LoRASculpt: Sculpting LoRA for Harmonizing General and Specialized Knowledge in Multimodal Large Language Models

Jian Liang*, Wenke Huang*, Guancheng Wan*, Qu Yang, Mang Ye†
National Engineering Research Center for Multimedia Software,
School of Computer Science, Wuhan University.
{jianliang, wenkehuang, yemang}@whu.edu.cn
https://github.com/LiangJian24/LoRASculpt

Abstract

While Multimodal Large Language Models (MLLMs) excel at generalizing across modalities and tasks, effectively adapting them to specific downstream tasks while simultaneously retaining both general and specialized knowledge remains challenging. Although Low-Rank Adaptation (LoRA) is widely used to efficiently acquire specialized knowledge in MLLMs, it introduces substantial harmful redundancy during visual instruction tuning, which exacerbates the forgetting of general knowledge and degrades downstream task performance. To address this issue, we propose **LoRASculpt** *to eliminate harmful redundant parameters, thereby harmonizing general and specialized knowledge. Specifically, under theoretical guarantees, we introduce sparse updates into LoRA to discard redundant parameters effectively. Furthermore, we propose a Conflict Mitigation Regularizer to refine the update trajectory of LoRA, mitigating knowledge conflicts with the pretrained weights. Extensive experimental results demonstrate that even at very high degree of sparsity ($\leq$ 5%), our method simultaneously enhances generalization and downstream task performance. This confirms that our approach effectively mitigates the catastrophic forgetting issue and further promotes knowledge harmonization in MLLMs.*

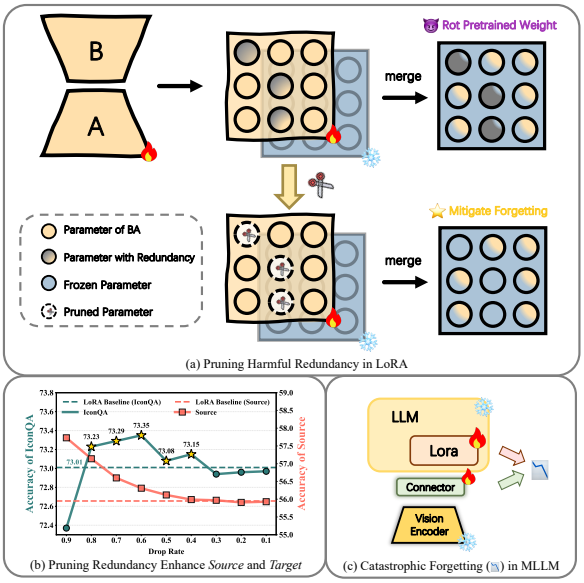


Figure 1. **Motivation**. Fine-tuning MLLM with LoRA on downstream tasks generates numerous harmful redundancy. (a) Illustration of post-pruning LoRA to reduce redundancy. (b) Simply pruning harmful redundancy in LoRA based on magnitude ***reduces forgetting of pretrained knowledge*** (*Source*), and even ***enhances downstream task performance*** (*Target*). (c) Current MLLMs suffer from catastrophic forgetting in both LLM and connector with fewer parameters, as detailed in Sec. 4.3.1.

1. Introduction

Multimodal large language models (MLLMs) are known for their strong generalization abilities, making them adaptable across modalities and tasks [7, 31, 35, 37, 63]. Visual instruction tuning [38] is often applied to enhance performance on specific downstream tasks. However, as the model size grows, full fine-tuning becomes resource-intensive. Consequently, LoRA-based methods [17, 30, 39, 79] have gained popularity for their efficiency, preserving the model architecture without additional inference costs. While LoRA induces less forgetting than full fine-tuning [3, 50], merging it with the pretrained model still modifies all parameters, posing a risk of severe generalization loss. The phenomenon of catastrophic forgetting in LoRA has been widely observed across several studies [9, 14, 69, 72, 84]. Furthermore, this problem is especially pronounced in MLLMs due to their complex architectures, cross-modal data differences, and task interference, posing heightened challenges in maintaining general knowl-

*Equal contributions
†Corresponding Author

edge while improving performance on downstream tasks [27, 55, 59, 77]. These issues make it challenging to balance general and specialized knowledge in deploying MLLMs, where both capabilities are essential.

To enable MLLMs to learn downstream tasks while preserving generalization, an intuitive approach is to minimize conflicts with pre-existing general knowledge during new task learning. However, although reducing conflicts can help preserve generalization, it may compromise performance on downstream tasks, raising the question: *How can we better balance this trade-off?* We observe a key phenomenon: large foundation models often exhibit ***significant parameter redundancy*** in delta weights after supervised fine-tuning (SFT) [73]. Furthermore, recent research has successfully eliminated redundant parameters in LoRA through post-pruning, showing that LoRA retains harmful redundancy after training, which not only fails to effectively serve the target task but also erodes pretrained knowledge [84]. Our additional experiments confirm that these harmful redundant parameters in LoRA are abundant and explicit: by simply applying post-pruning to the trained LoRA based on parameter magnitudes, both the general and specialized capabilities of MLLMs can be simultaneously improved, as is shown in Fig. 1 (a) and (b). However, as sparsity increases, post-pruning methods tend to reduce downstream task performance [78]. Motivated by the substantial harmful redundancy in LoRA after SFT, we consider **sculpting LoRA for redundancy reduction** during training, to ensure both generalization and specialization ability.

However, mitigating LoRA redundancy faces two crucial challenges: First, unlike standard parameter modules, sparsity cannot be applied directly to the product matrix BA of LoRA during training, as the pruned BA cannot be decomposed back into B and A. Additionally, directly sparsifying B and A does not necessarily yield a sparse product matrix BA [75]. Thus, this leads to the **LoRA Sparsity Dilemma**, which fails to ensure the necessary sparsity of BA. This raises the first challenge: **I)** *How to ensure the necessary sparsity of LoRA for redundancy reduction?* Second, although removing redundant parameters can restore pretrained general knowledge and thus mitigate forgetting, it does not address a key issue: the remaining task-specific parameters may still risk conflicting with important parameters in the pretrained model. LoRA updates faithfully follow the current empirical loss minimization, thus encountering optimization interference from pretrained knowledge behavior, which brings the **LoRA Optimization Conflict**, leading to general knowledge forgetting. Thus, this raises the second key challenge: **II)** *How to calibrate LoRA update trajectory to alleviate knowledge conflicts?*

In response to these challenges, we introduce a novel framework named **LoRASculpt**, which integrates sparse updates into LoRA for redundancy reduction, and introduces a Conflict Mitigation Regularizer for knowledge harmonization. Specifically, to address challenge **I)** , we implement *Sparsifying LoRA for Redundancy Reduction*, introducing sparsity into the training process of LoRA with rigorous theoretical guarantees. We prove the expected sparsity of the product of two sparse low-rank matrices and derive an upper bound for the probability of exceeding the expected sparsity, thus effectively resolving the LoRA Sparsity Dilemma. For challenge **II)** , we implement *Regularizing LoRA for Knowledge Harmonization*. A pretrained knowledge-guided Conflict Mitigation Regularizer is proposed to adjust the optimization trajectory of LoRA, directing task-specific knowledge injection into parameter locations that are less critical in the pretrained model, thereby minimizing the LoRA Optimization Conflict. We further discuss the synergy between the two components in mitigating conflicts between general and specialized knowledge.

