MindBridge: Scalable and Cross-Model Knowledge Editing via Memory-Augmented Modality

Shuaike Li, Kai Zhang*, Qi Liu, Enhong Chen
State Key Laboratory of Cognitive Intelligence,
University of Science and Technology of China, Hefei, China
lishuaike767@gmail.com
{kkzhang08, qiliuql, cheneh}@ustc.edu.cn

Abstract

Knowledge editing is a technique for efficiently and accurately updating the knowledge of large language models (LLMs) to alleviate obsolescence and correct errors. However, most existing methods overfit to specific models, causing edited knowledge to be discarded during each LLM update and requiring frequent re-editing, which is particularly burdensome in today's rapidly evolving open-source community. To address this issue, we propose the problem of cross-model knowledge editing and introduce **MindBridge**, a scalable solution inspired by the low coupling between modality processing and LLMs in multi-modal models. MindBridge introduces the novel concept of **memory modality**, which encodes edited knowledge as an independent modality. It first performs LLM-agnostic pre-training of the memory modality and then integrates it with various LLMs. Extensive experiments on multiple LLMs and popular knowledge editing datasets demonstrate that MindBridge achieves superior performance even in editing tens of thousands of knowledge entries and can flexibly adapt to different LLMs. Our code is available at here.

1 Introduction

Large language models (LLMs) have revolutionized natural language processing, demonstrating remarkable abilities in understanding and generation (Achiam et al., 2023; Touvron et al., 2023; Brown et al., 2020). These models leverage the knowledge acquired during pre-training to answer user queries. However, since the knowledge embedded in LLMs is stored in static parameters, their internal knowledge needs to be updated to keep pace with the ever-changing world and avoid obsolescence. Additionally, for personalized user information or domain-specific knowledge, customizing LLM output also requires updating the model's knowledge. Traditional methods, such as fine-tuning, continual learning, or retraining, are computationally expensive and may inevitably degrade the model's general capabilities(Kalajdzievski, 2024; Wang et al., 2023). Fortunately, recent advancements in knowledge editing offer a promising solution, enabling efficient and precise modification of model knowledge at a low computational cost.

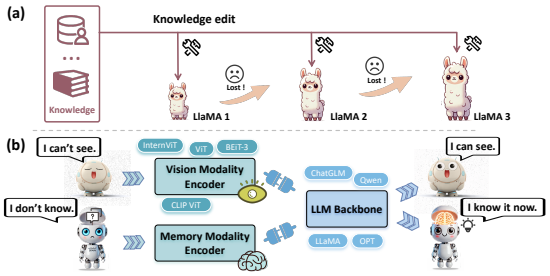


Figure 1: (a) The cross-model knowledge editing problem. Current knowledge editing methods discard previously edited knowledge after every LLM update (e.g., when the base model is updated alongside LLaMA), requiring frequent re-editing, which is labor-intensive. ***This motivates us to explore whether edited knowledge can transcend individual models, i.e., achieve cross-model knowledge editing***. (b) Analogizing memory modality to visual modality. Different visual modality encoders exhibit low coupling with various LLM backbones, and after efficient modality alignment, they enable LLMs to see. ***Inspired by this, we propose memory modality, which decouples knowledge from a single model, allowing knowledge editing through modality bridging with different LLMs.***

Existing knowledge editing methods generally fall into two categories (Yao et al., 2023). The first one preserves LLM parameters, such as memory-based methods (Mitchell et al., 2022a; Madaan et al., 2022a) or methods that add extra parameters to the model (Huang et al., 2023; Dong et al., 2022). The second category modifies the parameters of the edited model, such as meta-learning-based methods (Mitchell et al., 2021; Tan et al., 2023) or locate-and-edit methods (Meng et al., 2022a; Fang et al., 2024). While these methods have made significant

*Corresponding author

Findings of the Association for Computational Linguistics: ACL 2025, pages 11999–12013

progress, most still face the issue of overfitting to a single LLM. If the knowledge to be edited is domain- or user-specific, re-editing this knowledge becomes necessary whenever the LLM is updated, which is both tedious and time-consuming, especially given the fast pace of updates in the open-source community. This motivates us to explore whether the edited knowledge can be loosely coupled with the target model and is no longer limited to a single LLM. We call this problem *cross-model knowledge editing*, as shown in Figure 1(a).

To address this issue, inspired by the work on multimodal large language models, we introduce the **memory modality**, a novel concept that encodes edited knowledge as a standalone modality to partially decouple a model's knowledge from the LLM itself. As illustrated in Figure 1(b), the analogy to visual modality offers a clearer understanding of memory modality. The current mainstream paradigm (Liu et al., 2024) involves concatenating pre-trained visual modality encoders (e.g., ViT (Dosovitskiy, 2020), CLIP ViT (Radford et al., 2021)) with various LLM backbones (e.g., LLaMA, Vicuna (Zheng et al., 2023b)) to enable LLMs with visual perception. The low-coupling characteristic between visual modality processing and LLMs allows for independent updates of both components (Chen et al., 2024; Jain et al., 2024). This low-coupling property is precisely what is needed to solve the problem of cross-model knowledge editing. By encoding editable knowledge as a separate memory modality, pre-training a memory modality encoder to handle this knowledge, and subsequently concatenating it with multiple LLMs, we achieve efficient cross-model knowledge editing.

Based on the idea of memory modality, we propose **MindBridge**, a two-stage solution for cross-model knowledge editing. The first stage is *memory modality pre-training*, where we introduce three training objectives: memory injection, memory association, and memory existence. These objectives are designed to train the memory modality encoder so that it can acquire relevant memories, perform memory association, and determine whether certain memories exist. The second stage is *memory modality bridging*, where we fine-tune a simple projector to achieve efficient cross-modal alignment, enabling the output of the memory modality to be understood and utilized by LLMs. This approach allows for efficient large-scale knowledge updates, with minimal modifications required when the LLMs are updated.

To test the effectiveness of MindBridge, we conduct extensive experiments on widely used knowledge editing datasets, ZsRE and Counterfact, using multiple LLMs, including GPT2-XL, GPT-J, and LLaMA3. Leveraging the low coupling characteristic between the memory modality and LLMs, MindBridge achieves efficient and scalable knowledge editing across different models, delivering excellent editing performance even in scenarios involving tens of thousands of knowledge edits. Our contributions can be summarized as follows:

- We introduce the novel problem of *cross-model knowledge editing*, addressing the critical challenge of discarding previously edited knowledge and repeatedly re-editing caused by the rapid iteration of LLMs.
- We propose MindBridge, a scalable and effective solution that leverages the innovative concept of a memory modality to achieve cross-model knowledge editing.
- Extensive experiments on knowledge editing datasets and multiple LLMs validate the effectiveness and scalability of MindBridge for cross-model knowledge editing, even when handling tens of thousands of edits.

2 Related Work

2.1 Knowledge Editing

Knowledge editing, aimed at inserting new knowledge or modifying existing knowledge in LLMs to alter their behavior, falls into two main paradigms (Yao et al., 2023). *Parameter-modifying editing* directly adjusts model parameters. This includes locate-then-edit strategies like ROME (Meng et al., 2022a), MEMIT (Meng et al., 2022b), and Alphaedit (Fang et al., 2024), which pinpoint relevant knowledge before fine-tuning parameters. Meta-learning approaches, such as KE (De Cao et al., 2021), MEND (Mitchell et al., 2021), and MALMEN (Tan et al., 2023), also belong to this paradigm, training hypernetworks to generate parameter updates.

In contrast, *parameter-preserving editing* maintains original model weights and uses external components for knowledge storage. T-Patcher (Huang et al., 2023) adds neurons to the last feed-forward layer to adjust output. SERAC (Mitchell et al., 2022b) trains a counterfactual model using a classifier to determine response relevance. Database-retrieval methods, including GRACE (Hartvigsen et al., 2024) (hidden vector database) and MELO

(LoRA block database), modify LLM computation based on retrieved knowledge. MemPrompt (Madaan et al., 2022b) and IKE (Zheng et al., 2023a) utilize retrieved demonstrations as prompts, leveraging in-context learning.

