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Multimodal large language models (MLLMs) are prone to non-factual or outdated knowledge issues, highlighting the importance of knowledge editing. Many benchmark has been proposed for researching multimodal knowledge editing. However, previous benchmarks focus on limited scenarios due to the lack of rigorous definition of multimodal knowledge. To better evaluate multimodal knowledge editing, we propose a decomposed definition of multimodal knowledge. Following the decomposed definition of multimodal knowledge, we introduce three scenarios and a novel requirement modality consistency. We construct MC-MKE, a fine-grained **M**ultimodal **K**nowledge **E**diting benchmark emphasizing **M**odality **C**onsistency through strict data selection. We evaluate several 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 However, these models struggle with the challenge that the knowledge stored within them could be inaccurate or outdated. Knowledge editing is aimed to solve the problem. 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 evaluation datasets overlook a key difference between multimodal knowledge and textual knowledge, which led them to ignore an additional requirements for multimodal knowledge editing. Specifically, a multimodal knowledge can be seen as a combination

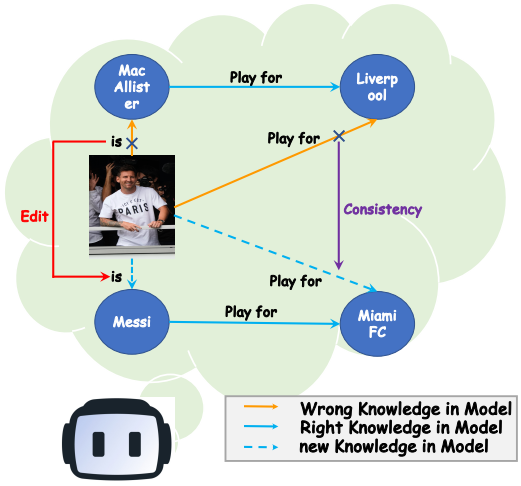


Figure 1: An example of multimodal knowledge editing. Knowledge editing corrects the knowledge, allowing it to accurately recognize the person as Messi instead of Mac Allister. At the same time, to ensure the consistency of multimodal knowledge, the edited model should also correctly understand that the person in the image plays for Miami FC instead of Liverpool.

of a visual knowledge part linking an image to the corresponding entity and a textual knowledge part related to the entity. To better handle and evaluate multimodal knowledge editing scenarios, we define multimodal knowledge in a decomposed format consisting of visual knowledge and textual knowledge in multimodal knowledge editing task. The decomposition of multimodal knowledge brings up the extra requirement modality **Consistency**.

Editing the knowledge requires ensuring consistency across the corresponding visual, textual and multimodal knowledge. For example, as shown in Figure 1, the wrong visual knowledge in model is (image of Messi, Mac Allister). Incorrect visual knowledge combined with textual knowledge (Mac Allister, play for, Liverpool) can lead to wrong multimodal knowledge. Knowledge editing needs to correct the corresponding visual knowledge so that the model can successfully recognize the person in the image as Messi. At the same time, it

is essential to ensure that the related multimodal knowledge is also consistently updated, meaning the corresponding multimodal knowledge should be corrected to (image of Messi, play for, Miami). We believe that a knowledge editing method should always ensure the consistency of knowledge across different modalities. This property is the essential difference between 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 three different scenarios of multimodal knowledge editing. Our benchmark provides various scenarios for multimodal knowledge editing and can more systematically and comprehensively evaluate the performance of a multimodal knowledge editing method in a fine-grained manner on Reliability, Locality, Generality and **Consistency** aspect.

We evaluate four of the most renowned multimodal knowledge editing methods including fine-tuning, MEND (Mitchell et al., 2022a), IKE (Zheng et al., 2023), and SERAC (Mitchell et al., 2022b) on the three subsets of different editing scenarios. 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 introduce a decomposed definition of multimodal knowledge in multimodal knowledge editing task. We introduce three scenarios and a novel requirement Consistency based on the definition.
- We present **MC-MKE**, a new multimodal knowledge editing benchmark with 112k train samples and 44k test samples that can evaluate multimodal editing methods on various properties under three different editing scenarios. The large-scale dataset ensure the quality of the test results and also provide abundant data resources for developing multimodal knowledge editing.
- We conduct experiments with various knowledge editing methods on **MC-MKE** under different scenarios. The results reveal the limitations of existing methods, especially for **Consistency**. We found that **editing the model component** corresponding to **type of edit knowledge** can obtain better results of Consistency.

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

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). Huang et al. (2024) developed VLKEB, 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 rigorous 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

Benchmark	Fine-grained	Edit_scenarios			Edit_requirements			
		IE	SRO	IRO	Reliability	Locality	Generality	Consistency
MMEdit	✗	✗	✗	✓	✓	✓	✓	✗
VLKEB	✓	✓	✗	✗	✓	✓	✓	✗
MIKE	✓	✓	✗	✗	✓	✓	✓	✗
MC-MKE	✓	✓	✓	✓	✓	✓	✓	✓

Table 1: Comparisons of current multimodal knowledge editing benchmarks, MMEdit (Cheng et al., 2024), VLKEB (Huang et al., 2024) and MIKE (Li et al., 2024). IE, SRO, and IRO represent different editing scenarios. ✓ and ✗ mean whether the benchmark can provide data of corresponding editing scenario. 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.

most powerful one now. We select some of these MLLMs for our research.

