MEMoE: Enhancing Model Editing with Mixture of Experts Adaptors

Anonymous ACL submission

Abstract

Model editing aims to efficiently modify the behavior of Large Language Models (LLMs) within a desired scope, while preserving their original capabilities. However, existing methods overlook the long-tail distribution of the knowledge to be edited, leading to compromised performance in reliability, generalization and locality. Through empirical analysis, we find that high-frequency knowledge tends to overfit, resulting in high reliability but poor locality, while long-tail knowledge suffers from sparse semantics, leading to degraded generalization. To address this, we propose MEMoE, an advanced model editing framework based on a Mixture of Experts (MoE) architecture, which aligns sparse parameter activations with long-tail knowledge distributions. MEMoE incorporates a single-layer frequency-specialized MoE mechanism to ensures different experts specialize in knowledge of varying frequencies, along with a dual-attention router that directs inputs to the appropriate expert based on the integrated semantic representations before and after editing. To mitigate overfitting to high-frequency knowledge and enhance the learning of long-tail knowledge, we introduce a balancing constraint loss. Experimental results show that MEMoE outperforms existing methods across various model types and editing tasks, while preserving the general abilities of LLMs on downstream tasks.

1 Introduction

Large Language Models (OpenAI, 2024; Anthropic, 2024; Google, 2024) learn a vast repository of world knowledge during pre-training, which can be accessed and utilized through natural language prompts (Petroni et al., 2019). However, due to the ever-evolving nature of the real world, these models must be regularly updated to correct outdated or incorrect information (Yao et al., 2023, 2024b). Retraining or fine-tuning LLMs to reflect such updates is often impractical, given the substantial resources and time required (Li et al., 2024b; Yao et al., 2024b). To address this challenge, the concept of model editing, also known as knowledge editing, has been introduced (Zhang et al., 2024c). This paradigm aims to efficiently modify a model's output for specific knowledge queries while preserving its overall performance on unrelated inputs. Recent works have explored various editing scenarios, including single editing (e.g., ROME (Meng et al., 2022)), batch editing (e.g., MEMIT (Meng et al., 2023)), and sequential editing (e.g., GRACE (Hartvigsen et al., 2023)).

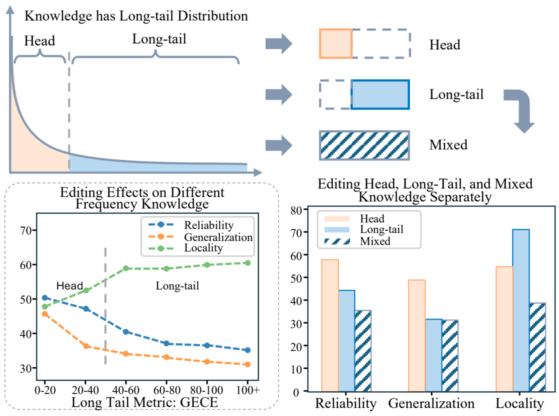

Figure 1: Model editing performance across knowledge frequency. The lower-left subfigure shows the performance differences when simultaneously editing knowledge of different frequencies. The lower-right subfigure compares the performance of separately editing head and long-tail knowledge versus simultaneous mixed-frequency editing.

Despite these advances, existing approaches largely ignore the long-tail distribution of knowledge to be edited, which significantly impacts performance across three critical dimensions: reliability, generalization, and locality. For example, high-frequency knowledge (e.g., "The president of the United States is Donald Trump") and long-tail knowledge (e.g., "Kruger National Park is located in the Mpumalanga province of South Africa") exhibit distinct patterns in large-scale pre-training corpora. However, current methods adopt a uniform parameter update strategy, failing to account for the frequency-specific characteristics of knowledge—a factor that becomes particularly critical when updating model knowledge with limited data.

Our empirical analysis (§2.2) reveals distinct editing behaviors: (1) High-frequency knowledge tends to overfit. Edits on head knowledge often achieve high reliability and generalization but suffer from reduced locality due to parameter drift (e.g., modifying "the president of America" may inadvertently affect related facts like "the population of America"). (2) Long-tail knowledge tends to underfit. While target knowledge can be injected successfully (e.g., correcting the location of a national park to the Mpumalanga province), the model struggles to generalize to related queries (e.g., "What is the capital of the province where Kruger Park is located?"). However, this also results in improved locality for edits on tail knowledge.

In light of these, we propose **MEMoE**, a **M**odel **E**diting framework based on a **M**ixture **o**f **E**xperts architecture. MEMoE introduces a single-layer frequency-specialized MoE structure that explicitly allocates different experts to handle knowledge at varying frequencies, aligning with the inherent long-tail distribution of knowledge. Additionally, we propose a dual-attention router that dynamically directs inputs to the appropriate expert by leveraging semantic representations both before and after editing. To further enhance performance, we introduce a balancing constraint loss that mitigates high-frequency overfitting and promotes effective learning of long-tail knowledge.

We validate MEMoE across three model families (GPT2, LLaMA2, and BLOOMZ) and two widely used editing benchmarks (ZsRE (Levy et al., 2017) and COUNTERFACT (Meng et al., 2022)). Experimental results demonstrate that MEMoE consistently outperforms existing editing methods while preserving the general capabilities of LLMs on downstream tasks.

The main contributions of this work are:

- We analyze how editing performance varies with the frequency of knowledge and demonstrate the benefit of frequency-aware editing.
- We propose MEMoE, a novel framework for model editing, featuring a frequency-specialized MoE structure, dual-attention routing, and a balancing constraint loss.
- Experimental results show the efficacy of our proposed method across various model types and editing tasks, while preserving the general abilities of LLMs on downstream tasks.

2 Preliminary and Analysis

2.1 Preliminary

Based on the prior works (Yao et al., 2023; Wang and Li, 2024a), the task of model editing involves effectively modify an initial base model f_θ (θ represents the model's parameters) into an edited model $f_{\theta'}$. The goal is to adjust the model's responses to a set of specified edit instances as desired, while preserving its behavior on all other instances (Li et al., 2024b). The intended edit descriptor is denoted as $\{(x_i^e, y_i^e)\}_{i\in[1,N]}$, where $f_\theta(x_i^e) \neq y_i^e$ and N represents the total number of editing instances. This set of intended instances is referred to as the editing scope I_{edit}, while the out-of-scope O_{edit} refers to inputs set that are not relevant to the editing examples. Formally, a successful editing can be expressed as:

$$f_{\theta'}(x_i) = \begin{cases} y_i^e & \text{if } x_i \in I_{edit} \\ f_\theta(x_i) & \text{if } x_i \in O_{edit} \end{cases} \quad (1)$$

Problem settings for model editing usually fall into four categories (Yao et al., 2023; Li et al., 2024b): single editing, batch editing, sequential editing and sequential batch editing.

1) **Single Editing** assesses model performance after a single knowledge update.:

$$\theta' \leftarrow \underset{\theta}{\operatorname{argmin}}(\| f_\theta(x_i^e) - y_i^e \|) \quad (2)$$

2) **Batch Editing** assesses model performance when multiple knowledge pieces are modified simultaneously ($n \leq N$ represents the batch size):

$$\theta' \leftarrow \underset{\theta}{\operatorname{argmin}} \sum\nolimits_{i=1}^{n} (\| f_\theta(x_i^e) - y_i^e \|) \quad (3)$$

3) **Sequential Editing** requires that every single edit is executed successively and evaluation conducted only after all edits are completed (Hartvigsen et al., 2023):

$$\theta' \leftarrow \underset{\theta}{\operatorname{argmin}} \sum\nolimits_{i=1}^{N} (\| f_\theta(x_i^e) - y_i^e \|) \quad (4)$$

4) **Sequential Batch Editing** aims to perform edits in a sequential manner and in batches (n represents the batch size, S represents sequential editing step):

$$\theta' \leftarrow \underset{\theta}{\operatorname{argmin}} \sum_{s=0}^{S} \sum_{i=s\times n}^{(s+1)\times n} (\| f_\theta(x_i^e) - y_i^e \|) \quad (5)$$

Based on the above settings, a successful model editor should meet requirements of the following three properties: Reliability, Generalization, and Locality (Yao et al., 2023). Formally, these can be expressed as:
1) **Reliability** measures the average accuracy of the post-edit model $f_{\theta'}$ on intended edits:

$$\mathbb{E}_{(x_i^e, y_i^e)\sim I_{edit}} \mathbb{1}\left\{\operatorname{argmax}_y f_{\theta'}\left(y \mid x_i^e\right) = y_i^e\right\} \quad (6)$$

2) **Generalization** measures the average accuracy of the model $f_{\theta'}$ on examples drawn uniformly from the equivalent neighborhood N_{edit} which includes input/output pairs related to I_{edit}:

$$\mathbb{E}_{(x_i, y_i^e)\sim N_{edit}} \mathbb{1}\left\{\operatorname{argmax}_y f_{\theta'}\left(y \mid x_i\right) = y_i^e\right\} \quad (7)$$

3) **Locality** is evaluated by the rate at which the predictions of the post-edit model $f_{\theta'}$ remain unchanged compared to the pre-edit model f_θ:

$$\mathbb{E}_{(x_i, y_i)\sim O_{edit}} \mathbb{1}\left\{f_{\theta'}\left(y \mid x_i\right) = f_\theta\left(y \mid x_i\right)\right\} \quad (8)$$

2.2 Empirical Analysis

Current model editing methods often overlook the long-tail distribution inherent in the knowledge to be edited. These approaches typically apply uniform editing strategies regardless of the frequency of the knowledge. In this section, we investigate how editing performance varies with knowledge frequency.

To quantify the frequency of knowledge, we adopt a newly proposed metric called Generative Expected Calibration Error (GECE) (Li et al., 2024a), which reflects the degree of semantic sparsity and frequency. The formal definition of GECE is as follows:

$$GECE = \frac{\left|M(pred, ref) - \frac{1}{n}\sum_{i=1}^{n} p(t_i)\right|}{\alpha \cdot \left[E(\bigtriangledown_{ins}) \cdot \bigtriangledown_{ins}\right]} \quad (9)$$

where $pred$ and ref represent the generated text and the ground truth, respectively, and $M(pred, ref)$ denotes the METEOR score (Banerjee and Lavie, 2005). The average token probability is given by $\frac{1}{n}\sum_{i=1}^{n} p(t_i)$ where $p(t_i)$ denotes the i-th token's probability produced by LLM, and n is the token sequence length. α represents the average word frequency, $\bigtriangledown_{ins}$ is the gradient with respect to the current instance, and $E(\bigtriangledown_{ins})$ is the mean gradient over the entire dataset. A larger GECE value indicates more long-tail knowledge. For example, the query "Who has played Raoul in The Phantom of the Opera" has a GECE of 112.7, while "Who was named African footballer of the year 2014" yields 34.6.