Extensive experiments on VQA and Captioning tasks demonstrate that `LoRASculpt` effectively mitigates generalization loss in MLLMs. Moreover, by appropriately mitigating the optimization conflict and pruning the redundancy of LoRA, our approach can even enhance performance on downstream tasks. We reveal that severe forgetting also occurs in connector module, which typically has fewer parameters than LLM module. Adapting our method to connectors with simple modifications can further boost performance. Our main contributions are summarized as follows:

❶ ***Re-examining Forgetting in MLLM Trained with LoRA.*** Our findings indicate that LoRA exhibits abundant harmful redundancy after SFT in MLLMs, which undermines both generalization and specialization performance, even within the connector with fewer parameters.

❷ ***Novel Parameter-Efficient Framework for Mitigating Forgetting in MLLMs.*** Building on the phenomenon of parameter redundancy in LoRA, we effectively mitigate forgetting by addressing the LoRA Sparsity Dilemma and LoRA Optimization Conflict.

❸ ***Theoretical Guarantees and Experimental Validation.*** We provide theoretical guarantees for introducing sparse updates into LoRA to reduce redundant parameters, and further demonstrate the effectiveness and robustness of our framework through comprehensive experiments.

2. Related Works

Multimodal Large Language Models. With the advancement of multimodal learning, the generalization of small-scale multimodal models has been widely explored [19, 21, 23], and recently extended to Multimodal Large Language Models (MLLMs) [1, 2, 24, 37, 38, 70]. MLLMs have significantly improved the multimodal understanding, cross-modal reasoning, and problem-solving abilities by integrating visual encoders [51, 76] with LLMs [8, 10, 60, 61], demonstrating strong generalization ability. The connec-

tor module such as MLPs and the Q-Former [32] enhances alignment between visual and language features, facilitating smooth and efficient interaction between the visual encoder and language model. Visual instruction tuning [38] further adapts MLLMs for specific downstream tasks, improving task-specific performance [22, 84].

Catastrophic Forgetting. Deep learning models often suffer from catastrophic forgetting, where previously learned knowledge is lost when learning new tasks [20, 21, 34, 46, 47, 62, 67]. Various continual learning algorithms have been proposed to address this issue, generally categorized into rehearsal-based [5, 41, 53], regularization-based [29, 40, 54], and architecture-based [52, 68, 71, 82] approaches. Some traditional methods have explored using pruning to mitigate forgetting [43, 44, 57, 65, 66]; however, these are mostly designed for small models and rely heavily on full-model fine-tuning, making them less applicable for fine-tuning large foundation models. In the era of large foundation models, the traditional forgetting problem shifts to maintaining the model's strong generalization ability after downstream task fine-tuning [9, 14, 84]. Additionally, these models are often closed-source, with only model weight accessible. Recently, SPU [80] demonstrates the feasibility of sparse updates for fine-tuning CLIP to mitigate general knowledge forgetting. Other studies have explored the forgetting issue when training LLMs with LoRA. LoRAMoE [9] and SLIM [14] apply MoE architectures within LoRA to address generalization loss in LLMs. CorDA [69] builds task-aware adapters through context-guided weight decomposition to preserve world knowledge. Model Tailor [84] pioneers a parameter-efficient solution to MLLM forgetting, but this post-training approach carries the risk of leading to suboptimal downstream performance [78].

3. Methodology

3.1. Preliminary

Low-Rank Adaptation (LoRA) [17] is a parameter-efficient fine-tuning method for adapting large foundation models with minimal computational cost. Instead of updating all parameters, LoRA learns weight changes by introducing low-rank matrices B and A, so that fine-tuned weights W are expressed as:

$$W = W_0 + \Delta W = W_0 + BA, \tag{1}$$

where W_0 is the pre-trained weight matrix, and $r \ll \min(p, q)$ denotes the low rank of LoRA.

Harmful Redundant Delta Weights after SFT. Recent study shows that the delta vectors after supervised fine-tuning (SFT) in LLMs are highly redundant. Removing most of these parameters has minimal impact on performance [74]. Model Tailor reached similar conclusions on MLLMs, suggesting that redundant parameters in LoRA may further degrade downstream task performance [84]. Our further experiments show that harmful redundancy in LoRA are abundant and explicit. Simply pruning low-magnitude parameters can mitigate generalization loss and even improve downstream task performance, as shown in Fig. 1. We restate this key phenomenon as follows:

> **Harmful Parameter Redundancy in LoRA**: *LoRA introduces substantial and explicit **harmful redundancy** during SFT, which not only diminishes the generalization ability of MLLM but can also fail to contribute positively on downstream tasks.*

3.2. Proposed Method

Inspired by our observations in Sec. 3.1 that LoRA generates substantial harmful redundancy when SFT on MLLMs, we propose `LoRASculpt` to finely sculpt LoRA, enabling the elimination of redundant parameters and achieving a balance between general and specialized knowledge. We divide our method into two main components: establishing the feasibility of sparsifying LoRA for redundancy reduction (Sec. 3.2.1), and regularizing LoRA for knowledge harmonization (Sec. 3.2.2). Finally, we present the adaptation of LoRASculpt for the MLLM connector.

3.2.1. Sparsifying LoRA for Redundancy Reduction

Given the presence of substantial harmful redundancy in LoRA, we aim to introduce sparsity by pruning redundant parameters. In this section, we demonstrate the feasibility of incorporating sparse updates into LoRA training, effectively compressing downstream task knowledge into a critical subset of sparse parameters.

However, as mentioned in challenge I), introducing sparsity in LoRA leads to the ***LoRA Sparsity Dilemma***—incorporating sparsity into LoRA is not trivial. We claim that, although the product of two sparse matrices is not necessarily sparse, under the conditions of low-rank matrices in LoRA and high sparsity, this sparsity can be theoretically bounded with formal guarantees.

To implement pruning, we need to evaluate parameter importance, a considerable amount of research has been conducted, which can be broadly categorized into magnitude-based [12, 15, 58] and gradient-based [11, 48, 66] approaches. Although gradient-based methods often combine magnitude and gradient information to improve importance estimation, when using LoRA in resource-constrained scenarios, minimizing memory usage and computational cost during fine-tuning becomes critical. Therefore, we adopt the simple yet effective and efficient importance evaluation method that leverages the magnitude of parameters as a proxy for importance. Specifically, after a warm-up period, we assess the magnitude of each parameter and prune the less important parameters within the low-rank

matrices A and B in LoRA, resulting in pruned matrices $\widetilde{A}$ and $\widetilde{B}$ as follows:

$$\begin{cases} \widetilde{A} = M_A \odot A \\ \widetilde{B} = M_B \odot B, \end{cases} \tag{2}$$

where M_A and M_B are sparsity masks defined by the sparsity levels s_A and s_B, respectively. Specifically, M_A and M_B are constructed as follows:

$$M_A = \text{Mask}(A, s_A), \quad M_B = \text{Mask}(B, s_B), \tag{3}$$

where $\text{Mask}(X, s)$ denotes a function that generates a binary mask for matrix X, retaining a fraction s of elements based on importance (i.e., magnitude). Unlike the magnitude of overall model, the magnitude of LoRA reflects accumulated gradient updates on downstream tasks, i.e., delta parameters [73]. This cumulative effect makes LoRA more robust than single-step gradient estimation, as it smooths out variations across steps and reduces noise sensitivity.

However, this introduces a potential issue: ***the product of two sparse matrices B and A, denoted as BA, is not necessarily sparse [75].*** If BA cannot maintain sparsity, then when BA is merged with the frozen pre-trained matrix W, it will result in modifications across all parameters of W. Consequently, this undermines the goal of minimizing changes to the pre-trained model's knowledge.

