R-MLLM: DYNAMIC MODEL EDITING FOR ROBUST MULTIMODAL LARGE LANGUAGE MODEL UNDER LIMITED DATA

Anonymous authors
Paper under double-blind review

ABSTRACT

In real-world scenarios, unknown visual corruptions may continue to emerge and degrade model performance. Existing robust Multimodal Large Language Models (MLLMs) are typically optimized for predefined corruptions and thus generalize poorly to unknown corruptions. Although adapting these models to new corruptions can improve their performance on newly emerging corruptions, it often leads to catastrophic forgetting of previously acquired robustness knowledge. To address these challenges, we formulate the visual robustness task as a continual learning problem and propose a model editing framework for data-efficient continual adaptation to visual corruptions. Specifically, we fine-tune the visual encoder with an additional robustness loss to obtain corrupted key-value pairs. We then propose a dynamic key-value pair construction strategy that adaptively weights samples according to their quality, enabling effective model updates with limited data. To preserve pretrained knowledge, we further introduce a principal component constraint that regularizes parameter updates during adaptation and mitigates catastrophic forgetting. Finally, we introduce VCBench, a benchmark covering vision questions under 17 common real-world visual corruptions, to evaluate the robustness and continual adaptation ability of MLLMs. Extensive experiments across multiple benchmarks show that our method improves robustness over strong baselines, achieving up to a 7.6% average accuracy gain on VCBench.

1 INTRODUCTION

MLLMs (Zhu et al., 2025b; Bai et al., 2025b;a; Li et al., 2024a) have shown strong performance on Visual Question Answering (VQA) tasks. However, real-world visual inputs may be affected by unknown visual corruptions, which typically degrade model performance in real-world deployment. Meanwhile, newly emerging unknown corruptions are often represented by only a limited number of samples, and adapting MLLMs with such limited data often leads to suboptimal performance.

Existing robust MLLMs (Schlarmann et al., 2024; Malik et al., 2025; Tang et al., 2026) are typically trained on a predefined set of corruptions and often show degraded performance on newly encountered corruptions. Although adapting these models to a newly encountered corruption can improve robustness to that corruption, it may also lead to catastrophic forgetting of old robustness knowledge learned from previous corruptions. Therefore, enabling MLLMs to continually adapt to new corruptions while avoiding catastrophic forgetting of knowledge acquired from previous corruptions is particularly crucial for their deployment in real-world scenarios. This naturally leads us to formulate the visual robustness task as a continual learning problem.

Under this formulation, each newly encountered visual corruption is treated as an incremental task, and the model is updated sequentially as new corruptions appear. The goal is to acquire robustness to each new corruption without forgetting old robustness knowledge. Continual learning (Wu et al., 2025; Zheng et al., 2023b; Liang & Li, 2024) has emerged as a promising paradigm for such sequential adaptation. Conventional continual learning methods (Yu et al., 2024a; Tang et al., 2024; Zhu et al., 2025a) enable MLLMs to adapt sequentially to different visual corruptions while reducing forgetting. However, they often struggle to balance pre-trained knowledge with newly acquired capabilities (He et al., 2025; Zhu et al., 2025a; Gutiérrez et al., 2025; Liu & Chang, 2025) and can

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.

be sensitive to task order, especially under limited-data settings. Model editing (Meng et al., 2022; 2023; Li et al., 2025b; 2024c; Gangadhar & Stratos, 2024) provides an alternative by updating only a small subset of parameters to incorporate new knowledge while preserving the model's original capabilities. Nevertheless, existing methods mainly focus on editing the language model (Zhang et al., 2024; Li et al., 2024c; Mitchell et al., 2022a; Fang et al., 2025), with limited attention to the visual encoder. Since visual corruptions can shift image features and impair visual representation extraction, editing only the language model cannot directly recover information lost in the visual encoder. This motivates us to develop a model editing method specifically for the visual encoder.

To alleviate the above problem, we propose a model editing framework for the visual encoder to enhance MLLM robustness using limited data. We fine-tune only the up-projection and down-projection layers of the visual encoder's FFN layers, which achieves better performance than full-parameter fine-tuning. To improve feature extraction from corrupted images, we align corrupted features with their clean counterparts using an additional robustness loss, reducing the feature shift caused by visual corruptions. After injecting new knowledge into the target layers through fine-tuning, we use the fine-tuned visual encoder to extract the corresponding layer inputs and outputs as key-value pairs. In limited-data scenarios, each sample has a relatively large influence on the model update, making the editing process particularly sensitive to low-quality samples. Assigning uniform weights to all key-value pairs may allow unreliable samples to have an excessive impact on parameter updates. To address this issue, we propose a dynamic key-value pair construction strategy that assigns adaptive weights according to the feature similarity between clean and corrupted samples, reducing the influence of low-quality pairs while emphasizing more reliable ones.

Model editing with limited corrupted samples may also cause the updated model to gradually deviate from its pre-trained knowledge, resulting in catastrophic forgetting. Existing model editing methods (Li et al., 2024c; Fang et al., 2025) often alleviate this problem by introducing additional key-value pairs derived from relevant original knowledge to constrain model updates. However, this requires extra samples for knowledge preservation. To mitigate this problem, we introduce a principal component constraint strategy that directly restricts parameter updates along principal directions associated with pre-trained knowledge. Specifically, we perform eigen-decomposition on the corresponding feature covariance matrix to identify these dominant principal directions and incorporate a principal component regularization term into the optimization objective. By limiting changes along these directions, our method further reduces interference with pre-trained knowledge during adaptation to visual corruptions, without additional knowledge-preservation samples.

Beyond methodological advancements, we observe that existing MLLM benchmarks (Fu et al., 2025; Yue et al., 2024; Lu et al., 2022; He et al., 2024; Masry et al., 2022; Yu et al., 2024b; Li et al., 2024b) fall short in assessing authentic visual robustness. Within these benchmarks, many knowledge-intensive questions can be resolved purely through language priors, allowing models to bypass visual perception entirely. To ensure a faithful evaluation, we propose **VCBench**, a novel visual corruption benchmark. First, we rigorously filter established benchmarks to isolate visual questions unsolvable by language priors alone. Then, we systematically apply 17 common visual corruptions across four real-world categories (Weather Degradations, Sensor Noise, Optical Blurs, and Digital Variations) to effectively evaluate MLLMs' robustness in real-world scenarios.

Compared with existing continual learning and model editing methods, our method directly edits the visual encoder with only a limited number of corrupted samples and updates only FFN projection parameters, enabling efficient adaptation to newly encountered visual corruptions. Also, our dynamic key-value construction strategy adaptively weights key-value pairs according to their quality, reducing the influence of unreliable samples and enabling more stable knowledge integration. Our principal component constraint directly preserves pre-trained knowledge during model updates without requiring additional preservation samples. In summary, our contributions are as follows:

- **Robust model editing for MLLM visual encoders:** We propose a model editing framework to improve visual robustness with limited data. It uses dynamic key-value pair construction to adaptively control the contribution of different samples and a principal component constraint to reduce catastrophic forgetting without requiring additional samples.
- **Authentic visual robustness benchmark:** We introduce VCBench, a visual corruption benchmark containing questions that cannot be solved by language priors alone and 17 corruptions from four real-world categories.

- **Strong performance across multiple benchmarks:** Extensive experiments show that our method outperforms continual learning, model editing, and robust MLLM baselines.

2 Related Work

Multimodal Large Language Model. MLLMs (Team et al., 2024; Achiam et al., 2023; Bai et al., 2025a; Chen et al., 2024) augment Large Language Models (LLMs) with visual perception, enabling joint reasoning over images and text for tasks such as visual question answering, grounded dialogue, and multimodal reasoning (Yi et al., 2025; Awais et al., 2025). Early systems typically connect a pretrained vision encoder, such as CLIP (Radford et al., 2021), with an LLM through lightweight projection or query-based alignment modules, as exemplified by BLIP-2 (Li et al., 2023b) and LLaVA (Liu et al., 2023). Recent works further strengthen cross-modal integration and scalable training. For instance, InternVL3 (Zhu et al., 2025b) adopts native multimodal pretraining to jointly learn visual and linguistic capabilities, while Qwen2.5-VL improves high-resolution perception and structured visual reasoning (Bai et al., 2025b). Moreover, emerging studies extend MLLMs beyond static images toward video-level spatial-temporal understanding and grounding (Wang et al., 2025; Lin et al., 2023), reflecting a broader shift toward more dynamic multimodal reasoning.

