MC-MKE: A Fine-Grained Multimodal Knowledge Editing Benchmark Emphasizing Modality Consistency

Anonymous ACL submission

Abstract

Multimodal large language models (MLLMs) are prone to non-factual or outdated knowledge issues, which can manifest as misreading and misrecognition errors due to the complexity of multimodal knowledge. Previous benchmarks have not systematically analyzed the performance of editing methods in correcting these two error types. To better represent and correct these errors, we decompose multimodal knowledge into its visual and textual components. Different error types correspond to different editing formats, which edits distinct part of the multimodal knowledge. We present MC-MKE, a fine-grained **M**ultimodal **K**nowledge **E**diting benchmark emphasizing **M**odality **C**onsistency. Our benchmark facilitates independent correction of misreading and misrecognition errors by editing the corresponding knowledge component. We evaluate three multimodal knowledge editing methods on MC-MKE, revealing their limitations, particularly in terms of modality consistency. Our work highlights the challenges posed by multimodal knowledge editing and motivates further research in developing effective techniques for this task.

1 Introduction

With the developments of multimodal large language models (MLLMs) , their application has become widespread across various fields. However, these models struggle with a challenge that the knowledge stored within them could be inaccurate or outdated. This issue manifests in two errors: misreading and misrecognition (Cheng et al., 2024). As shown in Figure 1, misrecognition occurs when a model mistakenly identifies an image, such as mistaking Mac Allister as Messi. On the other hand, misreading refers to incorrect textual knowledge, such as misremembering Messi's football team. Recent researches have introduced knowledge editing in multimodal contexts to address these issues.

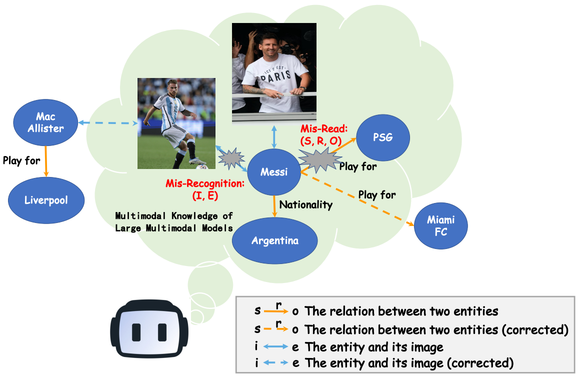


Figure 1: An illustration of multimodal knowledge and the two types of multimodal errors: misrecognizing a picture of Mac Allister as Messi, and misreading Messi's football team.

Following the conventional definition of knowledge-editing in LLMs, a few studies have proposed benchmarks for knowledge editing in MLLMs (Cheng et al., 2024; Huang et al., 2024; Li et al., 2024). However, these benchmarks over-simplify the evaluation of multimodal knowledge editing, and do not distinguish the differences between misreading and misrecognition errors(Cheng et al., 2024; Huang et al., 2024). Mixing evaluation of the two types of errors leads to inaccurate assessments of knowledge editing methods in real-world scenarios. Methods may appear to successfully inject objective multimodal knowledge, but actually conduct incorrect edits. Take the misreading error in Figure 1 for an example, where a MLLM misrecognizes the image of Messi to Mac Allister, leading to the erroneous knowledge that "the person in the image plays for Liverpool". If a knowledge editing method falsely injecting an knowledge triplet (*Mac Allister, Play for, Inter Miami*), it may still achieve great performance on prior benchmarks, since the multimodal knowledge (*Image of Messi, Play for, Inter Miami*) is actually corrected.

To better handle and evaluate these two types of knowledge editing scenarios, we for the first time define the multimodal knowledge in a decomposed format consist of visual knowledge and textual knowledge. In this way, the misreading and misrecognition errors can be distinguished, and thereby be independently corrected by editing different knowledge components. The decomposition of multimodal knowledge also brings up another requirement **Consistency**. We believe that a knowledge editing method should always ensure the consistency of knowledge across different modalities. This property is the essential difference between the multimodal knowledge editing and uni-modal knowledge editing.

Following the decomposed definition of multimodal knowledge, we propose a multimodal knowledge editing benchmark emphasizing modality consistency (**MC-MKE**). MC-MKE consists of three subsets, corresponding to the three different formats of multimodal knowledge. Our benchmark aligns more closely with multimodal knowledge editing in real-life scenarios and can more systematically and comprehensively evaluate the performance of a multimodal knowledge editing method in a fine-grained manner.

We evaluate three of most renowned multimodal knowledge editing methods including fine-tuning, MEND(Mitchell et al., 2022) and IKE(Zheng et al., 2023) on the three subsets of different editing formats. We find that the performance of these methods is far from satisfaction on MC-MKE. None of them can achieve great performance on all three different editing formats, especially for the consistency metric. It is demonstrated that multimodal knowledge editing is still challenging and requires further exploration.

In summary, our contributions are as follows[1]:

- We first propose a decomposed definition of multimodal knowledge according to different multimodal knowledge error types.
- We present **MC-MKE**, a new multimodal knowledge editing benchmark that can evaluate Reliability, Locality, Generality, and Consistency of multimodal editing methods under different editing formats.
- We conduct experiments with various knowledge editing methods on **MC-MKE**. The results reveal the limitations of existing methods especially for modality consistency. **Different** from previous research, we find that editing the corresponding component sometimes yields better performance.

[1]Our code and data will be released to the community to facilitate future research.

