LPEdit: Locality-Preserving Knowledge Editing for MultiModal Large Language Models

Anonymous ACL submission

Abstract

Large Language Models (LLMs) and Vision Large Language Models (VLLMs) demonstrate impressive abilities in comprehending natural language and interpreting visual information, but they can also preserve outdated or incorrect information in both forms. Existing knowledge editing methods can efficiently update erroneous text information in LLMs, avoiding the need for full retraining. The locality in multimodal knowledge editing refers to editing that should affect only the targeted outputs while preserving the model's behavior on unrelated inputs in both textual and visual modalities. Existing methods often overlook this principle and do not explicitly design to preserve the consistency of responses on unrelated information. Here, we propose LPEdit, a novel method that leverages the null space projection on key layers to focus the editing on conveyed visual information without influencing unrelated knowledge. Experiments show that our method achieves strong performance across different models and datasets. Moreover, our work advances the understanding and development of locality in multimodal knowledge editing.

1 Introduction

LLMs store vast amounts of factual knowledge acquired from large-scale pretraining corpora (OpenAI, 2023; Touvron et al., 2023). However, these corpora often contain outdated or incorrect information, prompting growing interest in model editing techniques that can efficiently revise specific knowledge without full retraining (Sajjad et al., 2022). This line of work addresses the need for dynamic updates in deployed models while avoiding the high cost of retraining (Roberts et al., 2020; Petroni et al., 2019).

Recent model editing efforts have mainly focused on the text modality and can be broadly classified into two categories. The first directly modifies model parameters to embed new knowledge (Meng et al., 2022, 2023; Cao et al., 2021; Mitchell et al., 2022a; Jiang et al., 2024), while the second introduces external mechanisms (e.g., memory modules or adapters) without altering internal weights (Zheng et al., 2023; Mitchell et al., 2022c; Hartvigsen et al., 2023; Huang et al., 2023).

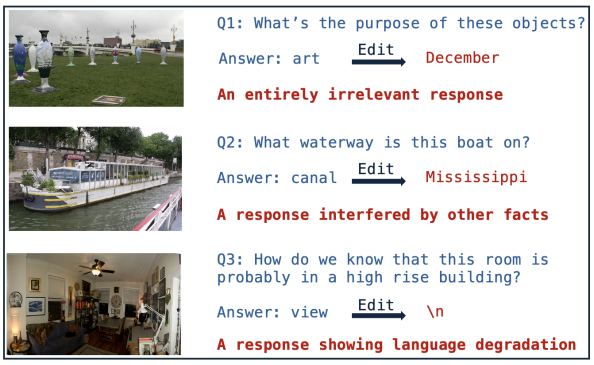

Figure 1: Multimodal knowledge editing leads to different types of incorrect responses on images and questions, from the EVQA dataset, unrelated to the editing target.

Editing VLLMs introduces unique challenges due to the complex interactions between visual and textual modalities, making it a relatively underexplored area. Prior work has extended text-based editing methods to VLLMs and proposed evaluation metrics (Cheng et al., 2023; Basu et al., 2024), but results suggest that such editors are often suboptimal, likely due to cross-modal dependencies beyond decoder weights. While recent methods have made notable progress in ensuring the correctness of edited answers, they often struggle to maintain the model's behavior on irrelevant samples. As a result, they may introduce unintended failures, such as off-topic responses, factual interference, or linguistic degradation (Gupta et al., 2024; Gu et al., 2024), shown in Figure 1. These limitations underscore the need for editing methods that not only correct target outputs but also maintain the model's overall reliability and behavior on unrelated inputs.

While methods using null space projection (Wang et al., 2021) to preserve original knowledge during editing have proven effective, they have not yet been applied to multimodal systems.

To achieve this, we introduce LPEdit, which adopt null space projection to ensure that the model's performance on unrelated inputs remains unaffected while correcting the target outputs. By applying null space projection to the MLP projection matrix, we constrain the parameter updates to occur only along directions without interfering with prior knowledge. Our method effectively guarantees that the edited model produces correct outputs for target inputs while maintaining stability on unrelated inputs. In this way, our approach not only corrects errors of editing outputs but also avoids introducing biases to irrelevant tasks or degrading the model's essential capabilities. Specifically, in multimodal tasks, we ensure that the interaction between visual and textual information is not disrupted by unnecessary interference.

Our method demonstrates strong performance across two distinct VLLM architectures, highlighting its generality and effectiveness. In particular, we observe consistent improvements in locality, the ability to constrain changes to the target region of knowledge without affecting unrelated outputs. We show that LPEdit enables effective editing while preserving the integrity of prior knowledge, marking a small step forward in the exploration of locality in multimodal knowledge editing.

2 Related work

Model editing has emerged as an essential research area focused on modifying the behavior of pre-trained large language models (LLMs) to integrate new knowledge or correct factual errors, all without the need for extensive retraining. In the text domain, knowledge editing has seen notable progress, with various methods successfully achieving precise updates while minimizing unintended changes to unrelated outputs. A more comprehensive review of both unimodal and multimodal model editing methods is provided in Appendix A.