Robust Visual Understanding. Visual corruptions and adversarial attacks substantially degrade the perception and reasoning reliability of MLLMs, prompting increasing attention to robustness (Ma et al., 2025; Tang et al., 2026; Schlarmann et al., 2024; Jiang et al., 2025a). Beyond conventional encoder fine-tuning, recent methods improve robustness through diverse strategies. Robust-LLaVA (Malik et al., 2025) incorporates large-scale adversarially pre-trained visual encoders to enhance resilience against various attacks while preserving clean-task performance (Malik et al., 2025). Meanwhile, agentic reasoning frameworks such as ORCA strengthen robustness at inference time by leveraging auxiliary vision models to mitigate hallucinations and cross-model inconsistencies without retraining the backbone LLM (Yu et al., 2025). Comprehensive surveys further summarize emerging threats and evaluation protocols for multimodal robustness (Jiang et al., 2025a).

Model Editing. Model editing updates knowledge in large language models without full retraining and can be broadly divided into parameter-preserving and parameter-modifying methods. *Parameter-preserving methods* keep backbone weights fixed and rely on auxiliary mechanisms, such as counterfactual models in SERAC (Mitchell et al., 2022b), lightweight modules in T-Patcher (Huang et al., 2023), or prompting and external memory in IKE, MemPrompt, WISE, and GRACE (Zheng et al., 2023a; Madaan et al., 2022; Wang et al., 2024; Hartvigsen et al., 2023). *Parameter-modifying methods* directly update internal weights. MEND (Mitchell et al., 2022a) predicts structured updates with a hypernetwork, while ROME, MEMIT, GLAME, and PMET (Meng et al., 2022; 2023; Zhang et al., 2024; Li et al., 2024c) follow a locate-then-edit paradigm for targeted modification. AlphaEdit (Fang et al., 2025) constrains updates within specific subspaces to reduce interference, while AnyEdit (Jiang et al., 2025b) improves generalization across diverse edits.

3 Method

Overview. In this section, we first formulate the problem setup, then fine-tune the visual encoder by a robustness loss to obtain robust key-value pairs, and present the dynamic key-value pairs construction strategy for adaptive key-value pair weighting. Then, we incorporate the principal component constraint strategy to mitigate catastrophic forgetting. Finally, we detail the robust optimization objective that yields a closed-form solution. The overall editing workflow is shown in Fig. 1.

Problem Setup. We consider a robust model editing setup across multiple types of visual degradations. Let $\mathcal{D} = \{D_1, D_2, \ldots, D_M\}$ denote a collection of datasets, where each D_m corresponds to a visual corruption task. We define the training set for the m-th corruption task as $S_m = \{(x^i_{clean}, x^i_{corr}, q^i, a^i)\}_{i=1}^{N_m}$, where N_m is the total number of samples in this subset, x^i_{clean} denotes the clean image, x^i_{corr} represents the corresponding visually corrupted image, q^i is the given question, and a^i is the associated ground-truth answer. For each corruption task, the robust adaptation process operates in two main stages. During the fine-tuning phase, we utilize the training set S_m to fine-tune the target layer of the visual encoder, denoted by $W \in \mathbb{R}^{d_{in} \times d_{out}}$. Subsequently, we extract its corresponding inputs and outputs as key-value pairs. During the model editing phase, these key-value pairs are utilized to compute the updated model weights W^*.

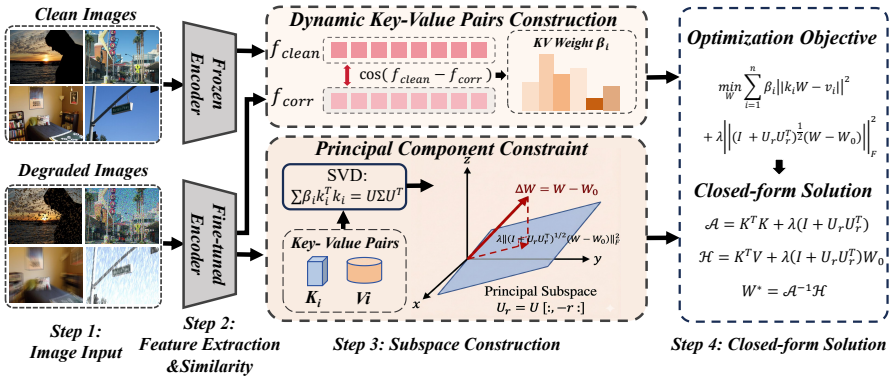

Figure 1: Overview of the proposed model editing framework. It consists of two core modules: (a) **Dynamic Key-Value Pairs Construction Strategy**, which adaptively weights key-value pairs based on sample quality to ensure that high-quality samples exert a dominant influence on the model update process. (b) **Principal Component Constraint Strategy**, which regularizes parameter updates along the leading principal directions to mitigate catastrophic forgetting.

Visual Encoder Fine-tuning. Following existing model editing methods that represent knowledge through key-value associations in FFN layers, we apply this formulation to the visual encoder by selecting the input and output projection matrices, W_{in} and W_{out}, as the editable parameters while keeping the remaining parameters frozen. Given a corrupted image x_{corr} and its clean counterpart x_{clean}, the trainable visual encoder extracts $\boldsymbol{f}_{corr}$, while a frozen copy of the encoder provides the reference feature $\boldsymbol{f}_{clean}$. To encourage the corrupted image to produce features consistent with those of the clean image, we define a feature alignment loss as $\mathcal{L}_r = 1 - \cos(\boldsymbol{f}_{corr}, \boldsymbol{f}_{clean})$. This loss is optimized together with the standard next-token prediction loss $\mathcal{L}_{next}$, so that the editable parameters $\theta_i = W_{in}, W_{out}$ are updated according to

$$\min_{\theta_i} \mathcal{L}_{next}(\theta_i; S_i) + \mathcal{L}_r(\theta_i; S_i). \tag{1}$$

Key-Value Pair Construction. After fine-tuning the visual encoder, we feed corrupted images from the training set S_i into the fine-tuned encoder and extract the inputs and outputs of the target layers as key-value pairs. In limited-data scenarios, each sample has a relatively large influence on model editing, making the update sensitive to low-quality samples. To reduce their impact, we assign each sample an adaptive weight based on the similarity between its corrupted and clean representations. Specifically, the fine-tuned encoder extracts $\boldsymbol{f}_{corr}$ from the corrupted image, while the frozen original encoder extracts the reference feature $\boldsymbol{f}_{clean}$ from the corresponding clean image. Their cosine similarity is used as the sample weight, $\beta_j = \frac{1+\cos(\boldsymbol{f}_{corr}, \boldsymbol{f}_{clean})}{2}$. A higher similarity indicates that the corrupted representation remains closer to its clean counterpart and is more reliable for model editing. To avoid explicitly storing token-wise features, we reshape the layer inputs and outputs of each sample j into matrices k_j and v_j and accumulate their weighted second-order statistics:

$$K^\top K = \sum_{j=1}^{|N|} \beta_j k_j^\top k_j, \quad K^\top V = \sum_{j=1}^{|N|} \beta_j k_j^\top v_j, \tag{2}$$

where $|N|$ denotes the number of samples across visual corruption tasks. In this way, more reliable samples contribute more to the weight update, while the influence of low-quality samples is reduced.