While effective for targeted edits, existing knowledge editing methods are often overfit to specific LLMs and struggle with cross-model, large-scale knowledge editing. After each iteration of the LLM, the previously edited knowledge is lost, leading to the need for frequent re-editing.

2.2 MultiModal Large Language Models

Recent advancements in multimodal large language models (MM-LMMs) have enabled LLMs to process diverse modality inputs like images, video, and audio (Li et al., 2023a,b; Chu et al., 2023). Due to the high cost of training MM-LLMs from scratch, a more efficient strategy involves integrating pre-trained unimodal foundation models with the LLM backbone. This approach typically follows a two-stage pipeline: Multimodal Pre-Training, which aligns modality encoder features with the LLM, and Multimodal Instruction-Tuning, which ensures instruction following and zero-shot generalization (Zhang et al., 2024a; Liu et al., 2024).

This decoupling of the LLM backbone from modality encoders enables the reuse or iterative updates of modality encoders without affecting the LLM (Chen et al., 2024; Oquab et al., 2023), offering insights for cross-model knowledge editing. Beyond traditional modalities, we propose the novel concept of memory modality to achieve similar decoupling, partially separating model knowledge and reasoning. This enables efficient knowledge editing, allowing for the retention or independent updating of previously edited knowledge even when the LLM backbone is updated.

3 Problem Formulation

The goal of knowledge editing is to modify the knowledge stored in a model. In this paper, we focus specifically on editing memories composed of factual knowledge. More concretely, for a triplet (s, r, o) consisting of a subject s, relation r, and object o (e.g., s = United States, r = President, o = Biden), we aim to insert a new triplet (s, r, o^*) (e.g., s = United States, r = President, o^* = Trump) into the LLM to replace the previous knowledge, i.e., $(s, r, o) \rightarrow (s, r, o^*)$, where these two triplets share the same subject and relation. Specifically, o^* can also represent knowledge that does not originally exist in the LLM, i.e., $(s, r, \emptyset) \rightarrow (s, r, o^*)$.

Given a set of knowledge to be edited $D_{\text{edit}} = \{(s_i, r_i, o_i^*) \mid i = 1, 2, \ldots, n\}$, a knowledge editing operation KE, and a model to be edited F, the goal of knowledge editing is to generate a new model F^* through the knowledge editing operation KE. This can be formulated as follows:

$$\begin{aligned} F^* &= KE(F, D_{edit}), \\ \text{s.t. } F^*(s,r) &= \begin{cases} o^*, & \text{if } (s,r,o^*) \in \mathcal{I}(D_{edit}), \\ F(s,r), & \text{if } (s,r,o^*) \in \mathcal{O}(D_{edit}). \end{cases} \end{aligned} \tag{1}$$

Here, $\mathcal{I}(D_{edit})$ denotes the knowledge set that requires editing and its neighborhood, such as paraphrasing and rewriting, with $D_{edit} \subseteq \mathcal{I}(D_{edit})$. Meanwhile, $\mathcal{O}(D_{edit})$ denotes the set of knowledge items unrelated to the edited knowledge. For simplicity, we will refer to these sets as $\mathcal{I}$ and $\mathcal{O}$ moving forward. In some studies, $\mathcal{I}$ is also referred to as in-scope examples, while $\mathcal{O}$ is termed out-of-scope examples (Mitchell et al., 2022b).

Equation 1 specifies that the edited model F^* should correctly predict the edited knowledge and its neighborhood. For inputs unrelated to the edited knowledge, F^* should maintain consistent predictions with the original model F. This ensures that knowledge editing updates target knowledge while minimizing interference with unrelated knowledge.

4 Method

As illustrated in Fig. 2, MindBridge is meticulously structured into two primary stages. The first stage, **Memory Modality Pre-training** (detailed in Section 4.1), is specifically designed to obtain the memory modality encoder, E_m. The second stage, **Memory Modality Bridging** (detailed in Section 4.2), is dedicated to training a projector to construct the memory-to-language bridging module, P_m. Ultimately, the edited model, F^*, can be mathematically represented by Eq. 2. In MindBridge, knowledge is encoded into memory modality features, denoted as x_{memory}, through the synergistic utilization of the memory modality encoder E_m and the bridging module P_m. These features are subsequently concatenated with the textual input (s, r) as soft prompts and provided to the LLM, F, for prediction.

$$\begin{aligned} x_{memory} &= P_m(E_m(s, r)), \\ F^*(s, r) &= F(x_{memory} \oplus (s, r)). \end{aligned} \tag{2}$$

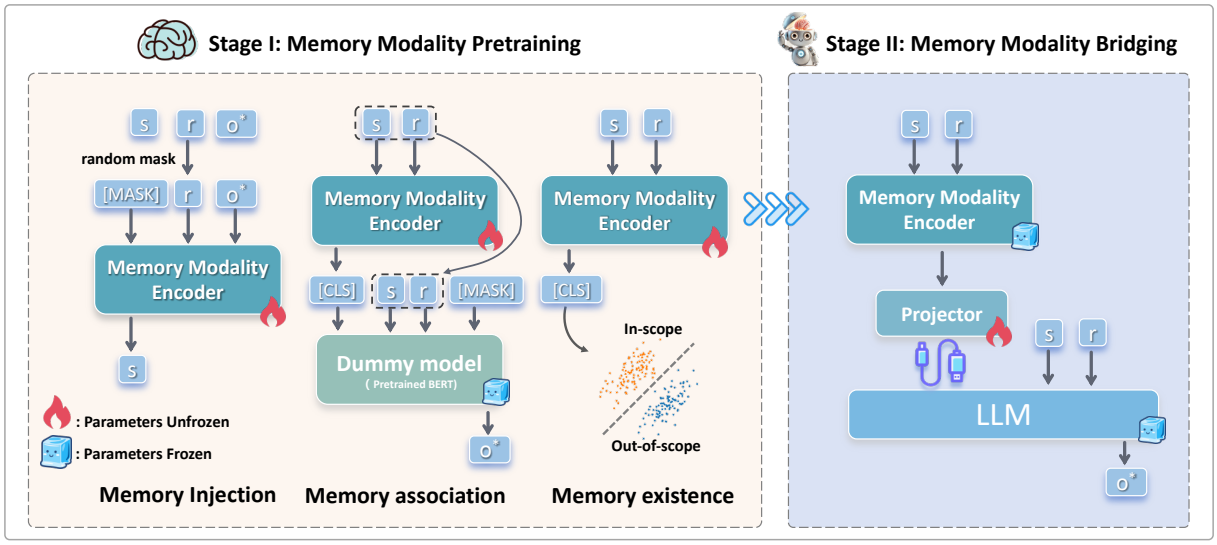


Figure 2: Overview of MindBridge. Given a massive collection of fact knowledge subject-relation-object triplets (s, r, o^*) intended for editing, we first perform ***stage I: Memory Modality Pre-training***. This phase utilizes three training objectives – memory injection, memory association, and memory existence – to develop a memory modality encoder capable of retaining relevant memories, making associations and determining whether specific memories exist. In ***stage II: Memory Modality Bridging***, we then train a projector to bridge the memory modality encoder with LLMs, allowing these models to obtain and effectively leverage the relevant memories.

To fully appreciate the design of MindBridge, we first review the general architecture of current mainstream multimodal large language models (MM-LLMs) (Liu et al., 2024; Bai et al., 2023). These models typically consist of three shared design elements: a modality encoder E, a modality-to-language alignment module P, and a frozen LLM F. The final constructed MM-LLMs resemble the edited model presented in Eq.2. During inference, MM-LLMs rely on the modality encoder E and the bridging module P to extract modality features and transform them into a text feature space understandable by LLMs. Drawing inspiration from this architecture's successful design, MindBridge adopts a similar architecture to implement the training and bridging of memory modalities. Consequently, the missing components in MindBridge are the modality encoder and the modality-to-language alignment module, which are addressed in its two stages respectively.