3 Multimodal Knowledge Editing

3.1 Definition of Multimodal Knowledge

We believe a piece of multimodal knowledge can be represented as a combination of visual knowledge from image recognition of an entity and textual knowledge triplet about the recognized entity. We use (i, e) and (s, r, o) to represent visual knowledge and textual knowledge, separately. 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 multimodal knowledge in the 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. A specific example can be seen in Figure 2, where the relevant relation and entities of the knowledge are highlighted in the sentence with corresponding colors.

3.2 Requirements of MMEdit Method

Consistency Consistency means that after any component (e, s, r, o) in multimodal knowledge 2 is edited, such a equation still holds like 3, where $\tilde{e}, \tilde{s}, \tilde{r}, \tilde{o}$ correspond to the possibly edited knowledge.

$$(i, \tilde{e}) \times_{\tilde{e}=\tilde{s}} (\tilde{s}, \tilde{r}, \tilde{o}) = (i, \tilde{r}, \tilde{o}) \quad (3)$$

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 output the edited knowledge under the exact input 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.

3.3 Edit scenarios of MMEdit

As shown in the example in Figure 2, we define three different edit scenarios: IE_edit, SRO_edit, and IRO_edit.

IE_edit IE_edit focuses on editing knowledge related to image-to-entity recognition, denoted as $(i, e \rightarrow \tilde{e})$. If we want to edit the model's recognition of an entity in an image, we input the image and modify the model's recognition output for this image to a new output (e.g. telling the model the person in Figure 2 is Messi rather than Mac Allister).

In IE_edit, after perfoming $(i, e \rightarrow \tilde{e})$, consistency means that, following Eq (3), the edited multimodal knowledge should be $(i, \tilde{r}, \tilde{o}) = (i, \tilde{e}) \times_{\tilde{e}=\tilde{s}} (\tilde{s}, \tilde{r}, \tilde{o})$, with $(i, \tilde{r}, \tilde{o})$ being the consistency knowledge to be checked. Using the same example, after the player in Figure 2 is edited from Mac Allister to Messi, the club which player in the image plays for should be changed from Liverpool to Miami FC correspondingly.

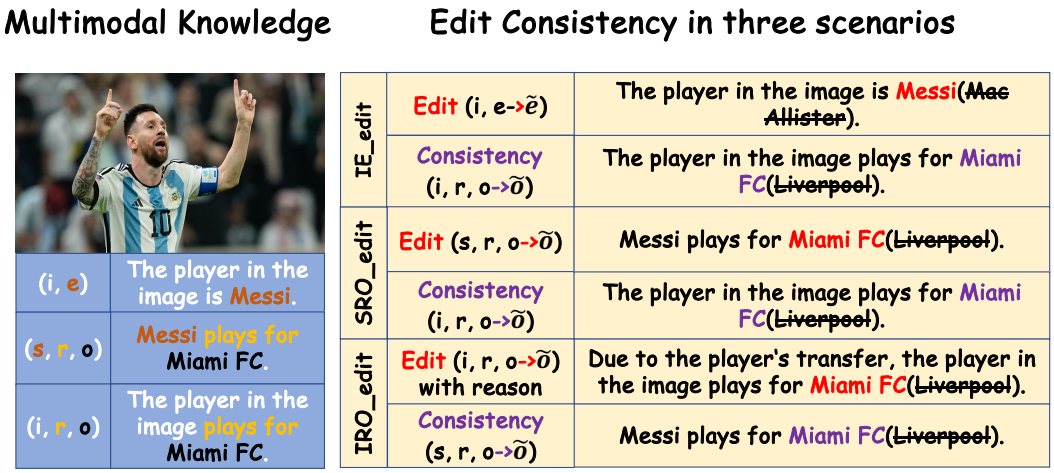


Figure 2: The left represents an example of visual knowledge (i, e), textual knowledge (s, r, o) and the corresponding multimodal knowledge (i, r, o). The right provides an overview of Consistency on different editing scenarios. The red represents the editing operation, while the purple indicates the requirement to maintain consistency in the corresponding scenario.

SRO_edit SRO_edit focuses on editing specific textual knowledge triplets $(s, r, o \rightarrow \tilde{o})$ without requiring image information, e.g., directly telling the model Messi plays for Miami FC instead of Liverpool. However, to unify the input format of multimodal large language models, we use a black image as visual input in SRO_edit (Subsequent experiments in Appendix A show that when using questions generated from textual knowledge as input, the type of input image does not significantly impact the accuracy of the answers as long as the image does not contain relative content.).