Principal Component Constraint Strategy. Model editing may cause the updated parameters to deviate from the original model, leading to the forgetting of pre-trained knowledge. Existing methods often introduce additional key-value pairs associated with original knowledge to constrain this deviation. Instead, we directly constrain parameter updates along the principal feature directions of the pre-trained model. Specifically, we characterize the feature distribution using the covariance matrix $K^\top K$ and perform eigen-decomposition as $K^\top K = U\Sigma U^\top$, where the eigenvectors in U represent the principal directions. We select the smallest number r of eigenvectors whose eigenvalues account for at least 98% of the total variance to construct the principal subspace U_r. Empirically, the feature spaces of corrupted and clean features are highly similar, and using principal directions from corrupted features yields comparable performance. Therefore, we only retain the corrupted

Algorithm 1 Robust Visual Encoder Editing

Input: Pre-trained MLLM (parameters Θ), editable parameters $W_0 \in \mathbb{R}^{d_{in} \times d_{out}}$, support set $\mathcal{S} = \{(\boldsymbol{x}_{corr}, \boldsymbol{x}_{clean}, q, a)\}$, Hyperparameters: λ, r,ρ
Output: Updated editable weight W^*

$K^\top K \leftarrow \mathbf{0}_{d_{in} \times d_{in}}, \quad K^\top V \leftarrow \mathbf{0}_{d_{in} \times d_{out}}$
for each mini-batch $(\boldsymbol{x}_{corr}, \boldsymbol{x}_{clean})$ from $\mathcal{S}$ **do**
  Forward pass to obtain features $\boldsymbol{f}_{corr}, \boldsymbol{f}_{clean}$ and KV pairs K_b, V_b
  $\boldsymbol{\beta} \leftarrow (1 + \cos(\boldsymbol{f}_{clean}, \boldsymbol{f}_{corr}))/2$ ▷ Compute sample-wise confidence weights
  $K^\top K \leftarrow K^\top K + \boldsymbol{\beta} K_b^\top K_b$ ▷ Accumulate weighted covariance $K^\top K$
  $K^\top V \leftarrow K^\top V + \boldsymbol{\beta} K_b^\top V_b$ ▷ Accumulate weighted cross-covariance $K^\top V$
end for
Perform eigen-decomposition: $K^\top K = U \Sigma U^\top$
Determine the smallest r satisfying: $\frac{\sum_{i=1}^{r} \sigma_i}{\sum_{i=1}^{d} \sigma_i} \geq \rho$
$U_r \leftarrow U[:, :r]$ ▷ Select top-r eigenvectors (principal directions)
$\mathcal{A} \leftarrow K^\top K + \lambda(I + U_r U_r^\top)$ ▷ Construct regularized LHS
$W^* \leftarrow \mathcal{A}^{-1}(K^\top V + \lambda(I + U_r U_r^\top) W_0)$ ▷ Solve via SPD linear system
return W^*

feature $K^\top K$ in our experiments. We then constrain parameter changes along this subspace using $\|U_r^\top (W - W_0)\|_F^2$, where W_0 and W denote the original and updated weights, respectively. This constraint reduces interference with pre-trained knowledge while improving robustness to visual corruptions without requiring additional preservation samples.

Optimization and Algorithm. The overall editing optimization objective is defined as:

$$\min_{W} \sum_{i=1}^{n} \beta_i \|k_i W - v_i\|^2 + \lambda \left\| (I + U_r U_r^\top)^{1/2} (W - W_0) \right\|_F^2, \tag{3}$$

where W_0 and W denote the original and updated parameters of the visual encoder, respectively. The scalar β_i is the dynamic KV weight for the i-th sample. U_r contains the top-r eigenvectors of the covariance matrix $K^\top K$, where r is the minimum number of components explaining 98% of the total variance, thereby capturing the principal directions of pre-trained knowledge. The hyperparameter λ balances new-knowledge fitting and knowledge preservation within this principal subspace. The complete robust model editing procedure is summarized in **Algorithm 1**. Taking the derivative of the robust optimization objective (Eq. 3) with respect to W and setting it to zero yields the following linear system:

$$\mathcal{A} = K^\top K + \lambda(I + U_r U_r^\top), \quad \mathcal{A} \in \mathbb{R}^{d_{in} \times d_{in}}. \tag{4}$$

For $\lambda > 0$, the matrix $\mathcal{A}$ is strictly symmetric and positive definite, guaranteeing invertibility. The closed-form solution for the optimal updated weights W^* is given by:

$$W^* = \mathcal{A}^{-1}\big(K^\top V + \lambda(I + U_r U_r^\top) W_0\big). \tag{5}$$

Here, $K^\top K$ and $K^\top V$ represent the aggregated dynamic key-value statistics. The term $\lambda(I + U_r U_r^\top) W_0$ acts as a regularized anchor, pulling the updated weights towards the original pre-trained parameters W_0, with a strictly stronger pull along the principal feature directions defined by $U_r U_r^\top$.

Finally, the edited projection is obtained as $W_{edit} = W^*$, optimally integrating the new robust knowledge while preserving pre-trained capabilities. This enables highly efficient optimization, ensuring stable model updates with constant memory cost.

4 VCBench: A Visual Corruption Benchmark for Evaluating Authentic MLLM Robustness

Existing benchmarks (Masry et al., 2022; Yu et al., 2024b; Li et al., 2024b) for MLLMs frequently include knowledge-intensive questions that can be answered correctly relying solely on language priors. Consequently, performance on such benchmarks fails to accurately reflect the true visual

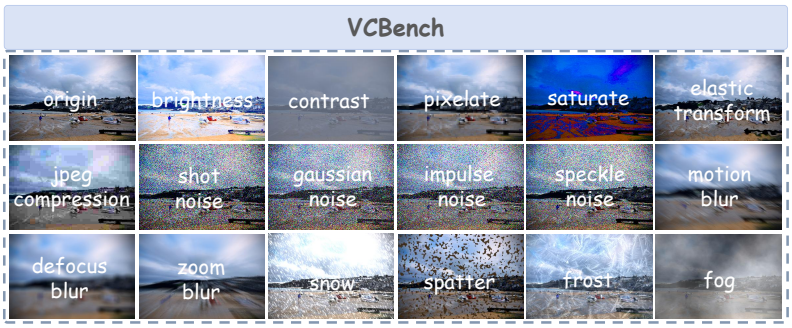

Figure 2: Overview of the proposed VCBench. The benchmark systematically categorizes 17 diverse real-world visual corruptions into four distinct groups: Optical Blurs, Digital Variations, Weather Degradations, and Sensor Noise.

robustness of MLLMs. To address this critical limitation, we propose a novel visual corruption benchmark: **VCBench**, to effectively evaluate the visual robustness of MLLMs.

Data Filtering. To guarantee that a model's correct prediction stems from robust visual comprehension rather than strong language priors, we rigorously filter existing benchmarks. Specifically, we compile data from five mainstream benchmarks: MMBench (Liu et al., 2024), SEEDBench (Li et al., 2023a), A-OKVQA (Schwenk et al., 2022), RealWorldQA (xAI, 2024), and WorldMedQA-V (Matos et al., 2025), and evaluate them using reference models. We specifically isolate instances where the reference model answers correctly on the clean image but fails on its perturbed image. This filtering rule ensures the exclusion of questions that can be resolved solely through language priors. Consequently, this curated subset of questions provides a more direct and accurate reflection of a model's authentic visual robustness.

Comprehensive Visual Corruption Generation. To the filtered data, we apply 17 common visual corruptions to generate the final evaluation data. To faithfully reflect real-world challenges, we categorize these corruptions into four groups: (1) Weather Degradations (e.g., snow, fog, spatter, frost) simulating harsh outdoor conditions; (2) Sensor Noise (e.g., gaussian noise, shot noise, impulse noise, speckle noise) mirroring low-light or hardware interference; (3) Optical Blurs (e.g., motion blur, defocus blur, zoom blur) capturing camera shake or focus failure; and (4) Digital & Illumination Variations (e.g., JPEG compression, contrast, pixelate, saturate, brightness, elastic transform) reflecting bandwidth constraints and extreme lighting. This taxonomy ensures our benchmark holistically evaluates MLLMs against physical, optical, and digital vulnerabilities.

5 Experiments

Training Setting. Our model is built upon Qwen2.5-VL-7B (Bai et al., 2025b). Our training set includes 17 common visual corruptions, with each corruption type utilizing 100 training samples. During the training phase, we fine-tune the `up_proj` and `down_proj` layers within the FFN layers of the visual encoder, while keeping the remaining model parameters frozen.

Benchmarks. We evaluate our method along three dimensions: (i) **Real-World Robustness** on R-Bench (Li et al., 2025a); (ii) **Synthetic Robustness** under diverse visual corruptions on VCBench and the Visual Robustness Benchmark for VQA (Ishmam et al., 2025); and (iii) **Pre-trained Knowledge Retention** on A-OKVQA (Schwenk et al., 2022), MMBench (Liu et al., 2024), RealWorldQA (xAI, 2024), and SEEDBench (Li et al., 2023a). This evaluation jointly measures robustness to real and synthetic corruptions and the preservation of general capabilities after editing.