4.1 Memory Modality Pre-training

In multimodal large language models, modality encoders are typically pre-trained in advance to extract features from modality-specific inputs, such as the commonly used vision modality encoders ViT (Dosovitskiy, 2020) and CLIP-ViT (Radford et al., 2021). Analogously, for the memory modality, pre-training a dedicated encoder, E_m, is indispensable. From the perspective of the knowledge editing workflow, this encoder should be designed to achieve the following three key objectives:

- *Objective 1*: Possess relevant memories for target knowledge, i.e., store tuples of $(s, r, o^*) \in D_{edit}$.
- *Objective 2*: Extract relevant memories based on provided context, i.e., retrieve memories associated with (o^*) from the given (s, r), where $(s, r, o^*) \in D_{edit}$.
- *Objective 3*: Distinguish whether relevant memories exist, i.e., to differentiate between the set $\mathcal{I}$ and the set $\mathcal{O}$.

We initialize E_m using a pre-trained BERT (Kenton and Toutanova, 2019) and use the [CLS] hidden state of its output as the extracted memory modality features, i.e., $E_m(s, r) = \text{BERT}_{[\text{CLS}]}(s, r)$. To achieve Objective 1, we adopt the Masked Language Model (MLM) training objective to inject the knowledge $(s, r, o^*) \in D_{edit}$ into E_m. Specifically, we randomly mask one element of the (s, r, o^*) triplet by replacing it with the [MASK] token and then have E_m reconstruct the masked element. We refer to this loss function as the memory-injection loss L_{inject}, which can be expressed as follows:

$$L_{\text{inject}} = -E_{(s,r,o^*)\sim D_{\text{edit}}} \Big[\sum_{x_i \in \{s,r,o^*\}} \log \mathbb{P}_{E_m}(x_i \mid [\text{MASK}], (s,r,o^*) \backslash x_i) \Big].$$

To achieve Objective 2, we need to enhance the memory modality features extracted from the [CLS] token so that they contain representations related to o^* based on (s, r). This process mirrors human associative memory, where observing a segment of text triggers recall of associated information. To adapt the memory modality encoder for this task, we feed the [CLS] representation output by $E_m(s, r)$ along with $(s, r, [\text{MASK}])$ into another dummy model M. The model M must predict o^* based on the associations made by the memory modality given (s, r). In this case, M is also a BERT model, sharing the same pre-trained initialization parameters with E_m, but its parameters remain frozen throughout the training process. We refer to this loss function as the memory-association loss $L_{\text{associate}}$, which can be formulated as follows:

$$L_{\text{associate}} = -E_{(s,r,o^*)\sim D_{\text{edit}}} \left[\log \mathbb{P}_M(o^* \mid E_m(s,r) \oplus (s,r,[\text{MASK}]))\right].$$

To achieve Objective 3, which involves determining whether relevant memories exist, we need to ensure that $E_m(s, r)$ produces distinct representations for the sets $\mathcal{I}$ and $\mathcal{O}$. We adopt a simple binary classification task to accomplish this goal. Specifically, we feed $E_m(s, r)$, where $(s, r, o^*) \in \mathcal{I} \cup \mathcal{O}$, into a classification head H composed of two linear layers to classify whether the input belongs to $\mathcal{I}$ or $\mathcal{O}$. This approach ensures that the internal representations of $\mathcal{I}$ and $\mathcal{O}$ are clearly differentiated and reside in distinct vector spaces. We refer to this loss function as the memory-existence loss L_{exist}, which can be expressed as follows:

$$\hat{y} = H(E_m(s,r)),$$
$$L_{exist} = -E_{(s,r,o^*)\sim \mathcal{I}\cup\mathcal{O}} \Big[y \log(\hat{y}) + (1-y)\log(1-\hat{y}) \Big],$$

where y is the true label indicating membership in $\mathcal{I}$ ($y = 1$) or $\mathcal{O}$ ($y = 0$).

Finally, the overall loss function for memory modality pre-training is shown in Equation 3, where λ_1, λ_2, and λ_3 are coefficients for different loss functions, with default values set to 1. Notably, the memory modality pre-training phase is entirely independent of the LLMs F that are to be edited. Therefore, it can be trained independently to embed a substantial amount of knowledge memory. As LLMs are updated or replaced over time, E_m can be reused across various models, facilitating efficient cross-model editing.

$$L_{\text{pre-training}} = \lambda_1 L_{\text{inject}} + \lambda_2 L_{\text{associate}} + \lambda_3 L_{\text{exist}}. \quad (3)$$

4.2 Memory Modality Bridging

After memory modality pre-training, we have obtained a memory modality encoder E_m, which has stored the relevant memories of (s, r, o^*) and is capable of extracting related memories and determining their existence based on context. Revisiting Equation 2, we observe that we still need a memory-to-language module P_m to bridge the memory modality to LLMs. This module allows LLMs to understand and interpret the memories.

In mainstream MM-LLM works (Li et al., 2023a; Liu et al., 2024; Jian et al., 2024), to obtain the modality-to-language alignment modules P, a modality-conditioned text generation loss is typically used as the training objective. Drawing inspiration from this, we also adopt this training objective, enabling the LLM F to correctly predict o^* based on the output of the memory modality and the prompt (composed of s and r), as specifically shown in Equation 4. Here, we use a simple two-layer fully connected network as P_m, while keeping the parameters of the LLM F and the memory modality encoder E_m frozen.

$$L_1 = -E_{(s,r,o^*)\sim D_{\text{edit}}} \left[\log \mathbb{P}_F(o^* \mid P_m(E_m(s,r)) \oplus (s,r))\right]. \quad (4)$$

However, training solely with this objective only enables LLMs to comprehend knowledge present in memory. For knowledge not contained in memories, the output from the memory modality may lead to unpredictable behavior. Ideally, for knowledge not present in memories, i.e., $(s, r, o^*) \in \mathcal{O}$, we expect the LLMs to ignore the memory modality's output and maintain their original predictions. To achieve this, we minimize the discrepancy between pre- and post-edit model predictions by reducing the Kullback-Leibler (KL) divergence of their prediction distributions(Cover, 1999), as illustrated in Equation 5. Benefiting from the pre-trained memory modality's ability to differentiate

representations of in-scope ($\mathcal{I}$) and out-of-scope ($\mathcal{O}$) knowledge, LLMs can more readily discern whether they possess the relevant memory.

$$L_2 = E_{(s,r,o^*)\sim\mathcal{O}} \left[\mathrm{KL}\left(\mathbb{P}_F(\cdot \mid (s,r)) \right.\right. \\ \left.\left. \| \, \mathbb{P}_{F^*}(\cdot \mid P_m(E_m(s,r)) \oplus (s,r))\right]. \quad (5)$$

In summary, we combine the two training objectives of the bridging phase into a single loss function, as shown in Equation 6, where λ_{kl} is the coefficient, defaulting to 1. Although the memory modality bridging phase is not LLM backbone agnostic, it only requires fine-tuning P_m. Given that the parameters involved in this fine-tuning are negligible compared to the total number of parameters, it enables an efficient and rapid implementation of memory modality bridging.

$$L_{\text{bridge}} = L_1 + \lambda_{kl} L_2. \quad (6)$$

5 Experiments

5.1 Experimental Setup

We first provide a brief overview of the datasets, models, metrics, and baseline methods used in our experiments. For more detailed information on the experimental setup, please refer to Appendix A.

