ComprehendEdit: A Comprehensive Dataset and Evaluation Framework for Multimodal Knowledge Editing

Yaohui Ma[1, 2, 3*], **Xiaopeng Hong**[1, 2], **Shizhou Zhang**[4], **Huiyun Li**[3, 5, 6], **Zhilin Zhu**[1, 2], **Wei Luo**[3], **Zhiheng Ma**[3, 5, 6†]

[1]Faculty of Computing, Harbin Institute of Technology
[2]Pengcheng Laboratory
[3]Shenzhen University of Advanced Technology
[4]School of Computer Science, Northwestern Polytechnical University
[5]Guangdong Provincial Key Laboratory of Computility Microelectronics
[6]Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences
{yhma, zhuzhilin}@stu.hit.edu.cn, hongxiaopeng@hit.edu.cn, szzhang@nwpu.edu.cn,
{hy.li, zh.ma}@siat.ac.cn, luow0014@e.ntu.edu.sg

Abstract

Large multimodal language models (MLLMs) have revolutionized natural language processing and visual understanding, but often contain outdated or inaccurate information. Current multimodal knowledge editing evaluations are limited in scope and potentially biased, focusing on narrow tasks and failing to assess the impact on in-domain samples. To address these issues, we introduce ComprehendEdit, a comprehensive benchmark comprising eight diverse tasks from multiple datasets. We propose two novel metrics: Knowledge Generalization Index (KGI) and Knowledge Preservation Index (KPI), which evaluate editing effects on in-domain samples without relying on AI-synthetic samples. Based on insights from our framework, we establish Hierarchical In-Context Editing (HICE), a baseline method employing a two-stage approach that balances performance across all metrics. This study provides a more comprehensive evaluation framework for multimodal knowledge editing, reveals unique challenges in this field, and offers a baseline method demonstrating improved performance. Our work opens new perspectives for future research and provides a foundation for developing more robust and effective editing techniques for MLLMs.

Codes — https://github.com/yaohui120/ComprehendEdit
Extended version — https://arxiv.org/abs/2412.12821

Introduction

The advent of large language models (LLMs) has transformed natural language processing (Zhao et al. 2023), while exposing limitations in maintaining up-to-date information and rectifying inaccuracies (Dhingra et al. 2022; Elazar et al. 2021; Cao et al. 2021). To address these challenges, knowledge editing methods (Zheng et al. 2023; Sun et al. 2024; Chen et al. 2024; De Cao, Aziz, and Titov 2021; Meng et al. 2022; Deng et al. 2024; Hu et al. 2024; Mitchell et al. 2022, 2021; Huang et al. 2023) enable updating outdated or incorrect knowledge within LLMs without complete retraining. These methods primarily focus on achieving reliability (successfully editing specified problems), generality (appropriately adjusting answers to similar questions), and locality (maintaining consistent responses to unrelated questions).

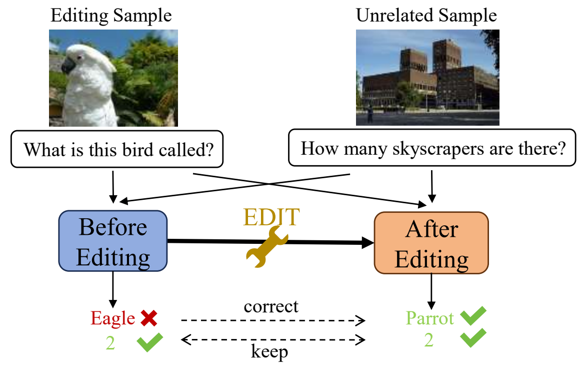

Figure 1: Concept of Multimodal Knowledge Editing. The framework is to correct the wrong answer for the editing sample ("Eagle" to "Parrot") while maintaining the output for unrelated samples ("2").

As multimodal large language models (MLLMs) emerge, new challenges arise in knowledge editing. While MLLMs like BLIP-2 OPT (Han et al. 2023), MiniGPT-4 (Zhu et al. 2023), Qwen-VL (Bai et al. 2023) and LLaVA-1.5 (Liu et al. 2024a) excel at answering questions about images, they still exhibit errors and misunderstandings. These inaccuracies stem from both language and vision modules (Liu et al. 2024b; Rawte et al. 2024; Tong et al. 2024; Jiang et al. 2023), necessitating multimodal-specific editing techniques.

Recent studies have established evaluation frameworks for multimodal knowledge editing through their MMEdit benchmark (including E-VQA and E-IC (Cheng et al.

[*]This work was done during the internship at Shenzhen Institutes of Advanced Technology.
[†]Corresponding author: Zhiheng Ma (zh.ma@siat.ac.cn)

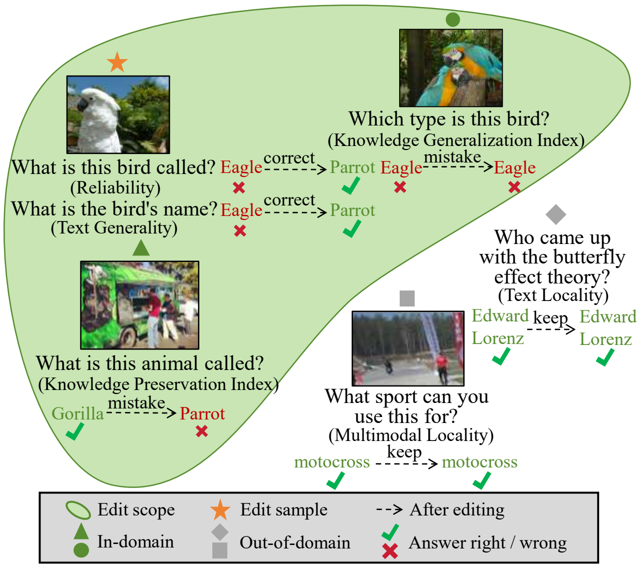

Figure 2: Knowledge Distortion in Multimodal Knowledge Editing. It shows how the model maintains correct outputs for out-of-domain samples but struggles with in-domain samples, highlighting the challenge in preserving and generalizing knowledge.