We claim that in low-rank adaptation (LoRA), adopting a high sparsity level allows the matrix BA to remain sparse with high probability. First, we present Theorem 3.1, which provides an expected sparsity estimation for the matrix BA.

Theorem 3.1. *Let $B \in \mathbb{R}^{p\times r}$ and $A \in \mathbb{R}^{r\times q}$ be two low rank matrices in LoRA, then the expected sparsity of the product matrix $BA \in \mathbb{R}^{p\times q}$ is given by:*

$$\mathbb{E}\left[s_{BA}\right] = 1 - (1 - s_B s_A)^r. \tag{4}$$

Proof. See Appendix A. □

Next, we established an upper bound on the probability that the actual sparsity of BA exceeds this expectation.

Theorem 3.2. *Let $B \in \mathbb{R}^{p\times r}$ and $A \in \mathbb{R}^{r\times q}$ be two low rank matrices in LoRA, where the sparsity of B is s_B and the sparsity of A is s_A. Define $C = BA$, with sparsity s_C. Then, for any $\delta > 0$:*

$$\mathbb{P}\left(|s_C - \mathbb{E}[s_C]| \geq \delta\right) \leq 2\exp\left(-\frac{2\delta^2 pq}{r(p+q)}\right), \tag{5}$$

where the expected sparsity $\mathbb{E}[s_C]$ is given by Theorem 3.1
Proof. See Appendix B. □

It is important to note that these theorems are meaningful only when the rank r is small, and the sparsity levels s_A and s_B are low. Fortunately, in LoRA, matrices A and B are inherently low-rank. Additionally, based on prior knowledge of the substantial redundancy in LoRA parameters post-training, the sparsity levels s_A and s_B can be set to relatively low values. These conditions ensure the practical of these theorems, suggesting that BA is sparse with high probability. We further validate these theoretical conclusions with experiments shown in Fig. 5.

3.2.2. Regularizing LoRA for Knowledge Harmonization

Under theoretical guarantees, we have successfully introduced sparse updates into LoRA; however, this does not resolve a critical issue: the selected sparse parameter subset may still conflict with important parameters in the pre-trained model. As described in challenge **II)**, LoRA updates strictly adhere to the empirical loss minimization of the current downstream task, which may leads to interference from pre-trained knowledge behavior during optimization, resulting in the ***LoRA Optimization Conflict*** and causing general knowledge forgetting.

To mitigate this optimization conflict, we propose *Conflict Mitigation Regularizer* to calibrate the LoRA update trajectory. Specifically, during training, we employ pre-trained knowledge as guidance to regularize LoRA, steering its updates away from the key regions of general knowledge in the pre-trained model and directing new knowledge into less critical areas of the pre-trained parameters. *Notably, it is the high sparsity achievable by LoRA (as mentioned in Sec. 3.2.1) that enables precise knowledge injection.*

As discussed in previous, the magnitude of pre-trained parameters serves as a crucial measure of importance. Moreover, during efficient fine-tuning with LoRA, gradients of the pre-trained model are inaccessible. Therefore, balancing feasibility and efficiency, we adopt magnitude as the measure of importance of pre-trained weights to generate Magnitude-Guided Retention Mask, as detailed below.

Magnitude-Guided Retention Mask. To identify the critical components of the pre-trained parameters and guide the subsequent LoRA update trajectory, we apply transformations to the pre-trained parameters to emphasize crucial knowledge, as shown in the following formulation.

$$S_{ij} = \psi(W_{ij}) = \left|\frac{1}{\log\left(|\widetilde{W}_{ij}| + \epsilon\right)}\right|, \tag{6}$$

$$M_{ij} = \tanh(\omega S_{ij}) = \frac{e^{\omega S_{ij}} - e^{-\omega S_{ij}}}{e^{\omega S_{ij}} + e^{-\omega S_{ij}}}, \tag{7}$$

where $\widetilde{W}$ denotes the normalization of pretrained weights, given by $\frac{W}{\|W\|_2}$; ϵ denotes a small constant for numerical stability; ω controls the steepness of the $\tanh$ function, and determines whether weights of different magnitudes are assigned importance values with smooth or sharp distinctions; and $i = 1, 2, \ldots, p$, $j = 1, 2, \ldots, q$ denote matrix indices.

The purpose of the above equations can be divided into two points. Eq. (6) alleviates $\log$ function to naturally compress the wide-ranging values of the weight parameters, resulting in a more balanced distribution of importance scores. Parameter with larger magnitude is mapped to higher importance score S. Eq. (7) rescales the impor-

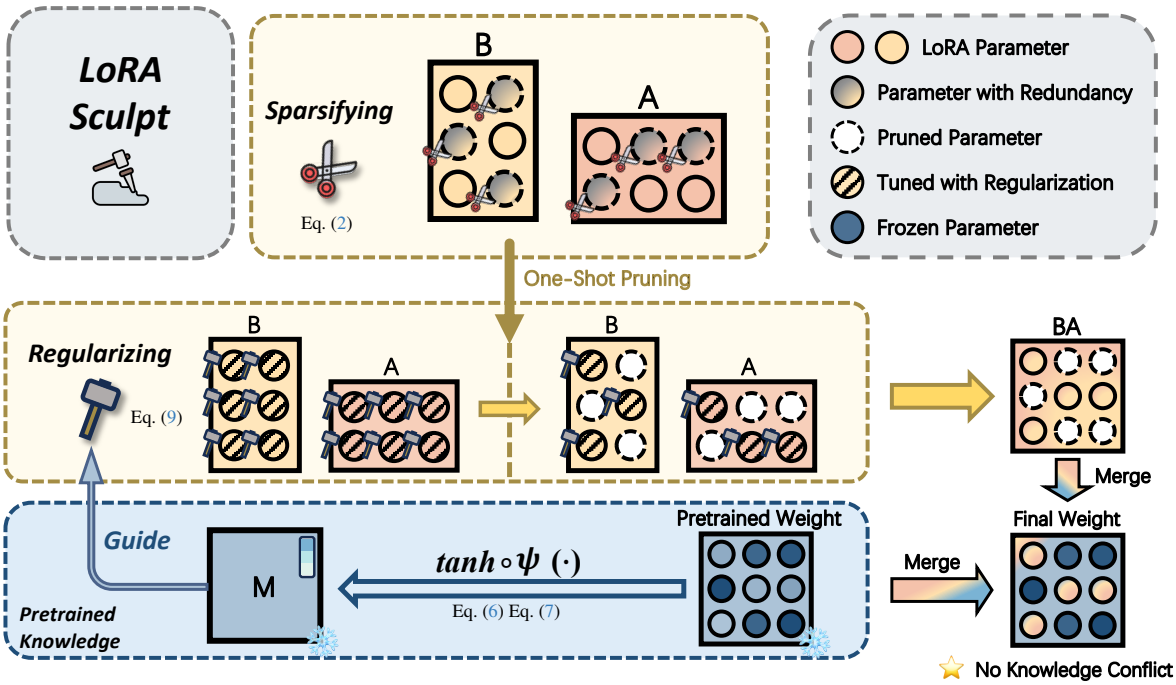


Figure 2. **Framework Illustration**. LoRASculpt consists of two components: *Sparsifying* and *Regularizing*. *Sparsifying* process aims to reduce redundancy by retaining only a sparse subset of parameters in the low-rank matrices. *Regularizing* process is guided by the pretrained-knowledge informed regularization, which adjusts the optimization trajectory to mitigate conflicts between the sparse LoRA subset and the pretrained knowledge, thereby promoting knowledge harmonization. The resulting sparse product matrix BA can be merged with pretrained weights without severe knowledge conflicts.

tance scores to the $(0, 1)$ range, placing them on a common scale, and facilitating subsequent knowledge-guided regularization. Thus, we obtain the Magnitude-Guided Retention Mask, which represents the strength of protection for the pre-trained knowledge.