Models and Datasets. We conducted experiments on three LLMs of varying sizes: GPT2-XL (1.5B) (Radford et al., 2019), GPT-J (6B) (Wang and Komatsuzaki, 2021), and LLaMA3 (8B) (Dubey et al., 2024). For the benchmark, we evaluated MindBridge on two commonly used knowledge editing datasets: the ZsRE dataset (Levy et al., 2017) and the Counterfact dataset (Meng et al., 2022a). For MindBridge, we designated the knowledge in the dataset that required editing as D_{edit}, and a portion of the knowledge that did not require editing as $\mathcal{O}$. Since the quantity of $\mathcal{I}$ in the dataset is small and typically used only for testing, we directly substitute D_{edit} for $\mathcal{I}$ during training.

Baseline Methods. We compare MindBridge with seven baseline methods, categorized into two groups: parameter-modifying knowledge editing approaches—Fine-Tuning (FT-L) (Meng et al., 2022a), r-ROME (Gupta et al., 2024a), MEMIT (Meng et al., 2022b), EMMET (Gupta et al., 2024c) and AlphaEdit (Fang et al., 2024); and parameter-preserving knowledge editing methods—GRACE (Hartvigsen et al., 2024) and WISE (Wang et al., 2024). For these baseline methods, we utilize EasyEdit (Zhang et al., 2024c) for replication and testing, applying the default parameter settings.

Metrics. Following previous work (Fang et al., 2024; Zhang et al., 2024c), we adopt three metrics to evaluate the performance of the edited model: **reliability** (edit success rate), **generalization** (paraphrase success rate), and **locality** (neighborhood success rate). These are abbreviated as Rel., Gen., and Loc., respectively. We further compute the average of these three metrics, denoted as Avg., to represent the overall editing performance.

5.2 Main Results

Table 1 shows the performance comparison between MindBridge and existing knowledge editing methods after editing 10,000 facts. We can observe that, compared to baselines, MindBridge demonstrates superior performance across multiple LLMs, different datasets, and almost all metrics. For example, on LLaMA3 and GPT-J, MindBridge outperforms the best baseline by more than 20% and 15% respectively on the Avg. metric, which measures the comprehensive editing performance. On the Gen. metric, MindBridge significantly outperforms other editing methods in all experiments. Furthermore, it is particularly important to note that MindBridge is the only editing method among these that can achieve cross-model editing. When the LLM is updated, MindBridge can retain a large amount of previously edited knowledge and quickly bridge the memory modality encoder to the new LLM, enabling rapid domain knowledge adaptation while maintaining excellent editing performance.

5.3 Further Analysis

Ablation Study for Memory Modality Pre-training. In the memory modality pre-training stage, we designed three training objectives for the modality encoder, each corresponding to its intended function. To validate the effectiveness of these three training objectives, we conducted an ablation study to evaluate the impact of each objective. We performed knowledge editing on 10,000 factual statements using GPT-J. The results, as shown in Table 2, demonstrate that employing only the L_{inject} objective already yields promising editing performance (with Rel. metric reaching 71.30% on the Counterfact dataset and 94.89% on the ZsRE dataset). However, the generalization and locality of the edits are limited. Upon incorporating the $L_{associate}$ training objective in addition to L_{inject}, the generalization capability of editing is enhanced (with the Gen. metric increasing by 29.1%↑ on the Counterfact dataset and 8.73%↑

Method	Model	Counterfact				ZsRE			
		Rel.↑	Gen.↑	Loc.↑	Avg.↑	Rel.↑	Gen.↑	Loc.↑	Avg.↑
Pre-edited		0.28	0.42	\	\	22.44	21.56	\	\
FT-L	GPT2-XL	0	0	22.84	7.61	15.60	15.83	28.21	19.88
WISE	GPT2-XL	1.79	1.99	50.38	18.06	24.88	24.84	99.98	49.90
AlphaEdit	GPT2-XL	66.00	30.90	42.98	46.62	51.82	42.97	53.72	49.50
EMMET	GPT2-XL	85.51	46.65	62.17	64.78	70.37	60.34	78.37	69.69
GRACE	GPT2-XL	**100**	0.39	**68.75**	56.38	**100**	3.17	**100**	67.72
r-ROME	GPT2-XL	0	0	0	0	0	0	0	0
MEMIT	GPT2-XL	45.36	39.70	66.25	44.08	47.20	24.70	60.34	50.44
MindBridge	GPT2-XL	95.60	**81.10**	38.18	**71.62**	78.14	**68.17**	85.67	**77.33**
Pre-edited		0.09	0.29	\	\	23.01	22.25	\	\
FT-L	GPT-J	10.58	7.39	1.03	6.34	15.73	14.04	9.47	13.08
WISE	GPT-J	18.38	12.08	3.85	11.44	36.88	34.53	99.51	56.97
AlphaEdit	GPT-J	92.00	46.45	57.40	65.28	76.95	52.37	62.02	63.78
EMMET	GPT-J	85.51	46.65	62.17	64.78	70.37	60.34	78.37	69.69
GRACE	GPT-J	**100**	0.29	**99.08**	66.46	**100**	3.13	**100**	67.71
r-ROME	GPT-J	0	0	0	0	0.09	0.08	0	0.05
MEMIT	GPT-J	95.80	57.64	61.87	71.77	86.17	69.70	75.96	77.28
MindBridge	GPT-J	94.60	**82.60**	93.56	**90.25**	99.06	**83.71**	95.77	**92.85**
Pre-edited		0.7	1.3	\	\	27.70	27.08	\	\
FT-L	LLaMA3	25.67	9.24	0.23	11.71	5.99	3.92	0.64	3.52
WISE	LLaMA3	16.08	10.68	3.87	10.21	29.18	29.18	99.40	52.07
AlphaEdit	LLaMA3	91.70	51.54	55.58	66.28	78.64	62.06	73.57	71.43
EMMET	LLaMA3	60.53	31.81	32.28	41.54	62.94	59.64	31.95	51.51
GRACE	LLaMA3	**100**	5.33	**100**	68.44	**100**	5.33	**100**	68.44
r-ROME	LLaMA3	0.09	0	0.15	0.08	0.70	0.50	1.01	0.74
MEMIT	LLaMA3	84.71	41.20	64.49	63.47	66.16	59.92	80.51	68.86
MindBridge	LLaMA3	93.85	**83.35**	92.14	**89.78**	99.03	**85.50**	92.65	**92.39**

Table 1: Comparison of MindBridge with existing methods after editing 10,000 facts. The best results are shown in bold, and the second best results are underlined.

on the ZsRE dataset). Furthermore, integrating the L_{exist} training objective alongside L_{inject} improves the locality of editing (with the Loc. metric increasing by 47.24%↑ on the Counterfact dataset and 19.05%↑ on the ZsRE dataset). These findings validate the effectiveness of each of the three training objectives. Ultimately, by combining all three training objectives, MindBridge achieves the best editing performance.

Impact on General Ability. Existing knowledge editing methods may more or less affect the general capabilities of models (Gu et al., 2024; Gupta et al., 2024b). To test the impact of MindBridge on the general capabilities of edited models, we evaluated the edited LLaMA3 (8B) on the General Language Understanding Evaluation (GLUE) benchmark (Wang, 2018) and compared it with AlphaEdit and MEMIT, which exhibit good performance in comprehensive editing effectiveness. As shown in Figure 3, MindBridge has the smallest change in F1-score compared to the pre-edited model on six tasks, indicating that MindBridge can maintain the general capabilities of the model even in the face of a large amount of knowledge editing.