2023)), which builds upon VQAv2 (Goyal et al. 2017) and COCO Caption (Chen et al. 2015). They assess methods on reliability in modifying target outputs, generality across rephrased questions (Du et al. 2021) and generated images (Rombach et al. 2022), and locality in preserving responses on out-of-domain datasets like NQ dataset (Kwiatkowski et al. 2019) and OK-VQA (Marino et al. 2019). Initial results are promising - transferring language model editing techniques to multimodal contexts has proven effective, with methods like MEND (Mitchell et al. 2021) achieving 98.51% reliability and 96.65% multimodal locality on E-VQA (Cheng et al. 2023).

However, we argue that current multimodal knowledge editing evaluations are incomplete and potentially biased for the following reasons:

1) Limited Task Coverage: Existing assessments like E-VQA focus on narrow tasks, failing to capture broad MLLM capabilities such as spatial reasoning.

2) AI-synthetic Content Issues: Generating equivalent images introduces unpredictable content shifts (Huang et al. 2024), while VQA questions offer limited rephrasing variations (e.g., "Is there a tree in front of the building?" vs "What is the status of the tree in relation to the building?").

3) Out-of-domain Only Evaluation: Current locality assessment uses only distant, unrelated samples, missing potential unintended changes to in-domain knowledge.

To address these limitations, we introduce a novel benchmark ComprehendEdit, with three key innovations:

1) Comprehensive Task Coverage: Eight diverse tasks derived from multiple datasets, ensuring broad evaluation of MLLM capabilities.

2) No Synthetic Content Dependency: Two novel metrics – Knowledge Generalization Index (KGI) and Knowledge Preservation Index (KPI) that evaluate editing effects without relying on AI-synthetic content.

3) In-domain Assessment: These metrics measure how editing affects similar knowledge within the same domain, providing crucial insights previously overlooked.

Within this evaluation framework, we thoroughly assessed existing multimodal knowledge editing methods. Our findings reveal that current approaches struggle to perform optimally across all metrics, indicating significant room for improvement. Many methods that excelled in previous evaluations performed poorly on the new metrics, demonstrating both the bias in earlier assessments and the considerable potential for advancement in multimodal knowledge editing techniques.

Based on the issues revealed by our evaluation framework, we establish a baseline method, Hierarchical In-Context Editing (HICE), and conduct comprehensive ablation studies to investigate various trade-offs in multimodal knowledge editing. HICE achieves comparable accuracy to previous state-of-the-art methods on existing metrics, while demonstrating superior and more balanced performance on the newly introduced metrics.

In summary, this study advances multimodal knowledge editing research by revealing unique challenges distinct from language-only editing, particularly in preserving and generalizing knowledge within the editing domain. Through our comprehensive benchmark ComprehendEdit, we establish new evaluation protocols for evaluating knowledge effects on editing-related samples, while providing a strong baseline method for systematic comparison. This work not only exposes previously overlooked deficiencies in current approaches but also establishes a foundation for developing more effective multimodal knowledge editing techniques.

Related Works

Knowledge Editing

Recent studies on knowledge editing can be classified into three categories: locate and update knowledge neurons, meta learning methods and memory based methods.

Locate and update methods focus on identifying and modifying specific neurons within a model. ROME (Meng et al. 2022) applies the interventions on activation to determine which neurons have the strongest effect on the prediction. KN (Dai et al. 2021) uses the integrated gradients method to calculate neuron contributions, thereby identifying knowledge neurons. T-patcher (Huang et al. 2023) locates and inserts trainable neurons into specific layers to alter the model's output. Additionally, UnKE (Deng et al. 2024) and WilKE (Hu et al. 2024) are not restricted to particular MLP layers or knowledge neurons. They search for parameters to edit across a broader range of locations.

Although these methods update only a few of neurons, they require substantial computation to identify the location of updated neurons, which increases the training cost. Additionally, their application in black box models is limited.

Meta learning methods employ auxiliary models to guide parameter updates. MEND (Mitchell et al. 2021) and

KE (De Cao, Aziz, and Titov 2021) both train additional modules to adjust gradients, ensuring that the optimized model minimally impacts predictions for unrelated inputs. SLAG (Hase et al. 2023) uses LSTM and MLPs to learn a set of weights for gradient modification.

Compared to locate-and-update methods, meta learning methods demonstrate superior locality. However, the memory and training time required for the additional networks are significant considerations.

Memory based methods store learnable parameters and editing samples within training set. SERAC (Mitchell et al. 2022) stores an editing sample and trains a counterfactual model to obtain the expected output. IKE (Zheng et al. 2023) first employ in-context learning (Brown 2020) for editing knowledge in language models. They construct demonstrations of each training sample, and select appropriate demonstrations as context to modify the model's output. DISCO (Sun et al. 2024) similarly use in-context learning to enhance the edited model's ability to utilize the edited knowledge for reasoning. Building upon these approaches, HICE introduces a two-stage process: it classifies samples as in-domain or out-of-domain before applying in-context learning. This classification step prevents the application of in-context learning to out-of-domain samples, thus avoiding potential interference from irrelevant demonstrations.

Multimodal Knowledge Editing

The advancement of Multimodal Large Language Models (MLLMs) demands new approaches to knowledge editing. Cheng et al. first introduced multimodal knowledge editing and developed a benchmark named MMEdit. They also established novel evaluation metrics: reliability, generality, and locality. Similarly, KEBench (Huang et al. 2024) extends existing metrics and introduces a portability metric to assess the model's ability to effectively apply edited knowledge to related content.

Benchmark for Multimodal Large Language Models

Recently, datasets used to evaluate multimodal large language models typically encompass assessments of various abilities, such as perception (e.g., object existence, quantity, and attributes) and reasoning (e.g., common sense reasoning and numerical calculation). However, most of these datasets are not suitable for knowledge editing evaluation. They either contain too few samples to support trainable methods (e.g., POPE (Li et al. 2023), MME (Fu et al. 2023)) or cannot be evaluated offline (e.g., VizWiz (Gurari et al. 2018), MMBench (Liu et al. 2023), MM-vet (Yu et al. 2023)).