Model Editing for VLLMs: While model editing has been widely studied for text-only LLMs, its extension to VLLMs remains limited. MMEdit (Cheng et al., 2023) first adapted LLM editing methods to MLLMs by constructing dedicated editing datasets and evaluation protocols. UniKE (Pan et al., 2024) is a unified multimodal editing framework that models knowledge as key-value memories and disentangles semantics and truthfulness to improve editing reliability. VisEdit (Chen et al., 2025) introduces an attribution-based editor that locates key visual layers and regions relevant to prompts for targeted VLLM editing. FGVEdit (Zeng et al., 2024) benchmark and MSCKE framework integrate multimodal cues to support fine-grained entity-level editing in images. ComprehendEdit (Ma et al., 2025)offers a comprehensive benchmark with eight tasks and two new metrics (KGI and KPI), along with HICE for balancing editing quality and preservation. MC-MKE benchmark (Zhang et al., 2024) focused on modality consistency by decomposing knowledge into visual and textual parts and defining three editing formats. MIKE (Li et al., 2024a) introduces a fine-grained multimodal entity editing dataset with over 1,000 entities and multi-step editing tasks to enhance efficiency and precision.

3 Method

3.1 Preliminary

For a VLLM $f_\theta \in \mathcal{F}$, given an edit sample (x_i^e, x_t^e, y^e) such that $f_\theta(x_i^e, x_t^e) \neq y^e$, with x_i^e and x_t^e representing the image and textual modal inputs, and o^e being the pre-edit target output. Given a VLLM editor $\mathcal{E}_{\text{edit}}(\cdot) : (f_\theta, x_i^e, x_t^e, y^e) \longrightarrow f'_\theta$, where $f'_\theta \in \mathcal{F}$ denotes the updated model, an effective multimodal knowledge editing method should satisfy the following three evaluation criteria.

Reliability evaluates whether the post-edit model can produce accurate outputs for the edited instances, where $\mathcal{D}_e$ is the set of edit samples and I denotes the indicator function:

$$E_{(x_i^e, x_t^e, y^e) \sim \mathcal{D}_{edit}} I\left\{f_{\theta_e}\left(x_i^e, x_t^e\right) = y^e\right\}$$

Generality measures the model's ability to adapt its predictions to variations semantically or visually related to the edited sample, encompassing both modal and textual generality; specifically, $\mathcal{D}_{tn}(x_t^e)$ and $\mathcal{D}_{in}(x_i^e)$ represent the textual and visual neighborhoods of the image x_i^e and prompt x_t^e, respectively.

$$E_{(x_i^e, x_t^e, y^e) \sim \mathcal{D}_e} E_{x_t^{tn} \sim \mathcal{D}_{tn}(x_t^e)} I\left\{f_{\theta_e}\left(x_i^e, x_t^{tn}\right) = y^e\right\}$$
$$E_{(x_i^e, x_t^e, y^e) \sim \mathcal{D}_e} E_{x_i^{in} \sim \mathcal{D}_{in}(x_i^e)} I\left\{f_{\theta_e}\left(x_i^{in}, x_t^e\right) = y^e\right\}$$

Locality requires the revised model to maintain consistent outputs with the original model on sam-

ples that are irrelevant to the edited instance, including both textual and visual locality:

$$E_{(x_i^e,x_t^e,o^e)\sim\mathcal{D}_e dit}\ E_{(x_t^{tu},o^{tu})\sim\mathcal{D}_{tu}(x_t^e)} I\left\{f_{\theta_e}(\emptyset,x_t^{tu})=f_\theta(\emptyset,x_t^{tu})\right\}$$

$$E_{(x_i^e,x_t^e,o^e)\sim\mathcal{D}_e dit}\ E_{(x_i^{iu},x_t^{iu},o^{iu})\sim\mathcal{D}_{iu}(x_i^e,x_t^e)} I\left\{f_{\theta_e}(x_i^{iu},x_t^{iu})=f_\theta(x_i^{iu},x_t^{iu})\right\}$$

$\mathcal{D}_{tu}(x_t^e)$ and $\mathcal{D}_{iu}(x_i^e,x_t^e)$ denote the sets of unrelated text-only and image samples, respectively.

3.2 Null Space Projection for Editing

To ensure that editing does not alter the model's responses to unrelated inputs, we constrain parameter updates to directions that are unlikely to interfere with existing representations. This is achieved by projecting the updates onto the approximate null space of the input representations at selected layers.

Given a layer input matrix $X \in R^{n\times d}$ where n denotes the number of input tokens and d is the feature dimension, we compute its uncentered covariance matrix $C = X^\top X$, which captures the raw correlations among input features without mean subtraction. Since not every covariance matrix possesses a strict null space in practice, we adopt a strategy by approximating the null space via singular value decomposition (SVD).

We perform SVD on C, yielding $U\Lambda U^\top$, where $U = [U_1, U_2]$ contains the left singular vectors, and $\Lambda = \text{diag}([\Lambda_1, \Lambda_2])$ is the diagonal matrix of singular values. The subspace spanned by U_1 corresponds to high-variance (informative) directions, while U_2 spans the directions associated with near-zero singular values. These latter directions define an approximate null space of the input covariance. To constrain the parameter update Δw to lie within this null space, we project it as:

$$\Delta w_{\text{null}} = U_2 U_2^\top \Delta w.$$

This projection ensures that the update does not alter the activations along directions that encode existing task knowledge. In our implementation, we apply this null space projection to the parameter updates of the MLP projection matrix in high-contribution layers, which are identified as particularly influential for model predictions. This ensures that the parameter updates occur only along the null space directions, preserving the model's core capabilities and stability of prior knowledge. As a result, this approach effectively enables multimodal knowledge editing while minimizing unintended side effects on unrelated inputs.

