DYNAMIC MIXTURE OF ADAPTIVE EXPERTS FOR LIFELONG MULTIMODAL KNOWLEDGE EDITING

Anonymous authors
Paper under double-blind review

ABSTRACT

Updating Multimodal Large Language Models (MLLMs) without catastrophic forgetting remains a fundamental challenge. Existing lifelong knowledge editing methods face two critical limitations: interference from previously learned knowledge, and knowledge capacity dilemma arising from the difficulty of predefining appropriate parameters. We propose Dynamic **M**ixture **o**f **A**daptive **E**xperts (**MoAE**), a novel framework that addresses these challenges through dynamic expert expansion and adaptive knowledge capacity. Each expert comprises a shared LoRA and multiple specific LoRAs that expand progressively based on expert utilization frequency. A hierarchical routing mechanism is proposed to select relevant experts and their corresponding specific LoRAs for each input. Extensive experiments on ComprehendEdit and VLKEB benchmarks across MiniGPT-4, LLaVA-1.5, and Qwen2.5-VL demonstrate that MoAE achieves superior overall performance on reliability, generality, and locality in lifelong editing scenarios.

1 INTRODUCTION

Multimodal large language models (MLLMs) have achieved remarkable success in integrating visual and textual information for diverse tasks such as image understanding, caption generation, and multimodal reasoning (Meng et al., 2024; Lu et al., 2023; Shi et al., 2024; Yan et al., 2024). However, these models are typically static post-training, rendering them unable to continuously incorporate new knowledge without retraining or degrading previously learned capabilities (Zhao et al., 2025; He et al., 2023; Chen et al., 2024a). This limitation has recently sparked growing interest in lifelong multimodal knowledge editing, which aims to update MLLMs with new facts while preserving prior knowledge (Cheng et al., 2023).

A fundamental challenge in this domain is the knowledge capacity dilemma (Allen-Zhu & Li, 2024). Existing methods typically rely on a pre-defined knowledge capacity, such as a fixed number of experts or LoRA adapters (Mitchell et al., 2021; Yu et al., 2024; Wang & Li, 2024; Wang et al., 2024; Li et al., 2025; Chen et al., 2025b; Cheng et al., 2025). This forces an intractable trade-off between knowledge capacity and efficiency. On one hand, under-provisioning capacity (e.g., a few experts) forces distinct facts to be "crammed" into shared parameters, causing knowledge interference and inefficient learning (Xu et al., 2025; Wang et al., 2024). On the other hand, over-provisioning capacity to prevent this interference leads to massive storage waste, as most experts or LoRAs remain under-utilized. As illustrated in Fig.1 (a), both the number and capacity of experts substantially influence the final editing success rate across varying sequence lengths. As the parameters increase, knowledge capacity becomes sufficient, but the performance gain correspondingly diminishes ($72.58\% \rightarrow 81.92\%(+9.34\%)$ vs $81.92\% \rightarrow 83.17\%(+1.25\%)$ when sequence $T = 50$, and $59\% \rightarrow 71.58\%(+12.58\%)$ vs $71.58\% \rightarrow 74\%(+2.42\%)$ when sequence $T = 100$). This indicates that it's infeasible to predefine an appropriate number and capacity of experts.

To escape this trade-off, we propose the Dynamic **M**ixture **o**f **A**daptive **E**xperts (MoAE), a framework that enables highly efficient, on-demand parameter allocation. Instead of a rigid, pre-defined budget, MoAE introduces a hierarchical expert framework that expands adaptively. First, it expands an expert only when new knowledge is semantically dissimilar to all existing knowledge clusters, avoiding redundant expert allocation. Second, each expert is decoupled into a shared LoRA (for common knowledge) and multiple specific LoRAs (for fine-grained facts). Crucially, the expert's capacity also expands adaptively by adding new specific LoRAs only as the samples it handled in-

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

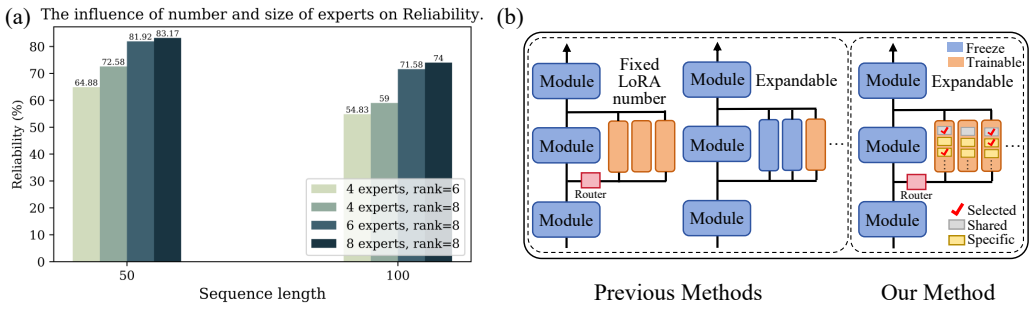


Figure 1: (a) For a constant sequence length, increasing the number of learnable parameters improves editing success rate (reliability). However, excessive parameters yield diminishing returns. Experiments are conducted using ELDER on VLKEB. (b) Comparison between previous methods and our method. MoAE dynamically expands decoupled experts, and each expert can adaptively expand capacity. For each sample, we selectively train experts based on the router's results.

creases. This "allocate-on-demand" design ensures parameters are allocated precisely where needed, maximizing parameter reuse while efficiently isolating distinct facts. To assign the appropriate experts and their parameters to the input, we employ a hierarchical routing mechanism to identify the most relevant experts and their specific LoRAs during inference. Concretely, we first compute distances between the representation and router embeddings to select the most relevant experts. Within each selected expert, we then compute distances with the representations of edited samples processed by each specific LoRA, selecting the specific LoRA corresponding to the nearest representation. The distinctions between conventional approaches and MoAE are depicted in Fig. 1 (b).