In SRO_edit, after performing $(s, r, o \rightarrow \tilde{o})$, consistency means that, following Eq 3, the edited multimodal knowledge should be $(i, e) \times_{e=s} (s, r, \tilde{o}) = (i, r, \tilde{o})$, with $(i, r, \tilde{o})$ being the consistency knowledge to be checked. Using the same example in Figure 2, after the club Messi plays for is edited from Liverpool to Miami FC, the club the player in the image plays for should also be changed from Liverpool to Miami FC correspondingly.

IRO_edit Many multimodal datasets and tasks only possess the final multimodal data (i, r, o) and may not contain its decomposed items (i, e) and (s, r, o), indicating that directly editing (i, r, o) is also a potential knowledge editing scenario. However, when $(i, r, o \rightarrow \tilde{o})$ is edited, we are actually **implicitly** editing its decomposed (i, e) or (s, r, o) knowledge. For example, we can actually perform either an implicit IE_edit $(i, e \rightarrow \tilde{e}) \times_{\tilde{e}=\tilde{s}} (\tilde{s}, r, \tilde{o}) = (i, r, o \rightarrow \tilde{o})$ or an implicit SRO_edit $(i, e) \times_{e=s} (s, r, o \rightarrow \tilde{o}) = (i, r, o \rightarrow \tilde{o})$ to achieve this target.

Therefore, a unique requirement in IRO_edit is that, even though the corresponding visual knowledge or textual knowledge is not explicitly identified, an effective editing method should implicitly understand and update which piece of knowledge should be edited, such as through utilizing reasons of this editing. For example in Figure 2, we are telling the model the person in the image plays for Miami FC instead of Liverpool due to player transfer, which indicates that the entity recognized in the image remains unchanged but the corresponding textual knowledge should be changed. An effective knowledge editing method should correctly comprehend this reason perform the correct implicit editing (in this case, the model should actually perform SRO_edit).

Theoretically, either $(i, e \rightarrow \tilde{e})$ or $(s, r, o \rightarrow \tilde{o})$ can be possible implicit ways of performing IRO_edit. However, there could be many non-unique $\tilde{e}$ which satisfy this requirements when IRO_edit is performed through $(i, e \rightarrow \tilde{e})$, so we only consider IRO_edit implicitly performed through $(s, r, o \rightarrow \tilde{o})$ in our research and our dataset provides automatically generated reasons to ensure this.

Therefore, the consistency in IRO_edit means that after $(i, r, o \rightarrow \tilde{o})$ is performed, following Eq 3, the edited multimodal knowledge should still be $(i, e) \times_{e=s} (s, r, o \rightarrow \tilde{o}) = (i, r, o \rightarrow \tilde{o})$, but this time with $(s, r, \tilde{o})$ being the consistency knowledge to be checked. Still taking Figure 2 as an example, after telling the model the person in the image plays for Miami FC instead of Liverpool, the club Messi plays for should also be modified to Miami FC from Liverpool.

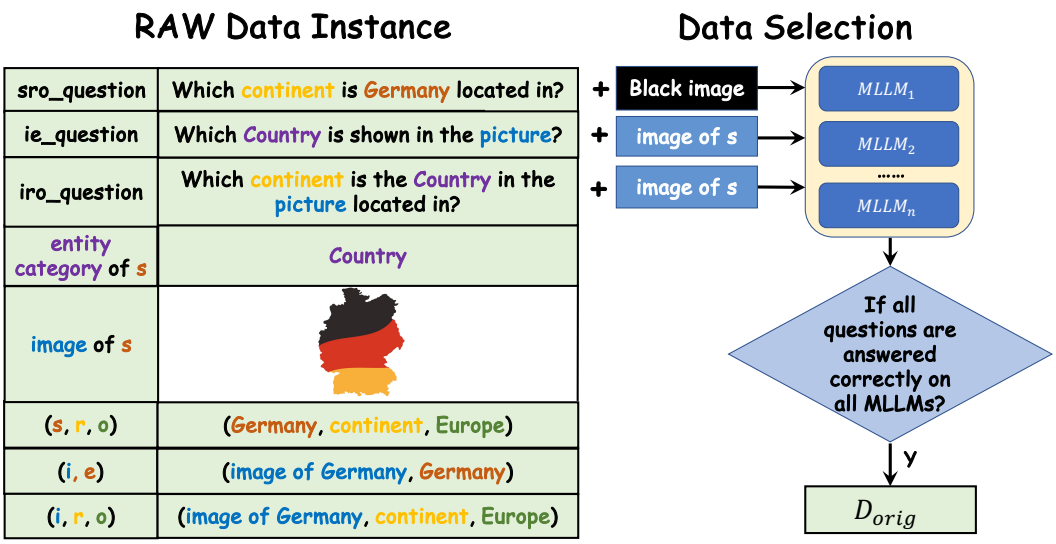


Figure 3: The left is the raw data we constructed from MQuAKE, where the sro_question and (s, r, o) triples are extracted from the MQuAKE dataset. The image of s is crawled and filtered from Google. We generate the entity categories and constructed ie_question and iro_question. The right is the data selection process, only the instance where all three questions can be successfully answered by all tested MLLMs(n=3), is retained in the original data D_{orig} for knowledge editing.

