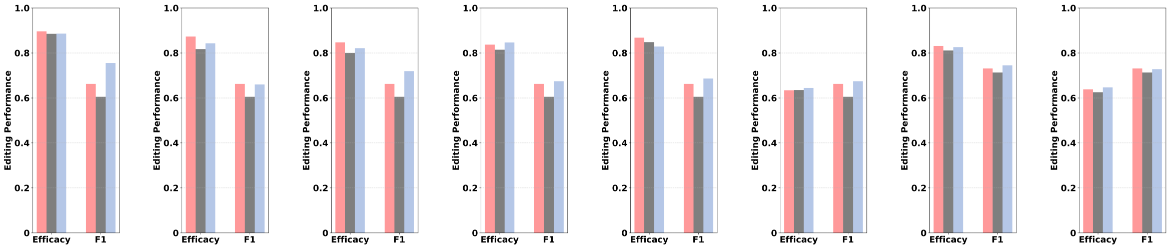
Without the proposed parameter updates the performance of multilingual sequential knowledge editing is lower than that of monolingual editing, confirming the effectiveness of the proposed null-space projection. Even for related languages (e.g., English-Dutch-German, French-Spanish), not integrating the null-space still degrades performance, reinforcing the effectiveness of our approach.

5 Related Work

Knowledge editing approaches for large language models (LLMs) can be broadly categorized into two paradigms: parameter-modifying and parameter-preserving, depending on whether the model weights are updated. We focus on the parameter-modifying paradigm. Prior research (Geva et al., 2021) has demonstrated that the MLP layers in Transformer models function as knowledge repositories, with specific neurons encoding editable factual associations.

Building on this insight, Meng et al. (2022) introduced ROME (Rank-One Model Editing), a pioneering two-stage framework that first identifies the MLP layer storing the target knowledge and then injects a single knowledge edit through rank-one weight perturbations. This approach was later extended by MEMIT (Mass-Editing Memory in Transformer) (Meng et al., 2023), which enhances scalability by enabling updates across multiple MLP layers to inject numerous knowledge edits.

Subsequent works identified a critical challenge in sequential editing: (Gu et al., 2024) attributed degradation of general abilities to parameter perturbations and introduced RECT, a regularization tech-



(a) EN (mzsre)) (b) DE (mzsre) (c) NL (mzsre) (d) ES (mzsre) (e) FR (mzsre) (f) ZH (mzsre) (g) EN (bzsre) (h) ZH (bzsre)

Figure 4: Performance comparison for all languages between multilingual knowledge editing model and the state-of-the-art monolingual model. 🟥: LangEdit, ⬛: AlphaEdit (Multi), 🟦: AlphaEdit (Mono).

nique constraining weight updates. Concurrently, Ma et al. (2025) established a mathematical connection between model degradation and the condition number (Smith, 1967) of the edited weights, introducing PRUNE to impose condition number-based constraints on weight updates. The monolingual AlphaEdit (Fang et al., 2025) projected the parameter perturbation of the MLP weights into the null space of knowledge preserved in the LLMs.

The above methods have shown success in monolingual knowledge editing but are limited to one language. To the best of our knowledge, we are the first work studying multilingual sequential knowledge editing, where the knowledge in different languages may interfere with each other.

Prior works (Wang et al., 2024a,b; Zhang et al., 2025) focus on single knowledge editing (Gu et al., 2024) instead of sequential knowledge editing. Single knowledge editing tests how well a model adapts to a single modification, while sequential knowledge editing checks whether the model can retain all previous edits and maintain overall performance after multiple consecutive changes.