Extensive experiments on the ComprehendEdit and VLKEB benchmarks across MiniGPT-4, LLaVA-1.5, and Qwen2.5-VL demonstrate that MoAE achieves superior overall performance. It simultaneously attains high reliability in editing, robust generality, and state-of-the-art locality in preserving unrelated knowledge.

Our main contributions are as follows:

1. We propose a dynamic mixture of adaptive experts framework that expands experts and their parameters jointly for lifelong multimodal knowledge editing, ensuring scalability and edit isolation.

2. To assign appropriate experts and their parameters, we introduce a hierarchical routing mechanism to dynamically select appropriate expert parameters for each sample.

3. Compared with existing baselines, our proposed MoAE achieves superior generality, locality, and continual adaptability across various backbones.

2 Related Works

2.1 Knowledge Editing

The emergence of multimodal knowledge editing has garnered substantial attention in recent years. Cheng et al. (2023) pioneered the multimodal knowledge editing framework, establishing foundational datasets and evaluation metrics. Subsequently, VLKEB (Huang et al., 2024) introduced enhanced datasets featuring comprehensive evaluations of knowledge application capabilities. To address the limitations of existing datasets about question diversity and evaluation comprehensiveness, ComprehendEdit (Ma et al., 2025) proposed a novel dataset alongside the Knowledge Preservation Index (KPI) for assessing knowledge preservation in ambiguous samples that have similar images or queries with the edited sample.

Meanwhile, recent advances have increasingly focused on lifelong knowledge editing paradigms. WISE (Wang et al., 2024) trains dedicated parameters for individual samples, subsequently merging

these parameters after processing samples within the sequence. GRACE (Hartvigsen et al., 2023) records intermediate layer representations and corresponding modifications for each editing sample. But it exhibits limited generalization capabilities. MELO (Yu et al., 2024) employs a predefined multi-LoRA architecture, sequentially updating LoRAs during the editing process. Notably, LoRA sharing occurs exclusively when answers of these questions are identical, resulting in parameter redundancy. LiveEdit (Chen et al., 2025b) trains a specialized expert generator to produce appropriate experts for individual inputs, though its effectiveness diminishes when processing samples that deviate from the training distribution.

Alternative approaches leverage Mixture of Experts (MoE) architectures to store edited knowledge. ELDER (Li et al., 2025) predefines expert quantities and incorporates guided loss to ensure consistent expert selection. But it requires pre-assign an expert allocation for each editing sample, which limits its application to sequence data stream. LEMOE (Wang & Li, 2024) addresses scalability through continuous expert module expansion, training novel experts while leveraging knowledge from existing ones. However, not all edited knowledge is beneficial for training new samples, limiting its effectiveness in lifelong learning scenarios.

2.2 Continual Learning

The continual learning methods can be categorized into replay-based methods, regularization-based methods and architecture-based methods. Here we introduce some classical methods based on expanding network structures. SD-LoRA (Wu et al., 2025), APT (Chen et al., 2025a) trains newly introduced parameters for each task and combination coefficients to integrate knowledge from previous tasks. But it's hard to determine optimal combination coefficients for novel samples during inference. SEMA (Wang et al., 2025) maintains task-specific parameters and corresponding input distributions, employing distribution similarity analysis to determine whether to update existing parameters or expand learnable parameters. LoRA-DRS (Liu & Chang, 2025) enforces orthogonality constraints on parameter updates relative to previous tasks' inputs, thereby preventing inter-task knowledge interference. DualPrompt (Wang et al., 2022) trains a shared LoRA alongside task-specific LoRAs, effectively decoupling inter-task shared knowledge from task-specific knowledge.

However, when applied to multimodal knowledge editing, these methods exhibit several limitations: (1) Substantial memory due to storage of high-dimensional representations and parameter updates; (2) reliance on training set limits performance on samples significantly different from training data; (3) orthogonal projection with respect to previous representations constitutes an overly restrictive constraint that may impede injected knowledge learning, since different samples often share similar knowledge while maintaining specific variations.

3 Method

This section introduces our proposed Mixture of Adaptive Experts framework. The design motivation stems from the non-parametric modeling principles of the Dirichlet Process Mixture Model (DPMM). However, when applied to multimodal lifelong knowledge editing, we have simplified the approach to accommodate high-dimensional representations and few sample characteristics, achieving dynamic expert expansion.