4 MC-MKE Benchmark Construction

Since pure textual knowledge editing datasets are constructed from textual knowledge triplets (s, r, o) and contain editing information $(s, r, o \rightarrow \tilde{o})$, we apply the textual knowledge editing dataset MQuAKE (Zhong et al., 2023) as the starting point to construct our multimodal knowledge editing dataset MC-MKE. MQuAKE, as a text knowledge editing dataset, contains knowledge triplets, related editing information and textual questions as test input. Each instance in MQuAKE corresponds to a textual knowledge triplet and its textual editing information.

4.1 Data Selection

Different from previous editing datasets, we performed strict data selection from the original MQuAKE dataset to achieve a high-quality dataset, to meet the requirement for rigorous consistency evaluation.

Based on the previous definition of multimodal knowledge, as seen in Eq (2), in order to rigorously evaluate the impact of knowledge editing on multimodal knowledge, we require that the model must be able to successfully answer the corresponding questions in all three scenarios for the original multimodal knowledge before editing. Otherwise, for example, even if the model successfully edits the visual knowledge $(i, e \rightarrow \tilde{e})$ and maintains consistency, but its textual knowledge $(\tilde{s}(\tilde{e}), r, \tilde{o})$ is incorrect, it will not be able to infer the multimodal knowledge $(i, r, \tilde{o})$. The specific process for constructing and filtering the raw data is shown in Figure 3. We first extract the textual knowledge triples (s, r, o) and their corresponding sro_question from the MQuAKE dataset. Based on experiment A, we use textual questions with black images as inputs. Next, we retrieve images of s from Google and apply the CLIP to filter the images, while employing the GPT to generate the entity category of s. We then construct the corresponding ie_question and iro_question to test whether the model correctly understand the corresponding (i, e) and (i, r, o) knowledge. We selected LLaVA-v1.5, InstructBLIP, and MiniGPT-v2 as the MLLMs for testing. Only the data where all three questions are successfully answered by all tested MLLMs will be retained as the raw data D_{orig}.

More details about data selection and generation quality assessment can be found in Appendix C.

4.2 Dataset Construction

Reliability Data Construction For multimodal knowledge in our filtered dataset D_{orig}, we sequentially construct editing data under different editing scenarios. For IE_edit, our editing inputs consist of images and automatically generated textual inputs. We choose to use an entity $\tilde{e}$ of the same category as the entity e as the editing target. Additionally, we require that D_{orig} contains the corresponding $(\tilde{s}(\tilde{e}), \tilde{r}, \tilde{o})$ data. If this condition is not guaran-

teed, the corresponding $\tilde{e}$ cannot be used as the edit target because consistency cannot be evaluated. For SRO_edit, our editing inputs consist of textual questions with black image, with the editing target being the corresponding new knowledge $\tilde{o}$ given in MQuAKE dataset. We also require that $\tilde{o}$ is of the same category as o. For IRO_edit, our editing input is constructed based on the input from SRO_edit, combined with entity categories 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. A statement which is constructed with the updated knowledge (as shown in the left part of Fig 2) will serve as the editing input according to different scenarios. When editing knowledge on IRO_edit scenario, we will add the possible reason generated from GPT into statement according to sec 3.3. The input of Reliability Data is then constructed from the edited knowledge.

Consistency Data Construction For an edited knowledge, we first find the corresponding consistency test knowledge, according to Sec 3.3 under different scenarios, and then construct the corresponding test input from the consistency test knowledge.

Locality Data Construction This part contains multiple locality inputs and corresponding locality outputs. For each edit, we randomly select different instances with knowledge unrelated to the current edited knowledge but of the same editing scenario as locality data to check whether this edit affects unrelated knowledge. Each edit corresponds to five test data instance for locality.

Generality Data Construction Generality data consists of Image Generality data and Text Generality data. In Image Generality, we construct test input with different images of the corresponding entity for each edit. We crawl similar images from the web and use CLIP to choose the most similar ones. Each edit corresponds to five test data instance for Image Generality. In Text Generality, we first construct test input from the edited knowledge and then generate different paraphrases with GPT as final test input. Each edit corresponds to five test data instance for Text Generality.

4.3 Benchmark statistics

We create MC-MKE consisting of a training set with 111904 samples and a test set with 44118 samples. Methods such as SERAC can apply training set to adjust their configuration. The test set consists of a total of 2884 pieces of knowledge across three different edit formats. The associated knowledge involves a large number of entities and relations, indicating the diversity of MC-MKE. It also has an average of 18.11 answer aliases per sample, significantly reducing misjudgments of the exact match metrics. Dataset statistics are presented in Table 2. More details and examples about our dataset can be found in Appendix C and D.

	Edit format	IE_edit	SRO_edit	IRO_edit	All
Train	#Data	3544	5968	5968	15480
	#Relation	37	30	30	37
	#Entity	3544	5230	5230	5407
	#Alias(avg.)	14.18	13.62	13.62	13.75
	#Image	21264	-	20790	22134
	#Category	142	342	342	343
	#Input Samples	28352	35808	47744	111904
Test	#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
	#Input Samples	15640	11784	16694	44118

Table 2: The statistic of different subsets of MC-MKE. `#Data` refers to the number of knowledge entries. `#Relation` refers to the types of relation on related knowledge.`#Entity` refers to the total number of entities appeared including s, o and e. `#Alias` refers to the number of answer aliases. `#Image` refers to the number of images. `#Input Samples` refers to the total number of test inputs.