For evaluation, we first sample six frequency buckets from the ZsRE and COUNTERFACT datasets, with 100 editing instances per bucket based on GECE scores. We then apply two representative model editing methods—MEMIT and GRACE—on LLaMA2-7B, and assess their performance using three widely adopted metrics: reliability, generalization, and locality. The averaged results across frequency groups are reported in lower-left of Figure 1.

Our results reveal a clear trend. Overall, the editing performance deteriorates as the knowledge becomes more long-tail. Specially, (1) **High-frequency knowledge** yields higher reliability and generalization but suffers from lower locality. This suggests strong parameter entanglement that supports accurate updates but also leads to parameter drift (Zhang et al., 2024a). The dense semantic interconnections of head knowledge within the model's parameter space facilitate effective editing and knowledge propagation, albeit at the expense of broader parametric influence. (2) **Long-tail knowledge**, by contrast, preserves locality more effectively but exhibits lower reliability and generalization. This phenomenon may arise from the sparse representation of tail knowledge in the model's parameter space due to limited training data, suggesting that tail knowledge might be underfitted (Mao et al., 2025). Even when target knowledge is successfully injected, the model struggles to generalize to related queries.

Moreover, we compare three editing regimes: editing only Head, only Long-tail, or Mixed knowledge. As shown in lower-right Figure 1, editing knowledge of similar frequency improves overall performance, suggesting that frequency-aware editing is beneficial. More experimental details can be found in Appendix B.

Therefore, we propose MEMoE, which leverages the sparse activation properties of MoE to enable frequency-aware knowledge editing by employing specialized expert modules designed for different frequency bands.

3 Our Approach: MEMoE

Based on the above insights, in this section, we provide a detailed introduction to MEMoE.

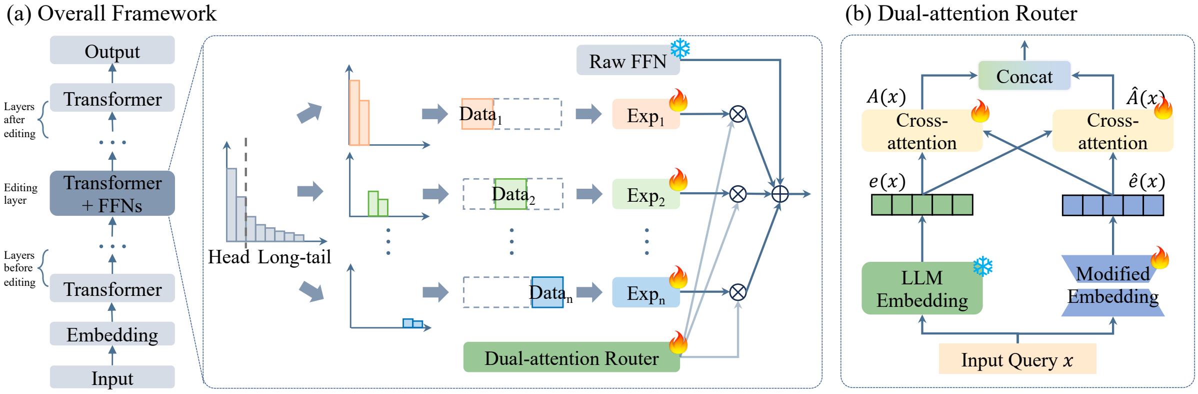

Figure 2: *Left:* The architecture of MEMoE, which implements model editing through parallel experts alongside the original FFN. *Right:* Overview of the dual-attention router structure.

3.1 Single-Layer Frequency-Specialized MoE

Inspired by traditional MoE (Jacobs et al., 1991), MEMoE introduces multiple parallel experts within the transformer feed-forward network (FFN) via a bypass mechanism, while freezing all the model's original parameters (left part of Figure 2). This module is applied in only one transformer block of the entire model. The choice to use the FFN module is not only due to its traditional role in MoE (Cai et al., 2024) but also aligns with recent experimental findings of knowledge probing technologies that the MLP layers within FFN store knowledge (Dai et al., 2022; Meng et al., 2022, 2023). The bypass mechanism preserves all the original parameters of the model, enhancing the locality of model editing.

Specially, let $\{E_i\}_{i=1}^{n}$ represent the set of n experts in the MEMoE layer, and let $g(i \mid x)$ represent a router that outputs the corresponding coefficients for each expert E_i based on the input x. The output h of the MEMoE layer can be expressed as:

$$
\begin{aligned}
h(x) &= \boldsymbol{W}_0 \cdot x + \lambda \sum_{i=1}^{t+1} g(i \mid x) E_i(x) \\
g(i \mid x) &= \mathrm{Top}_k\left(\frac{e^{r(x)_i}}{\sum e^{r(x)_j}}\right)
\end{aligned}
\tag{10}
$$

where $\boldsymbol{W}_0$ is the frozen original FFN parameters, $r(x)$ is the routing strategy and is modeled by one MLP in conventional MoE. λ is a non-negative weighting coefficient used to balance the old and new knowledge, usually set to 1.

Furthermore, considering the varying editing effects of different frequency knowledge observed in §2.2, we hypothesize that learning a relatively uniform distribution may be easier than learning an imbalanced distribution. However, since the amount of data for model editing is typically small, allowing the model to learn frequency-based knowledge handling independently may encounter challenges such as cold-start issues. Therefore, we explicitly assign knowledge of different frequencies to distinct experts within the MoE for learning. By leveraging sparse parameter activation patterns in conjunction with the long-tail distribution of knowledge, we ensure that different experts specialize in knowledge of varying frequencies.

Specially, let $\{x_i\}_{i=1}^{N}$ denote the dataset of N editing data points, sorted in ascending order based on their GECE values (Equation 9): $GECE(x_1) \leq GECE(x_2) \leq \cdots \leq GECE(x_N)$. We aim to assign the data to the n experts based on their long-tail distribution. Therefore, we divide the dataset into two parts:

1) The first $p\%$ of the data points (high-frequency knowledge) are assigned to the first expert e_1:

$$
D_1 = \{x_i \mid 1 \leq i \leq \lfloor pN \rfloor\} \tag{11}
$$

2) The remaining $(1-p)\%$ of the data points (long-tail knowledge) are distributed among the remaining $n-1$ experts using a balanced clustering approach. Let D_j denote the data assigned to expert e_j ($j = 2, \ldots, n$). The objective function for the balanced clustering is given by:

$$
\begin{aligned}
J = &\sum_{j=2}^{n} \sum_{x_i \in D_j} \|GECE(x_i) - \mu_j\|^2 \\
&+ \lambda \sum_{j=2}^{n} \left| |D_j| - \frac{\lceil (1-p)N \rceil}{n-1} \right|
\end{aligned}
\tag{12}
$$

where μ_j is the mean value of cluster D_j, and λ controls the strength of the balance constraint.

This distribution ensures that the first expert specializes in the high-frequency knowledge and the remaining specialize in long-tail knowledge, maintaining a balanced workload across experts.

3.2 Dual-Attention Router

In general, the semantic embedding space contains vast amounts of knowledge. During the editing process, the semantic representation of the knowledge being edited undergoes changes. Based on this, we propose the dual-attention router, which integrates the semantic representation of the input query both before and after the model editing to achieve accurate routing (right part of Figure 2).

Specifically, for an input instance x, the LLM embedding $e(x) \in \mathbb{R}^d$ can be obtained by extracting the last hidden representation of the final token in the input sequence. Given that the amount of data involved in model editing is typically small (Zhang et al., 2024c; Wang et al., 2024), directly modifying the embedding space could lead to collapse. Instead, we use an adapter module to approximate the edited semantic change:

$$\hat{e}(x) = \boldsymbol{W}^{\text{up}}(\boldsymbol{W}^{\text{down}} \cdot e(x) + b_1) + b_2 \qquad (13)$$

where $\boldsymbol{W}^{\text{down}} \in \mathbb{R}^{d_p \times d}$, $\boldsymbol{W}^{\text{up}} \in \mathbb{R}^{d \times d_p}$, $b_1 \in \mathbb{R}^{d_p}$ and $b_2 \in \mathbb{R}^d$ are weight matrice and bias of adapter.

Then, we introduce dual attention to integrate the two semantic embeddings. To simplify the description, we illustrate this with the interaction based on the pre-edited embeddings; the process for the other is similar. Let $\hat{e}(x)$ be the query, and $e(x)$ serve as both the key and value in the attention mechanism. Define $Q = W_q \cdot \hat{e}(x)$, $K = W_k \cdot e(x)$, and $V = W_v \cdot e(x)$, where $W_q, W_k, W_v \in \mathbb{R}^{d \times d}$ are the weight matrices. The embedded sequence after interaction can then be expressed as follows:

$$A(x) = \text{Softmax}(\frac{QK^T}{\sqrt{d}})V \qquad (14)$$

Finally, routing decisions are made based on the two resulting semantic representations: $A(x)$ (derived from the interaction with the pre-edited embedding) and $\hat{A}(x)$ (from the interaction with the post-edited embedding).

$$r(x) = \text{Softmax}(\boldsymbol{W}_r \cdot (\alpha A(x) + \beta \hat{A}(x))) \qquad (15)$$

where α, β and $\boldsymbol{W}_r$ is learnable weights. It is worth noting that this router directs all tokens in the same instance to one expert, thereby guaranteeing equal treatment of the entire knowledge.

3.3 Balancing Constraint Loss

To address the distinct learning dynamics between high-frequency and long-tail knowledge, we propose a balancing constraint loss that combines adaptive weighting with parameter-space regularization. Our key insight arises from two fundamental observations: (1) High-frequency knowledge tends to dominate gradient directions due to its dense parameter associations in pretrained models (Wang and Li, 2024b), while long-tail knowledge updates are often overshadowed by these dominant signals (Kandpal et al., 2023), resulting in an imbalanced parameter update landscape. (2) Directly applying larger learning rates to long-tail samples may destabilize the well-formed semantic manifold of pretrained models, particularly harming the locality preservation of high-frequency knowledge. Given that model editing typically involves fewer parameter updates, we find that incorporating parameter regularization significantly alleviates these issues.

Specifically, given a batch of editing samples $\{(x_i^e, y_i^e)\}_{i=1}^N$, we first introduce an adaptive weight into the original model loss function:

$$\mathcal{L}_{\text{model}} = \sum_{i=1}^{N} -w(g_i) \cdot \log f_\theta(y_i^e \mid x_i^e) \qquad (16)$$

where g_i represents the normalized long-tail scores $GECE(x_i)$ for short, and the adaptive weight $w(\cdot)$ follows a sigmoidal transition:

$$w(x) = \frac{1}{2}(1 + \tanh(\gamma_1(x - \tau))) \qquad (17)$$

This introduces soft thresholds (with $\tau = 0.6, \gamma_1 = 1$ in practice) to gradually suppress high-frequency samples ($x < \tau$) while amplifying gradient signals for long-tail knowledge ($x > \tau$).