3.1 Background

Following established frameworks in prior work (Cheng et al., 2023), we formalize the lifelong knowledge editing paradigm as follows. Consider a sequential stream of editing samples $\mathcal{D}_T = \{(x_i, y_i)\}_{i=1}^{T}$, where x_i represents the multimodal input comprising both image and textual components, y_i denotes the corresponding answer, θ_0 represents the original model parameters, and θ_i denotes the updated model parameters after incorporating the i-th editing sample. The lifelong knowledge editing process can be formalized as:

$$\theta_i = \mathrm{Alg}(x_i, y_i; \theta_{i-1}), i = 1, 2, \cdots, T \tag{1}$$

where $\mathrm{Alg}(x_i, y_i; \theta_{i-1})$ represents editing the model parameters θ_{i-1} based on the current editing sample. In the context of additional module-based approaches, θ_0 and θ_T denote the initial pa-

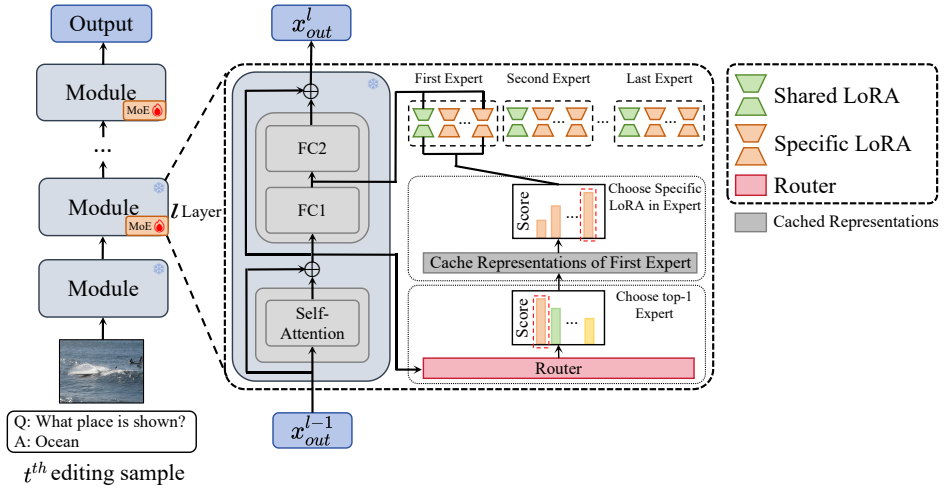


Figure 2: Illustration of editing process, and we take $k = 1$ as an example. When editing each sample, we first compute the distance between the representation and the routers to determine whether expert expansion is necessary. Subsequently, we select the top-k experts by allocation scores and evaluate whether these experts satisfy the expansion criterion for specific LoRAs. Finally, forward propagation is executed. During inference, we select a specific LoRA based on the similarity between the representation of the input and those of the editing samples previously processed by the selected expert.

rameters of the additional modules and the final parameters after processing all editing samples, respectively.

In lifelong knowledge editing, existing methods suffer from two critical limitations. First, newly introduced editing samples may exhibit substantial discrepancies from previously edited samples. When updating parameters trained on edited samples with the current sample, these discrepancies frequently induce significant knowledge interference, consequently leading to catastrophic forgetting of acquired knowledge. Second, determining the optimal parameter budget presents a fundamental challenge: insufficient learnable parameters limit knowledge capacity, resulting in interference between new and old knowledge, while excessive parameters introduce storage redundancy.

To address above limitations, existing parameter expansion methods employ least square method to obtain a classifier or train autoencoders to capture input distributions, training a set of parameters for each distributions (Tong et al., 2025; Wang et al., 2025). These approaches expand parameters only when the input exhibits significant deviation from recorded distributions. However, such methods necessitate explicit task boundaries or category labels, which are unavailable in lifelong knowledge editing scenarios. Consequently, it's necessary to develop an adaptive parameter expansion mechanism that operates without task or category supervision.

Dirichlet Process Mixture Models (DPMM) provide a natural modeling framework for this scenario. DPMM assumes data originates from an underlying infinite mixture distribution, where the number of clusters grows automatically with incoming data (Görür & Edward Rasmussen, 2010). Each new sample is either assigned to an existing cluster or triggers the creation of a new cluster, with assignment probabilities dependent on existing cluster sizes and prior weights.

Inspired by DPMM, we adopt a simplified distance-thresholded online clustering strategy to circumvent the challenges of applying DPMM in high-dimensional feature spaces: (1) *Covariance estimation instability*: Insufficient samples preclude accurate estimation of covariance matrices σ for high-dimensional features; (2) *Weight-induced expansion collapse*: Following standard DPMM updates, larger clusters exhibit higher probabilities for processing new samples, leading to expansion collapse as empirically observed (as shown in Appendix B). In fact, the simplified method is equivalent to introducing two assumptions: (1) Unit covariance matrices for all input distributions, i.e., $\Sigma = I$; (2) The weights of all distributions are set to 1. The details are in Appendix A.

3.2 FRAMEWORK

Our proposed framework incorporates routers and experts across multiple transformer layers. Following the design of ELDER (Li et al., 2025), we initialize a limited number of routers and experts in selected layers, where each router corresponds to a latent knowledge cluster. Let N_{t-1}^l denote the number of routers and experts in injected layer l after processing sample x_{t-1}, where $l = 0$ means the first injected layer. Given the intermediate representation $h^l(x_t)$ of input sample x_t at layer l, we first determine whether the existing experts can handle it. If the minimum distance between $h^l(x_t)$ and all existing routers $R^l = \{R_i^l\}_{i=1}^{N_{t-1}^l}$ exceeds a threshold ϵ, we trigger expansion by instantiating a new router and an expert; otherwise, no expansion occurs. After this process, l layer yields N_t^l routers and experts.

