

Curvature-Guided Mixing for MLLM Adaptation

Jinglong Yang^{1,2†}, Jiaxuan He^{1†}, Wenjian Huang¹, Zhan Zhuang², and
Jianguo Zhang^{1*}

¹ Research Institute of Trustworthy Autonomous Systems and Department of
Computer Science and Engineering,
Southern University of Science and Technology

² Department of Computer Science, City University of Hong Kong

Abstract. Fine-tuning Multimodal Large Language Models (MLLMs) on specialized tasks often leads to catastrophic forgetting of their general capabilities. Existing model merging methods to combat this are often heuristic or use sub-optimal objectives. We propose Curvature-Guided Mixing (CGM), a theoretically grounded framework that merges pre-trained and fine-tuned models. CGM formulates a joint optimization objective and uses a second-order (Hessian) approximation of the loss landscapes to analytically derive an optimal, closed-form “soft mixing” ratio. This ratio intelligently blends parameters based on their relative task-specific curvatures. We also introduce CGM[†], a robust “hard mixing” variant that performs sparse parameter selection guided by a novel, curvature-aware score. Experiments on LLaVA-1.5 and Qwen-2.5VL across multiple downstream tasks show that CGM and CGM[†] consistently improve the trade-off between task specialization and general knowledge retention over existing methods. Code is available at github.com/zsyzjl/CGM-ECCV-2026.

Keywords: Continual learning · MLLM adaptation · model merging

1 Introduction

Multimodal Large Language Models (MLLMs) have emerged as powerful foundation models capable of joint understanding, generation, and planning across vision and language modalities [1, 4, 18, 21, 47]. Their impressive performance stems from pre-training on web-scale datasets, which endows them with a broad, general-purpose knowledge base. While this enables strong zero-shot performance on many tasks, adapting these models to specialized or novel domains requires targeted fine-tuning. This adaptation process, however, often leads to a critical issue: the model’s newly acquired skills come at the expense of losing its foundational, pre-trained abilities—a phenomenon known as catastrophic

[†] These authors contributed equally.

^{*} Corresponding author.

Funding: This work is supported by National Natural Science Foundation of China (Grant No. 62276121) and Innovation Team and Talents Cultivation Program of National Administration of Traditional Chinese Medicine, No. ZYYCXTD-D-202403.

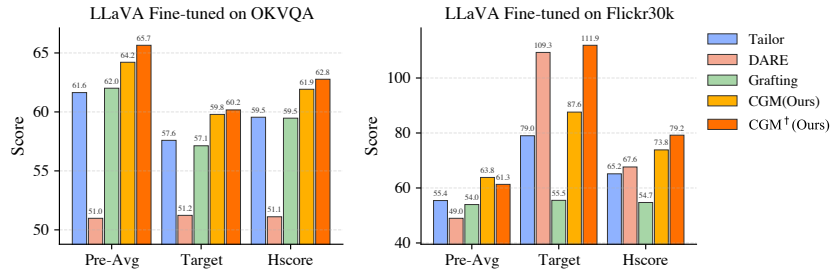


Fig. 1: Performance comparison of our methods (CGM and CGM[†]) against baselines for LLaVA fine-tuned on OKVQA. We evaluate general knowledge retention (Pre-Avg: average performance of pre-training tasks), specialization on the new task (Target), and the harmonic mean of both (Hscore) to measure the overall balance.

forgetting [15, 25, 44]. The central challenge, therefore, is to develop a methodology that can effectively instill new, task-specific knowledge into an MLLM while safeguarding the vast and robust general intelligence it already possesses.

Several recent works have explored this challenge through model merging [13, 46] and parameter-efficient fine-tuning [20, 24, 35, 37]. However, these approaches remain limited in effectively balancing task adaptation and knowledge retention. For example, Spider [13] merges parameters using heuristic scoring functions without a solid theoretical foundation, while Model Tailor [46] employs second-order information [7] but optimizes only for downstream performance. As a result, these methods tend to overfit to the new task and neglect the preservation of pre-trained general knowledge, underscoring the need for a principled framework that jointly optimizes both objectives.

To overcome these limitations, we introduce **Curvature-Guided Mixing (CGM)**, a theoretically grounded framework for merging pre-trained and fine-tuned models. Our approach begins by formulating a clear objective: to find a new set of weights that simultaneously minimizes the loss with respect to both the fine-tuning and general pre-training tasks. We model the loss landscapes around the fine-tuned and pre-trained optima using a second-order Taylor approximation, which captures the local geometry, or “curvature”, of each loss surface via its respective Hessian matrix.

To illustrate this, consider Figure 2, which visualizes two hypothetical loss landscapes corresponding to the fine-tuning and pre-training tasks. A naive interpolation between their optima would likely fall into a region of high loss for both. Guided by local curvature, our CGM method navigates this anisotropic landscape by identifying that the first parameter direction, critical for the fine-tuning task (due to its high curvature), should be adopted from the fine-tuned model, while the second direction, critical for the pre-training task, should be retained from the pre-trained model. This enables CGM to effectively integrate the “skills” learned during fine-tuning with the foundational knowledge from pre-training, achieving low loss on both tasks and mitigating catastrophic forgetting.

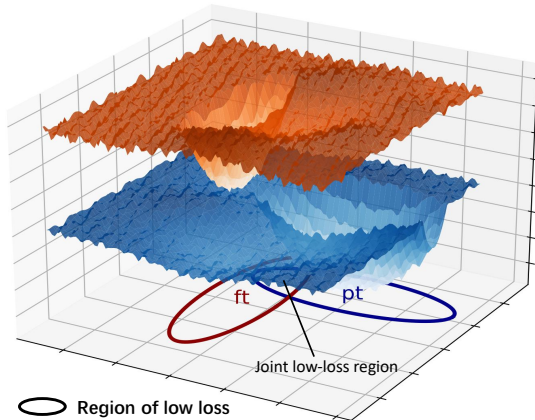


Fig.2: A conceptual illustration of the motivation behind CGM. The fine-tuning loss (orange) and pre-training loss (blue) landscapes exhibit conflicting anisotropic curvatures: the fine-tuning loss is sharp along one axis, whereas the pre-training loss is sharp along the other. CGM leverages this second-order geometry to locate a balanced joint minimum, as detailed in Section 3.

By mathematically formulating and solving this joint optimization problem (as detailed in Section 3), we derive a closed-form per-parameter mixing ratio. This “soft mixing” rule provides a non-heuristic and theoretically grounded solution. It elegantly demonstrates that the optimal mixing ratio for each parameter is directly determined by its relative curvature in the two loss landscapes.

While soft mixing provides a closed-form solution for the approximated loss, it is fundamentally a dense interpolation that updates every parameter. This approach carries a significant risk: a dense update can be disruptive, potentially destroying existing knowledge. We hypothesize that a sparse update strategy is more robust. The intuition is to preserve the vast majority of the model’s parameters and only modify the most critical subset. This approach, which is analogous to imposing an $\mathbf{L}_0$ constraint on the update, ensures a minimal, targeted modification.

To implement this, we propose a “hard mixing” strategy, termed **CGM[†]**. This method reframes the problem as a sparse selection task. Our method starts from the fine-tuned model—which has acquired the new skill but suffered from forgetting—and treats the pre-trained parameters as a “knowledge reservoir” to be sparsely re-integrated. For each parameter, a discrete decision is made on whether to revert to the pre-trained value. We derive a simple, curvature-aware score to rank parameters in the learnable layers. By selecting only the top- $K\%$ of parameters (where K denotes the sparsity ratio) most critical for general knowledge and least critical for the new task, CGM[†] performs a sparse, targeted reversion that effectively preserves foundational knowledge while maintaining the newly acquired skill and avoiding the pitfalls of dense interpolation.