Performance on VCBench. As shown in Table 1, we compare our method with general MLLMs, continual learning methods, robust MLLMs, and model editing methods. Compared with Qwen2.5-VL-7B (Bai et al., 2025b), our method improves the average accuracy by 7.56%. It also outperforms the continual learning method SD-LoRA (Wu et al., 2025), the robust MLLM Robust-R1 (Tang et al., 2026), and the model editing method AlphaEdit (Fang et al., 2025) by 3.19%, 2.89%, and 4.27%, respectively. To evaluate the applicability of our framework across model scales, we further apply it to Qwen2.5-VL-3B (Bai et al., 2025b), where it improves the average accuracy on VCBench

Table 1: Comparison with general MLLMs, continual learning methods, and model editing methods on VCBench grouped by four main categories: Noise, Blur, Weather, and Digital. The best results are highlighted in **bold**. The fine-tuned upper bound (Oracle) is shown in gray for reference.

Method	Noise	Blur	Weather	Digital	Avg.
Qwen2.5-VL-3B (Bai et al., 2025b)	12.36	17.69	23.14	28.58	21.88
Qwen2.5-VL-7B (Bai et al., 2025b)	13.36	17.09	25.28	31.63	23.64
MEMIT (Meng et al., 2023)	16.38	19.84	27.42	33.69	26.07
AlphaEdit (Fang et al., 2025)	16.84	20.38	26.85	34.09	26.93
ROME (Meng et al., 2022)	16.02	18.48	27.46	33.06	25.49
NSE (Jiang et al., 2025c)	15.60	20.05	25.53	33.58	26.20
ZSCL (Zheng et al., 2023b)	17.82	19.52	25.92	29.37	24.30
EWC (Kirkpatrick et al., 2017)	16.76	18.84	26.33	28.82	23.78
SD-LoRA (Wu et al., 2025)	17.78	22.71	28.56	35.89	28.01
WISE (Wang et al., 2024)	15.58	20.19	30.00	35.07	25.21
GRACE (Hartvigsen et al., 2023)	14.67	20.62	28.90	34.41	24.65
Robust-R1 (Tang et al., 2026)	17.10	22.10	27.17	32.32	26.02
Robust-LLaVA (Malik et al., 2025)	18.25	21.13	24.65	27.41	22.86
Ours (Qwen2.5-VL-3B)	**18.92**$^{+6.56}$	24.27$^{+6.58}$	35.39$^{+12.25}$	33.36$^{+4.78}$	28.91$^{+7.03}$
Ours (Qwen2.5-VL-7B)	18.64$^{+5.28}$	**26.30**$^{+9.21}$	**36.22**$^{+10.94}$	**38.21**$^{+6.58}$	**31.20**$^{+7.56}$
Oracle	19.29	27.44	39.20	39.47	32.50

Table 2: Comparison result on R-Bench on MCQ/VQA/CAP tasks at varying difficulty levels (Low, Mid, High). The best results are highlighted in **bold**.

Method	Data	Params.	MCQ			VQA			CAP			Overall
			Low	Mid	High	Low	Mid	High	Low	Mid	High	
Qwen2.5-VL-3B (Bai et al., 2025b)	-	-	0.6411	0.6022	0.5732	0.4872	0.4854	0.4904	0.3778	0.3704	0.3330	0.4845
Gemma3-4B (Team et al., 2025)	-	-	0.5823	0.5776	0.5060	0.4865	0.4630	0.4419	0.4048	0.3746	0.3480	0.4649
InternVL-4B (Chen et al., 2024)	-	-	0.6235	0.6024	0.5914	**0.4982**	0.4539	**0.5108**	0.3667	0.3041	0.2851	0.4706
TeCoA (Mao et al., 2023)	1280K	304M	0.4647	0.4223	0.4024	0.4687	0.3994	0.4461	0.2111	0.2195	0.1937	0.3586
Robust CLIP (Schlarmann et al., 2024)	1280K	304M	0.4705	0.4658	0.4024	0.4503	0.4339	0.4743	0.2290	0.2219	0.1983	0.3718
Robust LLaVA (Malik et al., 2025)	1200K	7000M	0.3352	0.2608	0.3048	0.2607	0.2212	0.2443	0.0068	0.0065	0.0067	0.1830
Robust-R1 (SFT) (Tang et al., 2026)	2.5K	3090M	0.6176	0.6087	0.5610	0.4804	0.4836	0.5012	0.4080	0.3858	0.3518	0.4886
Robust-R1 (SFT and RL) (Tang et al., 2026)	10K	3090M	**0.6529**	**0.6391**	**0.6097**	0.4914	**0.4909**	0.4980	0.4068	0.3781	0.3484	**0.5017**
Ours	**1.7K**	**280M**	0.6471	0.6087	0.5610	0.4534	0.4721	0.4916	**0.4599**	**0.4462**	**0.3703**	0.5014

by 7.03%. We also evaluate independently fine-tuned models on each visual corruption task as an empirical upper bound. Our method achieves an average accuracy of 31.20%, compared with the upper bound of 32.50%, corresponding to a performance retention rate of 96.0%. These results indicate that our method consistently improves robustness while retaining most of the performance achieved by task-specific fine-tuning.

Performance on R-Bench. As shown in Table 2, we evaluate our method on R-Bench across three distinct tasks (MCQ, VQA, and CAP) at varying difficulty levels (Low, Mid, and High). We compare our approach with typical general MLLMs and robust MLLMs. Compared with the base model Qwen2.5-VL-3B, integrating our editing strategy yields an absolute overall accuracy improvement of 1.69%. Furthermore, when evaluated against traditional robust VLMs, TeCoA (Mao et al., 2023) and Robust CLIP (Schlarmann et al., 2024), our method achieves substantial improvements of 14.28% and 12.96%, respectively. Notably, our method effectively matches the performance of the current SOTA robust MLLM, Robust-R1, despite the latter relying on highly resource-intensive supervised fine-tuning and reinforcement learning phases and a massive number of trainable parameters. The experimental results demonstrate that our model editing strategy not only preserves foundational QA capabilities but also fundamentally enhances the model's visual robustness.

Performance on Standard Benchmarks. To further evaluate pre-trained knowledge retention after editing, we compare our method with existing continual learning and model editing approaches on four standard benchmarks: A-OKVQA, MMBench, RealWorldQA, and SEEDBench. For answer extraction, we use a rule-based strategy: (i) identify responses formatted as "The correct answer is: [X]"; (ii) if no match is found, check whether the response begins with an option letter; and (iii) otherwise, use the last occurring option letter as the final prediction. We use the original Qwen2.5-VL-7B as an empirical upper bound. As shown in Table 3, our method achieves 86.11% on A-OKVQA and 83.41% on MMBench, consistently outperforming AlphaEdit. Compared with ZSCL (Zheng et al., 2023b), it improves average accuracy by 21.16% on A-OKVQA and 22.04% on MMBench. These results show that our principal component constraint effectively mitigates catastrophic forgetting while improving visual robustness and preserving pre-trained capabilities.

Table 3: Comparison result on standard benchmarks. The gray row serves as the theoretical upper bound for performance on original clean data. The best results are highlighted in **bold**.

Method	SEEDBench	A-OKVQA	RealWorldQA	MMBench
Qwen2.5-VL-7B	77.08	88.73	67.97	86.23
AlphaEdit (Fang et al., 2025)	**75.30**	85.50	62.88	81.17
MEMIT (Meng et al., 2023)	74.72	85.33	60.92	81.67
ROME (Meng et al., 2022)	62.18	75.34	51.07	74.20
NSE (Jiang et al., 2025c)	74.13	83.61	59.41	82.03
ZSCL (Zheng et al., 2023b)	62.49	64.95	54.92	61.37
EWC (Kirkpatrick et al., 2017)	54.31	59.42	46.29	58.26
SD-LoRA (Wu et al., 2025)	72.36	83.63	61.62	82.01
Ours	75.13	**86.11**	**64.18**	**83.41**

Table 4: Ablation study on VCBench. Rob. Loss denotes robustness loss. SVD denotes the principal component constraint strategy. Dyn. KV denotes the dynamic key-value pairs construction strategy. We report average performance grouped by corruption categories: Noise, Blur, Weather, and Digital. The second row shows results from optimizing the model editing objective $\min_W \sum_{i=1}^{n} \|k_i W - v_i\|^2 + \lambda \|W - W_0\|_F^2$. The best result is highlighted in **bold**.