Other datasets that can be used have limited types of model capability evaluations. GQA (Hudson and Manning 2019) offers vast samples but they are mostly limited to object existence, object recognition, object attributes and scene information. Other datasets focus exclusively on evaluating a certain capability of the model. For instance, TextVQA (Singh et al. 2019) focuses on assessing the model's ability to recognize text. TallyQA (Acharya, Kafle, and Kanan 2019) consists of object counting questions, with complex types that require content understanding. VSR (Liu, Emerson, and Collier 2023) emphasizes the spatial relationship between objects, encompassing dozens of relationships. MathVista (Lu et al. 2023) collects various graphs and tables, all of which require numerical reasoning to answer correctly. In order to overcome these drawbacks and to evaluate multimodal knowledge editing methods more comprehensively, we construct a novel benchmark ComprehendEdit.

Proposed Method

Dataset

Cheng et al. was the first to propose the multimodal editing problem and developed two tasks E-VQA and E-IC. However, Huang et al. identified content shifts in the generated images within these datasets, leading to inaccurate locality assessments. Additionally, due to the extensive capabilities of MLLMs, a single-source dataset is inadequate for evaluating knowledge editing methods comprehensively. Existing datasets inadequately address diverse editing challenges due to their limited variety in question types and sample diversity, as shown in appendix. Commonly used MLLMs evaluation datasets, such as VizWiz (Gurari et al. 2018), MMBench (Liu et al. 2023), MME (Fu et al. 2023), MM-vet (Yu et al. 2023), POPE (Li et al. 2023), are unsuitable for evaluating knowledge editing methods due to insufficient training samples or inability to support offline evaluation. To overcome these limitations, we propose a new benchmark ComprehendEdit, which comprises 8 tasks derived from diverse datasets. The details of the dataset are shown in Table 1.

Task	Training set	Testing set	Source
Object Existence	1,471	491	GQA
Object Recognition	2,227	735	GQA
Object Attributes	2,282	705	GQA
Object Counting	1,506	503	TallyQA
Scene Information	2,067	787	GQA
Spatial Relationship	1,709	530	VSR
Text Recognition	1,554	519	TextVQA
Numerical Inference	634	212	MathVista
Total	13,450	4,482	

Table 1: Task Distribution in ComprehendEdit. It details the number of training and testing samples for each task.

The ComprehendEdit benchmark encompasses eight diverse tasks. The ratio of training data to test data in each task is approximately 3:1, with a total of 17,932 samples. Examples from the dataset and detailed construction of each task are provided in the appendix.

For measuring text generality, we use a pre-trained model (such as ChatGLM (Du et al. 2021)) to generate equivalent inputs (rephrased questions). Additionally, we also utilize samples from the NQ dataset (Kwiatkowski et al. 2019) and OK-VQA dataset (Marino et al. 2019) to measure text locality (T-L) and multimodal locality (M-L), respectively, following previous benchmarks (Cheng et al. 2023).

Task Formulation

The goal of multimodal knowledge editing is to adjust the output of a multimodal language model for a specific sample. To formalize this goal, we consider an editing dataset $\mathcal{D}_e$ containing N samples. Each editing sample s comprises an image content i_e, a text question x_e, and a ground-truth y_e, represented as (i_e, x_e, y_e). Additionally, for each s, a rephrased question, a locality sample and a multimodal locality sample are also provided. The parameters of the model f before and after editing are denoted as θ_o, θ_e, respectively.

Conventional Evaluation Metrics

Reliability. Reliability ($\mathcal{M}_{rel}$) measures how effectively a model's knowledge can be edited. Given an editing dataset $\mathcal{D}_e$, for each sample (i_e, x_e, y_e), the goal is to modify the model f with parameters θ_o such that its output changes from the original incorrect prediction $y_o = f(i_e, x_e; \theta_o)$ to the desired correct answer y_e after editing (with parameters θ_e). Reliability is formally defined as:

$$\mathcal{M}_{rel} = \mathop{\mathbb{E}}_{(i_e,x_e,y_e)\in\mathcal{D}_e} \mathbb{I}(f(i_e, x_e; \theta_e) = y_e), \tag{1}$$

where $\mathbb{I}(\cdot)$ is an indicator function that returns 1 if the edited model's output matches the target answer and 0 otherwise.

Generality. Generality assesses whether the editing effects can transfer to semantically equivalent inputs. Beyond the original editing sample (i_e, x_e), the model should maintain correct behavior on variations of both the question and image while preserving the same meaning.

Following Cheng et al., we generate equivalent variations using pre-trained models: rephrased questions x_r using LLMs (Du et al. 2021) (e.g., "What color is the floor?" $\rightarrow$ "What color is the ground?"), and alternative images i_r using diffusion models (Rombach et al. 2022). Let $\mathcal{N}(x_e)$ and $\mathcal{N}(i_e)$ denote the sets of generated questions and images respectively. We evaluate text generality (T-G) and multimodal generality (M-G) as:

$$\mathcal{M}^{txt}_{general} = \mathop{\mathbb{E}}_{\substack{(i_e,x_e,y_e)\in\mathcal{D}_e \\ x\in\mathcal{N}(x_e)}} \mathbb{I}(f(i_e, x; \theta_e) = y_e), \tag{2}$$

$$\mathcal{M}^{img}_{general} = \mathop{\mathbb{E}}_{\substack{(i,x_e,y_e)\in\mathcal{D}_e \\ i\in\mathcal{N}(i_e)}} \mathbb{I}(f(i, x_e; \theta_e) = y_e). \tag{3}$$

where $\mathcal{M}^{txt}_{general}$, $\mathcal{M}^{img}_{general}$ measures performance on rephrased questions and generated images independently.

Locality. Locality measures whether knowledge editing preserves the model's behavior on unrelated inputs. Following Cheng et al., we evaluate locality using external datasets: NQ dataset (Kwiatkowski et al. 2019) for text questions and OK-VQA dataset (Marino et al. 2019) for image-text questions. Text locality (T-L) and multimodal locality (M-L) are defined as:

$$\mathcal{M}^{txt}_{loc} = \mathop{\mathbb{E}}_{(x,y)\in\mathcal{D}_{loc}} \mathbb{I}(f(x; \theta_e) = f(x; \theta_o)), \tag{4}$$

$$\mathcal{M}^{img}_{loc} = \mathop{\mathbb{E}}_{(i,x,y)\in\mathcal{D}_{loc\text{-}v}} \mathbb{I}(f(i, x; \theta_e) = f(i, x; \theta_o)), \tag{5}$$

where $\mathcal{D}_{loc}$ and $\mathcal{D}_{loc\text{-}v}$ are datasets containing samples significantly different from the edited samples. Note that locality measures output consistency rather than correctness - the original outputs $f(x; \theta_o)$ or $f(i, x; \theta_o)$ may be incorrect.