In summary, our main contributions are as follows:

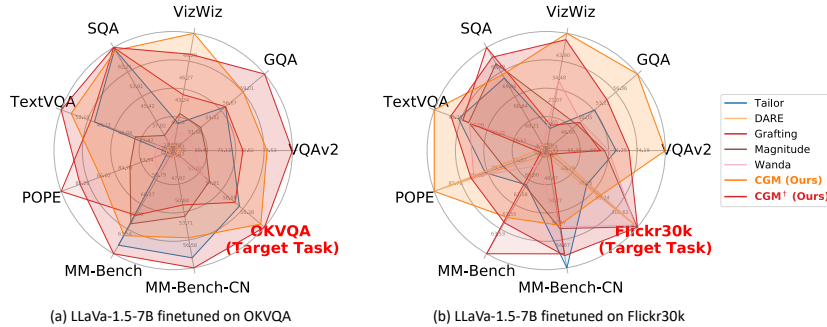


Fig. 3: Radar plots illustrating the performance trade-off between downstream adaptation and general knowledge retention. The “Target Task” axis shows performance on the fine-tuned task, while all other axes measure general pre-trained capabilities. Our methods, **CGM** and **CGM[†]** demonstrate superior balance by achieving high target-task performance while simultaneously preserving pre-trained knowledge.

- We propose **Curvature-Guided Mixing (CGM)**, a novel and theoretically grounded method that optimally merges fine-tuned and pre-trained models by minimizing a joint, curvature-aware loss objective.
- We introduce **CGM[†]**, a variant that reformulates the merging task as a sparse parameter selection problem, providing an efficient and effective mechanism for re-injecting pre-trained knowledge to mitigate forgetting.
- We conduct extensive experiments across diverse datasets and MLLM backbones, demonstrating that our proposed methods substantially outperform prior approaches in balancing downstream task performance and the preservation of general capabilities.

2 Related Work

2.1 Catastrophic Forgetting in MLLMs

Multimodal Large Language Models (MLLMs) [1, 18, 21, 47] have achieved remarkable success by leveraging large-scale pre-training to acquire broad, general-purpose knowledge. To adapt these models to specialized downstream tasks, fine-tuning is a necessary step [12, 13]. However, this adaptation process often leads to catastrophic forgetting [25, 44], a well-documented phenomenon where performance on its original general tasks degrades significantly after learning a new task.

Recent studies have begun to investigate this problem specifically within the MLLM context. Some work focuses on quantifying this effect in continual instruction tuning settings [3, 5, 33], while others observe that anti-forgetting techniques developed for LLMs show limited effectiveness when directly applied to MLLMs [46]. Our work tackles this core challenge: achieving single-step adap-

tation to new tasks while rigorously preserving the broad general knowledge obtained during pre-training.

2.2 Strategies for Knowledge Preservation

Existing methods to mitigate catastrophic forgetting for MLLM can be broadly categorized into three families:

Regularization-based Methods. This classic continual learning approach introduces auxiliary loss terms to penalize significant changes to parameters deemed important for old tasks [16, 43]. However, these methods often require modifications to the fine-tuning loss objective and can be complex to balance with the primary task loss.

Parameter-Efficient Fine-Tuning (PEFT). This popular family of methods freezes the vast majority of the pre-trained model and introduces a small set of new, trainable parameters for each task. These can be additive modules like Adapter [10] and LoRA [11]. These approaches are effective at isolating task-specific knowledge and reducing interference (especially in multi-step adaptation [36, 39, 40, 42]), and related parameter-efficient editing strategies have also been explored for diffusion models [6].

Partial-based Updating and Model Merging. This third category aims to create a merged model by selectively updating a subset of the pre-trained weights. The core challenge lies in determining *which* parameters to update and *how* to merge them. [13] assesses parameter importance using a heuristic combination of zeroth-order (magnitude) and first-order (gradient) information. Its reliance on a hand-crafted scoring function lacks a rigorous theoretical justification for why their specific formulation is optimal. Model Tailor [46] leverages the Hessian matrix to identify a sparse “model patch.” Inspired by model pruning techniques like SparseGPT [7], its methodology selects parameters deemed critical for the downstream task. However, this method is designed to minimize loss solely on the fine-tuning task, lacking an explicit objective to balance the preservation of pre-trained knowledge.

In contrast to this prior art, our CGM framework provides a theoretically-grounded solution derived directly from a clear, joint optimization objective. This allows us to derive a non-heuristic, closed-form “soft mixing” rule and a “hard mixing” score that elegantly uses relative curvature to determine the optimal merge.

Relation to Fisher Merging. We also relate our soft mixing to Fisher Merging [28]. Soft mixing resembles Fisher Merging when the curvature is computed using the Fisher Information Matrix (FIM). However, the derivations differ fundamentally: Fisher Merging follows from a Laplace approximation of the posterior, whereas CGM minimizes a joint weighted loss objective. It is theoretically intriguing that distinct assumptions converge to a similar expression. Crucially, CGM is a more general framework: it is not limited to the FIM for curvature estimation, and our further exploration shows, approximating the true Hessian diagonal via Hutchinson trace estimation and Hessian-vector products yields stronger performance.

Algorithm 1 Curvature-Guided Mixing (CGM) Pipeline

Require: Pre-trained weights $\mathbf{w}^{\text{pt}}$, fine-tuning data $\mathcal{D}_{\text{ft}}$, calibration set $\mathcal{C}$ (sampling from pre-trained tasks)

Output: Merged weights $\mathbf{w}^*$

- 1: **Pre-training analysis:** estimate diagonal $\mathbf{H}^{\text{pt}}$ on $\mathbf{w}^{\text{pt}}$ with $\mathcal{C}$
 - 2: **Fine-tuning:** SFT on $\mathcal{D}_{\text{ft}}$ to obtain $\mathbf{w}^{\text{ft}}$, estimate diagonal $\mathbf{H}^{\text{ft}}$ on $\mathbf{w}^{\text{ft}}$ with $\mathcal{D}_{\text{ft}}$
 - 3: **Mixing (choose one):**
 - 4: **Soft-Mixing (CGM):** compute λ_i from Eq. 8, set $w_i^* = w_i^{\text{ft}} + \lambda_i(w_i^{\text{pt}} - w_i^{\text{ft}})$
 - 5: **Hard-Mixing (CGM[†]):** score each parameter with c_i (Eq. 12); revert the lowest-score $K\%$ of parameters to w_i^{pt} , keep the rest as w_i^{ft}
-

3 Methodology

We consider the problem of merging a pre-trained model $\mathbf{w}^{\text{pt}}$ and its fine-tuned variant $\mathbf{w}^{\text{ft}}$ into a unified model $\mathbf{w}^*$ that performs well on the target task while preserving general capabilities. In this section, we first formulate this as a joint optimization problem balancing the two loss landscapes, then introduce our method along with its variant.

3.1 Loss Landscape Approximation

We begin by modeling the local geometry of the loss landscapes around the two optimal weight configurations: $\mathbf{w}^{\text{pt}}$ for general tasks and $\mathbf{w}^{\text{ft}}$ for the specific fine-tuning task.