3.2.1 REPRESENTATION REPLAY FOR ROUTER FORGETTING

Nevertheless, router expansion introduces a critical challenge: newly introduced routers may disrupt expert assignments for previously edited samples, resulting in editing errors—a phenomenon we term *router forgetting*. To mitigate this issue, we maintain a memory bank M that stores the representation $h^0(x_t)$ and corresponding expert indices $e^l(x)$ across all layers for each edited sample. Notably, each sample requires storing only a single representation, consistent with the approach adopted in GRACE (Hartvigsen et al., 2023). Following each router expansion, we perform layer-wise fine-tuning of existing routers before freezing them and subsequently training the associated experts. We employ a margin loss:

$$\begin{aligned} \mathcal{L}_m^l &= \frac{1}{|M|} \sum_{x \in M} \max(0, m + \text{neg} - \text{pos}), \\ \text{pos} &= \min(\{s(x, R_i^l) | i \in e^l(x)\}) \\ \text{neg} &= \max(\{s(x, R_i^l) | i \notin e^l(x), i = 1, \cdots, N_t^l\}), \\ s(a, b) &= \|a - b\|_2^{-1}, \end{aligned} \tag{2}$$

Among them, $\mathcal{L}_m^l$ is the calculated margin loss for the router at layer l, x is the representation of edited samples, m is the threshold hyperparameter, and $s(x, R_i^l)$ is the allocation scores of representation x and i-th expert in layer l. By adjusting existing routers with $\mathcal{L}_m^l$, router forgetting can be avoided effectively. In practical implementation, we tend to replay samples at the decision boundary, where the corresponding expert routing is easily affected. Specifically, we choose the top 20% of representations with the smallest pos$-$neg for replay.

3.2.2 HIERARCHICAL EXPERTS EXPANSION

After the expansion and router finetuning, we choose the top-k experts by representation $h^0(x_t)$ to update using this sample:

$$\begin{aligned} \triangle W^l &= \sum_{i \in top\text{-}k(w^l)} w_i^l E_i^l(h^0(x_t)), \\ w_i^l &= \text{softmax}(s(h^0(x_t), R^l))[i], \end{aligned} \tag{3}$$

where $\triangle W^l$ denotes the updated matrix of layer l. w_i^l is the weight for expert output $E_i^l(h^0(x_t))$, and $s(h^0(x_t), R^l)$ is the allocation score of $h^0(x_t)$ and experts in l layer.

While samples assigned to the same expert exhibit certain similarities, there is still intra-cluster variation within these samples. To effectively accommodate this intra-cluster variation, we decompose the LoRA of each expert $E_i^l(\cdot)$ into two distinct components: *a shared LoRA* $A_i^s B_i^s$ and *a specific LoRA* A_i^{spc}, B_i^{spc}, where $A_i^s, A_i^{spc} \in \mathrm{R}^{d \times r}, B_i^s, B_i^{spc} \in \mathrm{R}^{r \times d}$. d, r is the representation dimension and rank of LoRA, respectively. The shared LoRA captures the common knowledge patterns across these samples, while the specific LoRA encodes sample-specific knowledge. However, we empirically observe an imbalance in expert utilization in experiments: certain experts are assigned to process substantially more samples (Appendix B). If each expert has the same number of parameters, the knowledge capacity of heavily-utilized experts becomes insufficient. To address this limitation, we implement a adaptive specific LoRA expansion mechanism. For every m samples processed by

expert $E_i^l(\cdot)$, a learnable specific LoRA with rank r is expanded to increase its knowledge capacity. During training, every m samples share a specific LoRA, and the specific LoRA indices assigned for these samples are recorded. Through this LoRA decomposition and adaptive expansion mechanism, we effectively mitigate interference among sample-specific knowledge.

We optimize the experts parameters via Cross-Entropy loss, and router embeddings via $\mathcal{L}_m^l$. During inference, the appropriate specific LoRA is selected through distance between the input representation and the stored representations associated with each specific LoRA, enabling knowledge retrieval based on sample characteristics.

3.2.3 Inference with Retrieval

ELDER proposes a deferral mechanism to determine whether an input requires editing (Li et al., 2025). This mechanism computes the Hamming distance between the expert assignment indices of the input and edited samples; if the distance falls below a threshold ϵ, indicating sufficient similarity to previously edited knowledge, the input undergoes editing; otherwise, the original model's output is used. However, we found that this mechanism exhibits substantial error in identifying locality samples that are similar to edited samples. Therefore, we augment this mechanism by incorporating image-text feature retrieval. Specifically, we employ pretrained models to extract and store image and text features for the $p\%$ of locality samples from the training set and all edited samples. During inference, we first identify the sample with the highest average similarity to the input's image and text features. If this sample is an edited sample, we subsequently apply the deferral mechanism; otherwise, we directly use the original model's output.

4 Experiments

4.1 Datasets & Backbones & Baselines

We conduct experiments on ComprehendEdit (Ma et al., 2025) and VLKEB (Huang et al., 2024) with sequence lengths of T=100, 400 and 1000, respectively. We employ MiniGPT-4 (Zhu et al., 2023), LLaVA-1.5 (Liu et al., 2024), and the recent Qwen2.5-VL (Bai et al., 2025) as backbones. The baseline methods for comparison include MELO (Yu et al., 2024), WISE (Wang et al., 2024), Merging (Fu et al., 2025), SD-LoRA (Wu et al., 2025), and ELDER (Li et al., 2025). We additionally evaluate an enhanced version achieved by ourselves, ELDER++, where the number of experts and rank are configured using the average values obtained after expansion in our method across each sequence.