Proposed Evaluation Metrics

While conventional metrics focus on rephrased questions and out-of-domain samples, they overlook crucial aspects of knowledge editing within the same domain. Moreover, they rely on synthetic data that can introduce measurement inaccuracies through content shifts and semantic mismatches. To address these limitations, we propose two complementary metrics that directly evaluate editing effects on original in-domain samples.

Given an editing sample s, let $\mathcal{D}(s)$ denote the set of samples from the same source dataset as s. We split $\mathcal{D}(s)$ into complementary subsets such that $\mathcal{D}(s) = \mathcal{D}_{KGI}(s) \cup \mathcal{D}_{KPI}(s)$, where $\mathcal{D}_{KGI}(s)$ contains samples that the original model answered incorrectly (exclude s), and $\mathcal{D}_{KPI}(s)$ contains samples that the original model answered correctly.

Knowledge Generalization Index (KGI) measures how well the editing improves model performance on previously misclassified in-domain samples. For instance, after correcting the model to identify a specific parrot instead of "eagle", KGI evaluates whether this correction generalizes to other misclassified images. Unlike traditional generalization metrics that rely on synthetic data (Huang et al. 2024), KGI uses real samples to avoid measurement artifacts:

$$\mathcal{M}_{KGI} = \mathop{\mathbb{E}}_{s\in\mathcal{D}_e} \mathop{\mathbb{E}}_{s'\in\mathcal{D}_{KGI}(s)} \mathbb{I}(f(i', x'; \theta_e) = y'), \tag{6}$$

Knowledge Preservation Index (KPI) assesses whether editing preserves the model's correct behavior on in-domain samples. It quantifies potential negative impacts where editing might disrupt previously correct predictions, such as changing a correct gorilla identification after editing bird-related knowledge. KPI is defined as:

$$\mathcal{M}_{KPI} = \mathop{\mathbb{E}}_{s\in\mathcal{D}_e} \mathop{\mathbb{E}}_{s'\in\mathcal{D}_{KPI}(s)} \mathbb{I}(f(i', x'; \theta_e) = y'), \tag{7}$$

where for both metrics, $s' = (i', x', y')$ represents an in-domain sample.

Similarity-based Sampling. While KGI and KPI provide comprehensive evaluation metrics, testing all in-domain samples after each editing operation incurs substantial computational costs. To address this efficiency challenge while maintaining metric effectiveness, we propose a similarity-based sampling strategy.

For each editing sample s, we select the k most similar and k most dissimilar samples from $\mathcal{D}_{KGI}(s)$ and $\mathcal{D}_{KPI}(s)$ based on either image or text similarity scores. This dual-ended sampling approach captures both the local and global effects of knowledge editing. Specifically, we compute: 1) Image-based metrics (I-KGI, I-KPI): using visual feature similarity between images; 2)Text-based metrics (T-KGI, T-KPI): using semantic similarity between questions.

This sampling strategy not only reduces computational overhead but also enables fine-grained analysis of how editing effects propagate differently through visual and linguistic domains. The high-similarity samples reveal local editing impacts, while low-similarity samples help assess potential far-reaching effects within the same domain.

Hierarchical In-Context Editing

Pre-trained models are sensitive to parameter changes, which can significantly impact their performance on in-domain samples. Two-stage methods (Mitchell et al. 2022; Hartvigsen et al. 2024; Yu et al. 2024) address this by first determining whether an input requires a modified output and then generating the corresponding result. While IKE (Zheng et al. 2023) leverages contextual capabilities without modifying parameters, it can affect outputs on external data due to unrelated demonstrations. Inspired by IKE and two-stage approaches, we propose Hierarchical In-Context Editing (HICE). This method first determines if an input falls within the edited scope, then outputs either the original or updated prediction. This approach leverages contextual learning for in-domain data while preserving locality on external samples.

Recent studies suggest that features extracted by pre-trained models can be well adapted to classification tasks (Panos et al. 2023; McDonnell et al. 2024). Based on this, we use a pre-trained language model h to extract text features for the first stage classification. To enhance classification accuracy, these features are projected into higher dimensions (McDonnell et al. 2024). Illustration of HICE is placed in the appendix.

Here we provide a detailed introduction to the method. For each sample s in $\mathcal{D}_e$ along with its rephrased question, locality sample and multimodal locality sample, we follow IKE to structure these questions and answers separately into the template "New Fact: $\{x\}$ $\{y\}$ \n Prompt: $\{x\}$ $\{y\}$" to construct four demonstrations. Each demonstration is labeled with a one-hot vector $Y \in \{0,1\}^{4N\times 2}$, where 0 (or 1) indicates whether it originates from a locality sample. The features of these demonstrations $F \in \mathbb{R}^{4N\times d}$ are exacted by h, where d is the dimension of features. These are projected into higher dimension $F_p = FW_r \in \mathbb{R}^{4N\times M}$ by a randomly initialized weight $W_r \in \mathbb{R}^{d\times M}$, where M is the projected feature dimension. The projected features F_p are used to train a classifier W^*. Using the form of least squares problem with penalty term, an appropriate classifier weight W^* can be obtained by solving

$$W^* = \underset{W}{arg\,min}\, \|Y - F_pW\|_2^2 + \lambda\|W\|_2^2, \tag{8}$$

The solution to the above problem is

$$W^* = (F_p^\top F_p + \lambda I)^{-1} F_p^\top Y \tag{9}$$

where λ is a coefficient of penalty term, and $I \in \mathbb{R}^{M\times M}$ is an identity matrix.

To reduce memory usage, we store a subset of training samples in a text memory M_1. Additionally, to enhance the classification accuracy of W^*, some hard-to-classify external samples' questions are stored as M_2.