The loss functions in the neighborhoods of these optima are approximated using a second-order Taylor expansion. Let $\ell_{\text{pt}}(\mathbf{w})$ and $\ell_{\text{ft}}(\mathbf{w})$ denote the loss functions for the general and fine-tuning tasks, respectively. Their local approximations are given by:

$$\begin{aligned} \ell_{\text{pt}}(\mathbf{w}) &= \ell_{\text{pt}}(\mathbf{w}^{\text{pt}}) + \nabla \ell_{\text{pt}}(\mathbf{w}^{\text{pt}})^\top (\mathbf{w} - \mathbf{w}^{\text{pt}}) \\ &\quad + \frac{1}{2}(\mathbf{w} - \mathbf{w}^{\text{pt}})^\top \mathbf{H}^{\text{pt}}(\mathbf{w} - \mathbf{w}^{\text{pt}}) + \mathcal{O}(\|\mathbf{w} - \mathbf{w}^{\text{pt}}\|^3), \end{aligned} \quad (1)$$

$$\begin{aligned} \ell_{\text{ft}}(\mathbf{w}) &= \ell_{\text{ft}}(\mathbf{w}^{\text{ft}}) + \nabla \ell_{\text{ft}}(\mathbf{w}^{\text{ft}})^\top (\mathbf{w} - \mathbf{w}^{\text{ft}}) \\ &\quad + \frac{1}{2}(\mathbf{w} - \mathbf{w}^{\text{ft}})^\top \mathbf{H}^{\text{ft}}(\mathbf{w} - \mathbf{w}^{\text{ft}}) + \mathcal{O}(\|\mathbf{w} - \mathbf{w}^{\text{ft}}\|^3), \end{aligned} \quad (2)$$

where $\nabla \ell$ is the first-order gradient vector, $\mathbf{H}$ is the Hessian matrix, and $\mathcal{O}(\cdot)$ represents the higher-order terms.

Our framework relies on two simplifying assumptions.

First, regarding the first-order terms: We assume that $\mathbf{w}^{\text{pt}}$ and $\mathbf{w}^{\text{ft}}$ are local minima obtained from training on their respective tasks. By definition, the gradient at any converged local minimum $\mathbf{w}_{\text{opt}}$ is zero. Therefore, $\nabla \ell_{\text{pt}}(\mathbf{w}^{\text{pt}}) = \mathbf{0}$ and $\nabla \ell_{\text{ft}}(\mathbf{w}^{\text{ft}}) = \mathbf{0}$, allowing the first-order terms to be safely omitted in both Taylor expansions.

Second, regarding the higher-order terms: our objective is to find a merged model $\mathbf{w}^*$ that balances performance on both the pre-training and fine-tuning tasks. Such a solution must lie within the joint low-loss region—that is, within the neighborhoods of $\mathbf{w}^{\text{pt}}$ and $\mathbf{w}^{\text{ft}}$. Therefore, a solution $\mathbf{w}^*$ within this region satisfies the locality condition of the Taylor expansion, allowing us to omit the higher-order terms $\mathcal{O}(\cdot)$.

Based on these two assumptions and ignoring the constant terms, our objective simplifies to minimizing the sum of two quadratic penalties:

$$\ell_{\text{pt}}(\mathbf{w}) \propto \frac{1}{2}(\mathbf{w} - \mathbf{w}^{\text{pt}})^\top \mathbf{H}^{\text{pt}}(\mathbf{w} - \mathbf{w}^{\text{pt}}), \quad (3)$$

$$\ell_{\text{ft}}(\mathbf{w}) \propto \frac{1}{2}(\mathbf{w} - \mathbf{w}^{\text{ft}})^\top \mathbf{H}^{\text{ft}}(\mathbf{w} - \mathbf{w}^{\text{ft}}), \quad (4)$$

where $\mathbf{H}^{\text{pt}}$ and $\mathbf{H}^{\text{ft}}$ denote the Hessian matrices capturing the local curvature of the respective loss surfaces. A larger Hessian eigenvalue corresponds to a sharper curvature, indicating that the loss is more sensitive to perturbations in that parameter direction.

To make the problem computationally tractable, we keep only the diagonal entries (denoted as h_i^{pt} and h_i^{ft}) of each Hessian and drop all off-diagonal terms. This simplification, common in curvature-aware optimization and second-order approximation methods [16, 17, 27], reduces the computational and storage cost from $\mathcal{O}(d^2)$ to $\mathcal{O}(d)$ and allows per-parameter decoupling of the objective.

3.2 CGM: Soft Mixing with Curvature Guidance

With the quadratic approximations of the loss functions, we define a joint objective to find a new set of weights $\mathbf{w}$ that simultaneously minimizes the increase in loss for both tasks:

$$\underset{\mathbf{w}}{\operatorname{argmin}} \ell(\mathbf{w}) = \frac{1}{2}(\mathbf{w} - \mathbf{w}^{\text{ft}})^\top \mathbf{H}^{\text{ft}}(\mathbf{w} - \mathbf{w}^{\text{ft}}) + \frac{\alpha}{2}(\mathbf{w} - \mathbf{w}^{\text{pt}})^\top \mathbf{H}^{\text{pt}}(\mathbf{w} - \mathbf{w}^{\text{pt}}). \quad (5)$$

Here, $\alpha > 0$ is a hyperparameter that balances preserving general knowledge (captured by $\mathbf{H}^{\text{pt}}$) and acquiring new task-specific skills (captured by $\mathbf{H}^{\text{ft}}$). This objective seeks a point in the joint low-loss region, weighted by their curvature.

We parameterize the merged weights $\mathbf{w}$ as a per-parameter linear interpolation starting from the fine-tuned weights. Let $\Delta = \mathbf{w}^{\text{pt}} - \mathbf{w}^{\text{ft}}$ be the “reversion vector” pointing back to the pre-trained weights. For each parameter i , the new weight is $w_i = w_i^{\text{ft}} + \lambda_i \Delta_i$, where $\lambda_i \in [0, 1]$ is the mixing ratio.

Under the diagonal Hessian approximation, the joint objective in Eq. 5 decouples into independent per-parameter optimization problems, each expressed as $\min_{\lambda_i} \ell_i(\lambda_i)$:

$$\min_{\lambda_i} \ell_i(\lambda_i) = \frac{1}{2} h_i^{\text{ft}} ((w_i^{\text{ft}} + \lambda_i \Delta_i) - w_i^{\text{ft}})^2 + \frac{\alpha}{2} h_i^{\text{pt}} ((w_i^{\text{ft}} + \lambda_i \Delta_i) - w_i^{\text{pt}})^2. \quad (6)$$

Substituting $w_i^{\text{pt}} = w_i^{\text{ft}} + \Delta_i$, this simplifies to:

$$\ell_i(\lambda_i) = \frac{1}{2} \Delta_i^2 [h_i^{\text{ft}} \lambda_i^2 + \alpha h_i^{\text{pt}} (\lambda_i - 1)^2]. \quad (7)$$

This yields a simple quadratic function of λ_i . By setting the derivative $\partial\ell_i/\partial\lambda_i$ to zero, we obtain the closed-form optimal mixing ratio for each parameter:

$$\lambda_i^* = \frac{\alpha h_i^{\text{pt}}}{h_i^{\text{ft}} + \alpha h_i^{\text{pt}}}. \quad (8)$$

We refer to this result as Curvature-Guided Mixing (CGM), a theoretically grounded “soft mixing” rule. It states that parameters with higher pre-training curvature (h_i^{pt}) lead to a larger contribution from the pre-trained model, while those with higher fine-tuning curvature (h_i^{ft}) retain more from the fine-tuned model.

We are intrigued to observe that if curvature is estimated using the empirical Fisher Information Matrix, the resulting soft-mixing expression closely resembles Fisher Merging [28]. However, the derivations differ fundamentally: Fisher Merging stems from a Laplace approximation of the posterior, whereas CGM minimizes a joint weighted loss objective. It is theoretically interesting that distinct assumptions converge to a similar form. Crucially, CGM is a more general framework and is not tied to any specific Hessian estimation method, and Fisher Merging can be viewed as a degeneralized case of CGM when the Hessian is approximated by the FIM. Our further exploration shows that approximating the true Hessian diagonal via Hutchinson trace estimation and Hessian-vector products yields stronger performance.