2 Related Works

2.1 Knowledge Editing

Knowledge editing aims to provide efficient and lightweight solutions for updating knowledge in models (Zhu et al., 2020). Several benchmarks have been developed for this task, including COUNTERFACT (Meng et al., 2022) for counterfactual knowledge, MQuake (Zhong et al., 2023) for multi-hop knowledge, AToKE (Yin et al., 2024) for retaining old knowledge, and WIKIUPDATE (Wu et al., 2024) for unstructured knowledge.

These benchmarks primarily address language model editing, leaving multimodal model editing underexplored. To address this gap, Cheng et al. (2024) introduced the MMEdit benchmark based on Visual QA (Antol et al., 2015) and Image Captioning (Herdade et al., 2019). Wu et al. (2024) developed KEBench, which uses multimodal Knowledge Graphs (Liu et al., 2019) to evaluate vision knowledge editing. Additionally, MIKE (Li et al., 2024) focuses on fine-grained multimodal entity knowledge editing. However, as shown in Table 1, all previous work has neglected the organization of multimodal knowledge and lacked a more careful definition of multimodal knowledge editing, which is what our work focuses on.

2.2 Multimodal Models

Multimodal large language models have developed rapidly in recent years. BLIP-2 (Li et al., 2023b) apply Q-Former architecture to transform image input into LLMs input tokens. LLaVA(Liu et al., 2024b) and LLaVA-v1.5(Liu et al., 2024a) utilize linear layers or perceptrons to map the vision features into the inputs of LLMs. Through instruction tuning on BLIP2, InstructBLIP(Dai et al., 2024) gains the ability to follow the instructions on different tasks. Notably, MiniGPT-4(Zhu et al., 2023) and MiniGPT-v2(Chen et al., 2023) are also powerful LVLMs that exhibit strong performance across various vision-language tasks. There are many other MLLMs such as mPLUG-Owl(Ye et al., 2023), Otter(Li et al., 2023a) and Qwen-VL (Bai et al., 2023). Among all MLLMs, GPT-4V(OpenAI, 2023) is the most powerful one now. We select some of these MLLMs on our research.

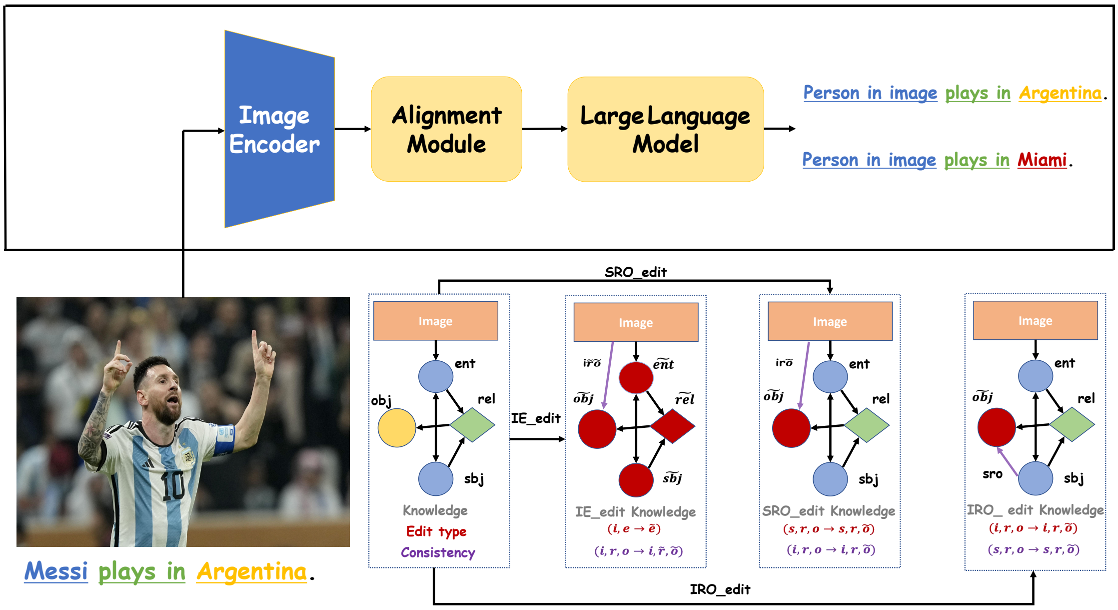


Figure 2: The upper represents editing different components of MLLMs. The bottom provides an overview of different editing formats. With an input image and its corresponding textual knowledge (s, r, o), we show three different editing formats. Although the final output is the same, the edited multimodal knowledge differs when editing its visual or textual knowledge, and the consistency property is also different given different edit inputs.

3 Multimodal Knowledge Editing

3.1 Definition of Multimodal Knowledge

There are two types of knowledge updating scenarios, namely misrecognition and misread. The misrecognition scenario refers to the model's recognized entity from the image being incorrect and needs correcting. So we define a visual knowledge (i, e) related to this scenario, where i represents an image and e represents the recognized entity.

In contrast, the misread scenario focuses on the model that successfully recognizes the entity in the image but fails to provide the correct object within the context of the entity and relation. In this scenario the corresponding textual knowledge (s, r, o) is related.

Therefore, we believe a piece of multimodal knowledge can be represented as a combination of visual knowledge (i, e) from image recognition of an entity and textual knowledge triplet (s, r, o) about the recognized entity. We finally decompose a piece of multimodal knowledge as:

$$K(i, e, s, r, o) = (i, e) \times_{e=s} (s, r, o) \quad (1)$$

Further, in many cross-modal datasets, most instances represent knowledge in the final form of (i, r, o) because there is no need to explicitly mention the intermediate entity e (and s). So another combined form of multimodal knowledge can be denoted as:

$$(i, e) \times_{e=s} (s, r, o) = (i, r, o) \quad (2)$$

In summary, $(i, e), (s, r, o), (i, r, o)$ are three types of knowledge involved in multimodal knowledge editing. However, regardless of the type of knowledge being edited, a good editing method must ensure that the consistency of multimodal knowledge is maintained after editing the corresponding type of knowledge.