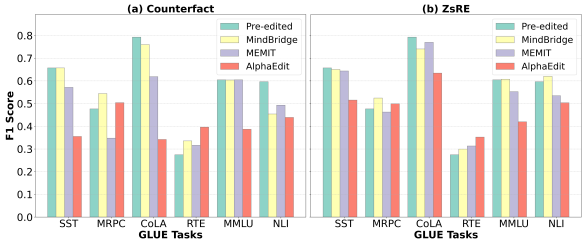
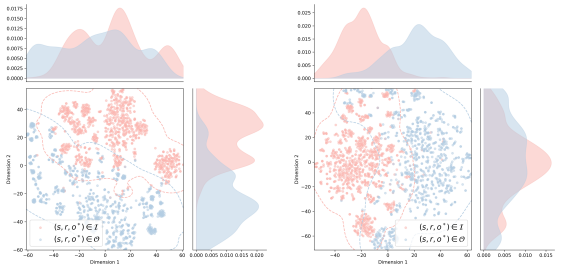


Figure 3: F1-score of LLaMA3 (8B) on the GLUE benchmark after editing 10,000 facts using MindBridge, AlphaEdit, and MEMIT. The evaluation includes six tasks: SST, MRPC, CoLA, RTE, MMLU, and NLI.

Visualization of In-scope and Out-of-scope Representations. To determine whether the memory modality can truly distinguish the presence of relevant memories, we randomly selected 1,000 examples from the in-scope and out-of-scope data of two datasets. Figure 4 shows the visualization of the representations extracted by the memory

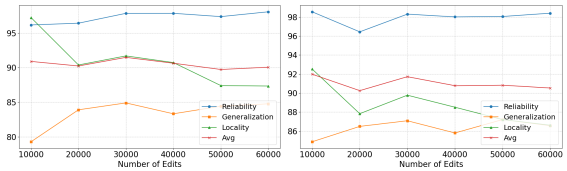
Training objectives	Counterfact				ZsRE			
	Rel.↑	Gen.↑	Loc.↑	Avg.↑	Rel.↑	Gen.↑	Loc.↑	Avg.↑
Pre-edited	0.2	0.5	\	\	24.20	22.93	\	\
L_{inject}	71.30	34.20	41.16	48.88	94.89	75.28	74.23	81.47
$L_{inject} + L_{associate}$	83.70	63.30	73.12	73.37	97.75	84.01	85.18	88.98
$L_{inject} + L_{exist}$	75.40	50.70	88.40	71.50	95.86	81.45	93.28	90.20
$L_{inject} + L_{associate} + L_{exist}$	**94.40**	**80.50**	**89.26**	**88.05**	**99.40**	**84.87**	**96.15**	**93.47**

Table 2: Ablation study of the three training objectives in MindBridge's memory modality pre-training. Edited LLM: GPT-J; 10,000 facts. Best results are highlighted in bold.



(a) Counterfact (b) ZsRE

Figure 4: Visualization of the dimensionality-reduced distributions of representations extracted by the memory modality encoder for $\mathcal{I}$ and $\mathcal{O}$.



(a) GPT-J (b) LLaMA3

Figure 5: Scaling MindBridge to 60,000 edits on ZsRE.

modality encoder, reduced to two dimensions using t-SNE. It can be observed that the data from $\mathcal{I}$ and $\mathcal{O}$ are separated in space, indicating that the memory modality encoder is capable of distinguishing whether relevant factual memories exist, thereby ensuring the locality of the edited LLM.

Scale Up to 60,000 Edits. We gradually extended MindBridge from 10,000 edits to 60,000 edits on the ZsRE dataset for GPT-J and LLaMA3, and tested their editing performance. As shown in Figure 5, despite the significant increase in the number of edits, MindBridge consistently achieves stable and superior editing performance. The Avg. metric remains largely unchanged, with only a slight decrease observed in the Locality metric.

Adaptability to Heterogeneous Architectures. Our prior evaluations focused on dense model architectures such as GPT-J and LLaMA3. To further probe MindBridge's cross-model adaptability, we extended our analysis to include a distinct heterogeneous architecture: the Mixture-of-Experts (MoE) model (Fedus et al., 2022). Table 3 details the outcomes of applying 10,000 knowledge edits to two prominent MoE models, Qwen1.5-MoE (Team, 2024) and DeepSeekMoE 16B (Dai et al., 2024). Notably, MindBridge demonstrates robust editing performance even on these different architectures, achieving an Avg. metric of approximately 90% for both. This result strongly corroborates MindBridge's capacity for effective cross-model knowledge editing, extending its applicability across diverse architectural paradigms.

Overfitting Tests. Prior research indicates that knowledge editing can introduce overfitting challenges (Wang et al., 2025; Liu et al., 2025). To evaluate MindBridge's susceptibility to this phenomenon, we adopted the data construction methodology from the EVOKE benchmark (Zhang et al., 2024b). Specifically, we assessed LLaMA3(8B) models, each modified with 10,000 edits on the Counterfact dataset using MEMIT, AlphaEdit, and MindBridge, respectively. The evaluation focused on two key tasks: Relation Specificity and Prefix Distraction, utilizing the DP (Direct Probability), CAP (Correct Answer Probability), and EOS (Editing Overfit Score) metrics as defined in (Zhang et al., 2024b). The results, presented in Table 4, reveal that MindBridge surpasses both MEMIT and AlphaEdit in DP and EOS metrics, indicating a superior resilience against overfitting.

Analysis of MindBridge's Editing Efficiency. MindBridge facilitates efficient knowledge editing during model updates by reusing its pre-trained memory modality encoder, requiring only a straightforward memory modality bridging step for new LLMs. To quantify this, we analyzed MindBridge's editing time against baseline methods, as shown in Figure 6, which depicts the time taken for 10,000 edits on LLaMA3 (8B) (conducted on a Linux server equipped with 8×NVIDIA A100

Dataset	Method	Qwen1.5-MoE				DeepSeekMoE 16B			
		Rel.↑	Gen.↑	Loc.↑	Avg.↑	Rel.↑	Gen.↑	Loc.↑	Avg.↑
ZSRE	Pre-edited	28.00	27.04	-	-	28.56	28.23	-	-
	MindBridge	99.08	84.30	84.21	89.20	96.11	80.11	93.84	90.02
Counterfact	Pre-edited	0.50	1.00	-	-	1.22	1.50	-	-
	MindBridge	94.00	83.00	89.20	88.73	94.02	79.57	88.93	87.50

Table 3: Performance of MindBridge on Mixture-of-Experts (MoE) models (number of edits: 10,000).

Method	Prefix Distraction			Relation Specificity		
	DP↓	CAP↑	EOS↑	DP↓	CAP↑	EOS↑
Pre-edited	7.12%	21.44%	69.60%	0.37%	14.46%	92.80%
MEMIT	7.37%	**25.26%**	70.60%	1.24%	**18.20%**	88.40%
AlphaEdit	7.91%	21.09%	63.60%	1.97%	15.12%	85.20%
MindBridge	**6.84%**	23.95%	**71.00%**	**0.46%**	15.81%	**93.00%**

Table 4: Comparison of MindBridge with MEMIT and AlphaEdit in overfitting tests on the Relation Specificity and Prefix Distraction tasks (Zhang et al., 2024b). Best results are highlighted in bold.

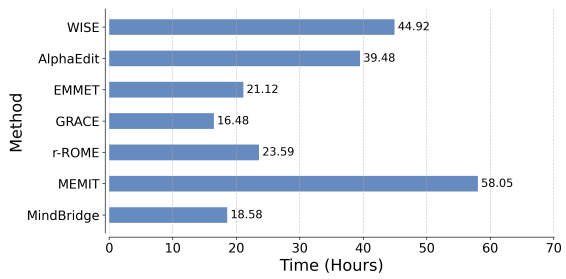
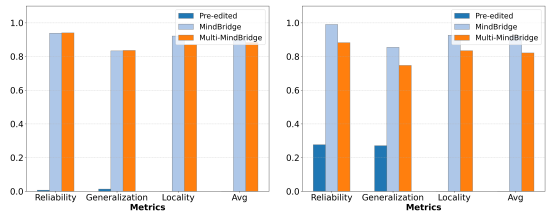


Figure 6: Time comparison of MindBridge with other knowledge editing methods for 10,000 edits on LLaMA3 (8B).