3.3 CGM[†]: Hard Mixing via Sparse Reversion

While soft mixing provides a closed-form solution to the approximated objective, it is a dense interpolation, which is often unnecessarily disruptive, as it modifies all parameters, potentially destroying existing knowledge. We argue that a more robust approach is to perform a sparse update: preserve the majority of the model’s parameters and only update the most critical subset. To achieve this, we propose a more robust “hard mixing” strategy, CGM[†], which reframes the problem as a sparse parameter selection task. Instead of blending weights, we make a binary choice for each parameter: either retain the fine-tuned weight w_i^{ft} or revert to the pre-trained one w_i^{pt} .

We introduce a binary mask $\mathbf{m} \in \{0, 1\}^d$, where d is the number of parameters, and define the merged model as $\mathbf{w}(\mathbf{m}) = \mathbf{w}^{\text{ft}} + \mathbf{m} \odot \Delta$, where $\Delta = \mathbf{w}^{\text{pt}} - \mathbf{w}^{\text{ft}}$. Here, $m_i = 1$ signifies reverting to the pre-trained weight, and $m_i = 0$ signifies keeping the fine-tuned weight. Substituting this into our joint objective (Eq. 5) yields:

$$\ell(\mathbf{m}) = \frac{1}{2} \sum_i h_i^{\text{ft}} ((w_i^{\text{ft}} + m_i \Delta_i) - w_i^{\text{ft}})^2 + \frac{\alpha}{2} \sum_i h_i^{\text{pt}} ((w_i^{\text{ft}} + m_i \Delta_i) - w_i^{\text{pt}})^2. \quad (9)$$

$$\ell(\mathbf{m}) = \frac{1}{2} \sum_i h_i^{\text{ft}} \Delta_i^2 m_i^2 + \frac{\alpha}{2} \sum_i h_i^{\text{pt}} \Delta_i^2 (m_i - 1)^2. \quad (10)$$

Since $m_i \in \{0, 1\}$, we have $m_i^2 = m_i$ and $(m_i - 1)^2 = 1 - m_i$. The objective simplifies to:

$$\ell(\mathbf{m}) = \frac{1}{2} \sum_i h_i^{\text{ft}} \Delta_i^2 m_i + \frac{\alpha}{2} \sum_i h_i^{\text{pt}} \Delta_i^2 (1 - m_i). \quad (11)$$

Rearranging the terms, we get:

$$\ell(\mathbf{m}) = \underbrace{\frac{\alpha}{2} \sum_i h_i^{\text{pt}} \Delta_i^2}_{\text{Constant w.r.t. } \mathbf{m}} + \sum_i \underbrace{\frac{1}{2} (h_i^{\text{ft}} - \alpha h_i^{\text{pt}}) \Delta_i^2}_{c_i} m_i. \quad (12)$$

Minimizing $\ell(\mathbf{m})$ is now equivalent to minimizing $\sum_i c_i m_i$, where $c_i = \frac{1}{2} (h_i^{\text{ft}} - \alpha h_i^{\text{pt}}) \Delta_i^2$ serves as a ranking score for updating the i -th parameter. A parameter with a small c_i (i.e., low h_i^{ft} and high h_i^{pt}) is one that is unimportant for the new task but crucial for the general-purpose pre-trained task, making it a prime candidate for reversion.

To enforce sparsity, we constrain the number of updated parameters to a budget K defined as a sparsity ratio (percentage of parameters updated). Let d be the total number of parameters and $K \in (0, 1]$; then $\|\mathbf{m}\|_0 = Kd$. The problem becomes:

$$\min_{\mathbf{m} \in \{0, 1\}^d, \|\mathbf{m}\|_0 = Kd} \sum_i c_i m_i. \quad (13)$$

This guides us to select the Kd parameters with the smallest c_i values (setting $m_i = 1$) while keeping the rest unchanged ($m_i = 0$). Thus, CGM[†] performs a sparse and modular update, effectively identifying and applying only the most critical parameter changes from the fine-tuning.

4 Experiments

4.1 Experimental Setup

Architectures and Datasets. We evaluate our methods on two representative MLLMs: LLaVA-1.5-7B [21] and Qwen-2.5VL-3B [1]. For each architecture, we partition the datasets into two categories to separately evaluate generalization capability and downstream adaptability. For LLaVA-1.5-7B [21], we fine-tune on OKVQA [26] and Flickr30k [38], representing visual question answering and image captioning, respectively, and evaluate generalization on a standard benchmark suite comprising VQAv2 [8], GQA [14], VizWiz [9], SQA [23], TextVQA [32], POPE [19], MM-Bench [22], and MM-Bench-CN [45]. Similarly, for Qwen-2.5VL-3B [1], we fine-tune on the Flickr30k [38] and LaTeX-OCR [31] datasets and assess retained general knowledge on the same benchmark suite, additionally including InfoVQA [29] and OKVQA [26] to provide a more comprehensive evaluation of multimodal reasoning and knowledge retention.

Compared Baselines. We evaluate our method against a diverse set of baselines, including the naive fine-tuning approach and model merging techniques.

- *Standard Fine-Tuning* [2]: Full task-specific fine-tuning without anti-forgetting strategies.
- *Tailor* [46]: Merges models by preserving task-critical parameters.
- *DARE* [41]: Uses random parameter selection and rescaling to balance generalization.
- *Grafting* [30]: Employs skill localization to identify sparse, task-critical weights.
- *Magnitude*: Heuristically reverts parameters with the smallest absolute changes to pre-trained values.
- *Wanda* [34]: Reverts parameters based on an importance score of weight magnitude and input activations.

Evaluation Metrics. We use (i) the score on the fine-tuning task (reflecting *specialization*) and (ii) the average score across a suite of general pre-training evaluation tasks (reflecting *generalization*). To holistically evaluate effectiveness in mitigating catastrophic forgetting in MLLMs, we adopt two aggregate metrics: **Average Performance** (Avg) and **Harmonic Mean Score** (Hscore). Formally, let S_{Target} denote the performance on the target task, and $S_{\text{Pre-Avg}} = \frac{1}{N} \sum_{i=1}^N S_i$ be the average performance over N general pre-training tasks. Then:

$$\text{Avg} = \frac{S_{\text{Target}} + N \cdot S_{\text{Pre-Avg}}}{N + 1}, \quad (14)$$

$$\text{Hscore} = \frac{2 \cdot S_{\text{Target}} \cdot S_{\text{Pre-Avg}}}{S_{\text{Target}} + S_{\text{Pre-Avg}}}. \quad (15)$$

Here, the Avg metric equally weights adaptation and generalization, while Hscore penalizes imbalanced performance.

Implementation Details. We follow the official codebases to fine-tune LLaVA-1.5-7B [21] and Qwen-2.5VL-3B [1]. For LLaVA-1.5, we fine-tune the last 12 layers of the language model and the visual projector for 1 epoch with a learning rate of $1\text{e-}4$ and a global batch size of 64. For Qwen-2.5VL, we fine-tune the last 6 layers and the visual projector for 3 epochs with a learning rate of $1\text{e-}5$ and the same batch size. We use the empirical Fisher Information Matrix as an efficient estimator of the diagonal Hessian. All experiments are run on 5 NVIDIA RTX 6000 Ada GPUs (48 GB each).