5 Experiments

5.1 MMEdit Methods

There have been many single-modal (text) knowledge editing methods for , while multi-modal knowledge editing methods have not been fully explored. Therefore, we select the following representative editing methods including Finetuning, MEND (Mitchell et al., 2022a), IKE (Zheng et al., 2023) and SERAC (Mitchell et al., 2022b) in single-modal knowledge editing for evaluation following previous setting(Cheng et al., 2024). More details of these editing methods can be seen in Appendix B.

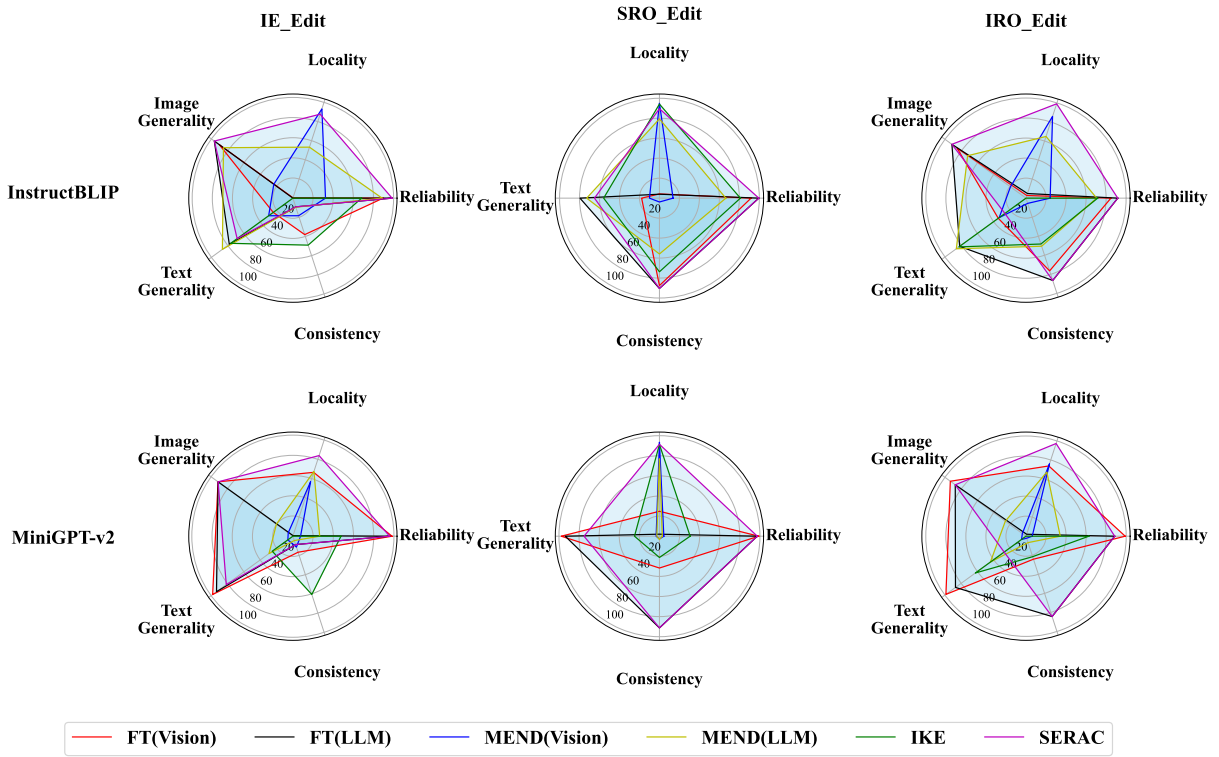


Figure 4: Results on **IE_edit**,**SRO_edit** and **IRO_edit** for tested editing methods on two MLLMs. Since models do not support multiple image inputs, we use 0 for Locality and Image Generality in IKE on IE_edit and IRO_edit. Detailed results can be found in appendix E.

5.2 Results & Analysis

Consistency of Existing Methods Based on the experimental results in Figure 4, we found that in IE_edit scenario, most knowledge editing methods perform poorly in terms of consistency across both models. In the SRO_edit and IRO_edit scenarios, the consistency performance is relatively higher. However, in SRO_edit and IRO_edit, the output of their corresponding consistency output matches the required edited output, with only the input information being different according to Sec 3.3. In these two editing scenarios, high Consistency without high Locality may come from overfitting. Only when a method achieves high Consistency across all three editing scenarios can its Consistency property be considered trustworthy.