(a) Counterfact (b) ZsRE

Figure 7: Editing performance of Multi-MindBridge, which simultaneously bridges two memory modality encoders trained on different datasets and evaluates them on a single dataset.

40GB PCIe GPUs). MindBridge (18.58 hours) exhibits superior editing speed compared to methods like WISE (44.92 hours), AlphaEdit (39.48 hours), and MEMIT (58.05 hours). While GRACE processes edits faster (16.48 hours), its limited editing effectiveness makes it less advantageous. For an analysis of MindBridge's parameter overhead, please refer to Appendix B.4.

From MindBridge to Multi-MindBridge. Just as multimodal large language models are not limited to one or two modalities and can simultaneously support multiple modalities (Shu et al., 2023; Zhang et al., 2023), we propose Multi-MindBridge, which explores the editing performance of bridging multiple distinct memory modality encoders to the same LLM. We bridge encoders pretrained on the Counterfact and ZsRE datasets (each with 10,000 edits) to the LLaMA3 (8B) model. The results are shown in Figure 7. It can be observed that compared to using a single encoder, the editing performance only slightly decreases but enables the LLM to simultaneously acquire the knowledge memories from both memory modality encoders. For more implementation details and exploratory experiments of Multi-MindBridge, please refer to Appendix B.2.

6 Conclusion

In this paper, we propose MindBridge, a scalable cross-model knowledge editing method designed to address the issue that most current knowledge editing methods are overfitted to a single model, leading to the problem of discarded edited knowledge and frequent re-editing with each model update. Based on the novel concept of memory modality, MindBridge enables edited knowledge to transcend individual models. Extensive experiments conducted on two popular knowledge editing datasets and various LLMs demonstrate the effectiveness and scalability of MindBridge.

7 Limitations

Although MindBridge has demonstrated promising results in cross-model knowledge editing, it still faces several limitations. One limitation stems from resource constraints. We were unable to conduct tests on larger-scale models, restricting our experiments to models with up to 8B parameters. Furthermore, our focus was primarily on factual knowledge. We did not delve into other forms of knowledge, such as conceptual knowledge, which we leave this for future work.

Acknowledgments

This research was partially supported by the National Natural Science Foundation of China (Grants No.62406303), Anhui Provincial Natural Science Foundation (No. 2308085QF229), Anhui Province Science and Technology Innovation Project (202423k09020010), the Fundamental Research Funds for the Central Universities (No. WK2150110034).

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Jinze Bai, Shuai Bai, Shusheng Yang, Shijie Wang, Sinan Tan, Peng Wang, Junyang Lin, Chang Zhou, and Jingren Zhou. 2023. Qwen-vl: A versatile vision-language model for understanding, localization, text reading, and beyond. *arXiv preprint arXiv:2308.12966*, 1(2):3.

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

Zhe Chen, Jiannan Wu, Wenhai Wang, Weijie Su, Guo Chen, Sen Xing, Muyan Zhong, Qinglong Zhang, Xizhou Zhu, Lewei Lu, et al. 2024. Internvl: Scaling up vision foundation models and aligning for generic visual-linguistic tasks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 24185–24198.

Yunfei Chu, Jin Xu, Xiaohuan Zhou, Qian Yang, Shiliang Zhang, Zhijie Yan, Chang Zhou, and Jingren Zhou. 2023. Qwen-audio: Advancing universal audio understanding via unified large-scale audio-language models. *arXiv preprint arXiv:2311.07919*.

Thomas M Cover. 1999. *Elements of information theory*. John Wiley & Sons.

Damai Dai, Chengqi Deng, Chenggang Zhao, RX Xu, Huazuo Gao, Deli Chen, Jiashi Li, Wangding Zeng, Xingkai Yu, Yu Wu, et al. 2024. Deepseekmoe: Towards ultimate expert specialization in mixture-of-experts language models. *arXiv preprint arXiv:2401.06066*.

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

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

Alexey Dosovitskiy. 2020. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

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

William Fedus, Barret Zoph, and Noam Shazeer. 2022. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. *Journal of Machine Learning Research*, 23(120):1–39.

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

Akshat Gupta, Sidharth Baskaran, and Gopala Anumanchipalli. 2024a. Rebuilding rome: Resolving model collapse during sequential model editing. *arXiv preprint arXiv:2403.07175*.

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

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

Tom Hartvigsen, Swami Sankaranarayanan, Hamid Palangi, Yoon Kim, and Marzyeh Ghassemi. 2024. Aging with grace: Lifelong model editing with discrete key-value adaptors. *Advances in Neural Information Processing Systems*, 36.

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

Jitesh Jain, Jianwei Yang, and Humphrey Shi. 2024. Vcoder: Versatile vision encoders for multimodal large language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 27992–28002.

Yiren Jian, Chongyang Gao, and Soroush Vosoughi. 2024. Bootstrapping vision-language learning with decoupled language pre-training. *Advances in Neural Information Processing Systems*, 36.

Damjan Kalajdzievski. 2024. Scaling laws for forgetting when fine-tuning large language models. *arXiv preprint arXiv:2401.05605*.

Jacob Devlin Ming-Wei Chang Kenton and Lee Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of naacL-HLT*, volume 1, page 2. Minneapolis, Minnesota.

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

Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. 2023a. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *International conference on machine learning*, pages 19730–19742. PMLR.

KunChang Li, Yinan He, Yi Wang, Yizhuo Li, Wenhai Wang, Ping Luo, Yali Wang, Limin Wang, and Yu Qiao. 2023b. Videochat: Chat-centric video understanding. *arXiv preprint arXiv:2305.06355*.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. 2024. Visual instruction tuning. *Advances in neural information processing systems*, 36.

Tianci Liu, Zihan Dong, Linjun Zhang, Haoyu Wang, and Jing Gao. 2025. Mitigating heterogeneous token overfitting in llm knowledge editing. *arXiv preprint arXiv:2502.00602*.

Aman Madaan, Niket Tandon, Peter Clark, and Yiming Yang. 2022a. Memory-assisted prompt editing to improve gpt-3 after deployment. *arXiv preprint arXiv:2201.06009*.

Aman Madaan, Niket Tandon, Peter Clark, and Yiming Yang. 2022b. Memory-assisted prompt editing to improve gpt-3 after deployment. *arXiv preprint arXiv:2201.06009*.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. 2022a. Locating and editing factual associations in gpt. *Advances in Neural Information Processing Systems*, 35:17359–17372.

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

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

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

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*, pages 15817–15831. PMLR.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. 2023. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. 2021. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR.

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

V Sanh. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*.

Fangxun Shu, Lei Zhang, Hao Jiang, and Cihang Xie. 2023. Audio-visual llm for video understanding. *arXiv preprint arXiv:2312.06720*.

Chenmien Tan, Ge Zhang, and Jie Fu. 2023. Massive editing for large language models via meta learning. *arXiv preprint arXiv:2311.04661*.

Qwen Team. 2024. Qwen1.5-moe: Matching 7b model performance with 1/3 activated parameters".

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Alex Wang. 2018. Glue: A multi-task benchmark and analysis platform for natural language understanding. *arXiv preprint arXiv:1804.07461*.

Ben Wang and Aran Komatsuzaki. 2021. GPT-J-6B: A 6 Billion Parameter Autoregressive Language Model. https://github.com/kingoflolz/mesh-transformer-jax.

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. *arXiv preprint arXiv:2405.14768*.

Pinzheng Wang, Zecheng Tang, Keyan Zhou, Juntao Li, Qiaoming Zhu, and Min Zhang. 2025. Revealing and mitigating over-attention in knowledge editing. *arXiv preprint arXiv:2502.14838*.