Computation and Memory Cost. Obtaining $\mathbf{w}^{\text{ft}}$ follows standard supervised fine-tuning. Both CGM (soft mixing) and CGM[†] (hard mixing) perform element-wise operations on the selected layers (the last 12 layers for LLaVA and the last 6 layers for Qwen), yielding $\mathcal{O}(d)$ complexity, where d is the number of parameters in those layers. The main additional cost is estimating diagonal Hessians. For $\mathbf{H}^{\text{pt}}$, we use a calibration set sampled from pre-training tasks (8 samples per task in practice) and compute squared gradients at $\mathbf{w}^{\text{pt}}$ to obtain an empirical FIM estimate. For $\mathbf{H}^{\text{ft}}$, we accumulate squared gradients over the fine-tuning dataset. Empirically, incorporating FIM estimation during fine-tuning reduces throughput from 3.81 to 3.51 samples/s ($\sim 7.9\%$ overhead), confirming negligible extra computation. For memory cost, this storage pattern is standard in model-merging methods: we keep only element-wise second-order statistics

Table 1: Main results on the **LLaVA-1.5-7B** backbone, fine-tuned on the **OKVQA** target task.

Method	Pre-trained Tasks								Target Task		Overall Metrics	
	VQAv2	GQA	VizWiz	SQA	TextVQA	POPE	MM-Bench	MM-Bench-CN	Pre-Avg	OKVQA	Hscore	Avg
Pre-trained Model	78.5	61.9	50.0	70.4	58.2	87.3	64.3	58.3	66.1	52.8	58.7	64.6
Fine-tuned Model	68.0	50.4	38.7	24.5	40.1	83.2	56.9	41.6	50.4	58.0	53.9	51.2
Tailor	71.7	56.1	40.1	69.8	50.5	82.3	64.4	58.2	61.6	57.6	59.6	61.2
DARE	67.7	49.6	37.2	28.6	39.9	82.4	57.2	45.2	51.0	51.2	51.1	51.0
Grafting	72.5	56.1	44.0	70.6	51.0	88.5	62.0	51.4	62.0	57.1	59.5	61.5
Magnitude	69.7	52.8	41.5	33.0	44.6	84.5	62.7	52.8	54.0	54.5	54.9	55.1
Wanda	68.8	51.2	39.1	30.3	41.4	83.1	58.8	45.1	52.2	53.9	53.1	52.4
CGM (Ours)	74.3 $\pm$ 0.04	58.5 $\pm$ 0.06	52.3 $\pm$ 0	69.3 $\pm$ 0.13	53.8 $\pm$ 0.35	86.2 $\pm$ 0.12	63.7 $\pm$ 0.27	55.6 $\pm$ 0.40	64.2 $\pm$ 0.03	59.8 $\pm$ 0.21	61.9 $\pm$ 0.11	63.7 $\pm$ 0.03
CGM[†] (Ours)	76.2 $\pm$ 0.04	61.4 $\pm$ 0.07	49.5 $\pm$ 0	70.6 $\pm$ 0.12	55.3 $\pm$ 0.11	87.4 $\pm$ 0.20	65.1 $\pm$ 0.32	59.5 $\pm$ 0.40	65.7 $\pm$ 0.05	60.2$\pm$0.22	62.8$\pm$0.12	65.0$\pm$0.07

^a The VizWiz server did not respond during our standard-deviation evaluation.

Table 2: Main results on the **LLaVA-1.5-7B** backbone, fine-tuned on the **Flickr30k** target task.

Method	Pre-trained Tasks								Target Task		Overall Metrics	
	VQAv2	GQA	VizWiz	SQA	TextVQA	POPE	MM-Bench	MM-Bench-CN	Pre-Avg	Flickr30k	Hscore	Avg
Pre-trained Model	78.5	61.9	50.0	70.4	58.2	87.3	64.3	58.3	66.1	11.73	19.93	60.08
Fine-tuned Model	67.8	46.7	22.4	67.5	37.9	76.1	59.7	50.8	53.6	53.57	53.59	53.6
Tailor	70.8	51.7	13.2	69.3	42.9	77.6	61.1	56.9	55.4	41.85	47.69	53.92
DARE	62.4	43.7	6.2	67.8	20.8	85.3	59.6	45.9	49.0	52.48	50.66	49.36
Grafting	68.7	49.0	14.5	69.9	41.8	67.9	64.5	55.5	54.0	30.56	39.02	51.37
Magnitude	69.3	48.6	22.1	69.6	40.2	74.5	61.2	53.0	54.8	<u>52.78</u>	53.77	54.57
Wanda	69.8	49.2	33.5	67.8	40.0	79.6	60.1	52.8	56.6	50.81	53.56	55.97
CGM (Ours)	77.1 $\pm$ 0.02	59.5 $\pm$ 0.09	53.3 $\pm$ 0	69.3 $\pm$ 0.25	49.5 $\pm$ 0.06	86.5 $\pm$ 0.26	62.7 $\pm$ 0.22	52.7 $\pm$ 0.69	63.8 $\pm$ 0.08	47.9 $\pm$ 0.74	54.71 $\pm$ 0.17	62.01$\pm$0.03
CGM[†] (Ours)	72.4 $\pm$ 0.02	54.0 $\pm$ 0.21	50.6 $\pm$ 0	69.7 $\pm$ 0.05	45.0 $\pm$ 0.30	80.1 $\pm$ 0.06	63.0 $\pm$ 0.13	55.7 $\pm$ 0.29	61.3 $\pm$ 0.09	51.91 $\pm$ 0.82	56.22$\pm$0.26	60.26$\pm$0.15

(the diagonal vectors $\mathbf{h}^{\text{pt}}$ and $\mathbf{h}^{\text{ft}}$). This is substantially more memory-efficient than methods that jointly store first-order, second-order, and other auxiliary statistics.

4.2 Main Results

Tables 1–4 summarize the main results across multiple datasets and backbones. We evaluate performance using Hscore and Avg, which reflect the trade-off between adaptation to new tasks and retention of general knowledge.

Across all experiments, existing approaches reveal a clear dilemma: the *Fine-tuned Model* achieves strong performance on the target task but suffers severe degradation on pre-trained tasks, while the *Pre-trained Model* retains general capabilities yet fails to adapt. Methods such as DARE and Wanda partially alleviate this issue but still incur substantial knowledge loss. In contrast, our methods *CGM* and *CGM[†]* achieve the best overall balance. For example, in Table 1, *CGM[†]* attains the highest Hscore of 62.8 while maintaining nearly full general knowledge (Pre-Avg 65.7). Although the target task scores in Tables 3 and 4 are not the highest, our methods still achieve the best overall metrics, significantly outperforming existing approaches in maintaining both specialization and generalization.

Table 3: Main results on the **Qwen-2.5VL-3B** backbone, fine-tuned on the **Flickr30k** target task.

Method	Pre-trained Tasks								Target Task		Overall Metrics	
	VQAv2	GQA	VizWiz	SQA	TextVQA	POPE	MM-Bench	MM-Bench-CN	Pre-Avg	Flickr30k	Hscore	Avg
Pre-trained Model	80.7	58.9	65.9	82.7	75.6	87.6	77.7	76.6	75.7	30.62	43.61	70.71
Fine-tuned Model	72.9	46.8	57.0	79.2	66.2	83.9	75.6	63.0	68.1	52.15	59.05	66.3
Tailor	70.3	46.4	56.1	79.7	61.2	84.6	75.6	60.7	66.8	47.97	55.85	64.73
DARE	73.4	47.8	54.2	78.2	64.0	86.6	73.7	57.0	66.9	51.04	57.89	65.11
Grafting	68.8	44.3	59.3	80.5	62.4	76.0	76.7	70.1	67.2	29.11	40.63	63.01
Magnitude	72.0	45.8	56.4	79.9	65.7	83.2	76.2	63.7	67.8	50.85	58.13	65.95
Wanda	72.8	46.8	57.2	79.3	66.1	83.9	75.4	63.1	68.1	<u>51.46</u>	58.61	66.23
CGM (Ours)	79.6 ± 0.02	56.1 ± 0.14	63.5 ± 0	82.0 ± 0.19	73.2 ± 0.15	86.9 ± 0.10	78.1 ± 0.18	76.8 ± 0.09	74.5 ± 0.08	49.1 ± 0.26	59.2 ± 0.05	71.69 ± 0.04
CGM[†] (Ours)	76.9 ± 0.02	53.0 ± 0.02	61.0 ± 0	82.2 ± 0.11	72.4 ± 0.71	86.5 ± 0.10	78.2 ± 0.10	76.7 ± 0.20	73.3 ± 0.10	50.09 ± 0.29	59.52 ± 0.09	<u>70.75 ± 0.08</u>

Table 4: Main results on the **Qwen-2.5VL-3B** backbone, fine-tuned on the **LaTeX-OCR** target task.