Rob. Loss	SVD	Dyn. KV	Noise	Blur	Weather	Digital	Avg.
×	×	×	13.36	17.09	25.28	31.63	23.64
✓	×	×	18.00	26.15	33.43	36.71	29.28
✓	✓	×	**18.72**	**27.85**	35.13	36.11	30.39
✓	✓	✓	18.64	26.30	**36.22**	**38.21**	**31.20**

Ablation Study. To validate the individual contributions of our proposed components, we conduct an incremental ablation study on VCBench. As shown in Table 4, we progressively integrate the robustness loss, dynamic key-value pairs construction strategy, and principal component constraint strategy into the Qwen2.5-VL-7B model. Initially, the base model achieves an average accuracy of 23.64%. Integrating the robustness loss improves the overall performance to 29.28%, demonstrating its effectiveness in encouraging the model to align corrupted image features into the clean feature space. Building upon this, the addition of the principal component constraint strategy boosts the overall performance to 30.39%, confirming that regularizing parameter updates along critical feature directions is essential for mitigating catastrophic forgetting when injecting new robust knowledge. Finally, incorporating the dynamic key-value pairs construction strategy further elevates the average accuracy to 31.20%, demonstrating that the dynamic weighting mechanism effectively magnifies the contribution of high-quality samples during parameter optimization.

Justification of PCA-based Knowledge Preservation. As shown in Table 5, the corrupted and clean $K^T K$ matrices have highly similar projection subspaces, with layer-wise similarities consistently above 0.91. This indicates that using corrupted-sample $K^T K$ can effectively approximate clean-sample $K^T K$ for knowledge preservation.

Different Fine-tuning Strategy Analysis. To determine the most efficient fine-tuning strategy for knowledge injection, we investigate the impact of fine-tuning different sub-modules within the visual encoder. As shown in Fig. 3, fine-tuning both the input and output projection layers within the FFN yields superior performance compared to fine-tuning either layer individually. This demonstrates that updating more modules facilitates a more effective injection of new knowledge. However, fine-tuning the entire visual encoder leads to a substantial degradation in performance, indicating that full fine-tuning of the encoder triggers severe overfitting. Consequently, the model largely forgets its pre-trained knowledge, leading to poor performance even on visual corruption tasks.

Directional Analysis of Model Updates. To better understand the behavior of our method during model editing, we analyze the direction of its parameter updates by comparing ΔW with the new-knowledge direction d_1 and the old-knowledge subspace derived from K_0. Specifically, we calculate the cosine similarity between the weight update $\Delta W = W - W_0$ and the optimal new-knowledge direction $d_1 = \arg\min_d \|K_1 d - (V_1 - K_1 W_0)\|_2^2$. Meanwhile, we quantify the projection of ΔW onto the old-knowledge subspace using the metric $\mathrm{OldProj} = \|K_0 \Delta W\|_F / (\|K_0\|_F \|\Delta W\|_F)$. As shown in Fig. 4, $\cos(\Delta W, d_1)$ consistently maintains a high value, while $\mathrm{OldProj}$ remains close to zero. This demonstrates that our method concentrates its updates along new-knowledge directions while preserving the integrity of the old-knowledge subspace.

Table 5: Projection subspace similarity between the $K^\top K$ matrices computed from clean and corrupted features across different layers.

Layer	0	4	8	16	24	32
Similarity	0.959	0.965	0.954	0.937	0.924	0.919

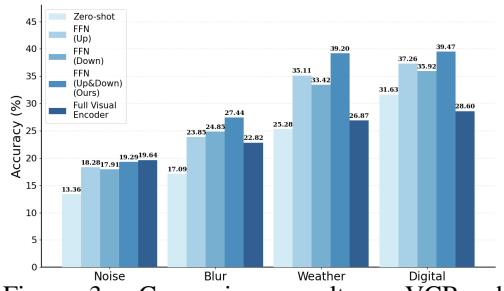

Figure 3: Comparison result on VCBench across different fine-tuning strategies within the visual encoder.

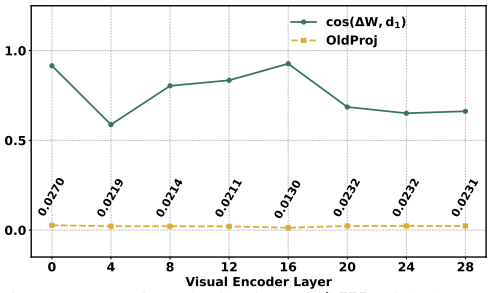

Figure 4: Cosine similarity of ΔW with the target direction d_1 and its projection on the old-knowledge subspace $OldProj$.

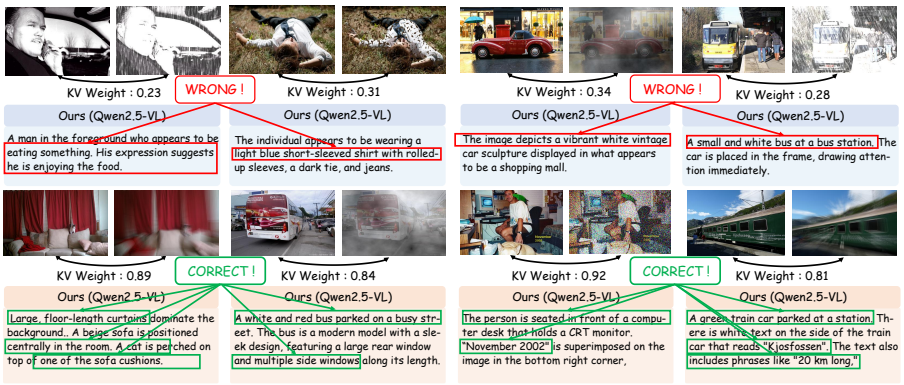

Figure 5: Qualitative validation of the dynamic KV weighting strategy. **Top row:** For the low KV weight samples, the generated caption exhibits severe hallucinations and semantic errors. **Bottom row:** For the high kv weight samples, the model produces an accurate and detailed caption. The results confirm that our strategy can accurately evaluate feature quality to achieve adaptive weighting.

Qualitative Validation of Dynamic KV Weighting. To qualitatively evaluate our dynamic key-value pair construction strategy, we use the fine-tuned visual encoder to extract features from corrupted images and generate corresponding captions. This allows us to examine the relationship between representation quality and the assigned KV weights. As shown in Fig. 5, the top-row sample receives a low KV weight and produces a caption with semantic errors and hallucinations, indicating poor feature quality. In contrast, the bottom-row sample receives a high KV weight and generates a more accurate and informative caption, suggesting better feature representations. These results show that the assigned KV weights are consistent with the quality of the extracted visual features across different corruption conditions.

6 Conclusion

In this paper, we present a model editing framework to enhance MLLM robustness under limited data. We first introduce a robustness loss that aligns corrupted features with the clean representation space. We then optimize knowledge injection through a dynamic key-value pair construction strategy, allowing high-quality samples to contribute more to model updates. Additionally, we apply a principal component constraint to regularize parameter optimization and mitigate catastrophic forgetting. Finally, we introduce VCBench, a visual corruption benchmark with 17 real-world corruptions, to evaluate visual robustness without language-prior bias. Extensive evaluations demonstrate the effectiveness of our framework, providing a scalable solution for building robust MLLMs.

AI USE STATEMENT

In this work, we used generative AI tools solely for language polishing and editing, including refining wording, rephrasing sentences, checking grammar, and enhancing readability. These tools were not used to formulate research ideas, design experiments, analyze data, interpret results, or make any other scientific contributions. We reviewed all AI-assisted content to ensure its correctness, clarity, and consistency. All research ideas were conceived by the authors, who also conducted the experiments and analyses and drew the conclusions. We take full responsibility for the final content of this work, including any text revised or polished with the assistance of generative AI tools.