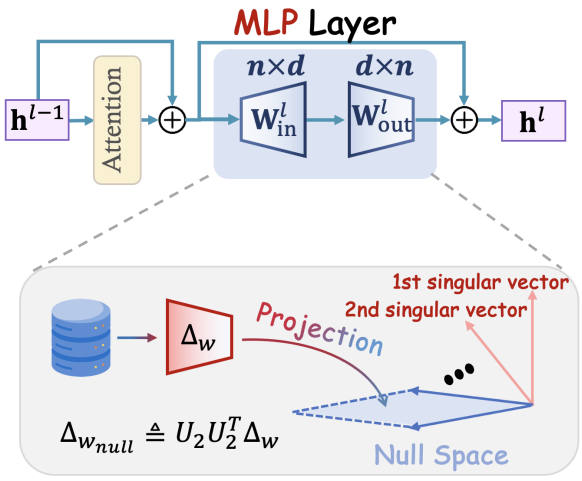

Figure 2: An illustration of our proposed method.

4 Experiments

4.1 Experiment setting

Datasets: In line with Cheng et al. (2023), we adopt EVQA (Editing Visual Question Answering) and E-IC (Editing Image Caption) as our benchmark datasets for evaluating editing performance.
VLLM Backbones: To ensure comprehensive evaluation in both model scale and architecture, we select two representative VLLMs for experimentation: BLIP2-OPT (2.7B) (Li et al., 2024b), LLaVA-V1.5 (7B) (Liu et al., 2023a).
Baselines: Since dedicated editing methods for VLLMs have not yet been proposed, existing approaches primarily adapt LLM editors for use in the multimodal setting (Cheng et al., 2023). We include several representative baselines in our evaluation: FTV (fine-tuning the visual encoder), TF-L (fine-tuning the final language model layer), IKE (Zheng et al., 2023), SERAC (Mitchell et al., 2022c), MEND (Mitchell et al., 2022a), TP (Huang et al., 2023), and LTE (Jiang et al., 2024). Full experimental details, including model configurations and training hyperparameters, are provided in Appendix B. Based on these settings, we systematically assess editing performance and further analyze the internal mechanisms of null space to confirm its effectiveness in boosting locality.

4.2 Analysis of Editing Performance

Table 1 presents the overall editing results. In the following, we provide a detailed experimental analysis from multiple perspectives.

In terms of method, LPEdit exhibits the strongest overall performance among all editing

Table 1: Performance comparison on E-VQA and E-IC benchmarks on BLIP2-OPT (2.7B) and LLaVA-V1.5 (7B). Rel., T-Gen., M-Gen., T-Loc., and M-Loc. denote Reliability, Textual Generality, Multimodal Generality, Textual Locality, and Multimodal Locality. Average is computed over the five metrics.

Model	Method	E-VQA						E-IC					
		Rel.	**T-Gen.**	**M-Gen.**	**T-Loc.**	**M-Loc.**	**Avg.**	**Rel.**	**T-Gen.**	**M-Gen.**	**T-Loc.**	**M-Loc.**	**Avg.**
BLIP2-OPT	FT-V	23.00	15.41	19.08	97.87	86.97	48.47	40.14	38.66	34.65	98.94	88.39	60.16
	FT-L	23.63	15.86	19.94	96.75	88.41	48.92	40.00	38.15	35.41	98.02	87.46	59.81
	IKE	97.31	89.80	90.17	12.38	1.77	58.35	95.38	76.48	81.16	12.67	2.05	53.55
	SERAC	89.25	90.41	88.47	100.00	0.31	73.69	92.59	93.79	89.68	100.00	0.45	75.30
	MEND	90.59	89.86	90.46	94.52	63.74	85.83	64.23	35.88	34.21	90.99	55.07	56.08
	TP	66.64	59.39	55.04	97.50	83.77	72.47	47.03	46.72	42.92	91.65	79.16	61.50
	LTE	95.78	96.18	95.15	93.09	83.56	92.75	94.50	93.76	92.66	93.22	86.70	92.17
	VisEdit	96.66	96.74	**97.39**	100.00	90.04	96.17	95.82	95.02	93.66	100.00	90.63	95.03
	LPE di t	**97.74**	**97.53**	96.86	**100.00**	**97.41**	**97.91**	**96.44**	**96.02**	**93.97**	**100.00**	**94.58**	**96.20**
LLaVA-V1.5	FT-V	29.92	28.19	25.87	98.32	89.97	54.45	52.02	50.85	46.23	98.30	91.12	67.70
	FT-L	30.06	29.03	25.89	98.54	90.66	54.84	51.91	50.25	46.98	97.59	93.78	68.10
	IKE	89.88	89.15	89.17	59.32	50.56	75.62	92.49	86.66	78.46	74.88	64.11	79.32
	SERAC	80.78	79.86	78.87	100.00	56.17	79.14	41.23	39.98	40.99	100.00	7.29	45.90
	MEND	90.06	90.52	90.52	89.43	80.11	88.13	91.90	92.43	90.81	89.46	85.44	90.01
	TP	47.35	46.97	43.05	93.62	89.30	64.06	57.86	56.23	54.28	63.35	87.04	63.75
	LTE	92.99	92.12	91.57	81.73	79.70	87.62	92.04	90.77	89.78	84.38	87.21	88.84
	VisEdit	95.12	**95.02**	93.85	100.00	93.97	95.59	95.06	94.19	93.12	100.00	94.74	95.42
	LPEdit	**95.15**	94.98	**93.92**	**100.00**	**98.39**	**96.49**	**95.23**	**94.50**	**93.43**	**100.00**	**97.41**	**96.11**

methods, particularly excelling in both locality metrics (T-Loc and M-Loc). VisEdit stands out as the most effective among existing baselines, leveraging precise visual-pathway manipulations to maintain strong performance across diverse evaluation dimensions. LTE also achieves competitive results through full-model fine-tuning, but shows limitations in maintaining stable performance on non-edited textual and visual outputs.