Next, we apply a regularization to impose stronger constraints on parameters primarily associated with high-frequency knowledge:

$$\mathcal{L}_{\text{balance}} = -\sum_{i=1}^{N} \frac{1}{1 + e^{\gamma_2 \cdot g_i}} \cdot \mathbb{D}_{\text{KL}}(f'_\theta \| f_\theta) \qquad (18)$$

where $\mathbb{D}_{\text{KL}}$ is the Kullback-Leibler Divergence, f_θ and f'_θ represent the model before and after editing. Similar to $w(x)$, the coefficient of $\mathcal{L}_{\text{balance}}$ also suppresses high-frequency samples while amplifying long-tail knowledge.

Third, we introduce router guidance loss to enforces clear routing decisions for different samples:

$$\mathcal{L}_{\text{router}} = -\sum_{i=1}^{N} (1 - r(x_i))^{\gamma_3} \log(r(x_i)) \qquad (19)$$

where $r(\cdot)$ is the router function (Equation 15), and γ_3 controls the suppression strength. A larger value of γ_3 leads to a stronger suppression of easy-to-learn knowledge.

Finally, the loss of MEMoE can be represented as follows:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{model}} + \mathcal{L}_{\text{balance}} + \mathcal{L}_{\text{router}} \tag{20}$$

4 Experiments

4.1 Experimental Setups

Datasets and Metrics: We use two widely used model editing datasets: ZsRE (Levy et al., 2017) and COUNTERFACT (Meng et al., 2022), with the split provided by (Zhang et al., 2024c; Yao et al., 2023). ZsRE is a context-free Question Answering (QA) dataset built upon zero-shot relation extraction and COUNTERFACT is a more challenging dataset that accounts for counter facts that start with low scores in comparison to correct facts. Further details are provided in Appendix C.1. In terms of evaluation metrics, we use the three metrics described in §2.1: Reliability (Rel.), Generalization (Gen.), Locality (Loc.), and the average scores over these metrics (Avg.).

Baselines: We compare the proposed method with mainstream model editing methods, which can be categorized into the following four types:

- **Fine-tuning based methods:** FT-L (Meng et al., 2022), FT-M (Hartvigsen et al., 2023), and LoRA (Hu et al., 2022). FT-L directly fine-tunes a single layer's FFN and FT-M is a small variation of FT-L using a different loss computation procedure. LoRA is a parameter-efficient fine-tuning method which decomposes the update gradient matrix into small rank matrices.
- **Locate and edit methods:** MEMIT (Meng et al., 2023). MEMIT treats the FFN as a linear associative memory and uses a minimum square error optimization to add new key-value associations to layer weights.
- **Meta-learning methods:** MEND (Mitchell et al., 2022a) and COMEBA-HK (Li et al., 2024b). MEND learns a hyper-network using additional training data to transform gradient obtained by standard fine-tuning, while COMEBA-HK (COMEBA for short) develop hook layers to identify the editing scope.
- **Memory based methods:** SERAC (Mitchell et al., 2022b) and GRACE (Hartvigsen et al., 2023). The SERAC uses an external cache to store explicit editing cases, while GRACE preserves the original model parameters and adopts a codebook to store relevant edits.

Implementation Details: We select GPT2-XL and LLaMA2-7B as the base models. The modification is applied to layer 16 for LLaMA2-7B and layer 18 for GPT2-XL (consistent with the findings of ROME (Meng et al., 2022)), with the number of experts set to 5 and $top_k = 1$ to yield the best experimental results within our computational resources. Further details of the baselines and the implementation are provided in the Appendix C. In this section, We opted for batch editing and sequential batch editing to evaluate the performance of MEMoE. Following previous research (Li et al., 2024b), batch editing uses a batch size of 30, while sequential batch editing uses a batch size of 10 for 1000 edits in total. We further report the experimental results for all editing types across various model types and sizes in §D.1.

4.2 Main Results

Batch Editing The results for batch editing are presented in the upper half of Table 1. Overall, MEMoE consistently outperforms the baselines across all datasets and base models. Although some methods, such as SERAC and GRACE, achieve high scores in certain metrics, these gains often come at the cost of significant drops in others. Notably, MEMoE demonstrates exceptional balance: it maintains high locality while achieving unparalleled reliability and generalization, effectively addressing the common trade-off between edit specificity and knowledge retention. These results validate MEMoE's design, where its MoE architecture isolates high-frequency and long-tail knowledge into specialized experts, enabling precise and conflict-free updates while preserving the model's original knowledge.

Sequential Batch Editing The results for sequential batch editing are shown in the lower half of Table 1, indicate that MEMoE achieves the best scores in most cases. Methods such as FT-L and LoRA, which are not specifically designed for sequential editing, tend to forget prior updates, leading to significantly lower scores. GRACE, although a strong baseline in reliably retaining previous edits through its discrete data adapter mapping, struggles with handling semantically equivalent inputs, resulting in poor generalization performance, as high-

Table 1: Batch editing and sequential batch editing results. **Bold** is the best result.

Method	Model	ZsRE				CounterFact			
		Reliability↑	Generalization↑	Locality↑	Average↑	Reliability↑	Generalization↑	Locality↑	Average↑
Batch Editing									
FT-L	GPT2-XL	16.85	16.34	71.55	34.91	0.27	0.34	85.18	28.60
FT-M	GPT2-XL	17.95	17.32	71.26	35.51	0.36	0.42	82.81	27.86
LoRA	GPT2-XL	30.10	29.08	80.54	46.57	5.64	3.46	69.45	26.18
MEMIT	GPT2-XL	60.29	44.22	86.85	63.78	80.52	23.56	90.66	64.92
MEND	GPT2-XL	2.16	2.11	20.34	8.20	0.13	0.03	4.22	1.46
COMEBA	GPT2-XL	76.58	62.28	90.58	76.48	84.62	40.07	96.51	73.73
SERAC	GPT2-XL	92.98	43.72	63.14	66.61	41.87	28.23	78.89	49.66
GRACE	GPT2-XL	70.19	37.45	92.83	66.82	89.21	85.25	91.75	88.74
MEMoE		**95.05**	**83.07**	**93.84**	**90.65**	**94.56**	**88.96**	**96.74**	**93.42**
FT-L	LLaMA2-7B	14.19	13.07	70.16	32.47	0.21	0.30	80.69	27.07
FT-M	LLaMA2-7B	16.57	15.62	70.15	34.11	0.29	0.38	81.83	27.50
LoRA	LLaMA2-7B	25.32	23.15	52.01	33.49	21.70	22.32	40.37	28.13
MEMIT	LLaMA2-7B	24.02	39.97	17.00	27.00	18.57	31.29	14.88	21.58
MEND	LLaMA2-7B	6.51	3.06	28.12	12.56	5.91	3.26	27.42	12.20
SERAC	LLaMA2-7B	89.08	36.29	71.82	65.73	50.67	27.34	82.05	53.35
GRACE	LLaMA2-7B	90.19	37.58	90.20	71.57	77.40	27.37	92.45	65.74
MEMoE		**92.43**	**85.03**	**94.12**	**90.53**	**95.96**	**80.28**	**95.53**	**90.59**
Sequential Batch Editing									
FT-L	GPT2-XL	3.79	2.48	6.60	4.29	1.00	1.00	6.00	2.67
FT-M	GPT2-XL	8.92	8.41	6.22	7.85	4.00	3.50	5.50	4.33
LoRA	GPT2-XL	0.96	1.29	0.03	0.76	0.50	0.02	0.50	0.34
MEMIT	GPT2-XL	34.88	32.96	70.74	46.19	56.00	37.00	31.00	41.33
MEND	GPT2-XL	20.95	18.29	87.69	42.31	4.01	2.01	6.08	4.03
COMEBA	GPT2-XL	59.97	54.81	89.45	68.08	81.24	29.79	50.83	53.95
SERAC	GPT2-XL	83.11	28.36	29.33	46.93	56.91	38.42	71.94	55.75
GRACE	GPT2-XL	81.38	7.47	89.46	59.44	77.68	15.16	87.09	59.97
MEMoE		79.74	**56.33**	**90.61**	**75.56**	**79.80**	**45.32**	**90.98**	**72.03**
FT-L	LLaMA2-7B	2.33	1.59	6.67	3.53	0.23	0.18	10.66	3.69
FT-M	LLaMA2-7B	6.72	4.37	7.78	6.29	0.33	0.70	8.54	3.19
LoRA	LLaMA2-7B	0.35	1.89	0.07	0.77	0.31	0.99	0.17	0.49
MEMIT	LLaMA2-7B	12.29	29.95	15.38	19.21	10.37	32.96	12.79	18.71
SERAC	LLaMA2-7B	67.78	33.98	34.55	45.44	20.21	14.05	34.90	23.05
GRACE	LLaMA2-7B	73.73	9.35	87.76	56.95	66.68	21.72	81.46	56.62
MEMoE		**74.24**	**36.64**	**90.45**	**67.11**	**78.51**	**33.77**	**84.40**	**65.56**

lighted in previous studies (Zhang et al., 2024b). Similarly, SERAC demonstrates strong reliability in editing but falls short in generalization. In contrast, although MEMoE is not explicitly optimized for sequential editing, it excels in preserving prior edits and effectively generalizing to rephrased inputs, further confirming its superior performance.

5 Discussion

In this section, we evaluate the impact of model editing on the generalization ability of the model in downstream tasks, along with some ablation studies. A more comprehensive analysis can be found in Appendix D, including performance across all four editing tasks for various model types and sizes (§D.1), computational analysis (§D.2), case studies (§D.6), and additional insights.