REFERENCES

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

Muhammad Awais, Muzammal Naseer, Salman Khan, Rao Muhammad Anwer, Hisham Cholakkal, Mubarak Shah, Ming-Hsuan Yang, and Fahad Shahbaz Khan. Foundation models defining a new era in vision: a survey and outlook. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 47(4):2245–2264, 2025.

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, et al. Qwen3-vl technical report. *arXiv preprint arXiv:2511.21631*, 2025a.

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Mingkun Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, Jiabo Ye, Xi Zhang, Tianbao Xie, Zesen Cheng, Hang Zhang, Zhibo Yang, Haiyang Xu, and Junyang Lin. Qwen2.5-vl technical report, 2025b.

Zhe Chen, Jiannan Wu, Wenhai Wang, Weijie Su, Guo Chen, Sen Xing, Muyan Zhong, Qinglong Zhang, Xizhou Zhu, Lewei Lu, et al. Internvl: Scaling up vision foundation models and aligning for generic visual-linguistic tasks. In *CVPR*, 2024.

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

Chaoyou Fu, Peixian Chen, Yunhang Shen, Yulei Qin, Mengdan Zhang, Xu Lin, Jinrui Yang, Xiawu Zheng, Ke Li, Xing Sun, et al. Mme: A comprehensive evaluation benchmark for multimodal large language models. In *NeurIPS*, 2025.

Govind Krishnan Gangadhar and Karl Stratos. Model editing by standard fine-tuning. In *ACL*, 2024.

Bernal Jiménez Gutiérrez, Yiheng Shu, Weijian Qi, Sizhe Zhou, and Yu Su. From rag to memory: Non-parametric continual learning for large language models. In *ICML*, 2025.

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

Jiangpeng He, Zhihao Duan, and Fengqing Zhu. Cl-lora: Continual low-rank adaptation for rehearsal-free class-incremental learning. In *CVPR*, 2025.

Zheqi He, Xinya Wu, Pengfei Zhou, Richeng Xuan, Guang Liu, Xi Yang, Qiannan Zhu, and Hua Huang. Cmmu: a benchmark for chinese multi-modal multi-type question understanding and reasoning. In *IJCAI*, 2024.

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

Farhan Ishmam, Ishmam Tashdeed, Talukder Asir Saadat, Hamjajul Ashmafee, Abu Raihan Mostofa Kamal, and Azam Hossain. Visual robustness benchmark for visual question answering (vqa). In *Proceedings of the Winter Conference on Applications of Computer Vision*, pp. 6623–6633, 2025.

Chengze Jiang, Zhuangzhuang Wang, Minjing Dong, and Jie Gui. Survey of adversarial robustness in multimodal large language models, 2025a.

Houcheng Jiang, Junfeng Fang, Ningyu Zhang, Guojun Ma, Mingyang Wan, Xiang Wang, Xiangnan He, and Tat seng Chua. Anyedit: Edit any knowledge encoded in language models, 2025b.

Houcheng Jiang, Junfeng Fang, Tianyu Zhang, Baolong Bi, An Zhang, Ruipeng Wang, Tao Liang, and Xiang Wang. Neuron-level sequential editing for large language models. In *ACL*, 2025c.

Dongjin Kim, Jaekyun Ko, Muhammad Kashif Ali, and Tae Hyun Kim. Idf: Iterative dynamic filtering networks for generalizable image denoising. In *ICCV*. IEEE, 2025.

James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, 2017.

Bohao Li, Rui Wang, Guangzhi Wang, Yuying Ge, Yixiao Ge, and Ying Shan. Seed-bench: Benchmarking multimodal llms with generative comprehension. *arXiv preprint arXiv:2307.16125*, 2023a.

Chunyi Li, Jianbo Zhang, Zicheng Zhang, Haoning Wu, Yuan Tian, Wei Sun, Guo Lu, Xiongkuo Min, Xiaohong Liu, Weisi Lin, et al. R-bench: Are your large multimodal model robust to real-world corruptions? *IEEE Journal of Selected Topics in Signal Processing*, 2025a.

Feng Li, Renrui Zhang, Hao Zhang, Yuanhan Zhang, Bo Li, Wei Li, Zejun Ma, and Chunyuan Li. Llava-next-interleave: Tackling multi-image, video, and 3d in large multimodal models. *arXiv preprint arXiv:2407.07895*, 2024a.

Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *International conference on machine learning*, pp. 19730–19742. PMLR, 2023b.

Kunchang Li, Yali Wang, Yinan He, Yizhuo Li, Yi Wang, Yi Liu, Zun Wang, Jilan Xu, Guo Chen, Ping Luo, et al. Mvbench: A comprehensive multi-modal video understanding benchmark. In *CVPR*, 2024b.

Xiaopeng Li, Shasha Li, Shezheng Song, Jing Yang, Jun Ma, and Jie Yu. Pmet: Precise model editing in a transformer. In *AAAI*, 2024c.

Xiaopeng Li, Shangwen Wang, Shasha Li, Jun Ma, Jie Yu, Xiaodong Liu, Jing Wang, Bin Ji, and Weimin Zhang. Model editing for llms4code: How far are we? In *ICSE*, 2025b.

Yan-Shuo Liang and Wu-Jun Li. Inflora: Interference-free low-rank adaptation for continual learning. In *CVPR*, 2024.

Kevin Qinghong Lin, Pengchuan Zhang, Joya Chen, Shraman Pramanick, Difei Gao, Alex Jinpeng Wang, Rui Yan, and Mike Zheng Shou. Univtg: Towards unified video-language temporal grounding. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 2794–2804, 2023.

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

Xuan Liu and Xiaobin Chang. Lora subtraction for drift-resistant space in exemplar-free continual learning. In *CVPR*, 2025.

Yuan Liu, Haodong Duan, Yuanhan Zhang, Bo Li, Songyang Zhang, Wangbo Zhao, Yike Yuan, Jiaqi Wang, Conghui He, Ziwei Liu, et al. Mmbench: Is your multi-modal model an all-around player? In *ECCV*, 2024.

Pan Lu, Swaroop Mishra, Tanglin Xia, Liang Qiu, Kai-Wei Chang, Song-Chun Zhu, Oyvind Tafjord, Peter Clark, and Ashwin Kalyan. Learn to explain: Multimodal reasoning via thought chains for science question answering. *NeurIPS*, 2022.

Ke Ma, Jiaqi Tang, Bin Guo, Fan Dang, Sicong Liu, Zhui Zhu, Lei Wu, Cheng Fang, Ying-Cong Chen, Zhiwen Yu, et al. Surgeon: Memory-adaptive fully test-time adaptation via dynamic activation sparsity. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 30514–30523, 2025.

Aman Madaan, Niket Tandon, Peter Clark, and Yiming Yang. Memory-assisted prompt editing to improve gpt-3 after deployment. In *EMNLP*, pp. 2833–2861, 2022.

Hashmat Shadab Malik, Fahad Shamshad, Muzammal Naseer, Karthik Nandakumar, Fahad Khan, and Salman Khan. Robust-llava: On the effectiveness of large-scale robust image encoders for multi-modal large language models. *arXiv preprint arXiv:2502.01576*, 2025.

Chengzhi Mao, Scott Geng, Junfeng Yang, Xin Wang, and Carl Vondrick. Understanding zero-shot adversarial robustness for large-scale models. In *ICLR*, 2023.

Ahmed Masry, Xuan Long Do, Jia Qing Tan, Shafiq Joty, and Enamul Hoque. Chartqa: A benchmark for question answering about charts with visual and logical reasoning. In *ACL*, 2022.

João Matos, Shan Chen, Siena Kathleen V Placino, Yingya Li, Juan Carlos Climent Pardo, Daphna Idan, Takeshi Tohyama, David Restrepo, Luis Filipe Nakayama, José María Millet Pascual-Leone, et al. Worldmedqa-v: a multilingual, multimodal medical examination dataset for multimodal language models evaluation. In *NAACL*, 2025.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. Locating and editing factual associations in gpt. *NeurIPS*, 35:17359–17372, 2022.

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

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

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

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pp. 8748–8763. PmLR, 2021.

Christian Schlarmann, Naman Deep Singh, Francesco Croce, and Matthias Hein. Robust clip: Unsupervised adversarial fine-tuning of vision embeddings for robust large vision-language models. In *ICML 2024*, 2024.