In terms of datasets, editing on E-VQA tends to perform better, as corrections typically involve a few key tokens. In contrast, E-IC requires full-sentence caption rewriting grounded in comprehensive image understanding, which poses greater challenges for maintaining locality.

In terms of models, most editors demonstrate greater stability on BLIP2-OPT, likely due to clearer modular separation between visual and language pathways, allowing local interventions to remain more contained. In contrast, LLaVA-V1.5 integrates visual features more deeply into the language decoder, making localized editing more difficult and leading to increased performance variation among editors. This highlights the importance of architectural compatibility in multimodal editing.

In terms of evaluation metrics, LPEdit achieves the highest performance on both textual and visual locality metrics, indicating strong consistency on unrelated samples across modalities. FT-V and VisEdit, which apply edits to the visual modality, naturally preserve the language generation pathway and maintain linguistic locality and overall output quality. In contrast, SERAC leverages a classifier to identify purely textual inputs, which helps maintain locality on the language side but fails for visual samples. Methods with lower overall performance often exhibit high locality at the expense of reliability and generality, suggesting that these objectives are difficult to satisfy simultaneously. In contrast, LPEdit demonstrates a more balanced performance: it significantly enhances locality while maintaining competitive reliability and generality.

5 Conclusions

We present LPEdit, a locality-preserving method using null-space projection for multimodal knowledge editing. By constraining parameter updates to directions that minimally affect unrelated knowledge, our method achieves accurate edits while preserving model behavior on non-target information. Experimental results across multiple models and tasks demonstrate its effectiveness and generalization ability. LPEdit is only a small step towards more general and reliable editing in multimodal systems, and locality-preserving capability is only the beginning. We hope this work encourages further research on knowledge editing techniques across diverse vision-and-language domains.

6 Limitations

Despite the effectiveness of LPEDIT, our study has several limitations. First, we evaluate our method on widely adopted VLLMs, but do not include the most recent models such as the latest LLaVA variants or Qwen2.5-VL, which may exhibit different cross-modal behaviors. Second, our experiments rely on datasets derived from COCO images, which have been extensively used in training and may already be memorized by some foundation models. As VLLMs continue to evolve, there is a growing need for new visual data and pretraining corpora that better reflect contemporary content and usage patterns. Third, although we introduce controlled variations in both visual and textual inputs, our evaluation remains grounded in two established benchmarks: EVQA and EIC. Recently proposed benchmarks target different aspects of multimodal reasoning, including new datasets, evaluation protocols, and task paradigms. Future work should explore whether LPEDIT generalizes well to these emerging settings and tasks.

7 Ethical Statement

This research follows ethical guidelines in both the collection and use of data and the use of open-source models. The datasets used in this study, including E-VQA and E-IC, are publicly available and were used in accordance with their respective licenses. We acknowledge that large-scale models may inherit and perpetuate biases present in training data, and we make efforts to minimize these biases by carefully curating the datasets and applying appropriate methods for evaluation. All experiments were conducted in compliance with ethical standards, ensuring that no personal or sensitive data was used in the analysis. We follow the usage protocols and licenses of the open-source models we build upon. We are committed to advancing research in a manner that promotes fairness, transparency, and accountability. This work was conducted during the author's internship at Bytedance, and the authors are required to adhere to the company's regulations. The authors are committed to ensuring that no proprietary data is leaked, and the codebase will only be made publicly available after undergoing a review process. The authors have already submitted the code for review, and it is currently under evaluation. We expect to release the specific code and related information in the next phase, following the approval process.

References

Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, Roman Ring, Eliza Rutherford, Serkan Cabi, Tengda Han, Zhitao Gong, Sina Samangooei, Marianne Monteiro, Jacob L. Menick, Sebastian Borgeaud, and 8 others. 2022. Flamingo: a visual language model for few-shot learning. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*.

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 2015, Santiago, Chile, December 7-13, 2015*, pages 2425–2433. IEEE Computer Society.

Samyadeep Basu, Martin Grayson, Cecily Morrison, Besmira Nushi, Soheil Feizi, and Daniela Massiceti. 2024. Understanding information storage and transfer in multi-modal large language models. In *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*.

Nicola De Cao, Wilker Aziz, and Ivan Titov. 2021. Editing factual knowledge in language models. In *EMNLP (1)*, pages 6491–6506.

Qizhou Chen, Taolin Zhang, Chengyu Wang, Xiaofeng He, Dakan Wang, and Tingting Liu. 2025. Attribution analysis meets model editing: Advancing knowledge correction in vision language models with visedit. In *AAAI-25, Sponsored by the Association for the Advancement of Artificial Intelligence, February 25 - March 4, 2025, Philadelphia, PA, USA*, pages 2168–2176. AAAI Press.