Conflict Mitigation Regularizer. To mitigate the potential conflicts between downstream and essential pre-trained knowledge, which could cause forgetting [29, 43], we further calibrate the LoRA update trajectory leveraging the retention mask M derived from Eq. (7). We propose a pre-trained knowledge guided Conflict Mitigation Regularizer by applying the Frobenius norm to the Hadamard product of the retention mask M and the low-rank LoRA matrix product BA. This reduces the magnitude in BA corresponding to positions with higher retention strengths in M, leading to smaller modifications to the pre-trained weight W when mergeing LoRA and thus preserving the essential general knowledge embedded in the pre-trained model. The Conflict Mitigation Regularizer is defined as follows.

$$\mathcal{L}_{CMR} = \|M \odot (BA)\|_F. \tag{8}$$

Thus, the final loss function is defined as follows.

$$\mathcal{L} = \mathcal{L}_{Task} + \alpha \cdot \mathcal{L}_{CMR}, \tag{9}$$

where the hyperparameter α is the balancing coefficient between the two losses, representing the degree of preservation of the pre-trained general knowledge.

Synergy Between the Two Components. We have proposed two components to address the *LoRA Sparsity Dilemma*, as mentioned in I), and the *LoRA Optimization Conflict*, as discussed in II). These components work synergistically toward a common goal: eliminating redundancy and identifying a critical sparse subset to alleviate knowledge conflicts. Specifically, we implement one-shot pruning during training, maintaining this sparse subset unchanged [1], and divide the training process into two phase: *knowledge warming* phase and *knowledge integration* phase. The proposed knowledge-guided regularization effectively functions in both phases: in the knowledge warming phase, it steers LoRA away from conflicts with critical pre-trained knowledge, aiding in the selection of the sparse parameter subset. In the knowledge integration phase, it leverages pre-trained knowledge to guide the optimization process, further harmonizing the selected sparse parameter subset with pre-trained knowledge to promote knowledge integration.

Theoretical Proof of Sparsity. In Theorem 3.1 and Theorem 3.2, we have demonstrated the expected sparsity of the resulting delta weight after introducing sparse updates into LoRA and further establish an upper bound for the probability that the actual sparsity exceeds this expectation. Furthermore, we claim that with the application of Knowledge-Guided Regularization, both the expected sparsity and its

[1] Crucial for fine-tuning with an unchanged sparse subset [13]

bound remain valid. Please refer to Appendix D for detailed proofs.

3.2.3. Method Adaptation in MLLM Connector

The connector is a key bridge between the visual encoder and language model in multimodal large language models, with the quality of conveyed visual information directly impacting overall performance [4, 33]. Therefore, full fine-tuning is often employed to connector when adapting MLLMs on downstream tasks [37, 38, 83].

Our experiment shown in Tab. 2 reveals that, despite the connector having much fewer parameters than the LLM, it still experiences substantial forgetting. Given the connector's key role in alignment, we make a simple modification to LoRASculpt for its adaptation in connector. Specifically, to prevent potential performance degradation, we avoid hard pruning on the connector and instead apply regularization for soft sparsity, as shown in the following equation.

$$\mathcal{L}_{CMR}^{\text{Con}} = \|M^{\text{Con}} \odot (BA)\|_1, \tag{10}$$

where $\|\cdot\|_1$ is L_1 norm, which encourages soft sparsity in BA. Thus, the final loss function when applying LoRASculpt into MLLM connector is defined as follows.

$$\mathcal{L} = \mathcal{L}_{Task} + \alpha \cdot \mathcal{L}_{CMR}^{\text{LLM}} + \beta \cdot \mathcal{L}_{CMR}^{\text{Con}}, \tag{11}$$

where α and β represent the degree of preservation of pre-trained general knowledge in the LLM and the connector, respectively; $\mathcal{L}_{CMR}^{\text{LLM}}$ is consistent with the formula in Eq. (8).

The framework of `LoRASculpt` is illustrate in Fig. 2, and the algorithm is outlined in Appendix E.

4. Experiments

4.1. Experimental Setup

Datasets and Architecture. For downstream knowledge acquisition, we fine-tune and evaluate on Visual Question Answering (VQA) and Captioning tasks using the IconQA [42] and COCO-Caption [36] datasets. We randomly sample 10k examples from each training dataset, following the resource setting in [6, 83]. To measure general knowledge forgetting in MLLM, we assess performance on four upstream datasets: OKVQA [45], OCRVQA [49], GQA [26], and TextVQA [56] after downstream task fine-tuning. Leveraging the representative open-source framework LLaVA-1.5 [37], we conduct detailed evaluations of compared LoRA-based methods at ranks 16, 32, and 64.

Implementation Details. All compared baselines are adapted based on LoRA and implemented within the LLaVA-1.5[2] framework. Consistent with several works [37, 83], we apply LoRA to all layers of LLM with a learning rate of 2e-4, and use full fine-tuning in the connector with a learning rate of 2e-5. The training epoch is set to 3, and the batch size is default set to 16. Regularizing LoRA for Knowledge Harmonization (RKH) is only applied to W_Q, W_K, W_V, to encourage higher sparsity and knowledge harmonization, as attention layers have been shown to be more significant and redundant than others [16]. All experiments are conducted on 4 NVIDIA 4090 GPUs.

Compared Baselines. We compare proposed method LoRASculpt against LoRA-based PEFT methods, regularization techniques, and post-training approaches, including: (a) LoRA [ICLR'22] [17] (b) DoRA [ICML'24] [39] (c) OrthReg [ECCV'24] [18] (d) L2-Regularization [PNAS'17] [28] (e) DARE [ICML'24] [73] (f) Model Tailor [ICML'24] [84]. Additional evaluation details are provided in Appendix F.

4.2. Comparison to State-of-the-Arts

Quantitative Results. We conduct extensive experiments comparing LoRASculpt with state-of-the-art baselines across different ranks on two downstream tasks, as shown in Tab. 1. Several key observations are summarized:

❶ ***LoRASculpt alleviates the trade-off between general and specialized knowledge.*** By sculpting LoRA with redundancy reduction and knowledge harmonization, it achieves better results on both *target* and *source* compared to baseline LoRA methods, consistently achieving state-of-the-art results across different ranks (16, 32, 64) and tasks.

❷ ***Existing methods carry the risk of reducing downstream task accuracy.*** Excluding cases of catastrophic forgetting at rank=64 on IconQA (discussed in Sec. 4.3.1), both regularization and post-pruning methods may underperform compared to the LoRA baseline. For instance, regularization methods generally show lower target performance on the COCO-Caption dataset, and post-pruning methods show similar results on IconQA.

❸ ***LoRASculpt demonstrates great robustness.*** As the LoRA rank increases, the risk of forgetting intensifies, and existing methods show similar declines in performance. In contrast, LoRASculpt consistently maintains stable results.

Results Across Different Epochs. As SFT progresses, forgetting issue tends to worsen over longer training durations [3]. Thus, we examine the results of different methods as training epochs increase on IconQA dataset. Fig. 3 shows how *target* and *source* accuracy change across epochs for Tailor [84], LoRA [17], and our method. As epochs increase, LoRA display a steady decline in *source* accuracy, reflecting worsening forgetting. Although Tailor mitigates forgetting, it compromises downstream performance. In contrast, our method maintains stable accuracy for both *target* and *source*, demonstrating greater robustness.