Method	Pre-trained Tasks								Target Task		Overall Metrics	
	VQAv2	GQA	VizWiz	SQA	TextVQA	POPE	InfoVQA	OKVQA	Pre-Avg	LaTeX-OCR	Hscore	Avg
Pre-trained Model	80.7	58.9	65.9	82.7	75.6	87.6	61.3	56.1	71.1	21.1	32.6	65.6
Fine-tuned Model	66.8	42.9	52.7	81.8	65.9	80.8	54.8	44.1	61.2	72.7	66.5	62.5
Tailor	69.7	45.5	54.4	82.3	70.4	82.3	58.2	46.7	63.7	57.0	60.2	62.9
DARE	66.3	44.1	50.9	78.5	63.3	82.9	53.1	42.7	60.2	74.8	66.7	61.8
Grafting	72.6	48.4	53.8	82.7	72.3	83.9	59.0	48.3	65.1	67.0	66.1	65.3
Magnitude	68.5	44.3	53.5	82.4	67.6	81.4	55.7	45.3	62.4	76.0	68.5	63.9
Wanda	67.1	43.1	52.9	82.2	66.4	81.0	55.1	44.3	61.5	74.8	67.5	63.0
CGM (Ours)	80.9 ± 0.02	58.9 ± 0.04	66.0 ± 0	82.7 ± 0.22	75.8 ± 0.18	87.6 ± 0.15	60.0 ± 0.46	57.1 ± 0.14	71.1 ± 0.05	70.9 ± 1.27	71.0 ± 0.64	71.1 ± 0.15
CGM[†] (Ours)	80.4 ± 0.01	57.8 ± 0.17	65.0 ± 0	82.7 ± 0.16	75.5 ± 0.18	86.6 ± 0.12	60.1 ± 0.16	56.6 ± 0.10	70.6 ± 0.03	<u>74.9 ± 0.09</u>	72.7 ± 0.04	<u>71.1 ± 0.03</u>

Table 5: Ablation study on the components of the CGM[†] score $c_i = \frac{1}{2}(h_i^{\text{ft}} - \alpha h_i^{\text{pt}})\Delta_i^2$. We compare four variants by selectively omitting components from the score. All variants update the same number of parameters ($K\%$ of the total).

Variant	Score $c_i \propto$	LLaVA - OKVQA		LLaVA - Flickr30k		Qwen3B - LaTeX-OCR	
		Hscore	Avg	Hscore	Avg	Hscore	Avg
1. Magnitude-only	Δ_i^2	54.0	54.0	53.59	53.61	67.6	62.9
2. Fine-tune+Magnitude	$h_i^{\text{ft}}\Delta_i^2$	53.1	52.4	53.59	53.61	67.2	62.7
3. Pre-train+Magnitude	$-\alpha h_i^{\text{pt}}\Delta_i^2$	61.7	65.1	<u>55.76</u>	<u>60.25</u>	<u>71.7</u>	<u>70.8</u>
4. CGM[†] (Full)	$(h_i^{\text{ft}} - \alpha h_i^{\text{pt}})\Delta_i^2$	62.8	<u>65.0</u>	56.22	60.26	72.7	71.1

4.3 Ablation Studies

We conduct an ablation study to verify the components of the curvature-aware score $c_i = \frac{1}{2}(h_i^{\text{ft}} - \alpha h_i^{\text{pt}})\Delta_i^2$ (Table 5). Relying solely on magnitude ($c_i \propto \Delta_i^2$) or fine-tuning curvature ($c_i \propto h_i^{\text{ft}}\Delta_i^2$) yields the lowest Hscores on LLaVA-OKVQA (54.0 and 53.1, respectively) due to severe overfitting and catastrophic forgetting. The pre-training variant ($c_i \propto -\alpha h_i^{\text{pt}}\Delta_i^2$) preserves general knowledge more effectively, improving the Hscore to 61.7. Ultimately, the full formulation consistently achieves the highest scores across all datasets. By jointly modeling both curvatures, it provides an optimal trade-off between adaptation to new tasks and pre-trained model inertia.

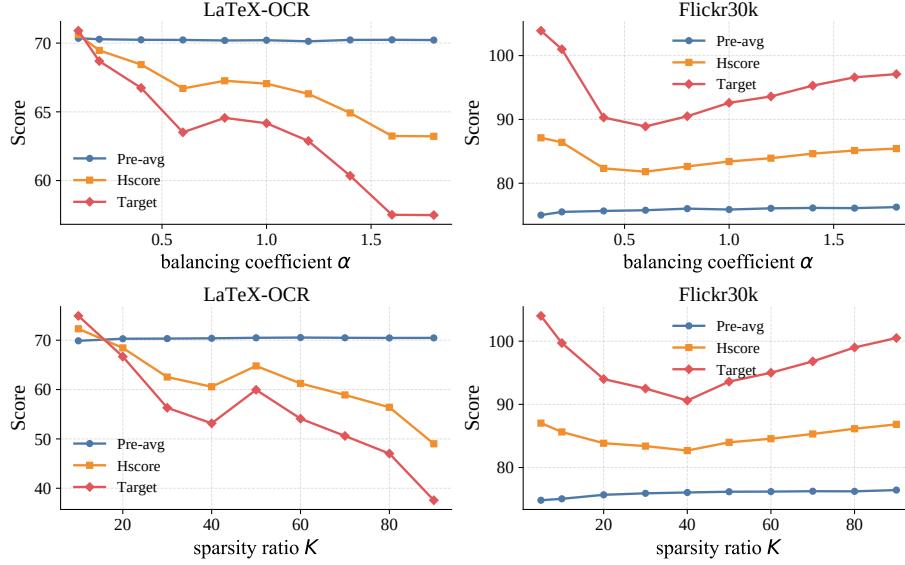


Fig. 4: Hyperparameter sensitivity analysis on the Qwen3B backbone for the LaTeX-OCR and Flickr30k tasks.

4.4 Hyperparameter Sensitivity

We conduct a sensitivity analysis on our two key hyperparameters, starting with the sparsity ratio K , which governs the fraction of pre-trained parameters retained. As shown in Figure 4, general knowledge preservation (Pre-Avg score) is remarkably robust, remaining nearly flat even as K varies from 10% to 90%. This demonstrates that retaining as few as 10% of the critically identified pre-trained parameters is sufficient to preserve almost all general knowledge and prevent catastrophic forgetting. Consequently, the primary role of K is to control the degree of downstream specialization rather than managing forgetting, with the optimal Hscore and target performance typically achieved at a 10% sparsity ratio.

Similarly, the balancing coefficient α modulates the trade-off between target task adaptation and pre-trained knowledge retention. Across varying values of α , the Pre-Avg score exhibits minimal variance, confirming that our joint objective inherently shields foundational knowledge. Therefore, α primarily influences target task performance, which in turn dictates the overall Hscore. The best balance is consistently found at smaller values ($\alpha \approx 0.1 \sim 0.15$), which provide sufficient adaptation to the new task while effectively maintaining inherent general capabilities.