Xinlei Chen, Hao Fang, Tsung-Yi Lin, Ramakrishna Vedantam, Saurabh Gupta, Piotr Dollár, and C. Lawrence Zitnick. 2015. Microsoft COCO captions: Data collection and evaluation server. *CoRR*, abs/1504.00325.

Siyuan Cheng, Bozhong Tian, Qingbin Liu, Xi Chen, Yongheng Wang, Huajun Chen, and Ningyu Zhang. 2023. Can we edit multimodal large language models? In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*, pages 13877–13888. Association for Computational Linguistics.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. 2021. An image is worth 16x16 words: Transformers for image

recognition at scale. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.

Yifan Du, Zikang Liu, Junyi Li, and Wayne Xin Zhao. 2022a. A survey of vision-language pre-trained models. In *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI 2022, Vienna, Austria, 23-29 July 2022*, pages 5436–5443. ijcai.org.

Zhengxiao Du, Yujie Qian, Xiao Liu, Ming Ding, Jiezhong Qiu, Zhilin Yang, and Jie Tang. 2022b. GLM: general language model pretraining with autoregressive blank infilling. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022*, pages 320–335. Association for Computational Linguistics.

Junfeng Fang, Houcheng Jiang, Kun Wang, Yunshan Ma, Xiang Wang, Xiangnan He, and Tat seng Chua. 2025. Alphaedit: Null-space constrained knowledge editing for language models. *ICLR*.

Yash Goyal, Tejas Khot, Aishwarya Agrawal, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. 2019. Making the V in VQA matter: Elevating the role of image understanding in visual question answering. *Int. J. Comput. Vis.*, 127(4):398–414.

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

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

Tom Hartvigsen, Swami Sankaranarayanan, Hamid Palangi, Yoon Kim, and Marzyeh Ghassemi. 2023. Aging with GRACE: lifelong model editing with discrete key-value adaptors. In *NeurIPS*.

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

Yuxin Jiang, Yufei Wang, Chuhan Wu, Wanjun Zhong, Xingshan Zeng, Jiahui Gao, Liangyou Li, Xin Jiang, Lifeng Shang, Ruiming Tang, Qun Liu, and Wei Wang. 2024. Learning to edit: Aligning llms with knowledge editing. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2024, Bangkok, Thailand, August 11-16, 2024*, pages 4689–4705. Association for Computational Linguistics.

Jiaqi Li, Miaozeng Du, Chuanyi Zhang, Yongrui Chen, Nan Hu, Guilin Qi, Haiyun Jiang, Siyuan Cheng, and Bozhong Tian. 2024a. MIKE: A new benchmark for fine-grained multimodal entity knowledge editing. In *Findings of the Association for Computational Linguistics, ACL 2024, Bangkok, Thailand and virtual meeting, August 11-16, 2024*, pages 5018–5029. Association for Computational Linguistics.

Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. 2024b. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. *arXiv preprint arXiv:2301.12597*.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. 2023a. Visual instruction tuning. In *NeurIPS*.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. 2023b. Visual instruction tuning. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*.

Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. 2019. Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 13–23.

Yaohui Ma, Xiaopeng Hong, Shizhou Zhang, Huiyun Li, Zhilin Zhu, Wei Luo, and Zhiheng Ma. 2025. Comprehendedit: A comprehensive dataset and evaluation framework for multimodal knowledge editing. In *AAAI-25, Sponsored by the Association for the Advancement of Artificial Intelligence, February 25 - March 4, 2025, Philadelphia, PA, USA*, pages 19323–19331. AAAI Press.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. 2022. Locating and editing factual associations in GPT. In *NeurIPS*.

Kevin Meng, Arnab Sen Sharma, Alex J. Andonian, Yonatan Belinkov, and David Bau. 2023. Mass-editing memory in a transformer. In *ICLR*.

Eric Mitchell, Charles Lin, Antoine Bosselut, Chelsea Finn, and Christopher D. Manning. 2022a. Fast model editing at scale. In *ICLR*.

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

Eric Mitchell, Charles Lin, Antoine Bosselut, Christopher D. Manning, and Chelsea Finn. 2022c. Memory-based model editing at scale. In *ICML*, pages 15817–15831.

OpenAI. 2023. GPT-4 technical report. *CoRR*, abs/2303.08774.

Kaihang Pan, Zhaoyu Fan, Juncheng Li, Qifan Yu, Hao Fei, Siliang Tang, Richang Hong, Hanwang Zhang, and Qianru Sun. 2024. Towards unified multimodal editing with enhanced knowledge collaboration. In

Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024.

Fabio Petroni, Tim Rocktäschel, Sebastian Riedel, Patrick S. H. Lewis, Anton Bakhtin, Yuxiang Wu, and Alexander H. Miller. 2019. Language models as knowledge bases? In *EMNLP/IJCNLP (1)*, pages 2463–2473.

Adam Roberts, Colin Raffel, and Noam Shazeer. 2020. How much knowledge can you pack into the parameters of a language model? In *EMNLP (1)*, pages 5418–5426.

Hassan Sajjad, Nadir Durrani, and Fahim Dalvi. 2022. Neuron-level interpretation of deep NLP models: A survey. *Trans. Assoc. Comput. Linguistics*, 10:1285–1303.

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

Shipeng Wang, Xiaorong Li, Jian Sun, and Zongben Xu. 2021. Training networks in null space of feature covariance for continual learning. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, virtual, June 19-25, 2021*, pages 184–193. Computer Vision Foundation / IEEE.