5.1 General Ability Test

Considering some current researches concern that model editing methods may significantly affect a model's general ability (Gu et al., 2024; Gupta et al., 2024; Pinter and Elhadad, 2023), we select eight representative task categories for evaluation, as outlined below following (Gu et al., 2024). For **reasoning**, we utilized the GSM8K dataset (Cobbe et al., 2021), with performance assessed by solve rate. **Natural language inference (NLI)** tasks were evaluated on the RTE dataset (Candela et al., 2006), with accuracy measured through two-way classification. For **open-domain question answering**, the Natural Question dataset (Kwiatkowski et al., 2019) was employed, evaluating exact match against reference answers after minor normalization as in (Chen et al., 2017) and (Lee et al., 2019). Similarly, **closed-domain QA** tasks were assessed using the BoolQ dataset (Clark et al., 2019), also measured by EM. **Dialogue** evaluation utilized the MuTual dataset (Cui et al., 2020), with results determined by selecting the most suitable response from four options, denoted as $\text{Recall}_4@1$ (Lowe et al., 2015). Evaluation for **summarization** tasks was conducted on the SAMSum dataset (Gliwa et al., 2019), using the average of ROUGE-1, ROUGE-2, and ROUGE-L as evaluation metrics. For **named entity recognition (NER)**, the CoNLL03 dataset (Sang and Meulder, 2003) was employed, with performance measured using entity-level F1-score. Lastly, for **sentiment analysis**, we utilized SST2 dataset (Socher et al., 2013), with accuracy as-

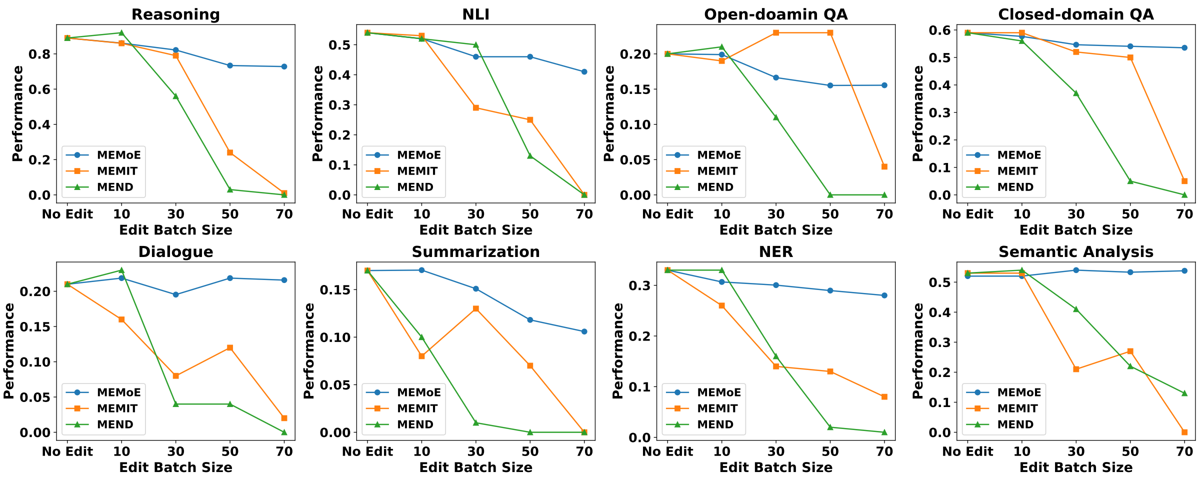

Figure 3: Performance on general tasks of edited models using MEMoE, MEMIT and MEND, with different batch sizes for editing.

Table 2: Results of ablation study. `ZsRE`. `LLaMA2-7B`.

	Rel.↑	Gen.↑	Loc.↑	Avg.↑
MEMoE	**92.43**	**85.03**	**94.12**	**90.53**
- MoE structure	38.10	35.08	83.54	52.24
- Data division strategy	87.24	82.64	92.45	87.44
- Dual-attention router	85.00	74.00	92.00	83.67
- Balancing constraint loss	84.29	83.44	92.10	86.61
- $\mathcal{L}_{\text{balance}}$	91.47	84.36	84.96	86.93
- $\mathcal{L}_{\text{router}}$	90.31	83.95	93.21	89.16

sessed through a two-way classification.

We conduct evaluations on LLaMA2-7B based on batch editing settings, progressively increasing the batch size to show the impact of more edited samples. The results are shown in the Figure 3. Compared to the MEMIT and MEND, the MEMoE yields consistently stable model performance under various batch editing conditions. With the increase in batch size and edited samples, both MEMIT and MEND significantly diminish the model's general ability, while the influence of MEMoE fluctuates within a smaller range. This further corroborates MEMoE's advantage in locality score in §4.2.

5.2 Ablation Study

We present ablation studies to evaluate the influence of each model component. First, we replace the sparse multi-expert structure (§3.1) with a dense adapter having similar parameters and also remove the proposed data division strategy (Equation 11-12). Second, we replace the dual-attention router (§3.2) with a conventional MoE router, a single MLP layer. Third, we substitute the Balancing Constraint Loss (§3.3) with the original cross-entropy loss function. Since the Balancing Constraint Loss is composed of three parts, we individually replace each part to assess its contribution.

The experimental results are shown in Table 2. The sparse multi-expert structure has the most significant impact on all evaluation metrics, as sparse activation is fundamental to MEMoE. Removing the data division strategy prevents the model from separately handling knowledge of varying frequencies, resulting in knowledge conflicts and a decrease in accuracy and generalization. The dual-attention router significantly affects both reliability and generalization, as proper routing of input information to the corresponding knowledge experts is essential for acquiring accurate knowledge and achieving generalization. As for the loss function, training the model with a simple cross-entropy loss leads to poor performance, and we also observe significant instability during training. Consistent with our design philosophy, the parameter-based regularization $\mathcal{L}_{balance}$ greatly impacts the model's locality. Without regularization, the model's generalization remains nearly unchanged, but locality significantly deteriorates. In contrast, the loss associated with the router $\mathcal{L}_{router}$ has a more comprehensive but smaller impact on overall performance.

6 Conclusion

In this paper, we present MEMoE, a model editing adapter utilizing MoE architecture, featuring a single-layer frequency-specialized MoE, dual-attention router and balancing constraint loss. Our approach emphasizes the critical role of knowledge frequency in editing performance and demonstrates the benefits of treating head and long-tail knowledge separately. Extensive experiments show that MEMoE consistently outperforms existing baselines while preserving the general capabilities of large language models on downstream tasks.

Limitation

First, although the proposed method demonstrates notable improvements, its performance in sequential batch editing tasks remains relatively limited and requires further refinement. In future work, we aim to design suitable continual learning strategies to mitigate the issue of catastrophic forgetting in sequential batch editing tasks. Additionally, the data division strategy outlined in §3.1 can be further explored to develop more refined approaches.

Second, the impressive performance of MEMoE highlights its promising potential for practical applications of model editing technology in specialized domains such as medicine and education. However, this study is confined to testing on mainstream model editing datasets. Future research could focus on evaluating its performance on domain-specific datasets to further advance the application of model editing technology. Furthermore, model editing techniques can be extended to various task types. Specifically, in addition to editing factual knowledge, they can be applied to address issues like eliminating hallucinations, mitigating biases, and protecting privacy. However, our experiments focus solely on general editing tasks, which are relatively well-explored and universally assessed in model editing, and do not tackle challenges such as reducing hallucinations.

Third, we focus on decoder-only autoregressive models, excluding encoder-decoder architectures, due to the widespread adoption of autoregressive models in contemporary mainstream systems (OpenAI, 2024; Touvron et al., 2023). Future research replicating our study with larger-scale models and alternative architectures would be valuable for confirming our findings.

References

Rahaf Aljundi, Punarjay Chakravarty, and Tinne Tuytelaars. 2017. Expert gate: Lifelong learning with a network of experts. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, pages 7120–7129. IEEE Computer Society.

Anthropic. 2024. Claude 3.5 sonnet.

Satanjeev Banerjee and Alon Lavie. 2005. METEOR: an automatic metric for MT evaluation with improved correlation with human judgments. In *Proceedings of the Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization@ACL 2005, Ann Arbor, Michigan, USA, June 29, 2005*, pages 65–72. Association for Computational Linguistics.

Weilin Cai, Juyong Jiang, Fan Wang, Jing Tang, Sunghun Kim, and Jiayi Huang. 2024. A survey on mixture of experts. *CoRR*, abs/2407.06204.

Joaquin Quiñonero Candela, Ido Dagan, Bernardo Magnini, and Florence d'Alché-Buc, editors. 2006. *Machine Learning Challenges, Evaluating Predictive Uncertainty, Visual Object Classification and Recognizing Textual Entailment, First PASCAL Machine Learning Challenges Workshop, MLCW 2005, Southampton, UK, April 11-13, 2005, Revised Selected Papers*, volume 3944 of *Lecture Notes in Computer Science*. Springer.

Danqi Chen, Adam Fisch, Jason Weston, and Antoine Bordes. 2017. Reading wikipedia to answer open-domain questions. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 1: Long Papers*, pages 1870–1879. Association for Computational Linguistics.

Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. 2019. Boolq: Exploring the surprising difficulty of natural yes/no questions. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 2924–2936. Association for Computational Linguistics.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. 2021. Training verifiers to solve math word problems. *CoRR*, abs/2110.14168.

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

Ronan Collobert, Samy Bengio, and Yoshua Bengio. 2001. A parallel mixture of svms for very large scale problems. In *Advances in Neural Information Processing Systems 14 [Neural Information Processing Systems: Natural and Synthetic, NIPS 2001, December 3-8, 2001, Vancouver, British Columbia, Canada]*, pages 633–640. MIT Press.

Leyang Cui, Yu Wu, Shujie Liu, Yue Zhang, and Ming Zhou. 2020. Mutual: A dataset for multi-turn dialogue reasoning. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 1406–1416. Association for Computational Linguistics.

Damai Dai, Li Dong, Yaru Hao, Zhifang Sui, Baobao Chang, and Furu Wei. 2022. Knowledge neurons in pretrained transformers. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022*, pages 8493–8502. Association for Computational Linguistics.

Payel Das, Subhajit Chaudhury, Elliot Nelson, Igor Melnyk, Sarathkrishna Swaminathan, Sihui Dai, Aurélie C. Lozano, Georgios Kollias, Vijil Chenthamarakshan, Jirí Navrátil, Soham Dan, and Pin-Yu Chen. 2024. Larimar: Large language models with episodic memory control. In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*. OpenReview.net.

Qingxiu Dong, Damai Dai, Yifan Song, Jingjing Xu, Zhifang Sui, and Lei Li. 2022. Calibrating factual knowledge in pretrained language models. In *Findings of the Association for Computational Linguistics: EMNLP 2022, Abu Dhabi, United Arab Emirates, December 7-11, 2022*, pages 5937–5947. Association for Computational Linguistics.

Nan Du, Yanping Huang, Andrew M. Dai, Simon Tong, Dmitry Lepikhin, Yuanzhong Xu, Maxim Krikun, Yanqi Zhou, Adams Wei Yu, Orhan Firat, Barret Zoph, Liam Fedus, Maarten P. Bosma, Zongwei Zhou, Tao Wang, Yu Emma Wang, Kellie Webster, Marie Pellat, Kevin Robinson, Kathleen S. Meier-Hellstern, Toju Duke, Lucas Dixon, Kun Zhang, Quoc V. Le, Yonghui Wu, Zhifeng Chen, and Claire Cui. 2022. Glam: Efficient scaling of language models with mixture-of-experts. In *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA*, volume 162 of *Proceedings of Machine Learning Research*, pages 5547–5569. PMLR.