According to Figure 4, the FT(Vision) maintains high Consistency among all training-based methods on InstructBLIP, indicating that the FT(Vision) is not solely overfitting to obtain Consistency under IE_edit scenario. Overall, IKE shows good Consistency while maintaining a certain degree of Locality. However, even IKE shows unsatisfactory consistency performance on the IE_edit scenario, and its consistency performance on MiniGPT-v2 on SRO_edit and IRO_edit scenario is even worse. This indicates that the current tested methods ignore Consistency during development, resulting in their inability to maintain high consistency across all scenarios.

Editing Methods on Different Scenarios There are some findings regarding different scenarios according to Figure 4. Some knowledge editing methods are sensitive to different scenarios on some aspects, while others are not. Specifically, the MEND knowledge editing method exhibits consistent characteristics across different models and scenarios, with similar shapes in the radar charts for various tests. It demonstrates high locality across all testing environments. However, its performance in reliability, generality, and consistency is poor. Overall, in the IE_edit scenario, MEND's reliability typically performs better. This may be because MEND conduct editing through mapping the corresponding changed knowledge to the corresponding parameter changes, and the mapping of (i, e) knowledge is relatively more straightforward. MEND may be easier to map the corresponding (i, e) knowledge changes to the corresponding parameter changes. w IKE places the edited knowledge into the input context, requiring the model to answer the questions based on the context. The variation in results is

caused by the different reasoning abilities required by the model in different scenarios. Additionally, the inherent reasoning capabilities of different models also affect the variation in IKE results. Overall, the performance of the IKE method is relatively unstable.

The FT method directly fits the edited input by training the specified parameters. Its consistency varies significantly across different scenarios. Considering its generally low locality in most cases, this may be caused by overfitting, as mentioned earlier. Its reliability and generality are relatively high across most scenarios, indicating that the FT method can successfully fit the corresponding edited knowledge. However, the ability of this method to protect other knowledge and maintain consistency still needs improvement.

SERAC performs well on most of the experiments, but its consistency remains low in the IE_edit scenario. We believe that, although the classifier SERAC applys can effectively distinguish between inputs related to edited knowledge and those that are not, it still cannot directly improve consistency. Even if the classifier identifies the need to use the counterfactual model to answer questions in the consistency test, the ability to respond to the consistency test still depends on the counterfactual model itself, which is obtained from the fine-tuning strategy. What's more, the performance of SERAC relies heavily on the classifier performance, whether the classifier can correctly identify the appropriate model for the given input. We find that in the Text Generality of IRO_edit, the classifier of SERAC often fails to properly classify the inputs for text generality, leading to selecting the wrong model and thereby reducing performance.

Editing Different Components Cheng et al. (2024) mentioned the visual module is harder to edit compared to the text module. Based on our experimental results, this point holds true in some experiments. For MEND, meta-learning requires predicting network changes corresponding to the knowledge edits, and editing the visual module to output the edited knowledge is more challenging. As a result, in most cases, using MEND(Vision) tends to result in lower Reliability according to Figure 4. However, while the MEND approach does help prevent the modification of irrelevant knowledge to some extent, editing the LLM module with MEND still often achieves lower locality as shown in Figure 4. Across the three datasets, FT(Vision) often achieves reliability similar to FT(LLM). On MiniGPT-v2, FT(Vision) results in higher locality.

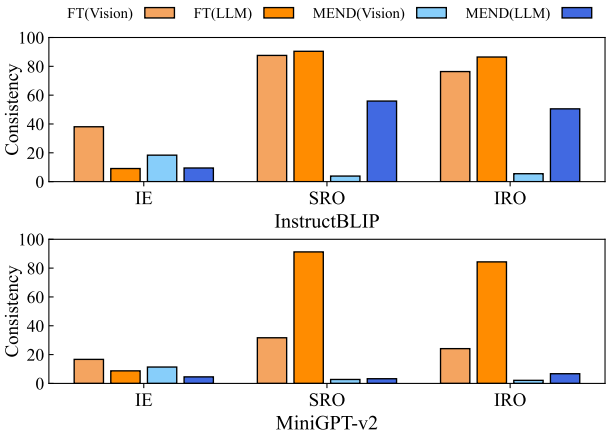


Figure 5: The Result of Consistency of FT and MEND when editing different component(Vision or LLM) of InstructBLIP and MiniGPT-v2 on three different scenarios.

Apart from previous difference, we also found that editing the corresponding part based on the edit scenario could yield better results in certain aspects according to Figure 5. From the consistency aspect, for the IE_edit data, the edited knowledge is primarily visual knowledge. Whether using the FT or MEND method, editing in the Vision module achieves better consistency. For the SRO and IRO scenarios, the edited knowledge is primarily textual, and in this case, both the FT and MEND methods, particularly in the LLM component of the model, achieve better consistency results. These results hold true for the tested models, InstructBlip and MiniGPT-v2. This may suggest that different types of knowledge are more closely related to the corresponding parts in MLLMs. Therefore, we believe that in the future, studying the appropriate editing methods for different types of knowledge should be an important direction.

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 scenarios, and components. We find that these methods have limitations, and cannot achieve perfect performance on different editing scenarios. 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 are related to limited knowledge editing methods and multimodal large language models. We only provide results on MLLMs with 7B checkpoint. We were unable to test larger checkpoints, due to resource constraints.As we study the latest MLLMs on four knowledge editing methods which have not been discussed in prior work, we need to implement them from scratch. We end up implement four knowledge editing methods, Finetuning, MEND, IKE and SERAC.