During inference, for each test sample (i,x,y), we first determine whether it requires updating by comparing its question x to those in M_2, and classify by W^*. If the maximum similarity between x and M_2 doesn't exceeds a threshold T, and it's classified as in-domain data, we retrieve k_0 similar demonstrations $\{s_i\}_{i=1}^{k_0}$ from M_1. These, combined with a demonstration s_o constructed from x,y, form a new question $x_{new} = [s_1; s_2; \cdots; s_{k_0}; s_o; x]$, which is then input as (i, x_{new}) to the model to obtain the updated output $f(i, x_{new}; \theta_o)$. Otherwise, we use the original model output $f(i, x; \theta_o)$.

Experiments

Benchmark and Evaluation Metrics

The evaluation metrics includes **Rel** (Reliability), **T-G** (Text Generality), **T-L** (Text Locality), **M-L** (Multimodal Locality) (Cheng et al. 2023) and **I-KGI**, **T-KGI**, **I-KPI**, **T-KPI**. Due to the content shifts in the rephrased images (Huang et al. 2024), we do not measure multimodal generality.

Comparison Methods

Our primary comparison targets methods includes Finetune vision model (FT-V), Finetune language model (FT-L), IKE (Zheng et al. 2023), SERAC (Mitchell et al. 2022) and MEND (Mitchell et al. 2021). The † symbol in the table indicates that we reproduced the results ourselves using the code provided by Cheng et al. and our own implementation.

Implementation Details

We conduct experiments on PyTorch with NVIDIA RTX 4090 GPUs. For FT method, we edit each test sample by fine-tuning the last layer of the language model (FT-L) or the vision model (FT-V). For other methods, we followed the experimental setting of Cheng et al.. Hyper-parameter values for these methods are provided in the appendix.

In the process of solving W^*, we used 80% of the training samples as training set, and reserved 20% as the validation set. The penalty term's coefficient λ is selected from $\{10^{-4}, 10^{-3}, \cdots, 10^3, 10^4\}$. We choose the one that performed best on the validation set as W^*. The dimension of randomly projected features M is set 10,000. The pre-trained language model h is all-MiniLM-L6-v2 (Reimers and Gurevych 2019), following Mitchell et al. and the pre-trained CLIP model we use is ViT-B/32 (Radford et al. 2021). When constructing text memory M_1, we use k-means clustering on CLIP-extracted features and select one sample per cluster. The number of cluster is set $5\% \times N$. We select $k_0 = 16$ similar samples from memory as context.

We use BLIP-2 OPT 2.7B and MiniGPT-4 7B to split the original samples into $\mathcal{D}_{KPI}$ and $\mathcal{D}_{KGI}$. When constructing $\mathcal{D}_{KPI}(s)$, $\mathcal{D}_{KGI}(s)$ for each test editing sample s, we consider the $k = 4$ nearest and farthest neighbors of the test sample s, and employ a pre-trained CLIP model (Radford et al. 2021) to extract features. We use the L2 norm of feature differences as the measure of similarity.

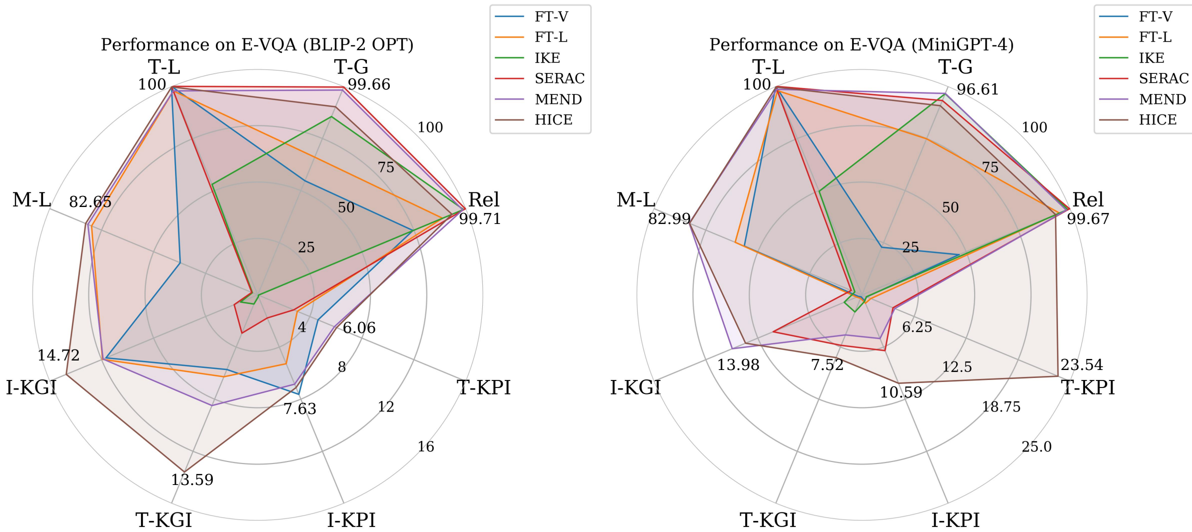

Figure 3: Performance comparison of knowledge editing methods on E-VQA benchmark. The range of values for Rel, T-G, T-L, M-L on two backbones are [0, 100], while the ranges of values for I-KGI, T-KGI, I-KPI, T-KPI are [0, 16] on BLIP-2 OPT and [0, 25] on MiniGPT-4.

Results

The results of different methods on E-VQA and ComprehendEdit are shown in Fig. 3 and Fig. 4, respectively. These figures reveal that all methods perform significantly on T-L, as the training set differs from out-of-domain data, which indicates that T-L poses little challenge.

ComprehendEdit's data, sourced from multiple datasets, exhibits greater internal variation compared to E-VQA's single-source data. Consequently, the samples in $\mathcal{D}_{KPI}(s)$ differ more from the edited sample s, making KPI more challenging in E-VQA.

FT-L and FT-V struggle to perform well on T-G, I-KGI, and T-KGI, as fine-tuning on a single sample limits the model's generalization capabilities. Fig. 3 and Fig. 4 demonstrate that indirectly fine-tuning the visual module is less effective than directly fine-tuning the language module, consistent with findings from (Cheng et al. 2023).

IKE constructs and selects demonstrations from memory for each test editing sample, combining them as context. This approach performs well on T-G since the context contains similar rephrased questions, which provide effective guidance. However, when processing samples from external datasets, demonstrations constructed from in-domain samples in the context interferes with the output, resulting to inferior performance on T-L and M-L compared to other methods. Moreover, IKE's poor performance on KGI and KPI indicates that the demonstrations used for the editing sample have limited effectiveness on other in-domain samples.