4.2 Implementation Details

All experiments are implemented on NVIDIA 4090 GPUs using the PyTorch framework. After editing each sequence, we restore the model to its initial state. The learning rate for expert parameters is set to 1e-3 for ComprehendEdit and 2e-4 for VLKEB, while the fine-tuning learning rate for routers is set to 1e-5. For each editing sample (batch size is 1), the maximum number of training iterations is set to 30. Following the configuration of ELDER, we incorporate modules at layers 22-26. Initially, we initialize a shared and a specific LoRA, and the rank of them is 2. Each expert expands by one specific LoRA of rank $r = 2$ after processing every $m = 5$ samples. To obtain ϵ, we employ k-means over the representations of training samples and average the radius of each cluster. The average radius estimates the intrinsic distribution of the representation space, providing a global scale prior. The proportion of used locality samples is set to $p = 20$. In the memory bank M, we store the image and text features, representations for expert retrieval, expert indices, and specific LoRA indices for each editing sample. The extractors for image and text features employ DINOv2 (Oquab et al., 2023) and reranker (Chen et al., 2024b), respectively.

4.3 Evaluation Metrics

The evaluation metrics include Reliability (Rel), Generality (includes text generality) and Locality (includes text locality, multimodal locality, and knowledge preservation index). Assume that after completing the editing of a sequence with length T, we obtain the edited model denoted as $f_T(\cdot)$,

Table 1: Results of sequential editing on LLaVA-1.5 7B. **Bold** and underline represent the first and second highest values, respectively.

		ComprehendEdit						VLKEB					
T	Method	Rel	T-G	T-L	M-L	KPI	**AVG**	Rel	T-G	T-L	M-L	KPI	**AVG**
100	MELO	50.68	23.89	96.75	97.26	**98.99**	57.41	50.17	16.17	**100**	**99.90**	**97.95**	55.21
	WISE	38.63	28.68	50.68	17.77	29.09	33.28	15.58	13.00	99.97	30.53	79.79	32.89
	Merging	49.40	39.00	93.11	83.24	75.41	57.44	49.40	39.00	93.11	83.24	75.41	57.44
	SD-LoRA	12.16	9.84	7.07	1.63	8.33	9.23	56.08	41.58	80.88	64.50	58.25	55.18
	ELDER	96.42	**78.26**	94.49	88.11	32.75	82.16	50.67	27.92	81.91	43.31	64.05	47.23
	ELDER++	98.42	73.79	99.72	93.71	38.86	83.21	81.08	**45.08**	98.63	79.13	72.79	69.89
	Our	**99.67**	66.00	**100**	**99.18**	85.21	**86.82**	**94.25**	41.58	**100**	98.68	94.25	**77.83**
400	MELO	48.81	24.94	89.71	82.21	**87.47**	53.40	50.00	15.83	**100**	**97.63**	**92.37**	54.17
	WISE	22.19	16.63	49.76	15.11	29.27	23.40	3.42	4.33	99.97	33.69	75.74	25.85
	Merging	46.75	37.25	95.41	83.10	74.05	56.06	0.88	1.13	93.51	86.53	86.30	30.26
	SD-LoRA	17.19	14.50	3.90	0.89	0.00	11.10	17.19	14.50	3.90	0.89	0.00	11.10
	ELDER	93.94	**75.19**	91.05	82.10	30.54	79.01	72.00	**46.25**	87.25	63.37	62.53	63.10
	ELDER++	98.56	65.25	**100**	97.31	80.27	85.45	66.67	30.33	99.49	66.87	66.67	58.23
	Our	**99.25**	71.63	**100**	**97.52**	74.83	**87.22**	**72.50**	28.42	**100**	93.72	79.31	**63.98**

and original model is f_0. The details about how to calculate these metrics are included in Appendix B.

In order to balance the evaluation of editing success rate, editing generalization, and locality, we weighted each metrics as the final indicator:

$$\text{AVG} = \frac{1}{3}[\text{Rel} + \text{T-G} + \frac{1}{3}(\text{T-L} + \text{M-L} + \text{KPI})]. \tag{4}$$

4.4 RESULTS

Results on LLaVA-1.5 are presented in Tab 1, respectively. Results on MiniGPT-4 and Qwen2.5-VL are included in Appendix B. MELO pre-defines a large number of LoRAs and trains them in sequence, due to its limited editing scope, achieves strong locality preservation, but sacrifices editing generalization. WISE and Merging train parameters on each sample and merges the trained parameters across samples; for longer sequences, this merging causes interference between parameters, thereby reducing the success rate of editing (Rel) and text generality. Meanwhile, it performs even worse on the challenging benchmark VLKEB. Compared with Merging, WISE introduces an additional constraint to determine whether the activation induced by an input significantly matches the trained parameters, and thus whether those parameters should be invoked. However, its weaker multimodal locality performance indicates that this mechanism does not generalize well to multimodal inputs.

SD-LORA trains one LoRA group per sample while freezing and leveraging all existing LoRAs; however, its editing performance deteriorates with longer sequences since not all previous knowledge is beneficial for learning current knowledge. Moreover, lacking explicit locality preservation mechanisms, its locality is adversely affected by edited knowledge.