Ethical Considerations

MC-MKE: 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, Tao Yu, Qiang Liu, Shu Wu, Liang Wang, and Tieniu Tan. 2024. Vlkeb: A large vision-language model knowledge editing benchmark. In *Advances in Neural Information Processing Systems*, volume 37, pages 9257–9280. Curran Associates, Inc.

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. 2022a. Fast model editing at scale. *Preprint*, arXiv:2110.11309.

Eric Mitchell, Charles Lin, Antoine Bosselut, Christopher D Manning, and Chelsea Finn. 2022b. Memory-based model editing at scale. In *Proceedings of the*

39th International Conference on Machine Learning, volume 162 of *Proceedings of Machine Learning Research*, pages 15817–15831. PMLR.

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 a preliminary experiment to explore different choices of the input visual images including black images, white images and random noise. The accuracy 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 Finetuning is one of the most widely used and apparent methods for 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 the LLM component, we only finetune the last layer. We list the hyper-parameters used for finetuning in Table 3. MiniGPT-v2 and InstructBLIP share the same hyper-parameters.

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

Table 3: Hyper-Parameters used for finetuning.

MEND Details Model Editor Networks with Gradient Decomposition (MEND) (Mitchell et al., 2022a) is an 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. Training process of MEND requires additional training data specific to the underlying model. Following (Mitchell et al., 2022a), 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 training 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).

IKE Details 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 limitation on the number of input images, we choose to implement the zero-shot version of IKE.

SERAC Details SERAC(Mitchell et al., 2022b) proposes a memory-based editing approach. The approach consists of a classifier and a counterfactual model. The classifier chooses whether to use the counterfactual model or not based on the relation between the given input and edit memory.

Since our tasks are multimodal, we use a neural network trained on the training set as the classifier. The neural network consists of a CLIP feature extraction layer and an MLP classification layer. We set the learning rate of the classification layer to 0.0005. Since consistency requires the model to have reasoning abilities, we opted to continue using the large model as the counterfactual model. Specifically, we employ a large model with its LLM part fine-tuned on the edited knowledge as the counterfactual model.

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

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_{orig}.

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.

Entity Category Generation Evaluation We employ ChatGPT to generate the category of a given entity. To verify the quality of categories generated by ChatGPT, we randomly sampled 200 items and invited two annotators to independently verify whether the entities mentioned in these items matched their respective categories. The average agreement between the annotators was 98%, with a consistency rate of 97%, indicating that the generated entity categories are highly reliable. An example of a generated entity category is: "google" : "company".

Training set Construction Except for not undergoing the original problem filtering, the construction of the train data is similar to that of the test set. We utilize some of the filtered data to construct training set. For the filtered data which are not in D_{orig}, we directly apply the question in MQuAKE dataset as text generality textual input, and use the images from google as image generality visual input.

Rephrase Generation Evaluation We employ ChatGPT to generate Generality data. To verify the quality of rephrases generated by ChatGPT, we randomly sampled 100 items each associated with 4 paraphrased sentences and asked two annotators to independently assess the quality of each paraphrased sentence, marking them as 0 for bad quality and 1 for good quality. The average scores for the 400 paraphrase results were 0.9675, respectively, with an agreement of 98%, demonstrating that the quality of our paraphrases is sufficiently reliable. An example of paraphrased sentence is: Origin : "Who performed Folsom Prison Blues?" Rephrase : "Who was the performer of Folsom Prison Blues?"

D Prompts

We designed specific prompts and instructions for GPT-3.5-turbo-16k to rephrase the textual input for

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 4: 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 5: Prompts and instructions used for generating fine-grained entity types.

the text generalization dataset and generate fine-grained entity types, as shown in Table 4 and Table 5, respectively.

We provide editing and testing inputs of different types of multimodal knowledge editing in Table 6, Table 7 and Table 8.

E Detailed Results

We present detailed experiment results in Table 9, 10, 11 corresponding to Figure 4.

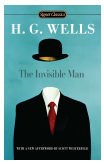
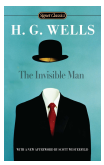
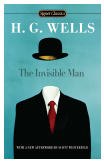
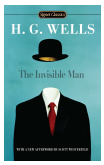
Input	Visual Inputs	Textual Inputs
Edit input		Question: The book in the picture is $\tilde{e}$: The Pilgrim's Progress
p_r		Question: The book in the picture is t_r: The Pilgrim's Progress Alias: Pilgrim's Progress, Land of Beulah, ...
p_c		Question: The book in the picture was written in the language of t_c: English Alias: en, eng, English language, ...
p_l		Question: Which TV channel is shown in the picture? t_l: ESPN Alias: Entertainment and Sports Programming Network
p_g^M		Question: The book in the picture is t_g^M: The Pilgrim's Progress Alias: Pilgrim's Progress, Land of Beulah, ...
p_g^T		Question: Which book is shown in the picture? t_g^T: The Pilgrim's Progress Alias: Pilgrim's Progress, Land of Beulah, ...