David Eigen, Marc'Aurelio Ranzato, and Ilya Sutskever. 2014. Learning factored representations in a deep mixture of experts. In *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Workshop Track Proceedings*.

William Fedus, Barret Zoph, and Noam Shazeer. 2022. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. *J. Mach. Learn. Res.*, 23:120:1–120:39.

Bogdan Gliwa, Iwona Mochol, Maciej Biesek, and Aleksander Wawer. 2019. Samsum corpus: A human-annotated dialogue dataset for abstractive summarization. *CoRR*, abs/1911.12237.

Google. 2024. Our next-generation model: Gemini 1.5.

Jia-Chen Gu, Hao-Xiang Xu, Jun-Yu Ma, Pan Lu, Zhen-Hua Ling, Kai-Wei Chang, and Nanyun Peng. 2024. Model editing can hurt general abilities of large language models. *CoRR*, abs/2401.04700.

Akshat Gupta, Anurag Rao, and Gopala Anumanchipalli. 2024. Model editing at scale leads to gradual and catastrophic forgetting. *CoRR*, abs/2401.07453.

Tom Hartvigsen, Swami Sankaranarayanan, Hamid Palangi, Yoon Kim, and Marzyeh Ghassemi. 2023. Aging with GRACE: lifelong model editing with discrete key-value adaptors. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*.

Hussein Hazimeh, Zhe Zhao, Aakanksha Chowdhery, Maheswaran Sathiamoorthy, Yihua Chen, Rahul Mazumder, Lichan Hong, and Ed H. Chi. 2021. Dselect-k: Differentiable selection in the mixture of experts with applications to multi-task learning. In *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pages 29335–29347.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. Lora: Low-rank adaptation of large language models. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net.

Zeyu Huang, Yikang Shen, Xiaofeng Zhang, Jie Zhou, Wenge Rong, and Zhang Xiong. 2023. Transformer-patcher: One mistake worth one neuron. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.

Robert A. Jacobs, Michael I. Jordan, Steven J. Nowlan, and Geoffrey E. Hinton. 1991. Adaptive mixtures of local experts. *Neural Comput.*, 3(1):79–87.

Albert Q. Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de Las Casas, Emma Bou Hanna, Florian Bressand, Gianna Lengyel, Guillaume Bour, Guillaume Lample, Lélio Renard Lavaud, Lucile Saulnier, Marie-Anne Lachaux, Pierre Stock, Sandeep Subramanian, Sophia Yang, Szymon Antoniak, Teven Le Scao, Théophile Gervet, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2024. Mixtral of experts. *CoRR*, abs/2401.04088.

Michael I. Jordan and Robert A. Jacobs. 1994. Hierarchical mixtures of experts and the EM algorithm. *Neural Comput.*, 6(2):181–214.

Tianjie Ju, Weiwei Sun, Wei Du, Xinwei Yuan, Zhaochun Ren, and Gongshen Liu. 2024. How large language models encode context knowledge? A layer-wise probing study. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation, LREC/COLING 2024, 20-25 May, 2024, Torino, Italy*, pages 8235–8246. ELRA and ICCL.

Nikhil Kandpal, Haikang Deng, Adam Roberts, Eric Wallace, and Colin Raffel. 2023. Large language models struggle to learn long-tail knowledge. In *International Conference on Machine Learning, ICML*

2023, 23-29 July 2023, Honolulu, Hawaii, USA, volume 202 of *Proceedings of Machine Learning Research*, pages 15696–15707. PMLR.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur P. Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. 2019. Natural questions: a benchmark for question answering research. *Trans. Assoc. Comput. Linguistics*, 7:452–466.

Kenton Lee, Ming-Wei Chang, and Kristina Toutanova. 2019. Latent retrieval for weakly supervised open domain question answering. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 6086–6096. Association for Computational Linguistics.

Dmitry Lepikhin, HyoukJoong Lee, Yuanzhong Xu, Dehao Chen, Orhan Firat, Yanping Huang, Maxim Krikun, Noam Shazeer, and Zhifeng Chen. 2021. Gshard: Scaling giant models with conditional computation and automatic sharding. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.

Omer Levy, Minjoon Seo, Eunsol Choi, and Luke Zettlemoyer. 2017. Zero-shot relation extraction via reading comprehension. In *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017), Vancouver, Canada, August 3-4, 2017*, pages 333–342. Association for Computational Linguistics.

Mike Lewis, Shruti Bhosale, Tim Dettmers, Naman Goyal, and Luke Zettlemoyer. 2021. BASE layers: Simplifying training of large, sparse models. In *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pages 6265–6274. PMLR.

Daliang Li, Ankit Singh Rawat, Manzil Zaheer, Xin Wang, Michal Lukasik, Andreas Veit, Felix X. Yu, and Sanjiv Kumar. 2023a. Large language models with controllable working memory. In *Findings of the Association for Computational Linguistics: ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 1774–1793. Association for Computational Linguistics.

Dongyang Li, Junbing Yan, Taolin Zhang, Chengyu Wang, Xiaofeng He, Longtao Huang, Hui Xue, and Jun Huang. 2024a. On the role of long-tail knowledge in retrieval augmented large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics, ACL 2024 - Short Papers, Bangkok, Thailand, August 11-16, 2024*, pages 120–126. Association for Computational Linguistics.

Jiaang Li, Quan Wang, Zhongnan Wang, Yongdong Zhang, and Zhendong Mao. 2025. ELDER: enhancing lifelong model editing with mixture-of-lora. In *AAAI-25, Sponsored by the Association for the Advancement of Artificial Intelligence, February 25 - March 4, 2025, Philadelphia, PA, USA*, pages 24440–24448. AAAI Press.

Shuaiyi Li, Yang Deng, Deng Cai, Hongyuan Lu, Liang Chen, and Wai Lam. 2024b. Consecutive model editing with batch alongside hook layers. *CoRR*, abs/2403.05330.

Xiaopeng Li, Shasha Li, Shezheng Song, Jing Yang, Jun Ma, and Jie Yu. 2023b. PMET: precise model editing in a transformer. *CoRR*, abs/2308.08742.

Ryan Lowe, Nissan Pow, Iulian Serban, and Joelle Pineau. 2015. The ubuntu dialogue corpus: A large dataset for research in unstructured multi-turn dialogue systems. In *Proceedings of the SIGDIAL 2015 Conference, The 16th Annual Meeting of the Special Interest Group on Discourse and Dialogue, 2-4 September 2015, Prague, Czech Republic*, pages 285–294. The Association for Computer Linguistics.

Aman Madaan, Niket Tandon, Peter Clark, and Yiming Yang. 2022. Memory-assisted prompt editing to improve GPT-3 after deployment. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, EMNLP 2022, Abu Dhabi, United Arab Emirates, December 7-11, 2022*, pages 2833–2861. Association for Computational Linguistics.

Zhengyang Mao, Wei Ju, Siyu Yi, Yifan Wang, Zhiping Xiao, Qingqing Long, Nan Yin, Xinwang Liu, and Ming Zhang. 2025. Learning knowledge-diverse experts for long-tailed graph classification. *ACM Trans. Knowl. Discov. Data*, 19(2).

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. 2022. Locating and editing factual associations in GPT. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*.

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

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

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

Shikhar Murty, Christopher D. Manning, Scott M. Lundberg, and Marco Túlio Ribeiro. 2022. Fixing model bugs with natural language patches. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, EMNLP 2022, Abu Dhabi, United Arab Emirates, December 7-11, 2022*, pages 11600–11613. Association for Computational Linguistics.

Shiwen Ni, Dingwei Chen, Chengming Li, Xiping Hu, Ruifeng Xu, and Min Yang. 2024. Forgetting before learning: Utilizing parametric arithmetic for knowledge updating in large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2024, Bangkok, Thailand, August 11-16, 2024*, pages 5716–5731. Association for Computational Linguistics.

OpenAI. 2024. Hello GPT-4o.

Fabio Petroni, Tim Rocktäschel, Sebastian Riedel, Patrick S. H. Lewis, Anton Bakhtin, Yuxiang Wu, and Alexander H. Miller. 2019. Language models as knowledge bases? 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 2019, Hong Kong, China, November 3-7, 2019*, pages 2463–2473. Association for Computational Linguistics.

Yuval Pinter and Michael Elhadad. 2023. Emptying the ocean with a spoon: Should we edit models? In *Findings of the Association for Computational Linguistics: EMNLP 2023, Singapore, December 6-10, 2023*, pages 15164–15172. Association for Computational Linguistics.

Carlos Riquelme, Joan Puigcerver, Basil Mustafa, Maxim Neumann, Rodolphe Jenatton, André Susano Pinto, Daniel Keysers, and Neil Houlsby. 2021. Scaling vision with sparse mixture of experts. In *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pages 8583–8595.

Erik F. Tjong Kim Sang and Fien De Meulder. 2003. Introduction to the conll-2003 shared task: Language-independent named entity recognition. In *Proceedings of the Seventh Conference on Natural Language Learning, CoNLL 2003, Held in cooperation with HLT-NAACL 2003, Edmonton, Canada, May 31 - June 1, 2003*, pages 142–147. ACL.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of BERT: smaller, faster, cheaper and lighter. *CoRR*, abs/1910.01108.

Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc V. Le, Geoffrey E. Hinton, and Jeff Dean. 2017. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Y. Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, EMNLP 2013, 18-21 October 2013, Grand Hyatt Seattle, Seattle, Washington, USA, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 1631–1642. ACL.

Lucas Theis and Matthias Bethge. 2015. Generative image modeling using spatial lstms. In *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*, pages 1927–1935.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton-Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurélien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023. Llama 2: Open foundation and fine-tuned chat models. *CoRR*, abs/2307.09288.

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

Renzhi Wang and Piji Li. 2024a. Lemoe: Advanced mixture of experts adaptor for lifelong model editing of large language models. In *Proceedings of the 2024*

Conference on Empirical Methods in Natural Language Processing, EMNLP 2024, Miami, FL, USA, November 12-16, 2024, pages 2551–2575. Association for Computational Linguistics.

Renzhi Wang and Piji Li. 2024b. Semantic are beacons: A semantic perspective for unveiling parameter-efficient fine-tuning in knowledge learning. *arXiv preprint arXiv:2405.18292*.

Haoyun Xu, Runzhe Zhan, Yingpeng Ma, Derek F. Wong, and Lidia S. Chao. 2025. Let's focus on neuron: Neuron-level supervised fine-tuning for large language model. In *Proceedings of the 31st International Conference on Computational Linguistics, COLING 2025, Abu Dhabi, UAE, January 19-24, 2025*, pages 9393–9406. 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. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*, pages 10222–10240. Association for Computational Linguistics.