4.3. Diagnostic Analysis

4.3.1. Catastrophic Forgetting in MLLM Connector

Increasing the rank of LoRA may raise the risk of overfitting and catastrophic forgetting [3, 64]. In our experiments in Tab. 1, we observed that when LoRA rank is set

[2] https://github.com/haotian-liu/LLaVA

Methods	IconQA							COCO-Caption						
	OKVQA	OCRVQA	GQA	TextVQA	*Source*	*Target*	*Avg*	OKVQA	OCRVQA	GQA	TextVQA	*Source*	*Target*	*Avg*
Zero-shot	57.99	66.20	61.93	58.23	61.09	23.18	42.13	57.99	66.20	61.93	58.23	61.09	40.40	50.74
LoRA Rank=16														
LoRA	51.10	54.90	55.56	49.48	52.76	84.14	68.45	47.34	60.55	56.91	45.61	52.60	112.22	82.41
DoRA	49.94	54.40	55.39	47.26	51.75	84.48	68.11	45.38	59.75	55.40	41.38	50.48	112.60	81.54
L2-Reg	49.96	51.40	55.15	47.47	51.00	84.66	67.83	46.05	61.50	56.60	43.54	51.92	111.83	81.88
Orth-Reg	53.12	56.10	57.43	51.00	54.41	84.52	69.47	48.11	59.85	56.83	46.69	52.87	111.87	82.37
Model Tailor	52.68	56.55	56.26	51.41	54.23	83.26	68.74	51.83	59.75	58.73	51.43	**55.44**	118.84	87.14
DARE	51.10	54.30	55.55	49.20	52.54	84.20	68.37	46.64	59.45	56.01	44.46	51.64	112.21	81.93
`LoRASculpt`	54.07	57.00	57.66	52.24	**55.24**	**85.02**	**70.13**$_{\uparrow 0.66}$	50.86	59.10	57.64	51.10	54.68	**121.26**	**87.97**$_{\uparrow 0.83}$
LoRA Rank=32														
LoRA	45.06	48.40	49.22	36.47	44.79	82.52	63.65	44.92	59.90	54.88	39.75	49.86	110.27	80.07
DoRA	46.35	48.50	40.80	33.01	42.17	84.82	63.49	44.10	60.05	54.69	40.17	49.75	109.25	79.50
L2-Reg	30.16	26.65	16.62	17.16	22.65	82.36	52.50	43.60	60.00	54.09	36.53	48.56	109.39	78.97
Orth-Reg	41.51	40.45	53.58	30.92	41.62	83.14	62.38	43.77	60.70	53.74	38.61	49.21	110.44	79.82
Model Tailor	50.07	53.85	53.04	46.92	50.97	81.76	66.37	50.99	60.60	58.54	49.65	**54.95**	117.64	86.29
DARE	44.39	46.85	48.75	34.84	43.71	82.28	62.99	42.98	58.65	53.92	38.61	48.54	108.57	78.56
`LoRASculpt`	53.52	59.50	57.63	53.76	**56.10**	**85.26**	**70.68**$_{\uparrow 4.31}$	49.99	58.65	57.63	50.73	54.25	**120.35**	**87.30**$_{\uparrow 1.01}$
LoRA Rank=64														
LoRA	0.04	0.00	20.22	0.28	5.14	37.24	21.19	31.51	54.80	42.35	28.13	39.20	107.90	73.55
DoRA	0.03	0.00	0.00	0.12	0.04	36.86	18.45	34.63	57.85	44.37	29.06	41.48	106.49	73.98
L2-Reg	27.46	31.75	17.98	9.66	21.71	81.79	51.75	40.95	59.05	52.83	36.00	47.21	106.41	76.81
Orth-Reg	39.04	35.90	25.59	21.21	30.44	81.71	56.07	45.78	60.70	55.69	43.58	51.44	108.16	79.80
Model Tailor	47.67	51.50	52.61	35.43	46.80	77.28	62.04	46.70	59.20	56.75	44.42	51.77	116.68	84.22
DARE	34.42	39.65	21.90	14.59	27.64	78.58	53.11	29.67	50.70	37.54	23.34	35.31	105.55	70.43
`LoRASculpt`	53.71	57.90	57.60	52.14	**55.34**	**84.74**	**70.04**$_{\uparrow 8.00}$	50.15	59.85	57.91	51.44	**54.84**	**120.39**	**87.61**$_{\uparrow 3.45}$

Table 1. **Comparison with State-of-the-Art Fine-Tuning Solutions for Multimodal Large Language Models (MLLMs)** on visual question answering task IconQA and image captioning task COCO-Caption. The optimal and sub-optimal results are denoted by boldface and underlining. ↑ means improved accuracy compared with the sub-optimal results. Please refer to Sec. 4.2 for detailed explanations.

LLM	Conn	Fine-Tune on IconQA						
		OKVQA	OCRVQA	GQA	TextVQA	*Source*	*Target*	*Avg*
Zero-shot		57.99	66.20	61.93	58.23	61.09	23.18	42.13
LoRA	FFT	0.04	0.00	20.22	0.28	5.14	37.24	21.19
LoRA	LoRA	0.76	2.45	0.01	2.48	1.43	81.60	41.51
LoRA	Freeze	51.10	52.15	54.17	45.91	50.83	84.61	67.72
Ours	**FFT**	53.71	57.90	57.60	52.14	55.34	84.74	70.04
Ours	**Ours**	56.91	64.15	60.93	56.22	**59.55**	**85.34**	**72.45**

Table 2. **Method Adaptation in MLLM Connector** shows significant catastrophic forgetting within the MLLM connector. Please refer to Sec. 4.3.1 for more details.

SRR	RKH	IconQA			COCO-Caption		
		Source	*Target*	*Avg*	*Source*	*Target*	*Avg*
Zero-shot		61.09	23.18	42.13	61.09	40.40	50.74
LoRA		44.79	82.52	63.65	49.86	110.27	80.07
✓		54.73	84.45	69.59	53.97	120.11	87.04
✓	✓	**56.10**	**85.26**	**70.68**	**54.25**	**120.35**	**87.30**

Table 3. **Ablative Study of Key Modules** for LoRASculpt, showing the effects of Sparsifying for Redundancy Reduction (SRR) and Regularizing for Knowledge Harmonization (RKH). Please refer to Sec. 4.3.2 for detailed discussion.

to 64 and fine-tuned for 3 epochs, both LoRA and DoRA without forgetting mitigation technique exhibit catastrophic forgetting. Our further experiments in Tab. 2 indicate that this issue is ***largely due to catastrophic forgetting within the MLLM connector.*** The results show that applying LoRA to the connector alleviates overfitting but still leads to catastrophic forgetting, while freezing the connector significantly reduces this effect. Additionally, our experiments demonstrate that applying our method to the connector with