3.2 Definition of MMEdit

We define three different edit formats, IE_edit, SRO_edit, IRO_edit.

IE_edit IE_edit is focused on editing knowledge related to image-to-entity recognition, denoted as (i, e). If we want to edit the model's recognition of an entity in an image, we input the image and modify the model's entity output for this image to a new output , which is $(i, e \to \tilde{e})$.

SRO_edit SRO_edit is focused on editing specific textual knowledge triplets (s, r, o). When we know the exact way to edit the corresponding textual knowledge tuple $(s, r, o \to \tilde{o})$, we do not need to find the corresponding multimodal data pair. Instead, we can directly use textual editing way. To ensure consistency in input format of multimodal language models, we use a black image as visual

Benchmark	Edit_formats			Fine-grained	Edit_requirements				
	IE	SRO	IRO		Reliability	Locality	Generality	Portability	Consistency
MMEdit	✗	✗	✓	✗	✓	✓	✓	✗	✗
KEBench	✓	✗	✗	✓	✓	✓	✓	✓	✗
MIKE	✗	✗	✓	✓	✓	✓	✓	✗	✗
MC-MKE	✓	✓	✓	✓	✓	✓	✓	✓	✓

Table 1: Comparisons of current multimodal knowledge editing benchmarks, MMEdit (Cheng et al., 2024), KEBench (Wu et al., 2024) and MIKE (Li et al., 2024). IE, SRO, IRO represent different editing formats. ✓ and ✗ mean whether the benchmark can provide data of corresponding editing format. In Fine-grained, ✓ means that the corresponding benchmark is constructed based on fine-grained entity information, while ✗ means that the benchmark is constructed around multimodal task data. Edit_requirements are the properties we expect from a good editing method. ✓ and ✗ indicate whether the benchmark contains the ability to test these properties of editing methods.

input. Subsequent experiments in appendix A have shown that when using questions generated from textual knowledge as input, the type of input image does not significantly impact the accuracy of the answers. In this case, the model's textual input is the same as the textual knowledge editing task.

IRO_edit In many multimodal datasets, numerous examples do not present the complete construction information of an instance of multimodal knowledge. We only possess the final multimodal data (i, r, o) and may not be able to accurately decompose it into the corresponding visual knowledge and textual knowledge. Nonetheless, we still need to edit such multimodal knowledge. Even though we may not explicitly identify the corresponding visual knowledge and textual knowledge, an effective method should implicitly understand and update the corresponding knowledge.

Therefore, we hope that a good multimodal knowledge editing method can maintain consistency, even when editing with the final multimodal knowledge input. Theoretically, modifying only $(i, r, o \rightarrow \tilde{o})$ should lead to consistency, whether through $(i, e \rightarrow \tilde{e})$ or $(s, r, o \rightarrow \tilde{o})$. However, there is an issue that there could be many non-unique e'. Our dataset provides automatically generated reasons to determine it is a modification of (s, r, o). A good editing method should automatically use the provided information to determine that the modification should be implemented on the corresponding textual knowledge triplet in IRO_edit of our benchmark.

3.3 Requirements of MMEdit Method

Consistency Consistency means that a piece of multimodal knowledge is answered consistently across different modalities after multimodal knowledge editing. In IE_edit, if we modify the corresponding visual knowledge $(i, e \rightarrow \tilde{e})$, consistency means that the corresponding multimodal knowledge should also change as: $(i, \tilde{r}, \tilde{o}) = (i, e \rightarrow \tilde{e}) \times_{\tilde{e}=\tilde{s}} (\tilde{s}, \tilde{r}, \tilde{o})$. In SRO_edit, if we modify the corresponding textual knowledge $(s, r, o \rightarrow \tilde{o})$ while keeping the visual knowledge unchanged, the corresponding multimodal knowledge will also be modified to $(i, e) \times_{e=s} (s, r, o \rightarrow \tilde{o}) = (i, r, o \rightarrow \tilde{o})$. In IRO_edit, due to the reasons mentioned above, our dataset provides information that allows the corresponding multimodal knowledge to change as follows: $(i, r, o \rightarrow \tilde{o}) = (i, e) \times_{e=s} (s, r, o \rightarrow \tilde{o})$. The definition of portability is similar to consistency in IE_edit. However, our consistency also includes situation on the SRO_edit and IRO_edit directions.

The property of consistency imposes higher demands on the multimodal knowledge editing method, requiring that the edited knowledge remains unified across different modalities in the multimodal model.

Reliability Reliability requirement of multimodal knowledge editing refers to the success rate of edits under the corresponding editing format.

Locality Locality means that multimodal editing should not affect unrelated knowledge when editing the corresponding knowledge.

Generality Generality means that after a piece of multimodal knowledge is edited, the model should not only respond to edited output under the format used for editing. It needs to provide correct edited responses under various generalizations, such as rephrased textual input or different images of the same entity.

4 MC-MKE Benchmark Construction

Since pure textual knowledge editing datasets are constructed from textual knowledge triplets (s, r, o) and contain editing information $(s, r, o \to \tilde{o})$, we opt for using the textual knowledge editing dataset MQuAKE as the starting point to construct our multimodal knowledge editing dataset MC-MKE. MQuAKE, as a text knowledge editing dataset, contains knowledge triplets and related editing information. Each instance in MQuAKE corresponds to a textual knowledge triplet and its textual editing information.