ELDER requires presetting the number and size of experts, and this method exhibits limited performance on VLKEB, as VLKEB features longer average answer lengths and consequently greater difficulty. With a fixed parameter budget, maintaining reliability becomes increasingly challenging for longer sequences. Compared with ELDER, although ELDER++ initializes a larger pool of learnable parameters and consequently achieves a substantial improvement in reliability on VLKEB, it still suffers from two key limitations. First, it is difficult to determine appropriate expert sizes and quantities. Second, its fixed expert capacity cannot adapt to the varying usage demands of different experts in streaming data. Compared to ELDER++, which directly defines appropriately sized and sufficient experts, our proposed method achieves higher reliability through progressive expert expansion. This improvement stems from the strategy of expanding experts only when existing experts struggle to handle the current sample. Consequently, the samples processed by each expert exhibit inherent similarities, thereby minimizing the knowledge interference among these samples. Fur-

thermore, MoAE decomposes each expert into finer-grained components that independently store the knowledge associated with these samples, which further mitigates the interference of sample-specific knowledge.

Table 2: Results of sequential editing (T=1000) on Llava1.5 7B with VLKEB. **Bold** and underline represent the first and second highest values, respectively.

T	Method	Rel	T-G	T-L	M-L	KPI	**AVG**
1000	MELO	59.3	15.8	96.8	91.1	73.8	54.1
	WISE	1.0	1.7	**99.4**	**98.9**	**99.7**	34.0
	Merging	-	-	-	-	-	-
	LEMOE	1.0	1.0	0.0	0.0	0.0	0.7
	RLEdit	31.8	**33.8**	88.3	51.5	85.3	46.9
	SD-LoRA	10.2	6.3	5.6	1.8	19.2	8.5
	ELDER++	49.9	17.2	**99.4**	59.6	63.2	47.1
	Our	**61.1**	26.2	99.0	88.3	66.0	**57.2**

In addition, we evaluate our method on substantially longer sequences ($T = 1000$) to further verify its effectiveness. The results are shown in Fig. 2. Although WISE demonstrates strong locality properties, its learned gating mechanism fails to reliably identify edited samples and activate the corresponding trained parameters, resulting in low reliability.

Meanwhile, LEMOE lacks an explicit mechanism to prevent router forgetting. When evaluated on longer sequences that better reflect practical scenarios (as opposed to the length of 5 used in the original paper), especially under settings with only one samples per batch, its editing performance drops significantly. The results of RLEdit indicate that continuously updating parameters on the base model still leads to a gradual degradation of editing effectiveness. Finally, compared with ELDER++, which employs an equal number of learnable parameters, our proposed method achieves substantially superior editing performance.

4.5 Ablation Study

This subsection discusses the ablation studies on each module and the hyperparameters of the proposed method. The experiments are conducted on VLKEB with MiniGPT-4 7B, and T=100.

Ablation studies on various modules of our method are presented in Table 3. The first row demonstrates the results without replay-based router fine-tuning during training. The second row indicates employing all specific LoRAs within an expert, rather than selecting only one. The third row represents removing the specific LoRA expansion module, with each expert initialized to the final average rank of MoAE. The fourth row represents the results without a gating mechanism, where insertion modules are applied to all inputs. The final row presents the results of our method. The absence of replay-based router fine-tuning significantly impairs both final reliability and generalization, as the expert assignments for previously edited samples are altered. Representation replay effectively corrects misassigned experts, thereby improving the overall reliability. In fact, after completing the editing of the entire sequence, less than 50% of the representations are involved in replay. The second and fifth rows show that using all task-specific LoRAs within an expert actually reduces accuracy, while the third and fifth rows indicate that directly initializing a large-capacity expert performs worse than gradual expansion. These results demonstrate that decomposing experts and progressively expanding specific LoRAs effectively mitigate knowledge interference. The fourth and fifth rows show that the gating mechanism effectively identifies ambiguous locality samples, thereby substantially improving locality performance.

The evolution of various metrics across different methods on editing length is illustrated in the Fig. 3. We evaluate all metrics on the edited samples after completing 10, 20, 50, and 100 edits. It can be seen that most methods perform well when the sequence length T is short, but various metrics are significantly affected when $T \geq 50$. The results demonstrate that our method exhibits superior anti-forgetting performance in terms of reliability and generality, primarily attributed to its continuously expanding knowledge capacity, whereas other methods exhibit varying degrees of forgetting. Moreover, compared to previous approaches, our method maintains relatively stable performance in locality preservation because of the gating mechanism. Notably, we observe that ELDER's performance

Table 3: Ablation study about each module of MoAE. The experiments are conducted on VLKEB with MiniGPT-4 7B, and sequence length T=100.

	Rel	T-G	T-L	M-L	KPI	AVG
wo Replay	83.42	36.33	100.00	99.06	96.31	72.74
wo spc LoRA	87.25	45.17	100.00	98.80	94.97	76.78
wo Expanding	85.00	44.33	100.00	98.87	94.97	75.76
wo Gating	92.75	56.08	94.66	85.12	76.98	78.14
MoAE	92.58	48.50	100.00	98.90	95.61	79.75