Xiao Wang, Yuansen Zhang, Tianze Chen, Songyang Gao, Senjie Jin, Xianjun Yang, Zhiheng Xi, Rui Zheng, Yicheng Zou, Tao Gui, et al. 2023. Trace: A comprehensive benchmark for continual learning in large language models. *arXiv preprint arXiv:2310.06762*.

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

Duzhen Zhang, Yahan Yu, Jiahua Dong, Chenxing Li, Dan Su, Chenhui Chu, and Dong Yu. 2024a. Mm-llms: Recent advances in multimodal large language models. *arXiv preprint arXiv:2401.13601*.

Hang Zhang, Xin Li, and Lidong Bing. 2023. Video-LLaMA: An instruction-tuned audio-visual language model for video understanding. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 543–553, Singapore. Association for Computational Linguistics.

Kai Zhang, Qi Liu, Zhenya Huang, Mingyue Cheng, Kun Zhang, Mengdi Zhang, Wei Wu, and Enhong Chen. 2022a. Graph adaptive semantic transfer for cross-domain sentiment classification. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1566–1576.

Kai Zhang, Qi Liu, Hao Qian, Biao Xiang, Qing Cui, Jun Zhou, and Enhong Chen. 2021. Eatn: An efficient adaptive transfer network for aspect-level sentiment analysis. *IEEE Transactions on Knowledge and Data Engineering*, 35(1):377–389.

Kai Zhang, Hefu Zhang, Qi Liu, Hongke Zhao, Hengshu Zhu, and Enhong Chen. 2019. Interactive attention transfer network for cross-domain sentiment classification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 5773–5780.

Kai Zhang, Kun Zhang, Mengdi Zhang, Hongke Zhao, Qi Liu, Wei Wu, and Enhong Chen. 2022b. Incorporating dynamic semantics into pre-trained language model for aspect-based sentiment analysis. *arXiv preprint arXiv:2203.16369*.

Mengqi Zhang, Xiaotian Ye, Qiang Liu, Pengjie Ren, Shu Wu, and Zhumin Chen. 2024b. Uncovering overfitting in large language model editing. *arXiv preprint arXiv:2410.07819*.

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

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

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, et al. 2023b. Judging llm-as-a-judge with mt-bench and chatbot arena. *Advances in Neural Information Processing Systems*, 36:46595–46623.

A Experimental Setup

In this section, we provide a more detailed introduction to the experimental setup, including the datasets used, a detailed explanation of the evaluation metrics, and a thorough description of the baselines.

A.1 Datasets

ZsRE. ZsRE (Levy et al., 2017) (Zero-Shot Relation Extraction) is a question-answering dataset widely used in knowledge editing tasks. Each data entry includes a question, the subject of the question, the updated answer, rephrased questions for testing the generalization of edits, and unrelated questions for testing the locality of edits. In the experiments comparing with baselines (see Section 5.2), we randomly selected 10,000 samples from this dataset as D_{edit}. Due to the limited amount of in-scope data $\mathcal{I}$ and to prevent contamination of test data, we directly used D_{edit} as a substitute for $\mathcal{I}$ during the training of MindBridge. The remaining samples that did not require editing were used as out-of-scope data $\mathcal{O}$. For other baseline methods, the edit data used were identical to those of MindBridge.

Counterfact. Counterfact (Meng et al., 2022a) is a more challenging knowledge editing dataset compared to ZsRE. It consists of incorrect facts that initially receive much lower scores than correct facts. Each entry in Counterfact includes a subject, an attribute of the subject to be edited, questions about attributes of non-identical subjects for testing edit locality, and paraphrases for testing

edit generalization. The construction of the editing data is similar to that of ZsRE.

A.2 Metrics

In this paper, we use three evaluation metrics—reliability, generalization, and locality—to represent performance. For simplicity, they are abbreviated as Rel., Gen., and Loc., respectively. Their specific formulas are as follows, where $\mathbb{1}(\cdot)$ denotes the indicator function.

$$Rel. = \frac{1}{|D_{edit}|} \sum_{(s,r,o^*)\in D_{edit}} \mathbb{1}(F^*(s,r) = o^*),$$

$$Gen. = \frac{1}{|\mathcal{I}|} \sum_{(s,r,o^*)\in \mathcal{I}} \mathbb{1}(F^*(s,r) = o^*),$$

$$Loc. = \frac{1}{|\mathcal{O}|} \sum_{(s,r,o^*)\in \mathcal{O}} \mathbb{1}(F^*(s,r) = F(s,r)).$$

Here, Rel. measures the proportion of edits that have been successfully applied to a model, Gen. indicates the proportion of edits to which the model generalizes after editing, and Loc. reflects the extent to which the edited model retains knowledge of unrelated facts. Note that the Loc. metric is only evaluated on the edited model. Higher values for all three metrics indicate better editing performance.

A.3 Baselines

Here, we introduce several knowledge editing baselines that are compared in this paper. We utilize EasyEdit(Zhang et al., 2024c) for the reproduction of these baselines. For the selection of hyperparameters among them, we follow the default settings provided in the code.

- **FT-L** is a fine-tuning method proposed by (Meng et al., 2022a). FT-L directly fine-tunes the feed-forward network (FFN) of a specific layer, identified by causal tracing in ROME, by maximizing the probability of all tokens in the target sequence through last token prediction.
- **GRACE** (Hartvigsen et al., 2024) is a lifelong editing method. It writes new mappings into a pre-trained model's latent space, creating a discrete, local codebook of edits without altering model weights. The phenomenon of GRACE's generalization collapse under extensive editing has been evidenced and discussed in (Wang et al., 2024).
- **WISE** (Wang et al., 2024) is a method specifically designed for lifelong model editing. It inserts side memory in the FFN layers to preserve edited memory and trained a router to select which memory module to activate. To further enhance the support for continual editing, WISE incorporated a knowledge-sharding mechanism to enable different edits to be maintained in distinct parameter subspaces.
- **r-ROME** is an improvement upon ROME. Prior work has demonstrated that ROME can suffer from disabling edits, leading to immediate model collapse (Gupta et al., 2024b). r-ROME identifies that this is caused by irregularities in ROME's implementation, specifically the asymmetric usage of key-vectors in its update equation, and proposes a more stable implementation, r-ROME.
- **MEMIT** (Meng et al., 2022b) is a scalable multi-layer update algorithm that employs explicitly calculated parameter updates to insert new memories. Building upon the direct editing approach of ROME, it designs an edit-distribution algorithm to distribute parameter updates uniformly across multiple layers of parameters. This enables MEMIT to update thousands of new pieces of knowledge.
- **EMMET** (Gupta et al., 2024c) unifies two editing methods, ROME and MEMIT, under a single optimization objective, namely, the preservation memorization objective. Furthermore, it improves upon ROME by employing equality constraints to support batched editing, achieving comparable performance to MEMIT.
- **AlphaEdit** (Fang et al., 2024) extends the locating-and-editing method by projecting the perturbation introduced during the editing process onto the null-space of the knowledge to be preserved. Subsequently, it applies this projection to the model parameters. This mechanism ensures the model's preservation of its original knowledge following the edit.

B More Experimental results

B.1 Impact of Different Memory Modality Encoders

We use BERT-Base (110M) as the default modality encoder. To evaluate how different-sized BERT

models affect editing performance, we compared DistillBERT (66M) (Sanh, 2019) and BERT-Large (340M). As shown in Table 5, DistillBERT delivers strong performance across GPT-J and LLaMA3. In most cases, BERT-Base and DistillBERT outperform the others, with DistillBERT achieving the best overall results on Counterfact and BERT-Base excelling on ZsRE. Despite its larger size, BERT-Large performs worse than the other two models.

B.2 Implementation Details and Further Experiments of Multi-MindBridge

In Section 5.3, inspired by the idea that multimodal large language models are not restricted to one or two modalities, we proposed the implementation of Multi-MindBridge. This allows LLMs to be jointly bridged by multiple memory modality encoders while possessing the corresponding knowledge memories. Here, we elaborate on its implementation details.