6 Conclusion

We introduced the multilingual sequential knowledge editing task and identified the negative interference arising from parameter changes during sequential updates across languages. To address this, we proposed LangEdit, a novel framework employing null-space constrained optimization to isolate language-specific parameter updates while preserving the model's multilingual generalization capabilities. LangEdit achieves this by constructing "language safeguards", which prevent edits in one language-specific knowledge from adversely affecting performance in another, without the need for additional language-specific modules. LangEdit offers a technically sound method based on null-space projection, specifically adapted for the multilingual sequential setting with dynamic projections. Extensive experiments conducted across six languages and three large language models architectures demonstrate the effectiveness of LangEdit, establishing state-of-the-art performance in multilingual sequential knowledge editing.

Limitations

While our study provides insight into multilingual sequential knowledge editing, three key limitations warrant further investigation: (1) Our experiments are conducted on 6 to 8B parameter LLMs. Extending this analysis to larger architectures (e.g., 70B-scale models) could reveal scaling effects in multilingual knowledge editing. Large models usually have more parameters, more complex structures, and have stronger multilingual capabilities. However, knowledge editing may face challenges, such as more parameters making it more difficult to locate specific knowledge areas, or more dispersed knowledge representation within the model. (2) The current evaluation focuses on constrained editing scenarios. Future work should explore a broader range of downstream applications to assess real-world deployment viability. (3) Our multilingual experiments are limited to 6 languages. Developing comprehensive multilingual benchmarks covering hundreds of languages would better test the boundaries of LangEdit and baselines.

Acknowledgements

This research was funded by the CHIST-ERA projects ANTIDOTE (ERA-NET CHIST-ERA IV FET PROACT JTC 2019) and AIDAVA (EU HORIZON-HLTH-2021-TOOL-06-03).

The computations described in this research were performed using the The Flemish Supercomputer Center Tier-1 HPC ser-

vice (https://www.vscentrum.be/).

References

Lynn Chua, Badih Ghazi, Yangsibo Huang, Pritish Kamath, Ravi Kumar, Pasin Manurangsi, Amer Sinha, Chulin Xie, and Chiyuan Zhang. 2024. Crosslingual capabilities and knowledge barriers in multilingual large language models. *arXiv preprint arXiv:2406.16135*.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online. Association for Computational Linguistics.

Junfeng Fang, Houcheng Jiang, Kun Wang, Yunshan Ma, Xiang Wang, Xiangnan He, and Tat-seng Chua. 2025. Alphaedit: Null-space constrained knowledge editing for language models. *International Conference on Learning Representations 2025*.

Patrick Fernandes, Behrooz Ghorbani, Xavier Garcia, Markus Freitag, and Orhan Firat. 2023. Scaling laws for multilingual neural machine translation. In *International Conference on Machine Learning*, pages 10053–10071. PMLR.

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

Werner H Greub. 2012. *Linear algebra*, volume 23. Springer Science & Business Media.

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.

Suchin Gururangan, Swabha Swayamdipta, Omer Levy, Roy Schwartz, Samuel Bowman, and Noah A. Smith. 2018. Annotation artifacts in natural language inference data. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 107–112, New Orleans, Louisiana. Association for Computational Linguistics.

Junjie Hu, Sebastian Ruder, Aditya Siddhant, Graham Neubig, Orhan Firat, and Melvin Johnson. 2020. Xtreme: A massively multilingual multi-task benchmark for evaluating cross-lingual generalisation. In *International Conference on Machine Learning*, pages 4411–4421. PMLR.

Patrick Lewis, Barlas Oguz, Ruty Rinott, Sebastian Riedel, and Holger Schwenk. 2020. MLQA: Evaluating cross-lingual extractive question answering. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7315–7330, Online. Association for Computational Linguistics.

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

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*.

Xiaoman Pan, Boliang Zhang, Jonathan May, Joel Nothman, Kevin Knight, and Heng Ji. 2017. Cross-lingual name tagging and linking for 282 languages. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1946–1958, Vancouver, Canada. Association for Computational Linguistics.

Aryan Singhal, Thomas Law, Coby Kassner, Ayushman Gupta, Evan Duan, Aviral Damle, and Ryan Luo Li. 2024. Multilingual fact-checking using LLMs. In *Proceedings of the Third Workshop on NLP for Positive Impact*, pages 13–31, Miami, Florida, USA. Association for Computational Linguistics.