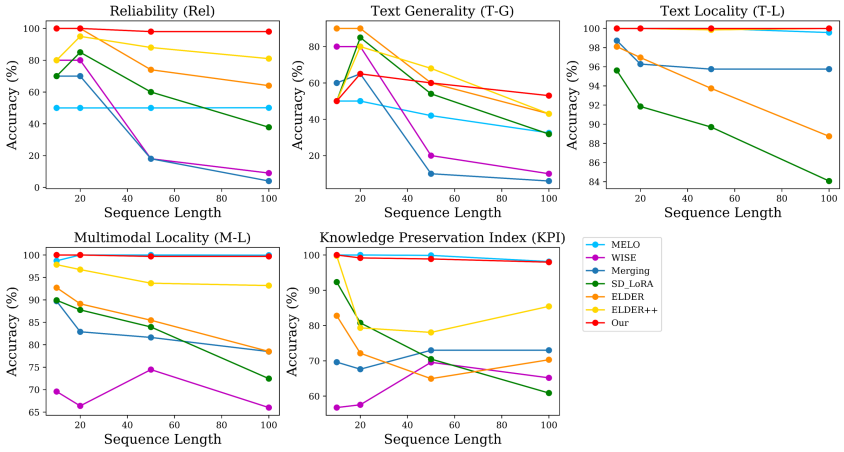


Figure 3: Changes in metrics during lifelong editing. The experiments are conducted on VLKEB with MiniGPT-4. We evaluate all metrics on the edited samples at T=10, 20, 50, 100.

on the KPI metric initially decreases and then increases with sequence length. This phenomenon occurs because, with fewer edited samples, ELDER's stored editing patterns are insufficient to distinguish ambiguous locality samples, resulting in decreased KPI. However, as the number of edited samples grows, the accumulated patterns facilitate better identification of ambiguous samples, consequently improving the KPI measured after completing a full sequence.

Due to space constraints, we present the sensitivity analyses on random seeds (which affect sequence order), the number of selected experts, the size and expansion frequency of specialized LoRAs, as well as the impact of different weights on expert activation frequencies in Appendix B.

5 Conclusion

This paper addresses the critical challenge of lifelong multimodal knowledge editing by proposing MoAE, a framework that enables MLLMs to continuously assimilate new knowledge without forgetting previously edited information or degrading on unedited samples. Our approach makes three key contributions: First, we introduce a Dirichlet Process Mixture Model-inspired dynamic expansion mechanism that adaptively grows the number and capacity of experts based on the distance between representations and router embeddings. Second, we design a decoupled expert architecture where specific LoRAs isolate fine-grained knowledge in samples. To prevent router forgetting caused by expert expansion, we employ representation replay to ensure stable expert assignments for historically edited samples. Extensive experiments across multiple benchmarks and backbones validate the effectiveness of MoAE, demonstrating substantial improvements in reliability, generality, and locality compared to existing methods.

AI use statement

In this work, we used generative AI tools for: 1. assist with translation, 2. support qualitative and thematic data analysis, 3. interpret results. We have not used generative AI tools for: 1. design or

provide feedback on research methodology or experiments, 2. help develop theoretical models or conceptual frameworks, 3. implement methods. And 1. generate synthetic data sets, 2. formulate mathematical claims, provide critical ingredients for proving mathematical claims, 3. assist in the writing of proofs, 4. propose or refine hypotheses, 5. clean and reformat datasets do not apply to this work. We take responsibility for the final content of this work, including text, claims, or artifacts produced with the aid of generative AI.

REFERENCES

Zeyuan Allen-Zhu and Yuanzhi Li. Physics of language models: Part 3.3, knowledge capacity scaling laws. *arXiv preprint arXiv:2404.05405*, 2024.

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, et al. Qwen2. 5-vl technical report. *arXiv preprint arXiv:2502.13923*, 2025.

Cheng Chen, Junchen Zhu, Xu Luo, Heng T Shen, Jingkuan Song, and Lianli Gao. Coin: A benchmark of continual instruction tuning for multimodel large language models. *Advances in Neural Information Processing Systems*, 37:57817–57840, 2024a.

Haoran Chen, Ping Wang, Zihan Zhou, Xu Zhang, Zuxuan Wu, and Yu-Gang Jiang. Achieving more with less: Additive prompt tuning for rehearsal-free class-incremental learning. *arXiv preprint arXiv:2503.07979*, 2025a.

Jianlv Chen, Shitao Xiao, Peitian Zhang, Kun Luo, Defu Lian, and Zheng Liu. Bge m3-embedding: Multi-lingual, multi-functionality, multi-granularity text embeddings through self-knowledge distillation. *arXiv preprint arXiv:2402.03216*, 2024b.

Qizhou Chen, Chengyu Wang, Dakan Wang, Taolin Zhang, Wangyue Li, and Xiaofeng He. Lifelong knowledge editing for vision language models with low-rank mixture-of-experts. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 9455–9466, 2025b.

Siyuan Cheng, Bozhong Tian, Qingbin Liu, Xi Chen, Yongheng Wang, Huajun Chen, and Ningyu Zhang. Can we edit multimodal large language models? *arXiv preprint arXiv:2310.08475*, 2023.

YuJu Cheng, Yu-Chu Yu, Kai-Po Chang, and Yu-Chiang Frank Wang. Serial lifelong editing via mixture of knowledge experts. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 30888–30903, 2025.

Zichuan Fu, Xian Wu, Guojing Li, Yingying Zhang, Yefeng Zheng, Tianshi Ming, Yejing Wang, Wanyu Wang, and Xiangyu Zhao. Model merging for knowledge editing. *arXiv preprint arXiv:2506.12384*, 2025.

Dilan Görür and Carl Edward Rasmussen. Dirichlet process gaussian mixture models: Choice of the base distribution. *Journal of Computer Science and Technology*, 25(4):653–664, 2010.