4.1 Data Selection

Unlike previous editing datasets, we performed filtering in three directions step by step on the original MQuAKE dataset D_{raw} to achieve a high-quality dataset.

First, we filter the data using a completely black image paired with generated questions. We selected data that our MLLMs could correctly answer. This step ensures that all the edited knowledge is originally known by the model to make sure we are "editing" instead of "learning". The filtered dataset is referred to as D_{filter_1}.

From D_{filter_1}, we obtain related images from Google, of the subject s in the textual knowledge triplets (s, r, o). We then used ChatGPT to generate fine-grained entity categories for these subjects and construct image queries using specific templates. If the subject in the image could be correctly recognized by all MLLMs, the data is then retained. This step ensures that all entities in our dataset are known by the models. This constitutes the dataset D_{filter_2}.

Finally, we replaced the subject in the questions generated from (s, r) with "the {category} in the picture" where {category} is the entity type previously generated by ChatGPT, seen in appendix D. If the combined question can be correctly answered by all models, the data is then retained. This step ensures the original multimodal knowledge consistency. The final retained multimodal knowledge $(i, r, o) = (i, e) \times (s, r, o)$ constitutes our knowledge editing source dataset D_{orig}.

More details are shown in appendix C.

4.2 Dataset Construction

Editing Dataset Construction For a multimodal knowledge $(i, r, o) = (i, e) \times_{e=s} (s, r, o)$ in our filtered multimodal knowledge source dataset D_{orig}, we sequentially construct editing data under different editing formats. For IE_edit, our editing inputs consist of images and automaticaly generated questions. We choose to use an entity $\tilde{e}$ of the same category as the entity e as the editing target. For SRO_edit, our editing inputs consist of generated questions, with the editing target being the corresponding new knowledge $\tilde{o}$ given in MQuAKE dataset. We require that $\tilde{o}$ is of the same entity category as o. For IRO_edit, our editing input is constructed based on the input from SRO_edit, combined with entity types and templates. The target $\tilde{o}$ is chosen from the corresponding data in the SRO_edit editing dataset. more strict requirements can be seen in appendix C.

Reliability Dataset Construction Our Reliability metric is calculated as shown in the following formula. D_e is the editing dataset corresponding to the editing format. For each piece of multimodal knowledge $k = (i, e) \times (s, r, o)$ in D_e, $\tilde{k}$ is the corresponding edited knowledge. p_r is the multimodal input used for testing the Reliability of the corresponding editting format. t_r is the target reliability output after knowledge editing. F is the multimodal model, and $\theta_{k\tilde{k}}$ represents the parameters of the model after editing a multimodal knowledge $k \to \tilde{k}$.

$$\text{Score}_R = \mathbb{E}_{(k,\tilde{k},p_r,t_r)\sim D_e}\left[\mathbb{1}_{F(p_r;\theta_{k\tilde{k}})=t_r}\right] \quad (3)$$

Consistency Dataset Construction Our consistency knowledge editing test data requires constructing according to different editing formats. In IE_edit, consistency is defined as $(i, e \to \tilde{e}) \times_{\tilde{e}=\tilde{s}} (\tilde{s}, \tilde{r}, \tilde{o}) = (i, r, o) \to (i, \tilde{r}, \tilde{o})$. Therefore, we construct the input p_c corresponding to the multimodal knowledge $(i, \tilde{r}, \tilde{o})$. The edited model should output the corresponding $\tilde{o}$ for this input to ensure consistency. In SRO_edit, we will edit the corresponding textual knowledge triplet $(s, r, o \to \tilde{o})$, and then construct the input p_c for multimodal knowledge $(i, r, \tilde{o})$ based on definition of consistency to test whether the edited model can provide a consistent edited answer $\tilde{o}$. In IRO_edit, for each piece of knowledge (i, r, o), we find its corresponding textual knowledge (s, r, o). After editing the multimodal knowledge $(i, r, o \to \tilde{o})$, we will analyze whether the corresponding textual knowledge $(s, r, o \to \tilde{o})$ provides a consistent response.

The consistency score is shown in the following formula. p_c is the multimodal input, $\theta_{k\tilde{k}}$ is the edited parameters, t_c is the corresponding consistency output in different editing format. Others are

Model	Method	Score$_R$	Score$_L$	Score$_G^T$	Score$_G^M$	Score$_C$
InstructBLIP	FT(Vision)	89.57	0.34	24.10	90.30	38.07
	FT(LLM)	98.60	3.00	77.43	96.77	10.15
	MEND(Vision)	32.39	93.15	29.73	23.43	18.37
	MEND(LLM)	88.58	53.23	86.49	85.21	9.46
	IKE	68.26	/	76.33	/	49.05
MiniGPT-v2	FT(Vision)	96.08	2.07	94.52	54.02	10.79
	FT(LLM)	95.87	0.78	93.91	93.20	10.80
	MEND(Vision)	4.34	26.08	4.23	5.13	6.81
	MEND(LLM)	45.21	24.30	44.41	26.17	11.74
	IKE	47.50	/	68.76	/	26.41

Table 2: Experimental results on IE_edit data for three editing methods editing two different model components on two MLLMs.

the same as (3).

$$\text{Score}_C = \mathbb{E}_{(k,\tilde{k},p_c,t_c)\sim D_e}\left[\mathbb{1}_{F(p_c;\theta_{k\tilde{k}})=t_c}\right] \quad (4)$$

Locality Dataset Construction In the edited datasets for the three editing formats, we used data unrelated to the current editing format but of the same type as locality data. In IE_edit, we randomly selected visual information (i_{loc}, e_{loc}) different from the current entity in D_{orig} as locality data. In SRO_edit, we randomly selected data $(s_{loc}, r_{loc}, o_{loc})$ different from the current textual knowledge triplet (s, r, o) in D_{orig} as locality data. In IRO_edit, we randomly selected multimodal knowledge $(i, e) \times_{e=s} (s, r, o)$ where i, e, s, r, and o are all different in D_{orig} to form locality data $(i_{loc}, e_{loc}) \times_{e_{loc}=s_{loc}} (s_{loc}, r_{loc}, o_{loc})$.