Yunzhi Yao, Ningyu Zhang, Zekun Xi, Mengru Wang, Ziwen Xu, Shumin Deng, and Huajun Chen. 2024a. Knowledge circuits in pretrained transformers. *CoRR*, abs/2405.17969.

Zihan Yao, Yu He, Tianyu Qi, and Ming Li. 2024b. Scalable model editing via customized expert networks. *CoRR*, abs/2404.02699.

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

Ted Zadouri, Ahmet Üstün, Arash Ahmadian, Beyza Ermis, Acyr Locatelli, and Sara Hooker. 2024. Pushing mixture of experts to the limit: Extremely parameter efficient moe for instruction tuning. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net.

Mengqi Zhang, Xiaotian Ye, Qiang Liu, Pengjie Ren, Shu Wu, and Zhumin Chen. 2024a. Uncovering overfitting in large language model editing. *CoRR*, abs/2410.07819.

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

Ningyu Zhang, Yunzhi Yao, Bozhong Tian, Peng Wang, Shumin Deng, Mengru Wang, Zekun Xi, Shengyu Mao, Jintian Zhang, Yuansheng Ni, Siyuan Cheng, Ziwen Xu, Xin Xu, Jia-Chen Gu, Yong Jiang, Pengjun Xie, Fei Huang, Lei Liang, Zhiqiang Zhang, Xiaowei Zhu, Jun Zhou, and Huajun Chen. 2024c. A comprehensive study of knowledge editing for large language models. *CoRR*, abs/2401.01286.

Taolin Zhang, Qizhou Chen, Dongyang Li, Chengyu Wang, Xiaofeng He, Longtao Huang, Hui Xue', and Jun Huang. 2024d. Dafnet: Dynamic auxiliary fusion for sequential model editing in large language models. In *Findings of the Association for Computational Linguistics, ACL 2024, Bangkok, Thailand and virtual meeting, August 11-16, 2024*, pages 1588–1602. Association for Computational Linguistics.

Ce Zheng, Lei Li, Qingxiu Dong, Yuxuan Fan, Zhiyong Wu, Jingjing Xu, and Baobao Chang. 2023. Can we edit factual knowledge by in-context learning? In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*, pages 4862–4876. Association for Computational Linguistics.

Yanqi Zhou, Tao Lei, Hanxiao Liu, Nan Du, Yanping Huang, Vincent Y. Zhao, Andrew M. Dai, Zhifeng Chen, Quoc V. Le, and James Laudon. 2022. Mixture-of-experts with expert choice routing. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*.

Simiao Zuo, Xiaodong Liu, Jian Jiao, Young Jin Kim, Hany Hassan, Ruofei Zhang, Jianfeng Gao, and Tuo Zhao. 2022. Taming sparsely activated transformer with stochastic experts. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net.

A Related Work

A.1 Model Editing

Model editing is a new and active research area where the goal is to make targeted changes to a pre-trained model's behavior (Zhang et al., 2024c). Given the fast-growing parameter sizes of LLMs, frequently updating LLMs with new knowledge through retraining is more and more expensive. Hence, it is vital to effectively edit the LLMs' knowledge without retraining (Cohen et al., 2024). Previous studies have explored multiple methods for editing the knowledge of LLMs, which can be broadly categorized into two streams based on whether it alters the parameters of the original model (Yao et al., 2023):

A.1.1 Preserve models' parameters:

1) Retrieve augmentation. This approach uses an external knowledge base to store new or correct knowledge. The new knowledge base is seamlessly integrated with the base model, facilitating the retrieval of pertinent information in response to prompts (Murty et al., 2022; Madaan et al., 2022; Li et al., 2023a). For example, IKE (Zheng et al., 2023) employs an in-context learning approach to adjust LLMs outputs using demonstrations sourced from the corpus guided by similarity, thus circumventing the need for gradient calculations.
2) Adding additional parameters. This paradigm introduces additional trainable parameters to represent new knowledge while keeping the original model parameters frozen (Wang et al., 2024; Yao et al., 2024b; Ni et al., 2024; Yu et al., 2024). T-Patcher (Huang et al., 2023) and CaliNET (Dong et al., 2022) inject neurons or patches into the final layer of the Feed-Forward Network (FFN), with T-Patcher assigning one neuron per erroneous prediction, and CaliNET leveraging multiple neurons to capture different knowledge instances. In contrast, GRACE (Hartvigsen et al., 2023) employs a discrete codebook to add and update knowledge entries over time, enabling dynamic adjustments to model outputs.

More recently, MoE-based approaches such as LEMoE and ELDER (Li et al., 2025) have demonstrated promising performance in lifelong model editing tasks. To the best of our knowledge, our work is the first to propose a knowledge editing framework grounded in a Mixture-of-Experts architecture, with an early version released in May 2024.

A.1.2 Modify models' parameters

This approach initially identifies parameters linked to specific knowledge and adjusts them directly (Zhang et al., 2024d; Xu et al., 2025; Das et al., 2024). The Knowledge Neuron (KN) technique (Dai et al., 2022) introduces a method for attributing knowledge to pinpoint the "knowledge neuron" and then updates these neurons accordingly. ROME (Meng et al., 2022) employs causal mediation analysis to pinpoint the area requiring modification. Both KN and ROME are limited to editing one factual association at a time. To address this limitation, MEMIT (Meng et al., 2023) builds upon ROME's framework, allowing for simultaneous editing across multiple cases. Building on MEMIT, PMET (Li et al., 2023b) incorporates attention values to achieve enhanced performance.

A.2 Mixture of Experts

The concept of mixture of experts (MoE), initially introduced in (Yao et al., 2024a; Jordan and Jacobs, 1994), has undergone extensive exploration and advancement as evidenced by subsequent studies (Aljundi et al., 2017; Collobert et al., 2001; Eigen et al., 2014; Theis and Bethge, 2015). The emergence of sparsely-gated MoE (Shazeer et al., 2017), particularly within the integration of transformer-based large language models (Lepikhin et al., 2021), has brought new vitality to this technology. Conventionally, the MoE replaces standard feed-forward neural network layers with sparsely activated expert modules. The MoE has been investigated thoroughly in the era of Large Language Model (Jiang et al., 2024), emerging as an effective way of increasing the model's capacity in parameter size while maintaining computational efficiency akin to its dense counterpart (Jacobs et al., 1991). In the context of MoE, there is a body of work focusing on improving the router (Hazimeh et al., 2021; Lewis et al., 2021; Zhou et al., 2022; Zuo et al., 2022) activating all expert through weighted average (Eigen et al., 2014) to sparsely select a single or k experts(Fedus et al., 2022; Du et al., 2022). Presently, token-level MoE architectures find widespread application in both pre-trained language models and vision-based models (Shazeer et al., 2017; Lepikhin et al., 2021; Du et al., 2022; Riquelme et al., 2021).

Table 3: An editing dataset example from ZsRE and CounterFact.

Dataset	Type	Text
ZsRE	$\mathbf{x}_i^e, \mathbf{y}_i^e$	Which continent is Berkner Island in? **South America**
	$\mathbf{x}_{\mathrm{loc}_i}, \mathbf{y}_{\mathrm{loc}}$	who gets the golden boot if its a tie? **shared**
	$\mathbf{x}_{\mathrm{gen}_i}, \mathbf{y}_i^e$	On which continent is Berkner Island located? **South America**
CounterFact	$\mathbf{x}_i^e, \mathbf{y}_i^e$	The mother tongue of Danielle Darrieux is **English**
	$\mathbf{x}_{\mathrm{loc}_i}, \mathbf{y}_{\mathrm{loc}}$	Where Danielle Darrieux is from, people speak the language of **English**
	$\mathbf{x}_{\mathrm{gen}_i}, \mathbf{y}_i^e$	Michel Rocard is a native speaker of **French**

B Details of Empirical Analysis

To ensure the validity and fairness of the empirical analysis in §2.2, we elaborate here on the experimental setup used for Figure 1.

We adopt two representative model editing methods—MEMIT (Meng et al., 2023) and GRACE (Hartvigsen et al., 2023)—on LLaMA2-7B across two standard datasets: ZsRE and CounterFact. The model layers selected for editing, learning rates, optimization strategies, and batch sizes are consistent with those reported in Appendix C.

GECE Computation. As described in Eq.9, GECE is used to quantify the long-tailness of each editing instance. For a given editting input x^e, origin model output y^o and target output y^e, GECE is computed using the predicted output y^o and the target ground truth y^e as inputs to the METEOR scoring and token-level confidence estimation, following the implementation in (Li et al., 2024a). A higher GECE implies lower prior frequency and more sparse semantic support.

Editing by Frequency Buckets. To study performance across knowledge frequencies (lower-left plot in Figure 1), we first partition the data into six groups based on GECE scores, with 100 instances per group (600 in total). The instances are then shuffled, and batch editing is performed using a standard batch size of 30. After editing, all samples are reassigned to their original GECE buckets, and we report the average reliability, generalization, and locality for each group. This setup allows us to isolate the correlation between editing performance and knowledge frequency.

Head / Tail / Mixed Comparison. In the second experiment (lower-right plot in Figure 1), we construct three disjoint subsets from each dataset: (1) Head: top 30% of edits by lowest GECE score, (2) Tail: bottom 30% (highest GECE), (3) Mixed: random selection from the entire distribution. We ensure equal batch sizes (30 edits) for all groups to eliminate batch-size confounds. Each group's editing performance is averaged over 5 runs.

Why Head is Easier to Edit? While it may seem counterintuitive that high-frequency knowledge is easier to edit, we hypothesize the following explanation based on repeated trials and probing analysis: For head knowledge, the model's decoder typically assigns a very high probability to the original answer y^o, making the editing objective narrowly focused—merely shifting probability mass to the desired target y^e. In contrast, tail knowledge suffers from output uncertainty: the model often assigns comparable scores to multiple incorrect completions, resulting in hallucinations and greater interference during optimization.

This phenomenon is consistent with findings in knowledge localization literature (Meng et al., 2023), where high-frequency knowledge is stored in more identifiable and editable parameter regions, whereas long-tail knowledge tends to be diffuse and harder to localize.

We acknowledge that further theoretical analysis is required to fully explain this behavior and leave this as an open research direction.