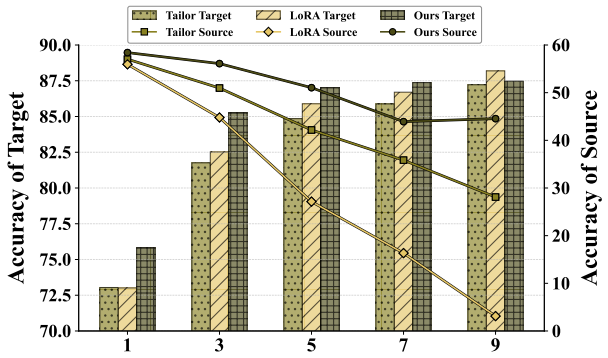
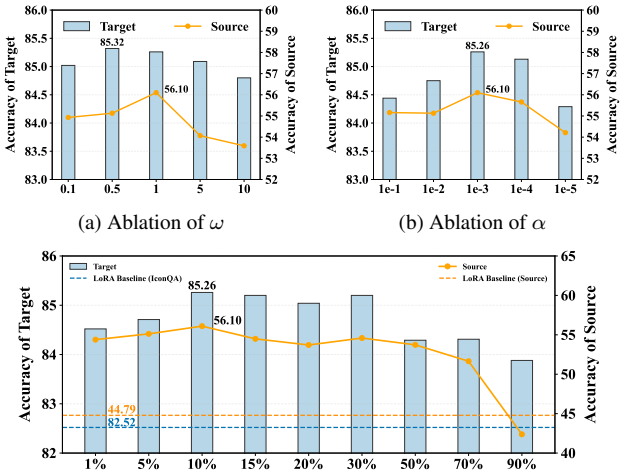


Figure 3. **Results Across Different Epochs** compared with SOTA method and baseline, showing LoRA obvious decline in *source* and Tailor compromise in *target* performance, while our method maintains stable for both. Please see details in Sec. 4.2

Eq. (11) achieves a better harmonization between general and specialized knowledge. For ablation study on the parameter β, which controls the sparsity strength for MLLM connector, please refer to the Appendix G.

4.3.2. Ablation Study

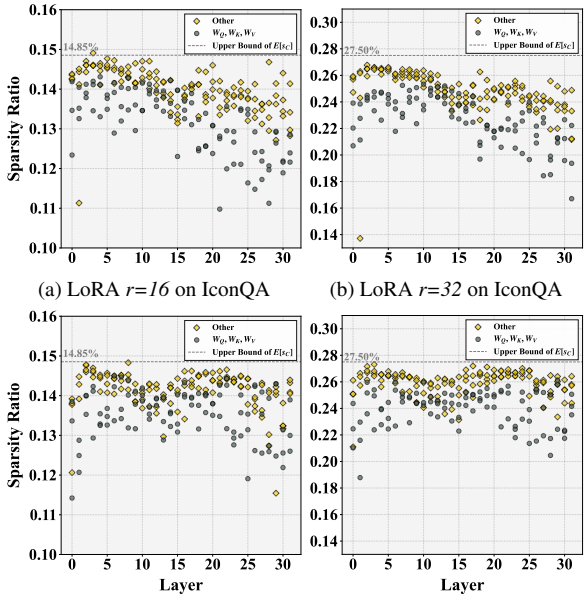
Ablation of Key Component. We present an ablation study of two key components of LoRASculpt in Tab. 3: including Sparsifying for Redundancy Reduction (SRR) and Regularizing for Knowledge Harmonization (RKH). Our experiments demonstrate the effectiveness of SRR in removing harmful redundant parameters in LoRA, while RKH further enhances the harmonization between the general and down-



(a) Ablation of ω (b) Ablation of α

(c) Ablation of Sparsity Ratio s

Figure 4. **Hyperparameter Study** for function steepness w (Eq. (7)), balancing coefficient α (Eq. (9)), and sparsity ratio s (Eq. (3)) when fix LoRA rank=32 and fine-tuning on IconQA. Please refer to Sec. 4.3.2 for detailed discussion.



(a) LoRA *r=16* on IconQA (b) LoRA *r=32* on IconQA

(c) LoRA *r=16* on COCO (d) LoRA *r=32* on COCO

Figure 5. **Actual Sparsity of LoRA in Each Layer** with the sparsity $s_A = s_B = 0.1$. Please refer to Sec. 4.3.3 for details.

stream task knowledge.

Ablation of Hyperparameters. We show the impact of the hyperparameters α (Eq. (9)) and ω (Eq. (7)) on the performance in Fig. 4b and Fig. 4a. We set the sparsity ratio of low-rank matrices A and B as $s_A = s_B = s$ (Eq. (3)), then evaluate performance across different sparsity in Fig. 4c. Optimal performance is achieved when $\omega = 1$, $\alpha = 10^{-3}$ and $s_A = s_B = s = 10\%$, which are used in main experiments Tab. 1. Further analysis of the figure reveals that when the sparsity ratio is set to 1%, performance on both source and target surpasses the LoRA baseline, indicating that LoRA contains highly redundancy after fine-tuning on MLLMs. This aligns with our observations in Sec. 3.1.

4.3.3. Empirical Validation of Proposed Theorem

In Sec. 3.2, we prove an upper bound (Theorem 3.1) on the expected sparsity of BA given fixed sparsity levels in the low-rank matrices A and B, and further provide an upper bound for the probability that actual sparsity exceeds this expectation (Theorem 3.2). We present experimental validation shown in Fig. 5, where the sparsity of the product matrix BA in nearly all layers falls below the expected sparsity derived from theoretical analysis (indicated by the gray line). Under these theoretical guarantees, our method successfully introduces sparsity into LoRA.

Further analysis of Fig. 5 reveals two insights: (1) The sparsity of LoRA in the MLP layers (11008×4096) is closer to the expected bound than in the W_Q, W_K, W_V layers (4096×4096), aligning with the conclusion of Theorem 3.2. (2) LoRASculpt applies sparsity to the low-rank matrices B and A, automatically enforcing varying degrees of redundancy reduction across layers. Recent studies on MLLM layer-wise roles suggest that shallow layers primarily integrate global image information, while deeper layers handle syntactic structures [25, 81, 85]. For IconQA dataset, where answers are simple single letters, LoRASculpt exhibits lower sparsity in deeper layers. Conversely, for the COCO-Caption dataset, which requires diverse answer formats, deeper layers retain more parameters to capture rich semantic information. This demonstrates the adaptive redundancy reduction of LoRASculpt rather than applying a fixed sparsity rate across all layers.

5. Conclusion

In this paper, we introduce `LoRASculpt`, a parameter-efficient framework designed to address the forgetting issue for MLLMs. Our method consists of two components: *Sparsifying LoRA for Redundancy Reduction*, which eliminates harmful redundancy, and *Regularizing LoRA for Knowledge Harmonization*, which balances general and task-specific knowledge to achieve better alignment. Through theoretical analysis and extensive experiments, we show that our method effectively retains general knowledge while enhancing downstream task performance.

Acknowledgement. This research is supported by the National Key Research and Development Project of China (2024YFC3308400), the National Natural Science Foundation of China (Grants 62361166629, 62176188, 623B2080), the Wuhan University Undergraduate Innovation Research Fund Project. The supercomputing system at the Supercomputing Center of Wuhan University supported the numerical calculations in this paper.