The locality score is shown in the following formula. p_l is the multimodal input, $\theta_{k\tilde{k}}$ is the edited parameters, t_l is the corresponding locality output in different editing format.

$$\text{Score}_L = \mathbb{E}_{(k,\tilde{k},p_l)\sim D_e}\left[\mathbb{1}_{F(p_l;\theta_{k\tilde{k}})=F(p_l;\theta)}\right] \quad (5)$$

Generality Dataset Construction For the three forms of multimodal knowledge editing IE_edit, SRO_edit, and IRO_edit we constructed corresponding generalization test datasets from both image and text perspectives. For the image generalization dataset, we used CLIP to process the images previously crawled from the web. Then, we calculated the relevance between the images and entities using the CLIP model and selected the top 5 most relevant images as the test images for entity image generalization. For the text generalization dataset, we use ChatGPT to rewrite 5 variations of the textual input to serve as the test inputs for text generalization. The prompts can be seen in appendix C.

The generality score is shown in the following formula. p_g^T, p_g^M is the multimodal input for text, and image generalization testing, respectively. $\theta_{k\tilde{k}}$ is the edited parameters. t_g^T, t_g^M are the corresponding text, and image generality output, respectively, in different editing formats.

$$\text{Score}_G^T = \mathbb{E}_{(k,\tilde{k},p_g^T,t_g^T)\sim D_e}\left[\mathbb{1}_{F(p_g^T;\theta_{k\tilde{k}})=t_g^T}\right] \quad (6)$$

$$\text{Score}_G^M = \mathbb{E}_{(k,\tilde{k},p_g^M,t_g^M)\sim D_e}\left[\mathbb{1}_{F(p_g^M;\theta_{k\tilde{k}})=t_g^M}\right] \quad (7)$$

Construction details about multimodal input p and corresponding t can be seen in appendix C.

Benchmark statistics Eventually, we create MC-MKE, cosist of a total of 2884 pieces of knowledge across three different edit formats. The associated knowledge involves a large amount of entities and relations, indicating the diversity of MC-MKE. It also has an average of 18.11 answer aliases per sample, significantly reducing misjudgements of the exact match metrics. More details about dataset statistics are presented in Table 4.

5 Experiments

5.1 Multimodal Large Language Models

InstructBLIP InstructBLIP is a multimodal large language model that consists of three modules. Its multimodal alignment module consists of a Qformer structure and a linear layer network to connect its vision and large language model module. We use InstructBLIP equipped with Vicuna-7B (Chiang et al., 2023).

MiniGPT-v2 MiniGPT-v2 utilizes a linear projection layer as an alignment module to map visual features to LLM feature space. Compared with InstructBLIP, MiniGPT-v2 has a smaller alignment module but still more input visual features. We

Model	Method	Score_R	Score_L	Score_G^T	Score_C
InstructBLIP	FT(Vision)	91.75	4.23	17.84	87.57
	FT(LLM)	99.49	3.95	79.59	90.43
	MEND(Vision)	13.64	95.03	10.00	3.86
	MEND(LLM)	66.49	79.34	72.85	55.90
	IKE	81.06	94.18	55.87	73.73
MiniGPT-v2	FT(Vision)	82.48	2.36	81.38	1.93
	FT(LLM)	97.34	2.63	96.00	94.49
	MEND(Vision)	4.78	86.53	4.94	6.72
	MEND(LLM)	71.89	19.93	69.79	6.41
	IKE	38.59	58.10	24.78	26.37

Table 3: Experimental results on SRO_edit data for three editing methods editing two different model components on two MLLMs.

Edit format	IE_edit	SRO_edit	IRO_edit	All
#Data	920	982	982	2884
#Relation	28	30	30	30
#Entity	810	1041	1041	1424
#Alias(avg.)	20.46	17.02	17.02	18.11
#Image	2358	-	1311	2550
#Category	49	76	76	76

Table 4: The statistic of different subsets of MC-MKE. `#Entity` refers to the total number of entities appeared including s, o and e. `#Alias` refers to the number of answer aliases.

use MiniGPT-v2 equipped with Llama-2-Chat-7B (Touvron et al., 2023).

5.2 MMEdit Methods

There have been many language knowledge editing methods, while multi-modality knowledge editing methods have not been fully explored. Therefore, we select the following representative editing methods in multimodal knowledge editing according to the editing requirements of the task and the applicability of the methods.

Finetune Finetune is one of the most widely used and apparent method to improving or modifying the abilities of pre-trained models and is also generally used as a baseline for knowledge editing. Since one can select the model component to finetune, it is natural to explore the differences between finetuning different model components. We focus on finetuning two parts: the alignment module and the LLM component of an MLLM. For LLM component, we only finetune the last layer of the LLM.