Given n pre-trained memory modality encoders $E_m^1, E_m^2, \ldots, E_m^n$ with different knowledge memories and their corresponding memory modality-language modules $P_m^1, P_m^2, \ldots, P_m^n$, similar to Equation 2, the model F_{multi}^* after editing with Multi-MindBridge can be expressed as follows:

$$\begin{aligned} x_{multi} &= \bigoplus_{i=1}^{n} P_m^i(E_m^i(s,r)), \\ F_{multi}^*(s,r) &= F(x_{multi} \oplus (s,r)). \end{aligned} \tag{7}$$

Here, $\bigoplus$ denotes the concatenation of outputs from the modality encoders E_m^1 to E_m^n to form x_{multi}. The resulting x_{multi} is then provided as a soft prompt to the LLM, together with the text prompt, a multi-source integration strategy with parallels in other NLP models that utilize interactive processing or fuse diverse data structures like sequences and graphs (e.g., (Zhang et al., 2019, 2022a)). Notably, in Multi-MindBridge, each memory modality encoder is independently trained using the same pipeline as MindBridge, and subsequently, they are combined and integrated into the edited LLM.

In Section 5.3, we tested the performance of Multi-MindBridge when bridging modality encoders trained on different datasets. The results showed that compared to using a single encoder, the editing performance only slightly decreased but allowed the LLM to simultaneously acquire the knowledge memory from two encoders. Here, we test the editing performance of bridging multiple

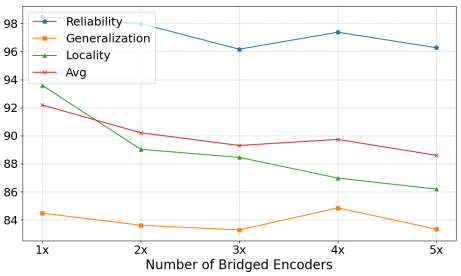


Figure 8: Editing performance of multiple memory modality encoders bridged to LLaMA3 on the ZsRE dataset, where each encoder handles 10,000 edits (non-overlapping).

encoders trained on the same dataset. Specifically, we selected the ZsRE dataset and randomly sampled 50,000 edits, training one memory modality encoder for every 10,000 edits (with no overlap between edits). We evaluated the editing performance of bridging 1 to 5 encoders simultaneously on LLaMA3. The experimental results are shown in Figure 8. As can be observed, as the number of simultaneously bridged encoders increases, the editing performance gradually decreases but remains satisfactory. Even when bridging up to 5 encoders, the Avg score still reaches 88.59%.

B.3 Impact of Editing Volume on General Capabilities

(Gupta et al., 2024b) observed that when using some model editing methods, the general capabilities of the model tend to decline as the number of edits increases, a challenge that relates to the broader goal of efficiently transferring and adapting learned capabilities across different domains or tasks (e.g., (Zhang et al., 2021)). Here, we test whether MindBridge exhibits a similar phenomenon. We conduct experiments using LLaMA3 (8B), gradually increasing the number of edits from 10,000 to 60,000. Similar to Section 5.3, we evaluate its performance on the GLUE benchmark, which includes six tasks: SST, MRPC, CoLA, RTE, MMLU, and NLI. The experimental results are shown in Figure 9. As can be seen, with the increase in the number of edits, the F1 scores for all tasks except NLI and CoLA remain stable, and in some cases, even outperform the pre-edited model. For NLI and CoLA, the F1 scores exhibit noticeable fluctuations as the number of edits increases, but they do not show a consistent downward trend. Overall, MindBridge demonstrates good scalability in preserving the model's general capabilities.

Modality Encoder	Model	Counterfact				ZsRE			
		Rel.↑	Gen.↑	Loc.↑	Avg.↑	Rel.↑	Gen.↑	Loc.↑	Avg.↑
DistilBERT (66M)	GPT-J	**98.21**	**90.64**	92.50	**93.79**	97.60	82.47	96.52	92.20
BERT-Base (110M)	GPT-J	94.71	83.57	**93.61**	90.63	**99.09**	**84.22**	95.95	**93.09**
BERT-Large (340M)	GPT-J	94.43	81.86	91.07	89.12	96.16	79.29	**97.22**	90.89
DistilBERT (66M)	LLaMA3	**97.71**	**89.31**	90.17	**92.39**	97.12	84.46	93.81	91.80
BERT-Base (110M)	LLaMA3	93.97	83.58	**92.08**	89.88	**98.50**	**85.02**	92.44	**91.99**
BERT-Large (340M)	LLaMA3	92.80	80.81	89.99	87.87	96.43	83.15	**95.34**	91.64

Table 5: Impact of different modality encoders on editing performance, 10,000 edits. Best results are shown in bold.

Model	Model Params (B)	Additional Params (B)	Total Add. Params (B)	% Increase
LLaMA3	8.030	E_m: 0.110, P_m: 0.008	0.118	1.46
GPT-J	6.050	E_m: 0.110, P_m: 0.010	0.120	1.93
Qwen1.5-MoE	14.316	E_m: 0.110, P_m: 0.004	0.114	0.80
DeepSeekMoE 16B	16.376	E_m: 0.110, P_m: 0.004	0.114	0.70

Table 6: Comparison of model parameters and additional parameters introduced by MindBridge across different LLMs. E_m denotes the memory modality encoder, and P_m denotes the bridging module. (B) stands for billions. Percentage increase is relative to the original model parameters.

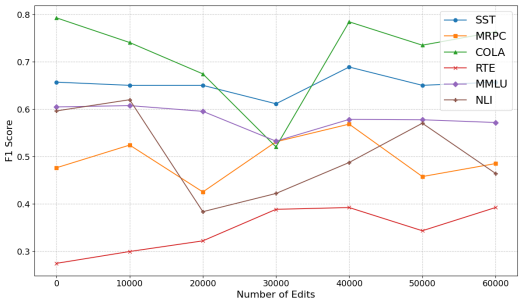


Figure 9: Change in F1-score on the GLUE benchmark for the edited LLaMA3 (8B) as the number of edited facts varies. 0 denotes the pre-edited model. Evaluation includes six tasks: SST, MRPC, CoLA, RTE, MMLU, and NLI.

B.4 Analysis of Additional Parameters in MindBridge

To evaluate MindBridge's computational footprint, this section analyzes the additional parameters introduced during knowledge editing across various Large Language Models (LLMs), utilizing BERT-Base as the default memory modality encoder. Table 6 provides a detailed breakdown.

MindBridge introduces two lightweight components: a shared memory modality encoder (E_m, 0.110B) and a model-specific bridging module (P_m, 0.004–0.010B). As shown in Table 6, the encoder remains consistent across all models, thanks to its reusability, which facilitates smooth knowledge transfer. In contrast, the bridging module is fine-tuned for each LLM during the memory modality bridging stage, with significantly fewer parameters—thereby ensuring parameter efficiency during this stage.

As a result, the overall parameter overhead is indeed minimal. For example, MindBridge adds only 0.118B parameters to LLaMA3 (8.030B), a 1.46% increase, and 0.120B to GPT-J (6.050B), or 1.93%. For larger Mixture-of-Experts models such as Qwen1.5-MoE (14.316B) and DeepSeekMoE 16B (16.376B), the relative increase drops even further to 0.80% and 0.70%, respectively, with only 0.114B added.

This analysis confirms MindBridge as a parameter-efficient knowledge editing solution, a goal pursued through various techniques when adapting pre-trained models for specific functionalities (e.g., (Zhang et al., 2022b)). The added parameters remain consistently low, with under 2% increase across all evaluated models and below 1% for larger MoE architectures, emphasizing its lightweight design and minimal impact on base model size. This efficiency facilitates seamless integration and contributes to MindBridge's broad applicability across diverse LLMs.