C Implementation Details

C.1 Dataset Details

ZsRE (Levy et al., 2017) is a context-free Question Answering (QA) dataset that has been extensively studied in the model editing literature (Meng et al., 2022, 2023; Mitchell et al., 2022b; Hartvigsen et al., 2023). Each record in this dataset includes an editing statement $\mathbf{x}_i^e$ with target answer $\mathbf{y}_i^e$, a paraphrase prompt $\mathbf{x}_{\mathrm{gen}_i}$ and and a locality prompt $\mathbf{x}_{\mathrm{loc}}$. We adopt the same train/test split as (Mitchell et al., 2022a), consisting of 163,196 training examples and 19,086 test examples. Notably, MEND is the only method that requires fitting a hyper network on the training set; other methods discard the training set and directly perform edits and evaluations on the test set. For our experiments, we randomly sampled 1k and 3k records from the test set to form the edit sets. Contrarily, CounterFact (Meng et al., 2022) presents a formidable challenge by focusing on counterfactual information, often yielding lower prediction accuracy compared to factual queries. This dataset constructs the out-of-scope instances by substituting the primary entity with a comparable descriptor while maintaining the same predicate (Yao et al., 2023).

An illustrative excerpt from the ZsRE dataset is presented in Table 3. Each entry within ZsRE comprises the subject string, the factual prompt for assessing reliability, the rephrase prompt for evaluating generality, and the locality prompt for assessing contextual relevance. It's important to note that the objective of the locality prompt isn't to predict the true answer, but rather to mirror the predictions made by the base model. Likewise, the fact, rephrase, and locality prompts within each entry of the COUNTERFACT dataset correspond to the evaluation of their respective metrics (Table 3). In our experiment, we use the original output of the base model (GPT2-XL and LLaMA2-7B) as the ground truth for evaluating locality metrics.

C.2 Implementation of Baselines

Following previous research (Li et al., 2024b), the implementation details of baselines are as follow:

Fine-tuning We implemented three fine-tuning methods. For **FT-L**, we followed the procedures outlined in (Meng et al., 2023, 2022), fine-tuning the mlp_{proj} parameter from layer 0 for GPT-2 XL and from layer 16 for LLaMA2-7B, as these configurations were found to yield optimal performance. **FT-M** is a slight variation of FT-L, differing primarily in its loss computation procedure for parameter optimization[1]. For both models, we performed 25 optimization steps using the AdamW optimizer (Kingma and Ba, 2015), with a learning rate of $5e^{-4}$. All other parameters for both models were kept at their default settings. **LoRA** (Hu et al., 2022) proposed a parameter-efficient fine-tuning method that decomposes the update gradient matrix into two small rank-n matrices, which reduces the required memory for LLM training to a large extent. In all experiments, we set the learning rate and the rank number to $1e^{-3}$ and 8, respectively. The α was chosen to be 32, and the dropout rate was 0.1. The number of update steps is 30 for GPT2-XL and 50 for LLaMA2-7B.

MEND MEND (Mitchell et al., 2022a) performs model editing by manipulating the gradients of language models. It trains a meta-network that utilizes a rank-1 decomposition of the model gradients to predict a new rank-1 update for the corresponding model weights. In this study, we train two meta-networks using the respective training splits from the ZsRE (Levy et al., 2017) and COUNTERFACT datasets for GPT-2 XL, adhering to the default hyperparameter settings. Due to the substantial computational resources required to train the meta-network for LLaMA2-7B, we did not conduct training for it.

SERAC SERAC (Mitchell et al., 2022b) developed a memory-augmented editing method that utilizes an external cache to store explicit editing instances. This method includes a scope classifier to determine whether an input sample falls within the editing scope and employs a small counterfactual model to edit in-scope cases. We independently train two models for GPT2-XL and LLaMA2-7B on their respective datasets. Consistent with the original methodology, we utilize distilbert-base-cased (Sanh et al., 2019) as the scope classifier across all models. All hyper-parameters remain at their default settings.

MEMIT The MEMIT (Meng et al., 2023) regards the feed-forward layer of a transformer as a linear associative memory. It employs a minimum square error optimization technique to incorporate new key-value associations into layer weights. Following the methodology outlined in the original paper, we adjust the layers within the identified critical path and determine the optimal value for the balance factor λ, as per the findings in the original research (Layer []). All other parameters for both models are configured in accordance with the specifications provided in (Meng et al., 2023, 2022).

GRACE GRACE (Hartvigsen et al., 2023) introduces a novel editing technique aimed at conserving the initial model parameters while integrating a dynamic codebook. This codebook evolves through incremental addition, splitting, and expansion of keys, facilitating the storage of pertinent modifications over time. We adhere to the meticulously crafted parameters outlined in the original study, configuring the optimization of the learning rate to a value of 1. The iterative process for optimizing these values spans 100 cycles, with an initial ϵ value set at 1.

COMEBA-HK The experimental results of COMEBA-HK on GPT2-XL are derived from their research paper (Li et al., 2024b). Due to the absence of experimental results on LLaMA2-7B in COMEBA-HK's paper and the lack of code disclosure, certain outcomes in our study do not include this approach.

[1]https://github.com/zjunlp/EasyEdit/blob/main/hparams/FT/gpt2-xl.yaml

Table 4: Results for all editing tasks across various model types and sizes. `ZsRE`.

Model	Single Editing				Batch Editing				Sequential Editing				Sequential Batch Editing			
	Rel.↑	Gen.↑	Loc.↑	Avg.↑	Rel.↑	Gen.↑	Loc.↑	Avg.↑	Rel.↑	Gen.↑	Loc.↑	Avg.↑	Rel.↑	Gen.↑	Loc.↑	Avg.↑
BLOOMZ-1.1B	100.00	82.26	99.10	93.79	90.74	84.02	91.61	88.79	52.80	34.32	96.37	61.16	71.14	32.96	88.45	64.18
BLOOMZ-1.7B	100.00	84.57	100.00	94.86	91.54	84.21	92.30	89.35	51.99	33.55	96.96	60.83	73.59	35.25	90.15	66.33
BLOOMZ-3B	100.00	86.68	100.00	95.56	92.05	84.92	93.16	90.04	49.54	31.90	96.51	59.31	70.65	34.30	96.73	67.23
LLaMA2-7B	100.00	92.03	100.00	97.34	92.43	85.03	94.12	90.53	56.72	31.02	97.54	61.76	74.24	36.64	97.45	69.44
LLaMA2-13B	100.00	91.06	100.00	97.02	92.17	84.29	93.42	89.96	56.78	30.55	97.38	61.57	72.45	36.15	97.36	68.65
LLaMA2-70B	100.00	89.77	100.00	96.59	91.58	83.90	92.44	89.31	50.74	30.18	95.98	58.96	71.83	34.01	95.51	67.11

C.3 Training Details of MEMoE

We select GPT2-XL and LLaMA2-7B as the base models. Modifications are applied to layer 16 of LLaMA2-7B and layer 18 of GPT2-XL (consistent with the findings of ROME (Meng et al., 2022)). The number of experts is set to 5, and $top_k = 1$ to achieve the best experimental results, which are determined based on the dataset characteristics and available computational resources. Regarding other hyperparameter choices, $\lambda = 1$ in Equation 12, $\tau = 0.6, \gamma_1 = 1$ in Equation 17, $\gamma_2 = 0.5$ in Equation 18, and $\gamma_3 = 1.5$ in Equation 19. We use the AdamW optimizer (Kingma and Ba, 2015), with $\beta_1 = 0.9$ and $\beta_2 = 0.95$, and a learning rate of $2e^{-4}$, employing a cosine learning rate scheduler. Additionally, we apply a linear warm-up to the learning rate scheduler for the first 10% of the training steps. The experiment is deployed on NVIDIA V100 GPU.

D More Results and Analyses

In this section, we present additional experiments and discussions about MEMoE. We test the performance of MEMoE across all four knowledge editing tasks on a range of model serious with different parameter sizes (§D.1). We then provide a computational analysis of MEMoE (§D.2). As a supplement to the ablation experiments in the main text (§5.2), we further investigate the impact of different model settings on performance (§D.3). Additionally, we evaluate the consistency of the proposed router in §D.4. Finally, we compare the performance differences between batch editing and sequential editing in larger data scenarios (§D.5), followed by a more detailed case study (§D.6).

D.1 Experiments across Model Types and Sizes

We apply the MEMoE framework to a diverse set of models, spanning both smaller and larger architectures, including BLOOMZ-1.1B/1.7B/3B, LLaMA2-7B/13B/70B, and evaluate its effectiveness across all four editing tasks outlined in §2.1. The experimental setup follows §4.1.

The results are presented in Figure 4. MEMoE consistently achieves impressive scores of around 90 in both single-editing and batch-editing tasks, indicating its robust performance across a wide range of settings. In the context of sequential editing scenario, while MEMoE demonstrates substantial improvement over the baseline, there remains significant potential for further optimization. Particularly sequential editing tasks present a notable challenge, as they involve modifying one piece of knowledge at a time, leading to a higher number of editing steps. This increase in steps exacerbates the model's tendency toward catastrophic forgetting, where earlier modifications are progressively overwritten by new edits. In contrast, sequential batch editing, where a batch of data is modified in a single step, significantly reduces the number of editing iterations, resulting in a marked improvement in performance. This highlights that, while MEMoE is effective, additional refinements are needed to fully exploit its potential in sequential editing, especially in addressing the issue of catastrophic forgetting.

Furthermore, we observe that MEMoE performs best on the 7B model, with performance declining as the model size increases or decreases. This trend can be attributed to the current expert configuration, such as the use of 5 experts, which was empirically selected as optimal for models around the 7B scale, as detailed in the main text. However, this configuration may not be ideal for larger models which contain significantly more parameters. These larger models may benefit from a greater number of experts to better align with their increased capacity. As discussed in §D.3, exploring alternative expert configurations tailored to larger models could lead to significant improvements in performance. This underscores the importance of selecting appropriate MEMoE structural settings based on the specific characteristics of the model, ensuring that the expert-based approach scales effectively with increasing model complexity.

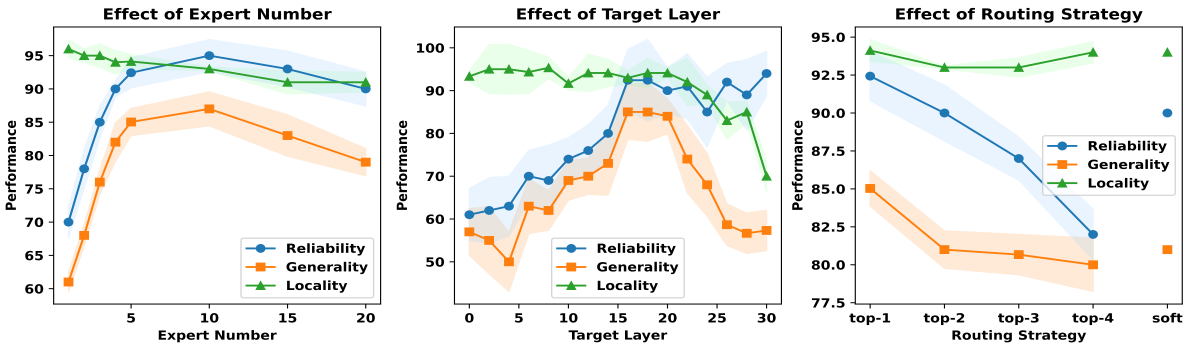

Figure 4: *Left:* Performance across different numbers of experts. *Middle:* Performance across different target model layers. *Right:* Effectiveness of activating experts. ZsRE. LLaMA2-7B.