Dustin Schwenk, Apoorv Khandelwal, Christopher Clark, Kenneth Marino, and Roozbeh Mottaghi. A-okvqa: A benchmark for visual question answering using world knowledge. In *ECCV*, 2022.

Jiaqi Tang, Jianmin Chen, Wei Wei, Xiaogang Xu, Runtao Liu, Xiangyu Wu, Qipeng Xie, Jiafei Wu, Lei Zhang, and Qifeng Chen. Robust-r1: Degradation-aware reasoning for robust visual understanding. In *AAAI*, 2026.

Longxiang Tang, Zhuotao Tian, Kai Li, Chunming He, Hantao Zhou, Hengshuang Zhao, Xiu Li, and Jiaya Jia. Mind the interference: Retaining pre-trained knowledge in parameter efficient continual learning of vision-language models. In *ECCV*, 2024.

Gemini Team, Petko Georgiev, Ving Ian Lei, Ryan Burnell, Libin Bai, Anmol Gulati, Garrett Tanzer, Damien Vincent, Zhufeng Pan, Shibo Wang, et al. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *arXiv preprint arXiv:2403.05530*, 2024.

Gemma Team, Aishwarya Kamath, Johan Ferret, Shreya Pathak, Nino Vieillard, Ramona Merhej, Sarah Perrin, et al. Gemma 3 technical report. *arXiv preprint arXiv:2503.19786*, 2025.

Jiankang Wang, Zhihan Zhang, Zhihang Liu, Yang Li, Jiannan Ge, Hongtao Xie, and Yongdong Zhang. Spacevllm: Endowing multimodal large language model with spatio-temporal video grounding capability. *arXiv preprint arXiv:2503.13983*, 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. *NeurIPS*, 37:53764–53797, 2024.

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. In *ICLR*, 2025.

xAI. Grok-1.5 vision preview, 2024.

Ziruo Yi, Ting Xiao, and Mark V Albert. A survey on multimodal large language models in radiology for report generation and visual question answering. *Information*, 16(2):136, 2025.

Chung-En Johnny Yu, Hsuan-Chih, Chen, Brian Jalaian, and Nathaniel D. Bastian. Orca: Agentic reasoning for hallucination and adversarial robustness in vision-language models, 2025.

Jiazuo Yu, Yunzhi Zhuge, Lu Zhang, Ping Hu, Dong Wang, Huchuan Lu, and You He. Boosting continual learning of vision-language models via mixture-of-experts adapters. In *CVPR*, 2024a.

Weihao Yu, Zhengyuan Yang, Linjie Li, Jianfeng Wang, Kevin Lin, Zicheng Liu, Xinchao Wang, and Lijuan Wang. Mm-vet: evaluating large multimodal models for integrated capabilities. In *ICML*, 2024b.

Xiang Yue, Yuansheng Ni, Kai Zhang, Tianyu Zheng, Ruoqi Liu, Ge Zhang, Samuel Stevens, Dongfu Jiang, Weiming Ren, Yuxuan Sun, et al. Mmmu: A massive multi-discipline multimodal understanding and reasoning benchmark for expert agi. In *CVPR*, 2024.

Mengqi Zhang, Xiaotian Ye, Qiang Liu, Pengjie Ren, Shu Wu, and Zhumin Chen. Knowledge graph enhanced large language model editing. In *EMNLP*, pp. 22647–22662, 2024.

Ce Zheng, Lei Li, Qingxiu Dong, Yuxuan Fan, Zhiyong Wu, Jingjing Xu, and Baobao Chang. Can we edit factual knowledge by in-context learning? In *EMNLP*, pp. 4862–4876, 2023a.

Zangwei Zheng, Mingyuan Ma, Kai Wang, Ziheng Qin, Xiangyu Yue, and Yang You. Preventing zero-shot transfer degradation in continual learning of vision-language models. In *ICCV*, 2023b.

Hao Zhu, Yifei Zhang, Junhao Dong, and Piotr Koniusz. Bilora: almost-orthogonal parameter spaces for continual learning. In *CVPR*, 2025a.

Jinguo Zhu, Weiyun Wang, Zhe Chen, Zhaoyang Liu, Shenglong Ye, Lixin Gu, Hao Tian, Yuchen Duan, Weijie Su, Jie Shao, et al. Internvl3: Exploring advanced training and test-time recipes for open-source multimodal models. *arXiv preprint arXiv:2504.10479*, 2025b.

A Appendix

A.1 Impact of Sample Quality.

To effectively evaluate the impact of sample quality on model editing, we employ samples of varying quality for model editing and measure their corresponding performance on VCBench. As shown in Table 6, the model edited with high-quality samples demonstrates superior robustness across all visual corruption scenarios compared to the one edited with low-quality samples, proving that utilizing high-quality samples effectively improves model editing performance.

Table 6: Comparison result on VCBench using different quality samples for model editing. To ensure a fair comparison, the number of samples in both subsets is 50. Selecting high-quality samples ($KV\ Weight > 0.8$) consistently yields better robustness against all corruption categories compared to low-quality samples ($KV\ Weight < 0.4$). Best results are highlighted in **bold**.

Sample Strategy	Noise	Blur	Weather	Digital	Avg.
$KV\ Weight < 0.4$	15.86	25.37	31.67	34.74	28.10
$KV\ Weight > 0.8$	**17.51**	**25.87**	**33.96**	**35.60**	**29.34**

Table 7: Comparison result on VCBench across different fine-tuning strategies within the visual encoder. The best results in each column are highlighted in **bold**.

Strategy	Noise	Blur	Weather	Digital	Avg.
Baseline (Zero-shot)	13.36	17.09	25.28	31.63	23.64
FFN (Up-proj only)	18.28	23.85	35.11	37.26	30.05
FFN (Down-proj only)	17.91	24.85	33.42	35.92	29.29
FFN (Up & Down-proj, Ours)	19.29	**27.44**	**39.20**	**39.47**	**32.50**
Full Visual Encoder	**19.64**	22.82	26.87	28.60	25.16

Table 8: Comparison of robustness retention after expanding perturbation training from four to eight corruptions. Here, $\mathrm{first4}$ denotes training with the first four perturbation types (Snow, Frost, Spatter, and Fog), while $\mathrm{all8}$ denotes training with all eight perturbation types. The downward values indicate performance degradation on previously learned corruptions after introducing additional perturbation training. All results are reported in percentage (%).

Method	Snow	Frost	Spatter	Fog	Zoom Blur	Speckle Noise	Defocus Blur	Shot Noise	Retention
Robust-CLIP$_{\mathrm{first4}}$	21.36	21.23	22.36	21.06	18.19	23.93	20.54	21.67	-
Robust-CLIP$_{\mathrm{all8}}$	21.15$_{\downarrow -0.21}$	20.71$_{\downarrow -0.52}$	21.49$_{\downarrow -0.87}$	19.15$_{\downarrow -1.91}$	21.14	23.06	21.84	22.28	95.9
Robust-R1$_{\mathrm{first4}}$	23.67	24.75	25.32	24.62	19.66	22.55	20.75	18.79	-
Robust-R1$_{\mathrm{all8}}$	22.31$_{\downarrow -1.36}$	22.81$_{\downarrow -1.94}$	24.60$_{\downarrow -0.72}$	22.40$_{\downarrow -2.22}$	23.10	23.70	23.94	22.83	93.7
Ours$_{\mathrm{first4}}$	30.64	35.19	43.50	42.83	21.68	23.13	26.45	16.43	-
Ours$_{\mathrm{all8}}$	30.20$_{\downarrow -0.44}$	34.79$_{\downarrow -0.40}$	42.73$_{\downarrow -0.77}$	42.45$_{\downarrow -0.38}$	23.90	24.73	28.67	16.95	**98.7**

A.2 Different Fine-tuning Strategy Analysis.

To determine the most efficient fine-tuning strategy for knowledge injection, we investigate the impact of fine-tuning different sub-modules within the visual encoder. As shown in Table 7, fine-tuning both the input and output projection layers within the FFN yields superior performance compared to fine-tuning either layer individually. This demonstrates that updating more modules facilitates a more effective injection of new knowledge. However, fine-tuning the entire visual encoder leads to a substantial degradation in performance, indicating that full fine-tuning of the encoder triggers severe overfitting. Consequently, the model largely forgets its pre-trained knowledge, leading to poor performance even on visual corruption tasks.