Zhen Zeng, Leijiang Gu, Xun Yang, Zhangling Duan, Zenglin Shi, and Meng Wang. 2024. Visual-oriented fine-grained knowledge editing for multimodal large language models. *CoRR*, abs/2411.12790.

Jinjin Zhang, Qiuyu Huang, Junjie Liu, Xiefan Guo, and Di Huang. 2025. Diffusion-4k: Ultra-high-resolution image synthesis with latent diffusion models. *CoRR*, abs/2503.18352.

Junzhe Zhang, Huixuan Zhang, Xunjian Yin, Baizhou Huang, Xu Zhang, Xinyu Hu, and Xiaojun Wan. 2024. MC-MKE: A fine-grained multimodal knowledge editing benchmark emphasizing modality consistency. *CoRR*, abs/2406.13219.

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? *CoRR*, abs/2305.12740.

Deyao Zhu, Jun Chen, Xiaoqian Shen, Xiang Li, and Mohamed Elhoseiny. 2024. Minigpt-4: Enhancing vision-language understanding with advanced large language models. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net.

A Related Work

A.1 Vision Language Models

Recent advances in large language models (LLMs) have catalyzed the growing interest in integrating vision modalities into language systems to form vision language large language models (VLLMs). A typical architecture involves coupling a pre-trained visual encoder, most commonly a Vision Transformer (ViT)(Dosovitskiy et al., 2021), with a frozen or lightly tuned LLM decoder. These systems are trained in a two-stage pipeline. The first phase involves aligning the image features with the token space of the LLM via lightweight feedforward adapters or more structured modules, such as the resampler (Li et al., 2024b). The second stage involves task-specific fine-tuning across a broad range of multimodal tasks such as visual question answering (Antol et al., 2015) and caption correction dataset (Cheng et al., 2023), adapted to interactive vision-language scenarios.

VLLMs can be broadly categorized by their modality fusion strategies into Modal Deep Fusion (MDF) and Modal Early Fusion (MEF) architectures (Du et al., 2022a). MDF approaches, such as ViLBERT (Lu et al., 2019) and Flamingo (Alayrac et al., 2022), incorporate visual information into the intermediate layers of the LLM through cross-modal attention. In contrast, MEF methods project image features into the input space of the LLM before language processing begins. For instance, BLIP-2 (Li et al., 2024b) and MiniGPT-4 (Zhu et al., 2024) employ a Q-Former module for visual compression, while LLaVA (Liu et al., 2023b) uses a single MLP layer to perform alignment. Due to its modular design and scalability, MEF has emerged as a more extensible and popular framework for building VLLMs.

Model Editing for LLMs: Model editing for large language models (LLMs) can be broadly categorized into approaches that either preserve or modify the model's internal parameters. Methods in the first category avoid changing the model by incorporating external mechanisms. For example, IKE (Zheng et al., 2023) adjusts model outputs via in-context demonstrations without any gradient updates, while SERAC (Mitchell et al., 2022c) isolates the editing process using a counterfactual model. T-Patcher (Huang et al., 2023) introduce additional neurons to correct specific errors or encode new knowledge. MELO (**?**) leverages a vector database to dynamically activate LoRA blocks based on retrieval. Similarly, GRACE (Hartvigsen et al., 2023) maintains an external codebook for sequential knowledge updates. In contrast, parameter-modifying approaches directly update the internal weights to embed new knowledge. LTE (Jiang et al., 2024)adapts LLMs through fine-tuning to enable them to execute editing instructions effectively. KE and MEND (Mitchell et al., 2022a) use hypernetworks trained via meta-learning to predict targeted weight changes efficiently. ROME (Meng et al., 2022) locates factual knowledge in specific layers using causal tracing and applies precise edits, while MEMIT (Meng et al., 2023) extends this to batch editing of multiple facts. AlphaEdit (Fang et al., 2025) introduces a novel approach that allows for precise and targeted modifications while preserving the model's overall performance.

B Experiments setting details

B.1 Dataset details

The E-VQA dataset, introduced by Cheng et al. (2023), is designed to fine-tune VLLMs by addressing errors found in samples from the VQA-v2 benchmark (Goyal et al., 2019). It contains 6,345 examples for training and 2,093 for testing. In this task, the model is given an image along with a relevant question and must generate an accurate textual response based on both visual and linguistic cues. Similarly, the E-IC dataset, also from Cheng et al. (2023), focuses on correcting mistakes in image captioning using samples from the COCO Caption dataset (Chen et al., 2015). This collection includes 2,849 training and 1,000 testing instances. The image captioning task requires the model to interpret the image content and produce a coherent and informative textual description. Due to some ambiguities in the inherent meaning of certain image captions and issues arising from prompt generation, some responses were incorrectly classified as correct. To address this, we performed a selective removal of the problematic examples in both the E-VQA and E-IC datasets. The final datasets used for our experiments consist of the samples that passed this cleaning process, with incorrect or ambiguous examples removed.

Each instance in these datasets comprises one core edit example, two for evaluating modality and textual generality, and two targeting modality and textual locality. To construct the generality examples, alternate versions of the original images and questions are produced using Stable

Diffusion (Zhang et al., 2025) and ChatGLM (Du et al., 2022b), respectively. For locality evaluation, unrelated images and questions are drawn from the OK-VQA dataset (Antol et al., 2015) and the Natural Questions (NQ) dataset (Mitchell et al., 2022b), ensuring a robust assessment of model specificity.