Table 5: The efficiency of editing method. Time indicates the wall clock time required for conducting 30 edits, while VRAM represents the graphics memory usage. ZsRE. LLaMA2-7B.

Method	VRAM	Training Time	Inference Time
FT-L	26G	28min	0.51 s/edit
LoRA	23G	4min	0.47 s/edit
MEMIT	33G	-	16.93 s/edit
MEND	46G	4h	0.64 s/edit
SERAC	42G	15h	0.85 s/edit
GRACE	28G	-	14.27 s/edit
MEMoE	24G	5min	0.72 s/edit

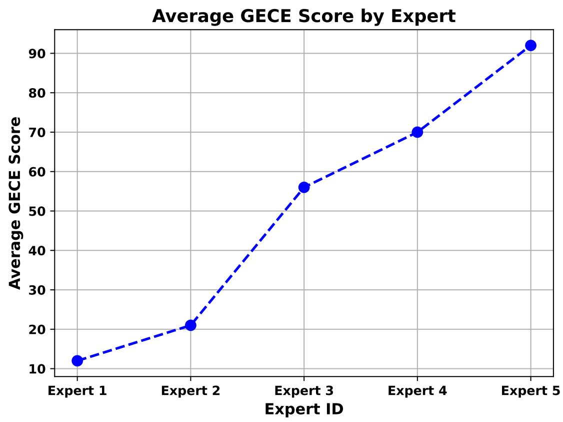

Figure 5: Performance comparison of model editing on different frequency knowledge.

D.2 Computational Analysis

The efficiency of model editing methods is a critical consideration for both research scalability and practical deployment. As illustrated in Table 5, MEMoE distinguishes itself by operating within a single 32GB V100 GPU, a feat enabled by its parameter-efficient design and streamlined computational workflow. This contrasts sharply with methods like MEND and SERAC, which demand 46GB and 42GB of VRAM respectively, largely due to their reliance on auxiliary networks or memory-intensive gradient computations. Such requirements pose barriers to accessibility, particularly for resource-constrained researchers. Three axes of efficiency merit discussion: (1) VRAM Utilization: MEMoE's 29GB footprint reflects its hybrid architecture, which synergizes lightweight adapter modules with sparse activation mechanisms. This contrasts with LoRA (23GB), which achieves lower memory use at the cost of expressivity through low-rank approximations, and MEMIT (33GB), which incurs overhead from mass-editing key-value associations in transformer layers. (2) Training Dynamics: MEMoE completes training in 5 minutes—orders of magnitude faster than MEND (4h) and SERAC (15h). This acceleration stems from its localized editing paradigm, which avoids global parameter updates through dynamic router networks. The absence of training phases in MEMIT and GRACE, while superficially advantageous, limits their applicability to scenarios requiring iterative model refinement. (3) Inference Latency: At 0.72s/edit, MEMoE closely approximates the baseline model's latency (0.47–0.85s/edit for parameter-preserving methods), outperforming approaches like MEMIT (16.93s/edit) that require traversal of edit-specific computational paths. This efficiency arises from MEMoE's non-serialized architecture, where router networks operate in parallel with frozen base model components. The proposed state caching mechanism could further reduce inference costs by amortizing routing computations across multiple edits.

D.3 Additional Ablation Study

Figure 4 presents an analysis of MEMOE's performance across different numbers of experts, target layers, and routing strategies. This section serves as a supplementary part of the ablation study discussed in §5.2, offering valuable insights into the impact of these factors on the model editing.

The left plot illustrates the impact of varying the number of experts. As the number of experts increases, both the accuracy and generalization score improve. The performance reaches near its peak when the number of experts is between 5 and 10. However, the locality of the model gradually decreases as more experts are introduced. This trade-off between performance and locality suggests that while increasing the number of experts enhances the model's capacity to edit and generalize, it comes at the cost of reduced locality, which is crucial for certain tasks. Considering both the editing performance and computational efficiency, we select 5 experts for the main experiments.

The middle plot examines the effect of selecting different target layers for editing. It reveals that both reliability and generalization reach their maximum at layers 16 to 20. Interestingly, generalization remains stable until layer 20, after which it starts to decline. This observation suggests that the bypass MoE structure could effectively maintains the influence of model editing within a specific range of layers. However, as approaching the output layers, the impact of model editing becomes more pronounced, beginning to significantly affect the locality score, potentially due to the accumulation of expert inputs in the final computation stages (Ju et al., 2024).

The right plot compares the performance of different routing strategies: soft merging (Zadouri et al., 2024) and discrete top-1, top-2, and top-3 routing schemes. The results show that top-1 routing consistently outperforms the others, providing the best overall performance. As the value of k increases, the number of experts involved in the computation rises, but the generalization performance declines, indicating that the broader utilization of experts may lead to a loss in coherence across the network. In contrast, the soft merging strategy, while slightly less effective than top-1 routing, offers a notable advantage over the other discrete strategies, suggesting that dynamic routing methods may have certain benefits over hard routing (Zadouri et al., 2024). Nevertheless, for model editing tasks, top-1 routing proves to be more effective. Additionally, discrete top-1 routing has an advantage in computational efficiency by requiring only one experts to be activated during inference.

D.4 Routing Consistency Analysis

In §3.1, we propose a data division strategy aimed at enabling different experts to specialize in knowledge of varying frequencies. In this section, we conduct an evaluation of the experts' specialization by measuring the average frequency of the knowledge processed by each expert. Specifically, we compute the average GECE values (Equation 9) corresponding to the knowledge handled by each expert during the inference phase, as illustrated in the figure. As observed, and in alignment with our design principles, the first expert exhibits a significantly lower GECE value compared to the others, indicating a specialization in processing high-frequency (or "head") knowledge. The subsequent experts, on the other hand, progressively specialize in handling long-tail knowledge, further validating the efficacy of our approach in promoting specialized expertise across experts.

Table 6: Comparison of batch editing and sequential batch editing on 1K edits. `ZsRE`. `LLaMA2-7B`.

Task Settings	Number	Rel.↑	Gen.↑	Loc.↑	Avg.↑
Batch	10	100	90.12	100.00	96.71
	100	91.03	81.59	93.31	88.64
	1000	80.02	56.92	89.53	75.49
Sequential Batch	1000	74.24	36.64	90.45	67.11

D.5 Batch Editing vs Sequential Editing

In §4.2, Table 4 clearly demonstrates MEMoE's superior performance in batch editing tasks compared to sequential editing. To provide a clearer comparison of its performance differences between batch editing and sequential editing, we progressively increased the batch size, with results presented in Table 6. MEMoE shows significant performance improvements when the batch size reaches 1,000, which is equivalent to the total batch size in sequential editing. Both reliability and locality remain above 80, while the generality score increases by 20.28 points. This improvement underscores the framework's advantage in batch editing, where simultaneous processing of multiple edits is more effectively managed. In contrast, the sequential batch editing method exhibits a notable performance decline, particularly in reliability and generality. Our analysis of these bad cases in sequential batch editing reveals that the underlying cause of this drop can be attributed to catastrophic forgetting: edits complete earlier are more prone to errors.

D.6 More Case Study

In Table 7, we present bad cases of using MEMoE to edit `LLaMA2-7B` on ZsRE dataset and mitigating

Table 7: Failure cases of MEMoE. ✓✗represents errors in part of the tokens, ✗represents complete output errors (i.e.,factual failures), and ✓indicates the expected exact match. Italics correspond to generality prompt. `ZsRE`. `LLaMA2-7B`.

	Prompt	Edit Target	Post-Edit Output
i	What level is Javan surili's iucn conservation status?	critically threatened	near threatened ✓✗
	What is Javan surilis ucn conservation status?	critically threatened	threatened ✓✗
	The point in time of Air France Flight 447 was when?	12 July 1944	12 July 1964 ✓✗
	When did Air France Flight 447 occur?	12 July 1944	12 July 1964 ✓✗
ii	Which war was William Babcock Hazen in?	World War II	US Civil War ✗
	What war did William Babcock Hazen go to?	World War II	Spanish Civil War ✗
	When was the inception of Parcelforce?	1961	1963 ✗
	When was Parcelforce formed?	1961	1931 ✗
iii	What team is Nicolas Raffault associated with?	Arizona Coyotes	Aqua ✗
	Which team is Nicolas Raffault associated with?	Arizona Coyotes	Arizona Coyotes ✓
	What sports team was Petteri Nummelin a member of?	Columbus Blue Bombers	Cleveland Monsters ✗
	In which sports team was Petteri Nummelin a member?	Columbus Blue Bombers	Columbus Blue Bombers ✓
iv	What level is Javan surili's iucn conservation status?	critically threatened	nearlly threatened ✓
	What state is Qaleh Lan in?	critically threatened	a ✗
	When did Battle of the Java Sea occur?	27 February 1942	27 February 1942 ✓
	When did the battle on the Java Sea begin?	27 February 1942	1942 ✓✗

these failures is critical for future work in model editing. We observe that:

i) errors occur only in part of the tokens, and these errors constitute a large proportion of the bad cases, indicating that the edits have not been sufficiently fitted. We wonder whether employing different learning rates and epochs for each batch in lifelong editing could alleviate this issue through more refined training.

ii) displays cases where the entire output is incorrect. These types of errors are the most common occurrences.

iv) presents cases of generalization failure. For example in prompt of last line, where the model answered "1942" which is partially correct, but did not fully follow the ground truth, indicating significant room for improvement in the accuracy of generalized edits.

Meanwhile, in iii) we surprisingly find that even when MEMoE errs on the edit prompt, it can correctly answer its paraphrase prompt. Upon closely examining these anomalous cases, we found that they predominantly pertain to question-answering scenarios within sports contexts, such as inquiries about a person's team affiliation. We hypothesize that this phenomenon may stem from the relatively limited number of teams in sports contexts, combined with the higher number of athletes and the occurrence of name duplication. Consequently, the model may accidentally provide correct answers to some of these questions.

In summary, MEMoE can handle contextual information correctly in some cases but falls short in specific editing instructions, suggesting that optimizing editing instructions (modifying the editing context) may be a direction for improvement.