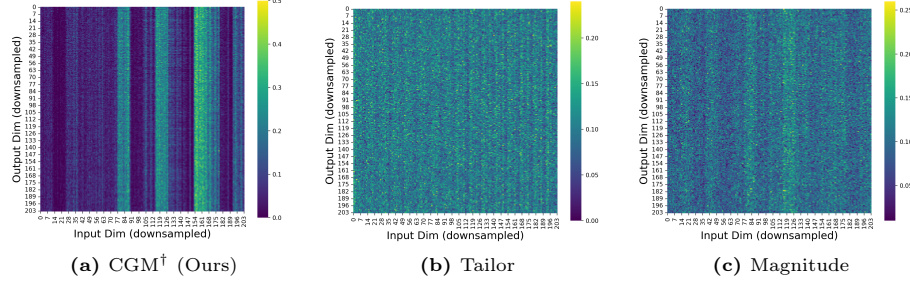


Fig. 5: Visualization of selection masks (10% sparsity) for different methods on the `self_attn.o_proj` layer. The masks are downsampled by patch-averaging for clarity. (a) Our $\text{CGM}^\dagger$ mask shows a highly structured pattern, selecting by input dimensions (vertical bands). (b) Tailor and (c) Magnitude masks are unstructured and noisy, selecting parameters diffusely across the matrix. This visually confirms that our curvature-guided score identifies coherent structural components for updates.

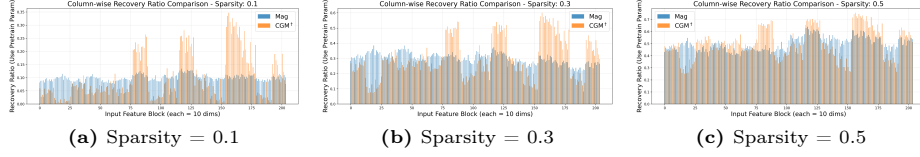


Fig. 6: Quantitative comparison of column-wise recovery ratios between $\text{CGM}^\dagger$ (orange) and Magnitude (Mag, blue) at varying update sparsity levels. The Y-axis represents the fraction of pre-trained parameters *kept*. Across all sparsity levels, $\text{CGM}^\dagger$ exhibits a non-uniform, structured selection that consistently targets or protects the same columns, whereas the Magnitude baseline remains uniform and diffuse.

4.5 Analysis of Selection Mask Structure

To understand the qualitative differences in parameter selection, Figure 5 visualizes downsampled masks on the `self_attn.o_proj` layer. Tailor and Magnitude produce diffuse, unstructured noise. In contrast, $\text{CGM}^\dagger$ exhibits distinct vertical bands, indicating a coherent, feature-level selection focused on specific input dimensions.

We quantify this via the *column-wise recovery ratio* in Figure 6. The Magnitude baseline applies updates uniformly across columns at all sparsity levels. Conversely, $\text{CGM}^\dagger$ consistently targets or protects specific structural column groups (e.g., blocks ~ 80 , ~ 120 , and ~ 160). This provides strong evidence that our curvature-guided score isolates structured, critically important parameter groups rather than relying on diffuse heuristics.

5 Conclusion

In this paper, we introduce Curvature-Guided Mixing (CGM) and its sparse variant, $\text{CGM}^\dagger$, as theoretically grounded solutions to catastrophic forgetting

in fine-tuned MLLMs. Our framework leverages a joint optimization objective and second-order information from the loss landscapes of both pre-training and fine-tuning tasks. This approach yields an optimal “soft mixing” ratio (CGM) and a robust, sparse “hard mixing” strategy (CGM[†]). Extensive experiments on the LLaVA and Qwen-VL models demonstrate that our methods significantly outperform prior art, establishing a new state-of-the-art in balancing task specialization and the preservation of general foundational knowledge. Our results confirm that leveraging loss landscape geometry provides a principled and effective approach to knowledge-preserving model adaptation.

References

1. Bai, S., Chen, K., Liu, X., Wang, J., Ge, W., Song, S., Dang, K., Wang, P., Wang, S., Tang, J., Zhong, H., Zhu, Y., Yang, M., Li, Z., Wan, J., Wang, P., Ding, W., Fu, Z., Xu, Y., Ye, J., Zhang, X., Xie, T., Cheng, Z., Zhang, H., Yang, Z., Xu, H., Lin, J.: Qwen2.5-vl technical report. arXiv preprint arXiv:2502.13923 (2025)
2. de Boer, P.T., Kroese, D.P., Mannor, S., Rubinstein, R.Y.: A tutorial on the cross-entropy method. *Annals of Operations Research* **134**, 19–67 (2005)
3. Cao, M., Liu, Y., Liu, Y., Wang, T., Dong, J., Ding, H., Zhang, X., Reid, I., Liang, X.: Continual llava: Continual instruction tuning in large vision-language models. arXiv preprint arXiv:2411.02564 (2024)
4. Cen, J., Wu, C., Liu, X., Yin, S., Pei, Y., Yang, J., Chen, Q., Duan, N., Zhang, J.: Using left and right brains together: Towards vision and language planning. In: ICML. pp. 5982–6001 (2024)
5. Chen, C., Zhu, J., Luo, X., Shen, H.T., Song, J., Gao, L.: Coin: A benchmark of continual instruction tuning for multimodal large language models. *NeurIPS* **37**, 57817–57840 (2024)
6. Chen, H., Yang, Y., Zhong, N., Ma, K.: Hiding images in diffusion models by editing learned score functions. In: CVPR. pp. 18663–18673 (2025)
7. Frantar, E., Alistarh, D.: SparseGPT: Massive language models can be accurately pruned in one-shot. In: ICML (2023)
8. Goyal, Y., Khot, T., Summers-Stay, D., Batra, D., Parikh, D.: Making the v in vqa matter: Elevating the role of image understanding in visual question answering. In: CVPR. pp. 6904–6913 (2017)
9. Gurari, D., Li, Q., Stangl, A.J., Guo, A., Lin, C., Grauman, K., Luo, J., Bigham, J.P.: Vizwiz grand challenge: Answering visual questions from blind people. In: CVPR. pp. 3608–3617 (2018)
10. Houlisby, N., Giurigu, A., Jastrzebski, S., Morrone, B., De Laroussilhe, Q., Gesmundo, A., Attariyan, M., Gelly, S.: Parameter-efficient transfer learning for nlp. In: ICML. pp. 2790–2799 (2019)
11. Hu, E.J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., Chen, W., et al.: Lora: Low-rank adaptation of large language models. In: ICLR (2022)
12. Huang, W., Liang, J., Guo, X., Fang, Y., Wan, G., Rong, X., Wen, C., Shi, Z., Li, Q., Zhu, D., Ma, Y., Liang, K., Yang, B., Li, H., Shao, J., Ye, M., Du, B.: Keeping yourself is important in downstream tuning multimodal large language model. arXiv preprint arXiv:2503.04543 (2025)
13. Huang, W., Liang, J., Shi, Z., Zhu, D., Wan, G., Li, H., Du, B., Tao, D., Ye, M.: Learn from downstream and be yourself in multimodal large language model fine-tuning. In: ICML (2025)
14. Hudson, D.A., Manning, C.D.: Gqa: A new dataset for real-world visual reasoning and compositional question answering. In: CVPR. pp. 6700–6709 (2019)
15. Jha, S., Gong, D., Yao, L.: CLAP4CLIP: Continual learning with probabilistic finetuning for vision-language models. In: NeurIPS (2024)
16. Kirkpatrick, J., Pascanu, R., Rabinowitz, N., Veness, J., Desjardins, G., Rusu, A.A., Milan, K., Quan, J., Ramalho, T., Grabska-Barwinska, A., et al.: Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences* **114**(13), 3521–3526 (2017)
17. LeCun, Y., Denker, J.S., Solla, S.A.: Optimal brain damage. In: *NeurIPS*. vol. 2, pp. 598–605 (1990)