References

[1] Jinze Bai, Shuai Bai, Shusheng Yang, Shijie Wang, Sinan Tan, Peng Wang, Junyang Lin, Chang Zhou, and Jingren Zhou. Qwen-vl: A versatile vision-language model for understanding, localization, text reading, and beyond. *arXiv preprint arXiv:2308.12966*, 2023. 2

[2] Yang Bai, Yucheng Ji, Min Cao, Jinqiao Wang, and Mang Ye. Chat-based person retrieval via dialogue-refined cross-modal alignment. In *CVPR*, 2025. 2

[3] Dan Biderman, Jose Gonzalez Ortiz, Jacob Portes, Mansheej Paul, Philip Greengard, Connor Jennings, Daniel King, Sam Havens, Vitaliy Chiley, Jonathan Frankle, et al. Lora learns less and forgets less. *arXiv preprint arXiv:2405.09673*, 2024. 1, 6

[4] Junbum Cha, Wooyoung Kang, Jonghwan Mun, and Byungseok Roh. Honeybee: Locality-enhanced projector for multimodal llm. In *CVPR*, pages 13817–13827, 2024. 6

[5] Arslan Chaudhry, Marcus Rohrbach, Mohamed Elhoseiny, Thalaiyasingam Ajanthan, Puneet K Dokania, Philip HS Torr, and Marc'Aurelio Ranzato. On tiny episodic memories in continual learning. *arXiv preprint arXiv:1902.10486*, 2019. 3

[6] Guanzheng Chen, Fangyu Liu, Zaiqiao Meng, and Shangsong Liang. Revisiting parameter-efficient tuning: Are we really there yet? *arXiv preprint arXiv:2202.07962*, 2022. 6

[7] 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*, pages 24185–24198, 2024. 1

[8] Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E Gonzalez, et al. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality. *See https://vicuna. lmsys. org (accessed 14 April 2023)*, 2(3):6, 2023. 2

[9] Shihan Dou, Enyu Zhou, Yan Liu, Songyang Gao, Wei Shen, Limao Xiong, Yuhao Zhou, Xiao Wang, Zhiheng Xi, Xiaoran Fan, et al. Loramoe: Alleviating world knowledge forgetting in large language models via moe-style plugin. In *ACL*, pages 1932–1945, 2024. 1, 3

[10] Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024. 2

[11] Yujie Feng, Xu Chu, Yongxin Xu, Guangyuan Shi, Bo Liu, and Xiao-Ming Wu. Tasl: Continual dialog state tracking via task skill localization and consolidation. In *ACL*, 2024. 3

[12] Jonathan Frankle and Michael Carbin. The lottery ticket hypothesis: Finding sparse, trainable neural networks. In *ICLR*, 2019. 3

[13] Demi Guo, Alexander M Rush, and Yoon Kim. Parameter-efficient transfer learning with diff pruning. In *ACL*, 2020. 5

[14] Jiayi Han, Liang Du, Hongwei Du, Xiangguo Zhou, Yiwen Wu, Weibo Zheng, and Donghong Han. Slim: Let llm learn more and forget less with soft lora and identity mixture. *arXiv preprint arXiv:2410.07739*, 2024. 1, 3

[15] Song Han, Jeff Pool, John Tran, and William Dally. Learning both weights and connections for efficient neural network. In *NeurIPS*, 2015. 3

[16] Shwai He, Guoheng Sun, Zheyu Shen, and Ang Li. What matters in transformers? not all attention is needed. *arXiv preprint arXiv:2406.15786*, 2024. 6

[17] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. In *ICLR*, 2022. 1, 3, 6

[18] Jiajun Hu, Jian Zhang, Lei Qi, Yinghuan Shi, and Yang Gao. Learn to preserve and diversify: Parameter-efficient group with orthogonal regularization for domain generalization. In *ECCV*, 2024. 6, 3

[19] Ming Hu, Peiheng Zhou, Zhihao Yue, Zhiwei Ling, Yihao Huang, Anran Li, Yang Liu, Xiang Lian, and Mingsong Chen. Fedcross: Towards accurate federated learning via multi-model cross-aggregation. In *ICDE*, pages 2137–2150. IEEE, 2024. 2

[20] Wenke Huang, Mang Ye, and Bo Du. Learn from others and be yourself in heterogeneous federated learning. In *CVPR*, 2022. 3

[21] Wenke Huang, Mang Ye, Zekun Shi, and Bo Du. Generalizable heterogeneous federated cross-correlation and instance similarity learning. *TPAMI*, 2023. 2, 3

[22] Wenke Huang, Jian Liang, Zekun Shi, Didi Zhu, Guancheng Wan, He Li, Bo Du, Dacheng Tao, and Mang Ye. Learn from downstream and be yourself in multimodal large language model fine-tuning. *arXiv preprint arXiv:2411.10928*, 2024. 3

[23] Wenke Huang, Mang Ye, Zekun Shi, Guancheng Wan, He Li, Bo Du, and Qiang Yang. A federated learning for generalization, robustness, fairness: A survey and benchmark. *TPAMI*, 2024. 2

[24] Wei Huang, Xingyu Zheng, Xudong Ma, Haotong Qin, Chengtao Lv, Hong Chen, Jie Luo, Xiaojuan Qi, Xianglong Liu, and Michele Magno. An empirical study of llama3 quantization: From llms to mllms. *Visual Intelligence*, 2(1): 36, 2024. 2

[25] Wenke Huang, Jian Liang, Xianda Guo, Yiyang Fang, Guancheng Wan, Xuankun Rong, Chi Wen, Zekun Shi, Qingyun Li, Didi Zhu, et al. Keeping yourself is important in downstream tuning multimodal large language model. *arXiv preprint arXiv:2503.04543*, 2025. 8

[26] Drew A Hudson and Christopher D Manning. Gqa: A new dataset for real-world visual reasoning and compositional question answering. In *CVPR*, pages 6700–6709, 2019. 6

[27] Yao Jiang, Xinyu Yan, Ge-Peng Ji, Keren Fu, Meijun Sun, Huan Xiong, Deng-Ping Fan, and Fahad Shahbaz Khan. Effectiveness assessment of recent large vision-language models. *Visual Intelligence*, 2(1):17, 2024. 2

[28] 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. *PNAS*, pages 3521–3526, 2017. 6, 3

[29] 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*, 114(13):3521–3526, 2017. 3, 5

[30] Dawid Jan Kopiczko, Tijmen Blankevoort, and Yuki M Asano. VeRA: Vector-based random matrix adaptation. In *ICLR*, 2024. 1

[31] Bo Li, Yuanhan Zhang, Dong Guo, Renrui Zhang, Feng Li, Hao Zhang, Kaichen Zhang, Yanwei Li, Ziwei Liu, and Chunyuan Li. Llava-onevision: Easy visual task transfer. *arXiv preprint arXiv:2408.03326*, 2024. 1

[32] 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 *ICML*, pages 19730–19742. PMLR, 2023. 3

[33] Wentong Li, Yuqian Yuan, Jian Liu, Dongqi Tang, Song Wang, Jianke Zhu, and Lei Zhang. Tokenpacker: Efficient visual projector for multimodal llm. *arXiv preprint arXiv:2407.02392*, 2024. 6

[34] Yichen Li, Haozhao Wang, Wenchao Xu, Tianzhe Xiao, Hong Liu, Minzhu Tu, Yuying Wang, Xin Yang, Rui Zhang, Shui Yu, et al. Unleashing the power of continual learning on non-centralized devices: A survey. *arXiv preprint arXiv:2412.13840*, 2024. 3

[35] Zhang Li, Biao Yang, Qiang Liu, Zhiyin Ma, Shuo Zhang, Jingxu Yang, Yabo Sun, Yuliang Liu, and Xiang Bai. Monkey: Image resolution and text label are important things for large multi-modal models. In *CVPR*, pages 26763–26773, 2024. 1

[36] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *ECCV*, pages 740–755, 2014. 6

[37] Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction tuning. In *CVPR*, 2023. 1, 2, 6

[38] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. In *NeurIPS*, 2023. 1, 2, 3, 6

[39] Shih-Yang Liu, Chien-Yi Wang, Hongxu Yin, Pavlo Molchanov, Yu-Chiang Frank Wang, Kwang-Ting Cheng, and Min-Hung Chen. Dora: Weight-decomposed low-rank adaptation. In *ICML*, 2024. 1, 6, 3

[40] Xialei Liu, Marc Masana, Luis Herranz, Joost Van de Weijer, Antonio M Lopez, and Andrew D Bagdanov. Rotate your networks: Better weight consolidation and less catastrophic forgetting. In *ICPR*, pages 2262–2268. IEEE, 2018. 3