Table 6: IE_edit multimodal input examples.

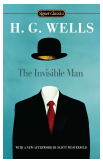
Input	Visual Inputs	Textual Inputs
Edit input	/	Question: Invisible Man was written in the language of $\tilde{o}$: Sanskrit
p_r	/	Question: Invisible Man was written in the language of t_r: Sanskrit Alias: Samskrta, Samskrtam, Sanskrit language, ...
p_c		Question: The book in the picture was written in the language of t_c: Sanskrit Alias: Samskrta, Samskrtam, Sanskrit language, ...
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: Which language was Invisible Man written in? t_g^T: Sanskrit Alias: Samskrta, Samskrtam, Sanskrit language, ...

Table 7: SRO_edit multimodal input examples.

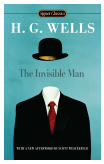
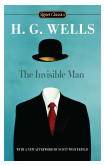
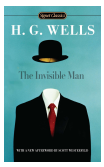
Input	Visual Inputs	Textual Inputs
Edit input		Question: The official work language of the book in the picture has changed. The book in the picture was written in the language of $\tilde{o}$: Sanskrit
p_r		Question: The book in the picture was written in the language of t_r: Sanskrit Alias: Samskrta, Samskrtam, Sanskrit language, ...
p_c	/	Question: Invisible Man was written in the language of t_c: Sanskrit Alias: Samskrta, Samskrtam, Sanskrit language, ...
p_l		Question: Who is the developer of the operating system in the picture? t_l: Microsoft Alias: MSFT, Microsoft Corp., ...
p_g^M		Question: The book in the picture was written in the language of t_g^M: Sanskrit Alias: Samskrta, Samskrtam, Sanskrit language, ...
p_g^T		Question: Which language was the book in the picture written in? t_g^T: Sanskrit Alias: Samskrta, Samskrtam, Sanskrit language, ...

Table 8: IRO_edit multimodal input examples.

Model	Edit Method	Reliability	Locality	Image Generality	Text Generality	Consistency
InstructBLIP	FT(Vision)	89.57	0.34	90.30	24.10	38.07
	FT(LLM)	**98.48**	0.03	**96.41**	78.04	9.09
	MEND(Vision)	32.39	**93.15**	23.43	29.73	18.37
	MEND(LLM)	88.58	53.23	85.21	**86.49**	9.46
	IKE	68.26	/	/	76.33	**49.05**
	SERAC	**98.48**	87.65	**96.41**	68.41	9.09
MiniGPT-v2	FT(Vision)	**98.04**	66.43	**91.52**	**98.13**	16.67
	FT(LLM)	95.76	0.59	91.48	93.41	8.71
	MEND(Vision)	7.57	56.73	5.69	6.17	11.36
	MEND(LLM)	26.52	67.34	20.17	29.19	4.54
	IKE	47.61	/	/	25.24	**60.60**
	SERAC	95.76	**83.85**	91.48	81.48	8.71

Table 9: Experimental results on **IE_edit** data for four editing methods editing two different model components on two MLLMs. The highest value is highlighted in **bold**.

Model	Edit Method	Reliability	Locality	Text Generality	Consistency
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
	SERAC	**99.49**	89.53	65.05	**90.43**
MiniGPT-v2	FT(Vision)	**98.78**	24.81	**97.68**	31.67
	FT(LLM)	97.35	2.01	93.73	**91.24**
	MEND(Vision)	4.37	**93.50**	3.29	2.74
	MEND(LLM)	2.85	76.96	2.62	3.25
	IKE	30.55	91.26	24.83	21.18
	SERAC	97.35	91.43	75.05	**91.24**

Table 10: Experimental results on **SRO_edit** data for four editing methods editing two different model components on two MLLMs. The highest value is highlighted in **bold**.

Model	Edit Method	Reliability	Locality	Image Generality	Text Generality	Consistency
InstructBLIP	FT(Vision)	84.83	2.75	85.07	34.25	76.37
	FT(LLM)	**91.65**	4.85	**91.47**	81.87	**86.46**
	MEND(Vision)	24.13	85.88	19.20	33.11	5.49
	MEND(LLM)	70.57	64.78	72.05	**86.00**	50.50
	IKE	71.59	/	/	82.83	48.17
	SERAC	**91.65**	**99.06**	**91.47**	26.01	**86.46**
MiniGPT-v2	FT(Vision)	**98.98**	73.71	**93.32**	**98.78**	24.13
	FT(LLM)	88.49	2.04	87.25	86.99	**84.32**
	MEND(Vision)	6.21	76.00	4.52	5.45	2.13
	MEND(LLM)	34.21	67.31	25.49	43.91	6.72
	IKE	62.73	/	/	62.48	21.49
	SERAC	88.49	**97.25**	87.25	26.92	**84.32**

Table 11: Experimental results on **IRO_edit** data for four editing methods editing two different model components on two MLLMs. The highest value is highlighted in **bold**.