MEND Model Editor Networks with Gradient Decomposition (MEND) (Mitchell et al., 2022) is a editor network mapping a single desired input-output knowledge pair to the corresponding parameter update of the original model. Specifically, the input-output knowledge pair provides a standard fine-tuning gradient as a starting point for editing updates. Then MEND directly transforms the gradient to a better parameter update ensuring both generality and locality.

IKE In-Context Knowledge Editing (IKE) (Zheng et al., 2023) enables knowledge editing by incorporating demonstration examples within the input data to update and acquire new factual knowledge without the requirement of further training. Considering the instruction-following ability of the MLLM and the limitation on the number of input images, we choose to implement the zero-shot model of IKE. More experimental details can be seen in appendix B.

5.3 Results & Analysis

Pros and Cons of Different Editing Methods We observe that across all editing formats and model modules, no existing editing method perfectly meets our editing requirements.

The Finetune method is characterized by high Reliability, demonstrating good Generality and Consistency when editing the LLM part in SRO_edit and IRO_edit. However, its Locality is very low, meaning it has a significant impact on unrelated knowledge.

MEND, although it also modifies the model's parameters, uses a meta-learning approach to control the model's changes to other unrelated knowledge. As shown from the results, MEND's Reliability is much lower than that of Finetune, but its Locality is extremely higher.

Model	Method	Score_R	Score_L	Score_G^T	Score_G^M	Score_C
InstructBLIP	FT(Vision)	84.83	2.75	34.25	85.07	76.37
	FT(LLM)	91.56	4.86	81.50	91.34	86.33
	MEND(Vision)	24.13	85.88	33.11	19.20	5.49
	MEND(LLM)	70.57	64.78	86.00	72.05	50.50
	IKE	71.59	/	82.83	/	48.17
Mini-GPTv2	FT(Vision)	86.86	3.12	86.33	52.87	6.01
	FT(LLM)	95.82	2.40	94.30	94.65	87.35
	MEND(Vision)	6.61	50.06	6.06	7.61	2.74
	MEND(LLM)	63.13	46.55	59.79	38.77	5.09
	IKE	17.51	/	38.57	/	11.09

Table 5: Experimental results on IRO_edit data for three editing methods editing two different model components on two MLLMs.

As for IKE, since many MLLMs do not support in-context learning for image inputs, we do not test it for Locality and M-Generality in IE_edit and IRO_edit. Since it relies on in-context learning, this method is inherently sensitive to prompts. Different types of editing formats correspond to different prompts, and different models have varying degrees of sensitivity to prompts, resulting in significant fluctuations in all of IKE's metrics. Overall, IKE performs better on InstructBLIP, achieving the highest Locality in SRO_edit. It also shows high Consistency on InstructBLIP, achieving the highest Consistency metric in IE_edit, indicating that the model can infer the edited multimodal knowledge $(i, \tilde{r}, \tilde{o})$ based on context and the image.

Consistency On Different Editing Formats In SRO_edit and IRO_edit, the output of their corresponding Consistency test data matches the required edited output, with only the input information being different. Therefore, if the model can correctly understand the different formats of questions, it can accurately answer the Consistency questions. In these two editing formats, Finetune and IKE achieve high Consistency on InstructBLIP. However, on MiniGPT-v2, these methods only maintain some degree of Consistency. In these two editing formats, high Consistency without high Locality may come from overfitting. Thus, to accurately assess the Consistency property, we need to analyze the IE_edit format. On InstructBLIP, the Finetune method maintains high Consistency with high Reliability, indicating that the Finetune method is not solely overfitting since $\tilde{e} \neq \tilde{o}$. Only when a method achieves high Consistency across all three editing formats can its Consistency property be considered trustworthy. Considering the results across the two models, IKE shows best Consistency, managing to maintain high Consistency while ensuring a certain degree of Locality.

Editing Different Components Cheng et al. (2024) mentioned the visual module is harder to edit compared to the text module. In SRO_edit and IRO_edit, apart from Locality, the effectiveness of editing the visual module is much lower than that of editing the LLM module. For Locality, editing the alignment module using MEND has a smaller impact, possibly because MEND's parameter fitting is not as strong, and the editing knowledge in SRO_edit and IRO_edit is the textual knowledge (s, r, o) triplet. However, in IE_edit, although editing the LLM module generally yields higher Reliability and slightly higher Generality, in the case of InstructBLIP, editing the LLM module has a lower Consistency. This indicates that editing the LLM module often leads to overfitting, where the model outputs the edited knowledge regardless of the input information. In contrast, editing the Vision module, although resulting in lower other metrics, still maintains high Consistency. This shows that in IE_edit, editing the visual module can still ensure the Consistency of the corresponding knowledge.

6 Conclusion

We refine the definition of multimodal knowledge and introduce a new benchmark MC-MKE. We conduct experiments to analyze the effectiveness of several multimodal knowledge editing methods across different models, editing formats, and components. We find that these methods have limitations, and cannot achieve perfect performance on different editing formats. To maintain consistency, it may be better to edit the model components corresponding to the specific knowledge part.

Limitations

The main limitations of our work is related to limited knowledge editing methods and multimodal large language models. We only provide results on the two latest MLLMs, including InstructBLIP and MiniGPT-v2, leaving many others behind. As we study the latest MLLMs on knowledge editing methods which have not been discussed in prior work, we only analyze three knowledge editing methods, Finetune, MEND and IKE.

Ethical Considerations

CMKoBe is a synthetic dataset constructed by randomly modifying the factual knowledge triplets, rather than being crafted by humans. The data samples could accidentally involve context which is toxic or offensive in nature. ChatGPT is used for data annotation and assisting writing.

References

Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C. Lawrence Zitnick, and Devi Parikh. 2015. Vqa: Visual question answering. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 2425–2433.