B.2 VLLMs details

BLIP2 (Li et al., 2024b) introduces a visual query transformer, Q-Former, which is learned through a two-stage pre-training process to capture key visual information and bridge the gap between the frozen visual encoder and the frozen language model. In this paper, we follow Cheng et al. (2023) and experiment with the BLIP2-OPT1 variant. LLaVA (Liu et al., 2023a) uses GPT-4 (OpenAI, 2023) to create an instruction tuning dataset for VLLM pre-training, aligning visual and linguistic representations by training only a two-layer MLP between the visual encoder and the LLaMA language model (Touvron et al., 2023). While BLIP2 compresses visual representations using Q-Former, LLaVA processes the entire visual input, preserving all visual information but at the cost of reduced inference efficiency.

B.3 Baseline methods

We include several representative baselines in our evaluation. For fine-tuning strategies, FT-V refers to fine-tuning the visual encoder of the VLLM on the edit sample, while FT-L fine-tunes only the final layer of the language model. IKE (Zheng et al., 2023) uses in-context learning with constructed demonstrations to steer the model's responses toward the desired edits. SERAC (Mitchell et al., 2022c) trains both a classifier and a counterfactual language model, redirecting inputs related to the edit sample to the counterfactual model for inference. MEND (Mitchell et al., 2022a) employs an MLP to predict parameter offsets for the FFN layer, conditioned on gradients from backpropagation with respect to the edit sample. TP (Huang et al., 2023) augments the model with a new neuron in the FFN layer that is trained specifically for the edit sample. LTE (Jiang et al., 2024) fine-tunes the language model to follow explicit editing instructions prepended to the input query. VisEdit (Chen et al., 2025) introduces a novel VLLM editor that effectively corrects knowledge by editing intermediate visual representations in regions important to the edit prompt based on attribution analysis.

B.4 Model setting and Training details

To maximize the extraction of visual information in VLLMs, our method is inserted prior to the high-contribution layers identified through our analysis. Specifically, our approach is applied at high-contribution layers from layer 19 in BLIP-OPT and from layer 18 in LLaVA-V1.5, resulting in 21M and 33M trainable parameters, respectively. The null-space threshold is set to 0.02. The learning rate is set to $\eta = 1 \times 10^{-4}$, with a batch size of $B = 4$ and a maximum of 200,000 training iterations. We save a model checkpoint every 500 iterations and select the checkpoint with the lowest loss for evaluation. We conduct our training on two Nvidia HGX H20 Enterprise 96GB. All reported results are averaged over five independent runs with different random seeds while keeping all other hyperparameters fixed.

C Method Details

C.1 SVD

Singular Value Decomposition (SVD) is a widely used technique for matrix factorization. Given any real matrix $A' \in R^{n \times d}$, SVD decomposes A' as

$$A' = (\mathbf{u_1}, \mathbf{u_1} \cdots, \mathbf{u_k})\, W \begin{pmatrix} \mathbf{v_1}^T \\ \mathbf{v_2}^T \\ \vdots \\ \mathbf{v_k}^T \end{pmatrix}$$

$$A' = \sigma_1 \mathbf{u_1} \mathbf{v_1}^T + \sigma_2 \mathbf{u_2} \mathbf{v_2}^T + \cdots + \sigma_k \mathbf{u_k} \mathbf{v_k}^T$$

where $\sigma_1, \ldots, \sigma_k$ are the singular values of the matrix. where $\mathbf{u}_i \in R^n$ are the left singular vectors, $\mathbf{v}_i \in R^d$ are the right singular vectors, and σ_i are the singular values for $i = 1, \ldots, k$, with $k = \min(n, d)$. Plus, $W \in R^{k \times k}$ is a diagonal matrix defined as

$$W = \begin{pmatrix} \sigma_1 & & 0 \\ & \ddots & \\ 0 & & \sigma_k \end{pmatrix},$$

In our method, we perform SVD on the uncentered covariance matrix of the layer input (i.e., $C = X^\top X$) to obtain its singular values and singular vectors. The directions corresponding to near-zero singular values (i.e., the singular vectors in the approximate null space) are then used to construct projection matrices for parameter update constraints. This decomposition is efficiently computed with standard linear algebra libraries such as NumPy or PyTorch.

C.2 Closed-Form Derivation of Null-Space Projected Update

As a theoretical supplement to the null space projection method described in the main paper, we derive a closed-form expression for the projected parameter update. This establishes a principled basis for ensuring that updates remain within the approximate null space while staying close to the original gradient.

To further formalize the update process, we aim to find a projected parameter update Δw_{null} that not only lies in the approximate null space spanned by U_2, but also minimally deviates from the unconstrained gradient direction. We define the following objective:

$$J = \left\| U_2 U_2^\top \Delta w - g \right\|^2 + \lambda \left\| U_2 U_2^\top \Delta w \right\|^2,$$

where g is the original unconstrained gradient update and λ is a regularization coefficient. The first term encourages the projected update to follow the original direction g, while the second term penalizes large steps in the null space.