A.3 Robustness Retention under Incremental Perturbation Learning.

We further evaluate whether robust MLLMs can preserve previously acquired robustness when learning additional corruptions. As shown in Table 8, expanding the training set from four to eight corruptions causes clear degradation on the originally learned corruptions for existing methods. Robust-CLIP and Robust-R1 achieve retention rates of only 95.9% and 93.7%, respectively, revealing the vulnerability to robustness forgetting during incremental training. In comparison, our method maintains a retention rate of 98.7%, with negligible performance degradation on the first four corruptions. Moreover, the overall robustness improves after incorporating new corruptions, suggesting that our approach not only alleviates catastrophic forgetting but also enables effective acquisition of new robustness capabilities.

Table 9: Comparison results on the Visual Robustness Benchmark for Visual Question Answering (VQA). The best results are highlighted in **bold**.

Method	Noise	Blur	Weather	Img Attr.	Physical	Digital
MEMIT (Meng et al., 2023)	38.99	44.48	47.55	54.21	46.28	43.36
AlphaEdit (Fang et al., 2025)	39.71	45.58	43.62	54.57	45.05	44.35
ROME (Meng et al., 2022)	37.60	43.35	44.51	52.38	45.17	45.69
SD-LoRA (Wu et al., 2025)	43.52	49.29	50.13	53.46	47.91	48.24
ZSCL (Zheng et al., 2023b)	37.94	42.55	43.24	52.16	43.85	44.08
WISE (Wang et al., 2024)	40.67	47.29	45.04	55.07	43.51	45.24
GRACE (Hartvigsen et al., 2023)	39.26	46.26	43.62	52.93	44.74	46.79
Robust LLaVA (Malik et al., 2025)	42.85	48.21	50.23	52.67	45.64	47.21
Robust-R1 (Tang et al., 2026)	47.10	50.08	53.31	58.53	53.83	53.01
Ours	**51.66**	**55.24**	**59.57**	**63.78**	**60.87**	**56.00**

Table 10: Computational efficiency of PCA, linear-system solving, and matrix inversion in our method.

d_{in}	PCA time	Solve time	Inverse time	Memory
3420	0.26 s	0.02 s	0.03 s	384.78 MB

A.4 Performance on Visual Robustness Benchmark for VQA.

Following the official construction protocol of the Visual Robustness Benchmark for VQA (Ishmam et al., 2025), we reproduce the benchmark to evaluate MLLM robustness across six corruption categories: Noise, Blur, Weather, Image Attribute, Physical, and Digital corruptions. As shown in Table 9, our method achieves the best performance across all categories. Compared with Robust-R1 (Tang et al., 2026), it improves accuracy from 47.10% to 51.66% on Noise, 50.08% to 55.24% on Blur, and 53.31% to 59.57% on Weather, while outperforming it by 5.25% and 7.04% on Image Attribute and Physical corruptions, respectively. On Physical corruption, our method also surpasses Robust LLaVA (Malik et al., 2025) and WISE (Wang et al., 2024) by 15.23% and 17.36%, respectively. On Digital corruption, it reaches 56.00%, exceeding Robust-R1 by 2.99%. These results show consistent robustness improvements across diverse visual corruptions.

Computational Efficiency of Inverting of $\mathcal{A}$. In our method, computing the inversion does not introduce significant computational latency. The matrix inversion is directly performed on $\mathcal{A} \in \mathbb{R}^{d_{in} \times d_{in}}$, rather than on the full weight matrix W. As shown in Table 10, the inversion of $\mathcal{A}$ is very fast in practice, introducing a small additional computational overhead during model editing.

Details of VCBench. VCBench consists of 19,567 corrupted instances constructed from MMBench, SEEDBench, A-OKVQA, RealWorldQA, and WorldMedQA-V. We use two MLLMs for filtering. We divide the 17 corruptions into 6 groups, and the final group having 2 corruptions. For each group, we retain samples that are answered correctly with the clean image but incorrectly under all corruptions in that group. The retained sets from all groups are then united. Formally, for model M_m, the retained set is:

$$\mathcal{D}_m = \{(x, q, a^*) : M_m(x_{\text{clean}}, q) = a^* \wedge \exists G_l \in \mathcal{G}, \forall c \in G_l, M_m(c(x), q) \neq a^*\}$$

Finally, the intersection of the retained sets from both models is taken:

$$\mathcal{D}_{\text{VCBench}} = \mathcal{D}_{\text{Qwen}} \cap \mathcal{D}_{\text{LLaVA}}$$

As shown in Table 11, existing visual corruption evaluation benchmarks have several issues that prevent them from accurately measuring model robustness. Specifically, they often include samples where the base model answers correctly on both clean and corrupted images, as well as samples where the base model answers incorrectly on both clean and corrupted images.

Neither type can effectively reflect the benefit of a robustness method. VCBench mainly filters out these two types of samples, thereby enabling a more faithful evaluation of the model's true robustness. This also explains why our method achieves a significant improvement on VCBench, while the gain on R-Bench is relatively small.

Table 11: Sample composition analysis of existing visual robustness benchmarks.

Benchmark	Clean Correct & Corrupted Correct	Clean Wrong & Corrupted Wrong	Total
R-Bench	776 (52.26%)	489 (32.93%)	1485
Visual Robustness Benchmark for VQA	7040 (44.00%)	3907 (24.42%)	16000

Table 12: Robustness comparison using real and denoised clean reference images. Results are reported as accuracy (%).

Method	**Gaussian Noise**	**Impulse Noise**	**Shot Noise**	**Speckle Noise**	**Avg.**
Qwen2.5-VL-3B	11.82	12.51	11.56	13.55	12.36
Robust-R1	15.64	16.68	16.42	19.64	17.10
Ours w/ Denoised Images	15.16	17.55	16.76	21.25	17.68
Ours w/ Clean Images	**16.33**	**18.33**	**18.42**	**22.59**	**18.92**

A.5 Robustness with Limited Clean References.

Our method requires only a small number of clean reference images, which are typically easy to collect in practical scenarios. For the few cases where even limited clean images are unavailable, we further investigate using images restored by a denoising model Kim et al. (2025) as substitutes. As shown in Table 12, denoised references achieve an average accuracy of 17.68%, slightly lower than the 18.92% obtained with real clean images, while still outperforming Robust-R1 at 17.10%. These results show that our method remains effective with limited clean references and can still achieve competitive robustness using generated substitutes when necessary.

Order Sensitivity in Continual Learning In real-world scenarios, the sequence of incoming corruptions is often random and cannot be predetermined. To simulate this practical setting, we evaluate the order sensitivity of different continual learning methods by randomly shuffling the corruption order. Specifically, we consider two representative corruption categories in VCBench: Weather degradations and Digital & Illumination variations. Weather degradations include realistic environmental perturbations, such as snow, fog, spatter, and frost, while Digital & Illumination variations contain common image processing and lighting changes, including JPEG compression, contrast, pixelate, saturate, brightness, and elastic transform. As shown in Fig. 6, each method is evaluated with ten different random corruption permutations, and we report the averaged accuracy with error bars representing the performance range across runs. It can be observed that existing continual adaptation approaches exhibit noticeable performance fluctuations under different corruption orders, indicating their sensitivity to task sequences. In contrast, our method demonstrates remarkable robustness to corruption order, as the knowledge from different corruptions is aggregated through the additive formulation, making the overall adaptation process inherently order-invariant.

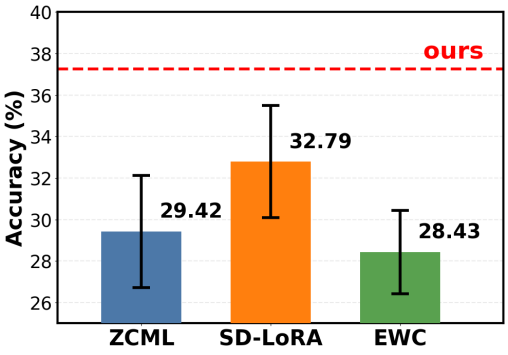

Figure 6: Order sensitivity analysis under different corruption sequences.