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.

Jun Chen, Deyao Zhu, Xiaoqian Shen, Xiang Li, Zechun Liu, Pengchuan Zhang, Raghuraman Krishnamoorthi, Vikas Chandra, Yunyang Xiong, and Mohamed Elhoseiny. 2023. Minigpt-v2: large language model as a unified interface for vision-language multi-task learning. *CoRR*, abs/2310.09478.

Siyuan Cheng, Bozhong Tian, Qingbin Liu, Xi Chen, Yongheng Wang, Huajun Chen, and Ningyu Zhang. 2024. Can we edit multimodal large language models? *Preprint*, arXiv:2310.08475.

Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E. Gonzalez, Ion Stoica, and Eric P. Xing. 2023. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality.

Wenliang Dai, Junnan Li, Dongxu Li, Anthony Meng Huat Tiong, Junqi Zhao, Weisheng Wang, Boyang Li, Pascale N Fung, and Steven Hoi. 2024. Instructblip: Towards general-purpose vision-language models with instruction tuning. *Advances in Neural Information Processing Systems*, 36.

Simao Herdade, Armin Kappeler, Kofi Boakye, and Joao Soares. 2019. Image captioning: Transforming objects into words. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Han Huang, Haitian Zhong, Qiang Liu, Shu Wu, Liang Wang, and Tieniu Tan. 2024. Kebench: A benchmark on knowledge editing for large vision-language models. *Preprint*, arXiv:2403.07350.

Bo Li, Yuanhan Zhang, Liangyu Chen, Jinghao Wang, Jingkang Yang, and Ziwei Liu. 2023a. Otter: A multi-modal model with in-context instruction tuning. *Preprint*, arXiv:2305.03726.

Jiaqi Li, Miaozeng Du, Chuanyi Zhang, Yongrui Chen, Nan Hu, Guilin Qi, Haiyun Jiang, Siyuan Cheng, and Bozhong Tian. 2024. Mike: A new benchmark for fine-grained multimodal entity knowledge editing. *Preprint*, arXiv:2402.14835.

Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. 2023b. 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.

Tsung-Yi Lin, Michael Maire, Serge Belongie, Lubomir Bourdev, Ross Girshick, James Hays, Pietro Perona, Deva Ramanan, C. Lawrence Zitnick, and Piotr Dollár. 2015. Microsoft coco: Common objects in context. *Preprint*, arXiv:1405.0312.

Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. 2024a. Improved baselines with visual instruction tuning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 26296–26306.

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

Ye Liu, Hui Li, Alberto Garcia-Duran, Mathias Niepert, Daniel Onoro-Rubio, and David S. Rosenblum. 2019. Mmkg: Multi-modal knowledge graphs. *Preprint*, arXiv:1903.05485.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. 2022. Locating and editing factual knowledge in GPT. *CoRR*, abs/2202.05262.

Eric Mitchell, Charles Lin, Antoine Bosselut, Chelsea Finn, and Christopher D. Manning. 2022. Fast model editing at scale. *Preprint*, arXiv:2110.11309.

OpenAI. 2023. Gpt-4 technical report. *Preprint*, arXiv:2303.08774.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. Llama: Open and efficient foundation language models. *Preprint*, arXiv:2302.13971.

Xiaobao Wu, Liangming Pan, William Yang Wang, and Anh Tuan Luu. 2024. Updating language models with unstructured facts: Towards practical knowledge editing. *Preprint*, arXiv:2402.18909.

Qinghao Ye, Haiyang Xu, Guohai Xu, Jiabo Ye, Ming Yan, Yiyang Zhou, Junyang Wang, Anwen Hu, Pengcheng Shi, Yaya Shi, et al. 2023. mplug-owl: Modularization empowers large language models with multimodality. *arXiv preprint arXiv:2304.14178*.

Xunjian Yin, Jin Jiang, Liming Yang, and Xiaojun Wan. 2024. History matters: Temporal knowledge editing in large language model. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 19413–19421.

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

Zexuan Zhong, Zhengxuan Wu, Christopher D. Manning, Christopher Potts, and Danqi Chen. 2023. Mquake: Assessing knowledge editing in language models via multi-hop questions. *Preprint*, arXiv:2305.14795.

Chen Zhu, Ankit Singh Rawat, Manzil Zaheer, Srinadh Bhojanapalli, Daliang Li, Felix Yu, and Sanjiv Kumar. 2020. Modifying memories in transformer models. *Preprint*, arXiv:2012.00363.

Deyao Zhu, Jun Chen, Xiaoqian Shen, Xiang Li, and Mohamed Elhoseiny. 2023. Minigpt-4: Enhancing vision-language understanding with advanced large language models. *CoRR*, abs/2304.10592.

A Pre-experiments

SRO_edit focuses on editing a textual knowledge triplets (s, r, o), inherently requiring no additional visual inputs. But to align with the standard input format of MLLMs, we input a black image as the visual placeholder. In this section, we present an preliminary experiment to explore different choices of the input visual images including black images, white images and random noise. The results of InstructBLIP with these three types of images on SRO_edit are 95.11, 96.53 and 94.70 respectively. It is shown that these uninformative images barely have influence on the results.

B Experiment Details

Finetuning Details We list the hyper-parameters used for finetuning in Table 6. MiniGPT-v2 and InstructBLIP share the same hyper-parameters.

Learning Rate	5e-4
Steps	16
Optimizer	AdamW
Weight Decay	0.05

Table 6: Hyper-Parameters used for finetuning.