Russell A Smith. 1967. The condition numbers of the matrix eigenvalue problem. *Numerische Mathematik*, 10:232–240.

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, Bangkok, Thailand. Association for Computational Linguistics.

Shipeng Wang, Xiaorong Li, Jian Sun, and Zongben Xu. 2021. Training networks in null space of feature covariance for continual learning. In *Proceedings of the IEEE/CVF conference on Computer Vision and Pattern Recognition*, pages 184–193.

Weixuan Wang, Barry Haddow, and Alexandra Birch. 2024b. Retrieval-augmented multilingual knowledge editing. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 335–354, Bangkok, Thailand. Association for Computational Linguistics.

Yabing Wang, Fan Wang, Jianfeng Dong, and Hao Luo. 2024c. Cl2cm: Improving cross-lingual cross-modal retrieval via cross-lingual knowledge transfer. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 5651–5659.

Zihao Wei, Jingcheng Deng, Liang Pang, Hanxing Ding, Huawei Shen, and Xueqi Cheng. 2025. MLaKE: Multilingual knowledge editing benchmark for large language models. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 4457–4473, Abu Dhabi, UAE. Association for Computational Linguistics.

Yinfei Yang, Yuan Zhang, Chris Tar, and Jason Baldridge. 2019. PAWS-X: A cross-lingual adversarial dataset for paraphrase identification. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3687–3692, Hong Kong, China. Association for Computational Linguistics.

Fuwei Zhang, Zhao Zhang, Xiang Ao, Dehong Gao, Fuzhen Zhuang, Yi Wei, and Qing He. 2022. Mind the gap: Cross-lingual information retrieval with hierarchical knowledge enhancement. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 4345–4353.

Xue Zhang, Yunlong Liang, Fandong Meng, Songming Zhang, Yufeng Chen, Jinan Xu, and Jie Zhou. 2025. Multilingual knowledge editing with language-agnostic factual neurons. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 5775–5788, Abu Dhabi, UAE. Association for Computational Linguistics.