SERAC trains a classifier to decide whether to use the output from the original model or a counterfactual model for a given input sample. It excels on T-L because the questions in the NQ dataset differ significantly from those in the E-VQA dataset, allowing the classifier to identify these external data and rely on the original model's output. However, SERAC underperforms on M-L due to the absence of constraints on multimodal locality during training.

MEND demonstrates strong performance on Rel, T-G, and it especially outperforms other methods on M-L. This is attributed to its use of knowledge distilling loss on external data during the training of an additional module, which preserves existing knowledge after editing. However, its performance on KGI is still limited, since it doesn't use in-domain data when calculating knowledge distilling loss. Additionally, MEND's KPI accuracy of on ComprehendEdit is significantly higher than on E-VQA, because there is a greater difference between the editing sample s and the samples in $\mathcal{D}_{KPI}(s)$ in ComprehendEdit. Consequently, after editing, benefiting from gradient projection, the model's knowledge used to answer questions in $\mathcal{D}_{KPI}(s)$ is less affected.

Although previous methods perform well on Rel, T-G, T-L, they are limited in M-L and neglect the edited model in-domain performance. Targeting the poor performannce of existing methods on M-L, KGI and KPI, HICE demonstrates significant advantages across these metrics on various datasets and multimodal language models (MLLMs). HICE achieve a balance between Rel, T-G and T-L, M-L, KGI, KPI. The key to HICE's significant performance on M-L lies in the usage of challenging sample memory M_2 and the classifier W^*, which accurately determines whether a test sample is related to the edited sample. For unrelated input samples, the model can generate outputs directly, preserving performance on out-of-domain samples. For related input samples, HICE search for similar demonstrations in memory M_1 and combine them as context, ensuring correct answers to questions related to the edited sample.

Despite HICE's advantages in KPI and KGI, there is considerable room for improvement. This limitation primarily stems from the fact that in multimodal models, questions

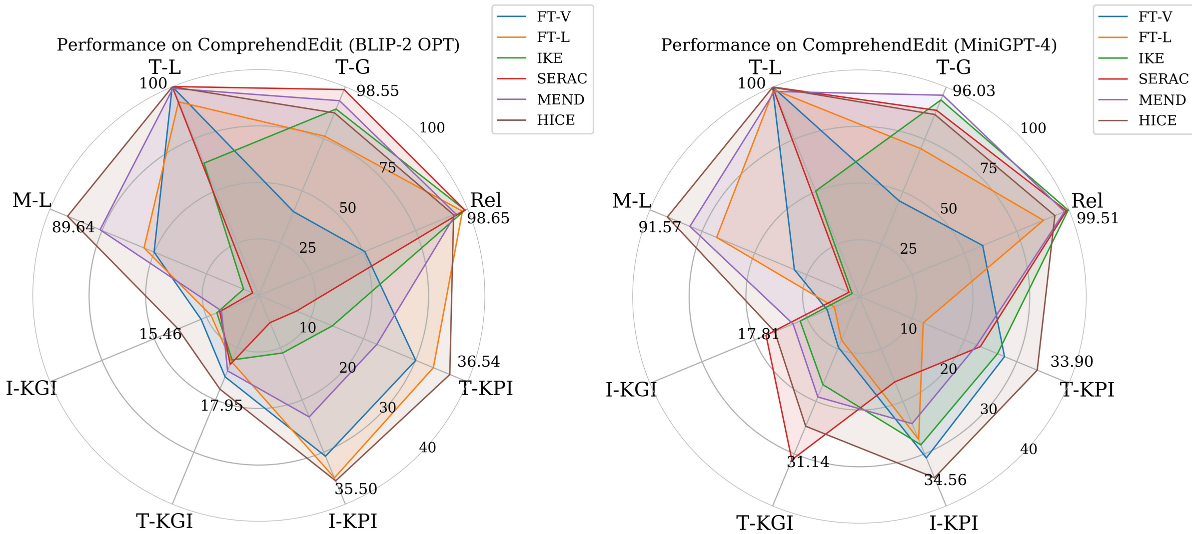

Figure 4: Performance comparison of knowledge editing methods on ComprehendEdit benchmark. The range of values for Rel, T-G, T-L, M-L on two backbones is [0, 100], while the range of values for I-KGI, T-KGI, I-KPI, T-KPI is [0, 40].

Module	Rel	T-G	T-L	M-L	I-KGI	T-KGI	I-KPI	T-KPI
baseline	75.86	15.44	98.1	37.63	3.90	2.89	2.59	1.08
$+M_1$	95.60	91.78	97.84	36.04	13.24	7.65	8.92	46.20
$+M_1+W_r$	95.94	92.26	99.64	52.56	13.29	7.65	8.85	46.20
$+M_1+M_2$	92.54	90.11	97.84	77.80	13.00	7.41	8.15	46.01
HICE	93.16	90.39	99.62	81.58	13.90	7.06	8.80	46.34

Table 2: The effect of each component of HICE.

are often closely tied to the input image. Relying solely on text-based context to address similar problems has inherent constraints. HICE shows substantial improvement in KPI because the demonstrations selected based on edited samples have lower correlation with samples further from the domain. As a result, the model can maintain accurate responses to these samples with minimal influence from unrelated demonstrations.

Ablation Study

To validate the effectiveness of each component in HICE, we conducted experiments on the E-VQA dataset using MiniGPT-4. For evaluating KGI and KPI, we selected the 1 nearest and 1 farthest neighbor.

The effect of each component of HICE. As shown in Table 2, "baseline" means we don't project features and utilize memory M_1 and M_2. The results of the first and second lines suggest that M_1 is the core part of HICE, and the demonstrations constructed from the training set is significantly beneficial for improving most indicators.

The last three lines suggest that W_r and M_2 are beneficial to improve the M-L, meaning the model is better at maintaining accurate responses for out-of-domain samples. This improvement occurs because the classifier effectively identifies out-of-domain samples, thus maintaining the model's output on these samples. However, there is a slight decrease in Rel and T-G metrics. This decline is likely due to the misidentification of a small number of in-domain samples, which results in the model not modifying its responses for these in-domain samples. Nevertheless, this slight reduction is acceptable given the substantial improvement in the model's performance on out-of-domain samples.