Taking the derivative of J with respect to Δw and setting it to zero yields the following first-order condition:

$$(1 + \lambda) U_2 U_2^\top \Delta w = U_2 U_2^\top g.$$

Solving for Δw, we obtain the optimal unconstrained update before projection:

$$\Delta w^* = \frac{1}{1 + \lambda} U_2 U_2^\top g.$$

Finally, the null-space-constrained update is given by projecting Δw^* back into the null space:

$$\Delta w_{\text{null}} = U_2 U_2^\top \Delta w^* = \frac{1}{1 + \lambda} U_2 U_2^\top g.$$

This closed-form solution ensures that the update remains within the approximate null space while staying close to the gradient signal, thereby preserving the model's behavior on unrelated inputs. Plus, this derivation follows the structure of projection-based constrained optimization and adapts to the multimodal representation space via null-space projection from input covariance.

D Visualization and Examples

D.1 Visual examples

Figure 5 illustrates two editing samples and their corresponding unrelated samples in visual form.

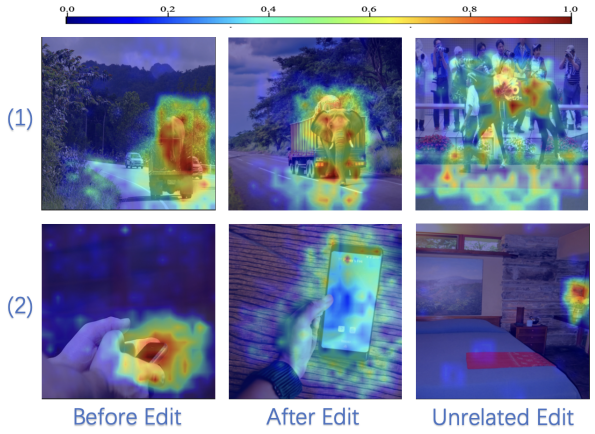

Figure 3: The visualization of editing samples in VLLMs.

In editing sample (1), the prompt is "`What is the animal on the road? It is`", with the expected answer being "`elephant`". In editing sample (2), the prompt is "`What is the person holding? It is`", with the expected answer being "`cell phone`". These samples serve as our targets for multimodal knowledge editing.

To assess whether our method maintains locality, we further present two unrelated examples. In the first unrelated sample, the prompt is "`What animal is on the track? It is`", with the correct answer "`horse`". In the second, the prompt is "`What color are the flowers in the vase? It is`", and the answer is "`yellow`".

From the heatmaps shown in both the *Original* and *Revised* visualizations, we observe that the model correctly focuses on the relevant visual regions specified in the prompt. Notably, even after editing, the model continues to attend to the correct visual cues in the unrelated examples. This indicates that our method not only succeeds in performing effective knowledge editing for the intended visual-textual pairs, but also preserves the model's behavior on unrelated inputs.

These visualizations provide qualitative evidence that our method enables consistent and localized multimodal editing in vision-language models (VLLMs), supporting the quantitative results reported in the main text.

D.2 Textual examples

To better illustrate the structure and intent of each task, we provide two representative examples for each of the two tasks: E-VQA and E-IC. These examples serve as concrete references to help readers understand how the prompts and target outputs

```
"src": "Why is this bike parked in a forest?",
"pred": "it is a motorcycle",
"rephrase": "What is the reason for this bike being stationary in a wooded area?",
"alt": "yes",
"image": "val2014/COCO_val2014_000000153083.jpg",
"image_rephrase": "val2014_image_rephrase/153083001_COCO_val2014_000000153083.png",
"loc": "nq question: who climbed up the beanstalk to get back his money bag hen and harp",
"loc_ans": "Jack",
"m_loc": "val2014/COCO_val2014_000000111032.jpg",
"m_loc_q": "What brand is this wine?",
"m_loc_a": "becker vineyards"


"src": "What is in the bowl?",
"pred": "grass",
"rephrase": "What is inside the bowl?",
"alt": "lettuce",
"image": "val2014/COCO_val2014_000000022112.jpg",
"image_rephrase": "val2014_image_rephrase/22112002_COCO_val2014_000000022112.png",
"loc": "nq question: dwarfs in snow white and the huntsman movie",
"loc_ans": "Ian McShane",
"m_loc": "val2014/COCO_val2014_000000509641.jpg",
"m_loc_q": "What would happen if these items fall to the ground?",
"m_loc_a": "shatter"
```


Figure 4: The two examples of VQA


```
"src": "a photo of",
"pred": "chopped carrots and a knife on a cutting board",
"rephrase": "offer a rich description of the image,",
"alt": "A knife and progressively finely chopped up carrots.",
"image": "val2014/COCO_val2014_000000006415.jpg",
"image_rephrase": "val2014_image_rephrase/COCO_val2014_000000006415.png",
"loc": "nq question: who is winner in telugu bigg boss 2",
"loc_ans": "Kaushal",
"m_loc": "val2014/COCO_val2014_000000493799.jpg",
"m_loc_q": "What are the white veggies?",
"m_loc_a": "leeks"


"src": "a photo of",
"pred": "a man riding a surfboard down a river",
"rephrase": "please provide a detailed description of this picture,",
"alt": "A man on a surfboard that is on a wave.",
"image": "val2014/COCO_val2014_000000127842.jpg",
"image_rephrase": "val2014_image_rephrase/COCO_val2014_000000127842.png",
"loc": "nq question: how long does a english springer spaniel live",
"loc_ans": "a typical lifespan of twelve to fourteen years",
"m_loc": "val2014/COCO_val2014_000000293452.jpg",
"m_loc_q": "What country or countries will you find these birds in?",
"m_loc_a": "new zealand"
```


Figure 5: The two examples of IC

are formulated in each setting. By examining these instances, one can gain clearer insight into how the multimodal task is formed during editing.