A Appendix

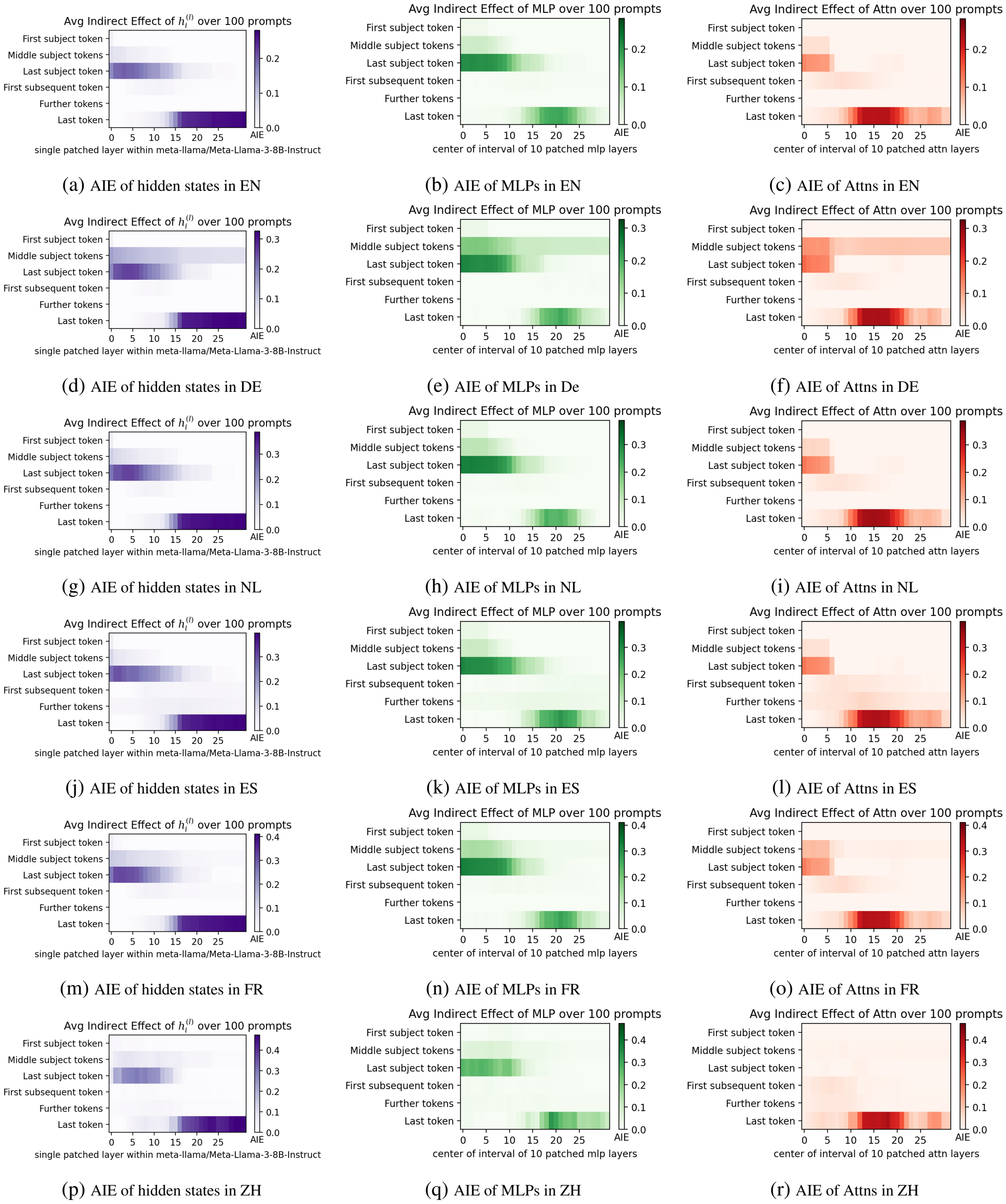
A.1 The analysis of factual knowledge storage

Assume that the factual knowledge stored in LLMs is represented in the (s, r, o) format where s denotes subjects, o represents objects and r is the relation between subjects and objects. (Meng et al., 2022) show that the MLP modules in the mid-layer encode subjects of the text in English and then generate outputs that retrieve updated knowledge from these layers. However, it remains uncertain whether knowledge in other languages follows the same pattern as described in (Meng et al., 2022), necessitating further investigation.

Firstly, we apply the causal tracing technique used in (Meng et al., 2022) to determine which hidden states have a causal impact on factual predictions by running the model multiple times, introducing interventions, and restoring specific states. The specific steps are as follows: **Clean Run**: The model processes the input without any interference, and the activation values of all hidden states are recorded. **Corrupted Run**: Noise (random perturbations) is introduced into the embeddings of the subject tokens to potentially cause the model to output incorrect factual predictions. **Corrupted-with-Restoration Run**: Based on the corrupted run, specific hidden states are restored to observe whether these states can restore the model's correct prediction. To quantify the causal contribution of each hidden state to factual predictions, we use average indirect effect (AIE) to measure the importance of specific states. Indirect Effect is the difference in predictions between the corrupted run and the corrupted-with-restoration run. The first column of Figure 5 shows that strong causal states appear in early layers at the last token of the subject for all six languages. The second and third columns of Figure 5 suggest that the MLP contributes stronger causality in early layers compared to the attention module. The opinion obtained by (Meng et al., 2022) is consistent with our finding, which is that the MLP modules in the mid-layers encode the subjects of knowledge in six languages and then generate the output of recalling memory objects.