Conclusion

This study proposes key challenges in multimodal knowledge editing by introducing ComprehendEdit, a comprehensive benchmark with diverse tasks, and novel metrics Knowledge Generalization Index and Knowledge Preservation Index to assess in-domain editing impacts. Our baseline method, Hierarchical In-Context Editing, demonstrates balanced performance across various metrics, revealing unique characteristics of multimodal editing and exposing deficiencies in existing methods. This work provides a reliable evaluation framework and baseline, paving the way for more effective editing techniques in large multimodal language models. While significant progress has been made, our study highlights areas for future improvement, particularly in addressing the intricate relationship between questions and images in multimodal contexts, opening new perspectives for advancing the field.

Acknowledgements

This work is funded by the National Natural Science Foundation of China (62076195, 62206271, 92473112), the Fundamental Research Funds for the Central Universities (AUGA5710011522), the Shenzhen Key Technical Projects under Grant JSGG20220831105801004, CJGJZD2022051714160501.

References

Acharya, M.; Kafle, K.; and Kanan, C. 2019. TallyQA: Answering complex counting questions. In *Proceedings of the AAAI conference on artificial intelligence*, 01, 8076–8084.

Bai, J.; Bai, S.; Yang, S.; Wang, S.; Tan, S.; Wang, P.; Lin, J.; Zhou, C.; and Zhou, J. 2023. Qwen-vl: A versatile vision-language model for understanding, localization, text reading, and beyond. *arXiv preprint arXiv:2308.12966*.

Brown, T. B. 2020. Language models are few-shot learners. *arXiv preprint ArXiv:2005.14165*.

Cao, B.; Lin, H.; Han, X.; Sun, L.; Yan, L.; Liao, M.; Xue, T.; and Xu, J. 2021. Knowledgeable or educated guess? revisiting language models as knowledge bases. *arXiv preprint arXiv:2106.09231*.

Chen, Q.; Zhang, T.; Li, D.; Huang, L.; Xue, H.; Wang, C.; and He, X. 2024. Lifelong Knowledge Editing for LLMs with Retrieval-Augmented Continuous Prompt Learning. *arXiv preprint arXiv:2405.03279*.

Chen, X.; Fang, H.; Lin, T.-Y.; Vedantam, R.; Gupta, S.; Dollár, P.; and Zitnick, C. L. 2015. Microsoft coco captions: Data collection and evaluation server. *arXiv preprint arXiv:1504.00325*.

Cheng, S.; Tian, B.; Liu, Q.; Chen, X.; Wang, Y.; Chen, H.; and Zhang, N. 2023. Can We Edit Multimodal Large Language Models? *arXiv preprint arXiv:2310.08475*.

Dai, D.; Dong, L.; Hao, Y.; Sui, Z.; Chang, B.; and Wei, F. 2021. Knowledge neurons in pretrained transformers. *arXiv preprint arXiv:2104.08696*.

De Cao, N.; Aziz, W.; and Titov, I. 2021. Editing factual knowledge in language models. *arXiv preprint arXiv:2104.08164*.

Deng, J.; Wei, Z.; Pang, L.; Ding, H.; Shen, H.; and Cheng, X. 2024. UnKE: Unstructured Knowledge Editing in Large Language Models. *arXiv preprint arXiv:2405.15349*.

Dhingra, B.; Cole, J. R.; Eisenschlos, J. M.; Gillick, D.; Eisenstein, J.; and Cohen, W. W. 2022. Time-aware language models as temporal knowledge bases. *Transactions of the Association for Computational Linguistics*, 10: 257–273.

Du, Z.; Qian, Y.; Liu, X.; Ding, M.; Qiu, J.; Yang, Z.; and Tang, J. 2021. Glm: General language model pretraining with autoregressive blank infilling. *arXiv preprint arXiv:2103.10360*.

Elazar, Y.; Kassner, N.; Ravfogel, S.; Ravichander, A.; Hovy, E.; Schütze, H.; and Goldberg, Y. 2021. Measuring and improving consistency in pretrained language models. *Transactions of the Association for Computational Linguistics*, 9: 1012–1031.

Fu, C.; Chen, P.; Shen, Y.; Qin, Y.; Zhang, M.; Lin, X.; Yang, J.; Zheng, X.; Li, K.; Sun, X.; et al. 2023. MME: A Comprehensive Evaluation Benchmark for Multimodal Large Language Models. *arXiv preprint arXiv:2306.13394*.

Goyal, Y.; Khot, T.; Summers-Stay, D.; Batra, D.; and Parikh, D. 2017. Making the v in vqa matter: Elevating the role of image understanding in visual question answering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 6904–6913.

Gurari, D.; Li, Q.; Stangl, A. J.; Guo, A.; Lin, C.; Grauman, K.; Luo, J.; and Bigham, J. P. 2018. Vizwiz grand challenge: Answering visual questions from blind people. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 3608–3617.

Han, X.; Li, R.; Li, X.; and Pan, J. Z. 2023. A divide and conquer framework for Knowledge Editing. *Knowledge-Based Systems*, 279: 110826.

Hartvigsen, T.; Sankaranarayanan, S.; Palangi, H.; Kim, Y.; and Ghassemi, M. 2024. Aging with grace: Lifelong model editing with discrete key-value adaptors. *Advances in Neural Information Processing Systems*, 36.

Hase, P.; Diab, M.; Celikyilmaz, A.; Li, X.; Kozareva, Z.; Stoyanov, V.; Bansal, M.; and Iyer, S. 2023. Methods for measuring, updating, and visualizing factual beliefs in language models. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, 2714–2731.

Hu, C.; Cao, P.; Chen, Y.; Liu, K.; and Zhao, J. 2024. WilKE: Wise-Layer Knowledge Editor for Lifelong Knowledge Editing. *arXiv preprint arXiv:2402.10987*.

Huang, H.; Zhong, H.; Liu, Q.; Wu, S.; Wang, L.; and Tan, T. 2024. KEBench: A Benchmark on Knowledge Editing for Large Vision-Language Models. *arXiv preprint arXiv:2403.07350*.