MEND Details Training process of MEND requires additional training data specific to the underlying model. Following (Mitchell et al., 2022), we construct an edit dataset and a locality dataset for both InstructBLIP and MiniGPT-v2. We leverage the data filtered in Section 4.1 as the edit dataset, sharing identical distribution with MC-MKE. Since both InstructBLIP and MiniGPT-v2 leverage MS COCO(Lin et al., 2015) for pretraining, we include it as the locality dataset. We search for three important hyper-parameters c_{loc}, c_{edit} and learning rate on each experimental setting for ten times. We found that MEND is very sensitive to hyperparameters, especially when the target module is small (e.g. the MEND(Vision) setting in our main experiment).

C Data Details

Entity Alias To facilitate entity evaluation, we collect alias of entities for all answers from the original dataset D_{raw}. However, since we will edit some of the subject entities, we also used alias data from Wiki as a supplement to construct the final entity alias library. All of our matching is performed with entities and their corresponding aliases.

Edit input Construction Details We choose to use an entity $\tilde{e}$ of the same category as the entity e and we require that the corresponding textual knowledge triplet $(\tilde{s}, \tilde{r}, \tilde{o})$, which $\tilde{s} = \tilde{e}$ exists in D_{filter_1}.

Locality Construction Details We ensure that these selected entities differ from those of the current knowledge. Formally, the knowledge $K_{loc}(i', e', s', r', o')$ for locality test of knowledge $K(i, e, s, r, o)$ must satisfy the condition $i' \neq i, e' \neq e, s' \neq s, r' \neq r, o' \neq o$. We randomly sample five pieces of knowledge to serve as the locality test data.

D Prompts

We designed specific prompts and instructions for GPT-3.5-turbo-16k to rephrase the textual input for the text generalization dataset and generate fine-grained entity types, as shown in Table 7 and Table

Prompts and Instructions
You are a helpful assistant. Please rephrase the following original text with 10 different and diverse expressions, maintaining exactly the same meanings. Note that you must not add any additional information and not delete or lose any information of the original text. Original Text: {source} 5 Rephrased Texts:

Table 7: Prompts and instructions used for rephrasing the textual input for the text generalization dataset.

Prompts and Instructions
You are a powerful fine-grained entity category generator. User will give the name of entity, and you will help answer the fine-grained categoty of the entity. The answer is the categoty only. There are some examples: Given entity Cameroon, a possible answer should be "country". Given entity David Beckham, a possible answer should be "person". Given entity The Great Gatsby, a possible answer should be "book". Given entity Producers' Showcase, a possible answer should be "TV show". Given entity Lady Madonna, a possible answer should be "song". Given entity Cox Enterprises, a possible answer should be "company". The given entity is { }, a possible answer is:

Table 8: Prompts and instructions used for generating fine-grained entity types.

8, respectively.

We provide editing and testing inputs of different types of multimodal knowledge editing in Table 9, Table 10 and Table 11.

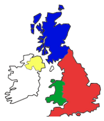
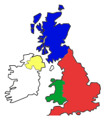
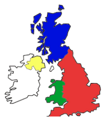
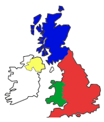
Input	Visual Inputs	Textual Inputs
Edit input		Question: The country in the picture is $\tilde{e}$: Lithuania
p_r		Question: The country in the picture is t_r: Lithuania Alias: Lietuva, Lietuvos Respublika, ...
p_c		Question: The capital of the country in the picture is t_c: Vilnius Alias: Vilnia, Vilna, Wilno, ...
p_l	ESPN	Question: Which TV channel is shown in the picture? t_l: ESPN Alias: Entertainment and Sports Programming Network
p_g^M		Question: The country in the picture is t_g^M: Lithuania Alias: Lietuva, Lietuvos Respublika, ...
p_g^T		Question: Can you tell me which country is depicted in the image? t_g^T: Lithuania Alias: Lietuva, Lietuvos Respublika, ...

Table 9: IE_edit multimodal input examples.

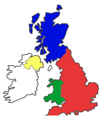
Input	Visual Inputs	Textual Inputs
Edit input	/	Question: The capital of United Kingdom is $\tilde{o}$: Rupnagar
p_r	/	Question: The capital of United Kingdom is t_r: Rupnagar Alias: Rupar, Ropar
p_c		Question: The capital of the country in the picture is t_c: Rupnagar Alias: Rupar, Ropar
p_l	/	Question: What is the country of citizenship of Warren Buffett? t_l: United States of America Alias: the United States, the United States of America, ...
p_g^T	/	Question: What is the capital of United Kingdom? t_g^T: Rupnagar Alias: Rupar, Ropar

Table 10: SRO_edit multimodal input examples.

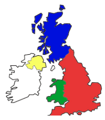
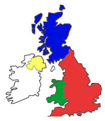
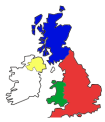
Input	Visual Inputs	Textual Inputs
Edit input		Question: As a result of World War III, the country in the picture moves its capital. The capital of the country in the picture is $\tilde{o}$: Rupnagar
p_r		Question: The capital of the country in the picture is t_r: Rupnagar Alias: Rupar, Ropar
p_c	/	Question: The capital of United Kingdom is t_c: Rupnagar Alias: Rupar, Ropar
p_l		Question: Which TV channel is shown in the picture? t_l: English Alias: English language, ...
p_g^M		Question: The capital of the country in the picture is t_g^M: Rupnagar Alias: Rupar, Ropar
p_g^T		Question: Can you tell me the capital of the country shown in the picture? t_g^T: Rupnagar Alias: Rupar, Ropar

Table 11: IRO_edit multimodal input examples.