18. Li, J., Li, D., Savarese, S., Hoi, S.: Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In: ICML (2023)
19. Li, Y., Du, Y., Zhou, K., Wang, J., Zhao, X., Wen, J.R.: Evaluating object hallucination in large vision-language models. In: EMNLP. pp. 292–305 (2023)
20. Liang, Y.S., Li, W.J.: Inflora: Interference-free low-rank adaptation for continual learning. In: CVPR. pp. 23638–23647 (2024)
21. Liu, H., Li, C., Li, Y., Lee, Y.J.: Improved baselines with visual instruction tuning. In: CVPR (2024)
22. Liu, Y., Duan, H., Zhang, Y., Li, B., Zhang, S., Zhao, W., Yuan, Y., Wang, J., He, C., Liu, Z., et al.: Mmbench: Is your multi-modal model an all-around player? In: ECCV. pp. 216–233 (2024)
23. Lu, P., Mishra, S., Xia, T., Qiu, L., Chang, K.W., Zhu, S.C., Tafjord, O., Clark, P., Kalyan, A.: Learn to explain: Multimodal reasoning via thought chains for science question answering. In: NeurIPS. vol. 35, pp. 2507–2521 (2022)
24. Luo, G., Yang, X., Dou, W., Wang, Z., Liu, J., Dai, J., Qiao, Y., Zhu, X.: Monointernvl: Pushing the boundaries of monolithic multimodal large language models with endogenous visual pre-training. In: CVPR (2025)
25. Luo, Y., Yang, Z., Meng, F., Li, Y., Zhou, J., Zhang, Y.: An empirical study of catastrophic forgetting in large language models during continual fine-tuning. *IEEE/ACM Transactions on Audio, Speech, and Language Processing* **33**, 3776–3786 (2025)
26. Marino, K., Rastegari, M., Farhadi, A., Mottaghi, R.: Ok-vqa: A visual question answering benchmark requiring external knowledge. In: CVPR. pp. 3195–3204 (2019)
27. Martens, J., Grosse, R.: Optimizing neural networks with kronecker-factored approximate curvature. In: ICML. pp. 2408–2417 (2015)
28. Matena, M., Raffel, C.: Merging models with fisher-weighted averaging. In: NeurIPS (2022)
29. Mathew, M., Bagal, V., Tito, R., Karatzas, D., Valveny, E., Jawahar, C.: Infographicvqa. In: WACV. pp. 1697–1706 (2022)
30. Panigrahi, A., Saunshi, N., Zhao, H., Arora, S.: Task-specific skill localization in fine-tuned language models. In: ICML. pp. 27011–27033 (2023)
31. Roboflow: Latex-ocr dataset (unsloth version). https://huggingface.co/datasets/unsloth/LaTeX_OCR (2024)
32. Singh, A., Natarajan, V., Shah, M., Jiang, Y., Chen, X., Batra, D., Parikh, D., Rohrbach, M.: Towards vqa models that can read. In: CVPR. pp. 8317–8326 (2019)
33. Srinivasan, T., Chang, T.Y., Pinto Alva, L., Chochlakis, G., Rostami, M., Thomason, J.: Climb: A continual learning benchmark for vision-and-language tasks. *NeurIPS* **35**, 29440–29453 (2022)
34. Sun, M., Liu, Z., Bair, A., Kolter, J.Z.: A simple and effective pruning approach for large language models. In: Workshop on Efficient Systems for Foundation Models @ ICML2023 (2023)
35. Wang, X., Zhuang, Z., Zhang, Y.: Plan: Proactive low-rank allocation for continual learning. In: ICCV. pp. 2909–2918 (2025)
36. Wu, J., Xiong, Y., Li, X., Xia, Y., Wang, R., Wang, Y., Yu, T., Kim, S., Rossi, R.A., Yao, L., Shang, J., McAuley, J.: Mitigating visual knowledge forgetting in MLLM instruction-tuning via modality-decoupled gradient descent. In: Findings of the Association for Computational Linguistics: EMNLP 2025. pp. 2282–2295 (2025)
37. Wu, Y., Piao, H., Huang, L., Wang, R., Li, W., Pfister, H., Meng, D., Ma, K., Wei, Y.: Sd-lora: Scalable decoupled low-rank adaptation for class incremental learning. In: ICLR (2025)

38. Young, P., Lai, A., Hodosh, M., Hockenmaier, J.: From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions. *Transactions of the Association for Computational Linguistics* **2**, 67–78 (2014)
39. Yu, J., Xiong, H., Zhang, L., Diao, H., Zhuge, Y., Hong, L., Wang, D., Lu, H., He, Y., Chen, L.: Llms can evolve continually on modality for x-modal reasoning. *NeurIPS* **37**, 49834–49858 (2024)
40. Yu, J., Zhuge, Y., Zhang, L., Hu, P., Wang, D., Lu, H., He, Y.: Boosting continual learning of vision-language models via mixture-of-experts adapters. In: *CVPR*. pp. 23219–23230 (2024)
41. Yu, L., Yu, B., Yu, H., Huang, F., Li, Y.: Language models are super mario: Absorbing abilities from homologous models as a free lunch. In: *ICML* (2024)
42. Zeng, F., Zhu, F., Guo, H., Zhang, X.Y., Liu, C.L.: Modalprompt: Towards efficient multimodal continual instruction tuning with dual-modality guided prompt. In: *EMNLP*. pp. 12126–12141 (2025)
43. Zenke, F., Poole, B., Ganguli, S.: Continual learning through synaptic intelligence. In: *ICML*. pp. 3987–3995 (2017)
44. Zhai, Y., Tong, S., Li, X., Cai, M., Qu, Q., Lee, Y.J., Ma, Y.: Investigating the catastrophic forgetting in multimodal large language model fine-tuning. In: *Conference on Parsimony and Learning* (2023)
45. Zhang, P., Dong, X., Wang, B., Cao, Y., Xu, C., Ouyang, L., Zhao, Z., Duan, H., Zhang, S., Ding, S., et al.: Internlm-xcomposer: A vision-language large model for advanced text-image comprehension and composition. *arXiv preprint arXiv:2309.15112* (2023)
46. Zhu, D., Sun, Z., Li, Z., Shen, T., Yan, K., Ding, S., Kuang, K., Wu, C.: Model tailor: Mitigating catastrophic forgetting in multi-modal large language models. In: *ICML* (2024)
47. Zhu, J., Wang, W., Chen, Z., Liu, Z., Ye, S., Gu, L., Tian, H., Duan, Y., Su, W., Shao, J., Gao, Z., Cui, E., Wang, X., Cao, Y., Liu, Y., Wei, X., Zhang, H., Wang, H., Xu, W., Li, H., Wang, J., Deng, N., Li, S., He, Y., Jiang, T., Luo, J., Wang, Y., He, C., Shi, B., Zhang, X., Shao, W., He, J., Xiong, Y., Qu, W., Sun, P., Jiao, P., Lv, H., Wu, L., Zhang, K., Deng, H., Ge, J., Chen, K., Wang, L., Dou, M., Lu, L., Zhu, X., Lu, T., Lin, D., Qiao, Y., Dai, J., Wang, W.: Internvl3: Exploring advanced training and test-time recipes for open-source multimodal models. *arXiv preprint arXiv:2504.10479* (2025)