A.2 Experimental results of variant models for multilingual editing of the LLM

We develop three variants and analyze their performance in Table 5: **AlphaEdit (Translation):** This variant edits only the English knowledge in the model using AlphaEdit and then translates the edited knowledge to other languages using Google Translate. While it underperforms AlphaEdit in direct editing evaluation (likely due to translation errors), it achieves stronger performance on multilingual generalization tasks. We assume that this is because focusing solely on English knowledge injection avoids interference from multilingual updates. **AlphaEdit (Multilingual):** We train the LLMs on multiple languages by first calculating knowledge updates for each language individually. Then, we combine all these updates by adding them together, forming unified updates that construct multilingual knowledge representations. Finally, we conduct knowledge editing by applying this representations to the weights of selected MLP layers in LLMs. This approach underperforms com-



(a) AIE of hidden states in EN

(b) AIE of MLPs in EN

(c) AIE of Attns in EN

(d) AIE of hidden states in DE

(e) AIE of MLPs in De

(f) AIE of Attns in DE

(g) AIE of hidden states in NL

(h) AIE of MLPs in NL

(i) AIE of Attns in NL

(j) AIE of hidden states in ES

(k) AIE of MLPs in ES

(l) AIE of Attns in ES

(m) AIE of hidden states in FR

(n) AIE of MLPs in FR

(o) AIE of Attns in FR

(p) AIE of hidden states in ZH

(q) AIE of MLPs in ZH

(r) AIE of Attns in ZH

Figure 5: Average Indirect Effects of all model components (Llama3-8B) of 100 factual knowledge in six languages.

pared to LangEdit across both editing accuracy and multilingual generalization, reinforcing the importance of multilingual sequential knowledge editing. **LangEdit with shuffled language order of the knowledge):** By randomizing the editing sequence for three times while preserving the content of the knowledge, we observe comparable performance to using the knowledge sequence without this shuffling. This suggests that while sequential editing is crucial, the specific ordering of languages carries minimal importance.

A.3 Package Version

Pytorch version is 2.3.0 and transformer is 3.9.1.

A.4 Computational Analysis

To evaluate the computational cost, we leverage the time per batch (100 edits) and memory cost. We conduct multilingual sequential knowledge editing on an NVIDIA A100-SXM4-80GB GPU. When we edit Llama3-8B on the MzsRE dataset, Time per Batch (s) and Memory (GB) for all models are shown in Table 2. When editing Qwen-7B on the MzsRE dataset, Time per Batch (s) and Memory (GB) for all models are shown in Table 4. When editing GPT-J-6B on the MzsRE dataset, Time per Batch (s) and Memory (GB) for all models are shown in Table 3.

Method	Time per Batch (s)	Memory (GB)
FT	69	36.8
ROME	1804	36.7
MEMIT	974	35.8
PRUNE	1073	35.8
RECT	991	35.8
AlphaEdit	1277	37.9
LangEdit	1549	41.5

Table 2: Computational cost of knowledge methods when editing the Llama3-8B.

A.5 The Analysis on Knowledge Sharing

To explore whether a fact expressed in one language remains consistent across other languages, we conduct the following experiment, illustrated with the example below. Imagine injecting new facts into an English-Spanish bilingual model:

- We edit the LLM with English data (e.g., "Carl Sagan → worked at BBC").
- We keep the original Spanish data unchanged.
- If the model correctly answers the Spanish query ’Dónde trabajó Carl Sagan?’ (’Where did Carl Sagan work?’)—despite never having seen this edit in Spanish—it demonstrates successful cross-lingual transfer enabled by LangEdit’s architecture.

Method	Time per Batch (s)	Memory (GB)
FT	68	28.1
ROME	1509	30.9
MEMIT	792	35.6
PRUNE	809	35.6
RECT	845	35.6
AlphaEdit	963	34.7
LangEdit	1098	37.9

Table 3: Computational cost of knowledge methods when editing the GPT-J-6B.