Huang, Z.; Shen, Y.; Zhang, X.; Zhou, J.; Rong, W.; and Xiong, Z. 2023. Transformer-patcher: One mistake worth one neuron. *arXiv preprint arXiv:2301.09785*.

Hudson, D. A.; and Manning, C. D. 2019. Gqa: A new dataset for real-world visual reasoning and compositional question answering. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 6700–6709.

Jiang, D.; Liu, Y.; Liu, S.; Zhang, X.; Li, J.; Xiong, H.; and Tian, Q. 2023. From clip to dino: Visual encoders shout in multi-modal large language models. *arXiv preprint arXiv:2310.08825*.

Kwiatkowski, T.; Palomaki, J.; Redfield, O.; Collins, M.; Parikh, A.; Alberti, C.; Epstein, D.; Polosukhin, I.; Devlin, J.; Lee, K.; et al. 2019. Natural questions: a benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7: 453–466.

Li, Y.; Du, Y.; Zhou, K.; Wang, J.; Zhao, W. X.; and Wen, J.-R. 2023. Evaluating object hallucination in large vision-language models. *arXiv preprint arXiv:2305.10355*.

Liu, F.; Emerson, G.; and Collier, N. 2023. Visual spatial reasoning. *Transactions of the Association for Computational Linguistics*, 11: 635–651.

Liu, H.; Li, C.; Wu, Q.; and Lee, Y. J. 2024a. Visual instruction tuning. *Advances in neural information processing systems*, 36.

Liu, H.; Xue, W.; Chen, Y.; Chen, D.; Zhao, X.; Wang, K.; Hou, L.; Li, R.; and Peng, W. 2024b. A survey on hallucination in large vision-language models. *arXiv preprint arXiv:2402.00253*.

Liu, Y.; Duan, H.; Zhang, Y.; Li, B.; Zhang, S.; Zhao, W.; Yuan, Y.; Wang, J.; He, C.; Liu, Z.; et al. 2023. Mmbench: Is your multi-modal model an all-around player? *arXiv preprint arXiv:2307.06281*.

Lu, P.; Bansal, H.; Xia, T.; Liu, J.; Li, C.; Hajishirzi, H.; Cheng, H.; Chang, K.-W.; Galley, M.; and Gao, J. 2023. Mathvista: Evaluating mathematical reasoning of foundation models in visual contexts. *arXiv preprint arXiv:2310.02255*.

Marino, K.; Rastegari, M.; Farhadi, A.; and Mottaghi, R. 2019. Ok-vqa: A visual question answering benchmark requiring external knowledge. In *Proceedings of the IEEE/cvf conference on computer vision and pattern recognition*, 3195–3204.

McDonnell, M. D.; Gong, D.; Parvaneh, A.; Abbasnejad, E.; and van den Hengel, A. 2024. Ranpac: Random projections and pre-trained models for continual learning. *Advances in Neural Information Processing Systems*, 36.

Meng, K.; Bau, D.; Andonian, A.; and Belinkov, Y. 2022. Locating and editing factual associations in GPT. *Advances in Neural Information Processing Systems*, 35: 17359–17372.

Mitchell, E.; Lin, C.; Bosselut, A.; Finn, C.; and Manning, C. D. 2021. Fast model editing at scale. *arXiv preprint arXiv:2110.11309*.

Mitchell, E.; Lin, C.; Bosselut, A.; Manning, C. D.; and Finn, C. 2022. Memory-based model editing at scale. In *International Conference on Machine Learning*, 15817–15831. PMLR.

Panos, A.; Kobe, Y.; Reino, D. O.; Aljundi, R.; and Turner, R. E. 2023. First session adaptation: A strong replay-free baseline for class-incremental learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 18820–18830.

Radford, A.; Kim, J. W.; Hallacy, C.; Ramesh, A.; Goh, G.; Agarwal, S.; Sastry, G.; Askell, A.; Mishkin, P.; Clark, J.; et al. 2021. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, 8748–8763. PMLR.

Rawte, V.; Rani, A.; Sharma, H.; Anand, N.; Rajbangshi, K.; Sheth, A.; and Das, A. 2024. Visual hallucination: Definition, quantification, and prescriptive remediations. *arXiv preprint arXiv:2403.17306*.

Reimers, N.; and Gurevych, I. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*.

Rombach, R.; Blattmann, A.; Lorenz, D.; Esser, P.; and Ommer, B. 2022. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 10684–10695.

Singh, A.; Natarajan, V.; Shah, M.; Jiang, Y.; Chen, X.; Batra, D.; Parikh, D.; and Rohrbach, M. 2019. Towards vqa models that can read. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 8317–8326.

Sun, Z.; Liu, Y.; Wang, J.; Meng, F.; Xu, J.; Chen, Y.; and Zhou, J. 2024. Outdated Issue Aware Decoding for Factual Knowledge Editing. *arXiv preprint arXiv:2406.02882*.

Tong, S.; Liu, Z.; Zhai, Y.; Ma, Y.; LeCun, Y.; and Xie, S. 2024. Eyes wide shut? exploring the visual shortcomings of multimodal llms. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 9568–9578.

Yu, L.; Chen, Q.; Zhou, J.; and He, L. 2024. Melo: Enhancing model editing with neuron-indexed dynamic lora. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, 19449–19457.

Yu, W.; Yang, Z.; Li, L.; Wang, J.; Lin, K.; Liu, Z.; Wang, X.; and Wang, L. 2023. Mm-vet: Evaluating large multimodal models for integrated capabilities. *arXiv preprint arXiv:2308.02490*.

Zhao, W. X.; Zhou, K.; Li, J.; Tang, T.; Wang, X.; Hou, Y.; Min, Y.; Zhang, B.; Zhang, J.; Dong, Z.; et al. 2023. A survey of large language models. *arXiv preprint arXiv:2303.18223*.

Zheng, C.; Li, L.; Dong, Q.; Fan, Y.; Wu, Z.; Xu, J.; and Chang, B. 2023. Can We Edit Factual Knowledge by In-Context Learning? *arXiv preprint arXiv:2305.12740*.

Zhu, D.; Chen, J.; Shen, X.; Li, X.; and Elhoseiny, M. 2023. Minigpt-4: Enhancing vision-language understanding with advanced large language models. *arXiv preprint arXiv:2304.10592*.