Tom Hartvigsen, Swami Sankaranarayanan, Hamid Palangi, Yoon Kim, and Marzyeh Ghassemi. Aging with grace: Lifelong model editing with discrete key-value adaptors. *Advances in Neural Information Processing Systems*, 36:47934–47959, 2023.

Jinghan He, Haiyun Guo, Ming Tang, and Jinqiao Wang. Continual instruction tuning for large multimodal models. *arXiv preprint arXiv:2311.16206*, 2023.

Han Huang, Haitian Zhong, Tao Yu, Qiang Liu, Shu Wu, Liang Wang, and Tieniu Tan. Vlkeb: A large vision-language model knowledge editing benchmark. *Advances in Neural Information Processing Systems*, 37:9257–9280, 2024.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, et al. Natural questions: a benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:453–466, 2019.

Jiaang Li, Quan Wang, Zhongnan Wang, Yongdong Zhang, and Zhendong Mao. Elder: Enhancing lifelong model editing with mixture-of-lora. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 24440–24448, 2025.

Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction tuning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 26296–26306, 2024.

Xuan Liu and Xiaobin Chang. Lora subtraction for drift-resistant space in exemplar-free continual learning. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 15308–15318, 2025.

Pan Lu, Hritik Bansal, Tony Xia, Jiacheng Liu, Chunyuan Li, Hannaneh Hajishirzi, Hao Cheng, Kai-Wei Chang, Michel Galley, and Jianfeng Gao. Mathvista: Evaluating mathematical reasoning of foundation models in visual contexts. *arXiv preprint arXiv:2310.02255*, 2023.

Yaohui Ma, Xiaopeng Hong, Shizhou Zhang, Huiyun Li, Zhilin Zhu, Wei Luo, and Zhiheng Ma. Comprehendedit: A comprehensive dataset and evaluation framework for multimodal knowledge editing. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 19323–19331, 2025.

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

Fanqing Meng, Jin Wang, Chuanhao Li, Quanfeng Lu, Hao Tian, Jiaqi Liao, Xizhou Zhu, Jifeng Dai, Yu Qiao, Ping Luo, et al. Mmiu: Multimodal multi-image understanding for evaluating large vision-language models. *arXiv preprint arXiv:2408.02718*, 2024.

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

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

Wenhao Shi, Zhiqiang Hu, Yi Bin, Junhua Liu, Yang Yang, See-Kiong Ng, Lidong Bing, and Roy Ka-Wei Lee. Math-llava: Bootstrapping mathematical reasoning for multimodal large language models. *arXiv preprint arXiv:2406.17294*, 2024.

Kai Tong, Kang Pan, Xiao Zhang, Erli Meng, Run He, Yawen Cui, Nuoyan Guo, and Huiping Zhuang. Analytic subspace routing: How recursive least squares works in continual learning of large language model. *arXiv preprint arXiv:2503.13575*, 2025.

Huiyi Wang, Haodong Lu, Lina Yao, and Dong Gong. Self-expansion of pre-trained models with mixture of adapters for continual learning. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 10087–10098, 2025.

Peng Wang, Zexi Li, Ningyu Zhang, Ziwen Xu, Yunzhi Yao, Yong Jiang, Pengjun Xie, Fei Huang, and Huajun Chen. Wise: Rethinking the knowledge memory for lifelong model editing of large language models. *Advances in Neural Information Processing Systems*, 37:53764–53797, 2024.

Renzhi Wang and Piji Li. Lemoe: Advanced mixture of experts adaptor for lifelong model editing of large language models. *arXiv preprint arXiv:2406.20030*, 2024.

Zifeng Wang, Zizhao Zhang, Sayna Ebrahimi, Ruoxi Sun, Han Zhang, Chen-Yu Lee, Xiaoqi Ren, Guolong Su, Vincent Perot, Jennifer Dy, et al. Dualprompt: Complementary prompting for rehearsal-free continual learning. In *European conference on computer vision*, pp. 631–648. Springer, 2022.

Yichen Wu, Hongming Piao, Long-Kai Huang, Renzhen Wang, Wanhua Li, Hanspeter Pfister, Deyu Meng, Kede Ma, and Ying Wei. Sd-lora: Scalable decoupled low-rank adaptation for class incremental learning. *arXiv preprint arXiv:2501.13198*, 2025.

Haoyu Xu, Pengxiang Lan, Enneng Yang, Guibing Guo, Jianzhe Zhao, Linying Jiang, and Xingwei Wang. Knowledge decoupling via orthogonal projection for lifelong editing of large language models. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 13194–13213, 2025.

Yibo Yan, Jiamin Su, Jianxiang He, Fangteng Fu, Xu Zheng, Yuanhuiyi Lyu, Kun Wang, Shen Wang, Qingsong Wen, and Xuming Hu. A survey of mathematical reasoning in the era of multimodal large language model: Benchmark, method & challenges. *arXiv preprint arXiv:2412.11936*, 2024.

Lang Yu, Qin Chen, Jie Zhou, and Liang He. Melo: Enhancing model editing with neuron-indexed dynamic lora. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 19449–19457, 2024.

Hongbo Zhao, Fei Zhu, Haiyang Guo, Meng Wang, Rundong Wang, Gaofeng Meng, and Zhaoxiang Zhang. Mllm-cl: Continual learning for multimodal large language models. *arXiv preprint arXiv:2506.05453*, 2025.

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