Method	Time per Batch (s)	Memory (GB)
FT	69	32.6
ROME	1183	36.5
MEMIT	450	35.7
PRUNE	479	35.7
RECT	485	35.7
AlphaEdit	524	34.4
LangEdit	793	38.3

Table 4: Computational cost of knowledge methods when editing the Qwen-7B.

The experimental results for exploring cross-lingual knowledge transfer with queries in the test language that have never been seen in the edits are shown in Table 6 and Table 7. From Table 6, we can observe that the evaluation scores of LangEdit (Spanish) and LangEdit (French) are higher than the Llama3 (Spanish) and Llama3 (French). The same phenomenon appears in Table 7. This indicates that the edited knowledge in a certain language can propagate to other languages when using LangEdit.

A.6 Experimental Results on the MLaKE Dataset

MLaKE is a multilingual benchmark designed to evaluate the performance of knowledge editing methods in large language models across different languages and reasoning complexities. It contains 4072 multi-hop and 5360 single-hop question-answer pairs in five languages: English, Chinese,

Methods	mzsre			XTREME ‡	bzsre			XTREME †
	Efficacy↑	**Generality↑**	**Specificity↑**	**F1↑**	**Efficacy↑**	**Generality↑**	**Specificity↑**	**F1↑**
AlphaEdit (translation)	63.01	55.57	24.24	75.53	53.91	47.74	24.66	74.51
AlphaEdit (multilingual)	45.35	37.53	11.23	14.53	39.15	22.21	9.98	4.66
LangEdit (shuffle order)	81.69	76.75	32.77	65.37	72.97	67.11	30.09	73.01
LangEdit	82.54	77.53	31.90	66.24	73.18	66.95	31.11	73.14

Table 5: Performance of variant models for multilingual editing using Llama3-8B and evaluated on the mzsre and bzsre datasets. XTREME‡ represents average F1 Scores on XTREME tasks after training on the mzsre dataset; XTREME† denotes the average F1 Scores after training on the bzsre dataset. All baselines are adapted for multilingual sequential knowledge editing.

Models	Test language	Efficacy	Generality	Specificity
LangEdit	French	56.41	53.91	30.54
Llama3 (unedited)	French	31.51	30.90	30.21
LangEdit	Spanish	55.89	53.84	31.54
Llama3 (unedited)	Spanish	31.71	31.62	31.33

Table 6: Experimental results of cross-lingual knowledge transfer when editing Llama3-8B in French and Spanish.

Models	Test language	Efficacy	Generality	Specificity
LangEdit	English	59.02	57.37	31.14
Llama3 (unedited)	English	31.03	30.92	30.97
LangEdit	German	58.12	56.30	30.16
Llama3 (unedited)	German	30.15	29.71	29.95
LangEdit	Dutch	56.33	53.31	30.82
Llama3 (unedited)	Dutch	30.62	30.14	30.37

Table 7: Experimental results of cross-lingual knowledge transfer when editing Llama3-8B in English, German and Dutch.

Japanese, French, and German. Each instance is based on fact chains aligned from Wikipedia, covering both shallow and complex reasoning paths. MLaKE enables systematic evaluation of cross-lingual transferability, multi-hop reasoning, and the limitations of current multilingual knowledge editing approaches. The MLaKE dataset does not provide subject items for the knowledge triplets. We conduct multilingual sequential knowledge editing on the MLaKE dataset in three languages (English, German and French), where we have manually annotated the subject items for a small subset (100 examples for each language). Experimental results on the MLaKE dataset are shown in the Table 8.

Models	**Single-hop**		**Multi-hop**	
	Efficacy	**XTREME (F1)**	**Efficacy**	**XTREME (F1)**
AlphaEdit (adapted)	90.62	69.96	88.48	68.17
LangEdit	91.44	70.89	90.05	69.09

Table 8: Comparison of models on Single-hop and Multi-hop tasks using Efficacy and XTREME (F1) metrics.