[41] David Lopez-Paz and Marc'Aurelio Ranzato. Gradient episodic memory for continual learning. *NeurIPS*, 30, 2017. 3

[42] Pan Lu, Liang Qiu, Jiaqi Chen, Tony Xia, Yizhou Zhao, Wei Zhang, Zhou Yu, Xiaodan Liang, and Song-Chun Zhu. Iconqa: A new benchmark for abstract diagram understanding and visual language reasoning. In *NeurIPS*, 2021. 6

[43] Arun Mallya and Svetlana Lazebnik. Packnet: Adding multiple tasks to a single network by iterative pruning. In *CVPR*, pages 7765–7773, 2018. 3, 5

[44] Arun Mallya, Dillon Davis, and Svetlana Lazebnik. Piggyback: Adapting a single network to multiple tasks by learning to mask weights. In *ECCV*, pages 67–82, 2018. 3

[45] Kenneth Marino, Mohammad Rastegari, Ali Farhadi, and Roozbeh Mottaghi. Ok-vqa: A visual question answering benchmark requiring external knowledge. In *CVPR*, 2019. 6

[46] James L McClelland, Bruce L McNaughton, and Randall C O'Reilly. Why there are complementary learning systems in the hippocampus and neocortex: insights from the successes and failures of connectionist models of learning and memory. *Psychological review*, 102(3):419, 1995. 3

[47] Michael McCloskey and Neal J Cohen. Catastrophic interference in connectionist networks: The sequential learning problem. In *Psychology of learning and motivation*, pages 109–165. Elsevier, 1989. 3

[48] Seyed Iman Mirzadeh, Mehrdad Farajtabar, Razvan Pascanu, and Hassan Ghasemzadeh. Understanding the role of training regimes in continual learning. In *NeurIPS*, pages 7308–7320, 2020. 3

[49] Anand Mishra, Shashank Shekhar, Ajeet Kumar Singh, and Anirban Chakraborty. Ocr-vqa: Visual question answering by reading text in images. In *ICDAR*, pages 947–952. IEEE, 2019. 6

[50] Shiwen Ni, Dingwei Chen, Chengming Li, Xiping Hu, Ruifeng Xu, and Min Yang. Forgetting before learning: Utilizing parametric arithmetic for knowledge updating in large language models. *arXiv preprint arXiv:2311.08011*, 2023. 1

[51] 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 *ICML*, pages 8748–8763, 2021. 2

[52] Anastasia Razdaibiedina, Yuning Mao, Rui Hou, Madian Khabsa, Mike Lewis, and Amjad Almahairi. Progressive prompts: Continual learning for language models. *arXiv preprint arXiv:2301.12314*, 2023. 3

[53] Matthew Riemer, Ignacio Cases, Robert Ajemian, Miao Liu, Irina Rish, Yuhai Tu, and Gerald Tesauro. Learning to learn without forgetting by maximizing transfer and minimizing interference. *arXiv preprint arXiv:1810.11910*, 2018. 3

[54] Jonathan Schwarz, Wojciech Czarnecki, Jelena Luketina, Agnieszka Grabska-Barwinska, Yee Whye Teh, Razvan Pascanu, and Raia Hadsell. Progress & compress: A scalable framework for continual learning. In *ICML*, pages 4528–4537. PMLR, 2018. 3

[55] Ying Shen, Zhiyang Xu, Qifan Wang, Yu Cheng, Wenpeng Yin, and Lifu Huang. Multimodal instruction tuning with conditional mixture of lora. *arXiv preprint arXiv:2402.15896*, 2024. 2

[56] Amanpreet Singh, Vivek Natarajan, Meet Shah, Yu Jiang, Xinlei Chen, Dhruv Batra, Devi Parikh, and Marcus Rohrbach. Towards vqa models that can read. In *CVPR*, pages 8317–8326, 2019. 6

[57] Ghada Sokar, Decebal Constantin Mocanu, and Mykola Pechenizkiy. Spacenet: Make free space for continual learning. *Neurocomputing*, 439:1–11, 2021. 3

[58] Mingjie Sun, Zhuang Liu, Anna Bair, and J Zico Kolter. A simple and effective pruning approach for large language models. In *ICLR*, 2023. 3

[59] Yi-Lin Sung, Jaehong Yoon, and Mohit Bansal. Ecoflap: Efficient coarse-to-fine layer-wise pruning for vision-language models. *arXiv preprint arXiv:2310.02998*, 2023. 2

[60] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023. 2

[61] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023. 2

[62] Liyuan Wang, Xingxing Zhang, Hang Su, and Jun Zhu. A comprehensive survey of continual learning: theory, method and application. 2024. 3

[63] Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, et al. Qwen2-vl: Enhancing vision-language model's perception of the world at any resolution. *arXiv preprint arXiv:2409.12191*, 2024. 1

[64] Sheng Wang, Liheng Chen, Jiyue Jiang, Boyang Xue, Lingpeng Kong, and Chuan Wu. Lora meets dropout under a unified framework. *arXiv preprint arXiv:2403.00812*, 2024. 6

[65] Zifeng Wang, Tong Jian, Kaushik Chowdhury, Yanzhi Wang, Jennifer Dy, and Stratis Ioannidis. Learn-prune-share for lifelong learning. pages 641–650. IEEE, 2020. 3

[66] Zifeng Wang, Zheng Zhan, Yifan Gong, Geng Yuan, Wei Niu, Tong Jian, Bin Ren, Stratis Ioannidis, Yanzhi Wang, and Jennifer Dy. Sparcl: Sparse continual learning on the edge. *NeurIPS*, 35:20366–20380, 2022. 3

[67] Zifeng Wang, Zizhao Zhang, Chen-Yu Lee, Han Zhang, Ruoxi Sun, Xiaoqi Ren, Guolong Su, Vincent Perot, Jennifer Dy, and Tomas Pfister. Learning to prompt for continual learning. In *CVPR*, pages 139–149, 2022. 3

[68] Zhicheng Wang, Yufang Liu, Tao Ji, Xiaoling Wang, Yuanbin Wu, Congcong Jiang, Ye Chao, Zhencong Han, Ling Wang, Xu Shao, et al. Rehearsal-free continual language learning via efficient parameter isolation. In *ACL*, pages 10933–10946, 2023. 3

[69] Yibo Yang, Xiaojie Li, Zhongzhu Zhou, Shuaiwen Leon Song, Jianlong Wu, Liqiang Nie, and Bernard Ghanem. Corda: Context-oriented decomposition adaptation of large language models. *arXiv preprint arXiv:2406.05223*, 2024. 1, 3

[70] Mang Ye, Xuankun Rong, Wenke Huang, Bo Du, Nenghai Yu, and Dacheng Tao. A survey of safety on large vision-language models: Attacks, defenses and evaluations. *arXiv preprint arXiv:2502.14881*, 2025. 2

[71] Jaehong Yoon, Eunho Yang, Jeongtae Lee, and Sung Ju Hwang. Lifelong learning with dynamically expandable networks. *arXiv preprint arXiv:1708.01547*, 2017. 3

[72] 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*, pages 23219–23230, 2024. 1

[73] Le Yu, Bowen Yu, Haiyang Yu, Fei Huang, and Yongbin Li. Language models are super mario: Absorbing abilities from homologous models as a free lunch. In *ICML*, 2024. 2, 4, 6, 3

[74] Le Yu, Bowen Yu, Haiyang Yu, Fei Huang, and Yongbin Li. Language models are super mario: Absorbing abilities from homologous models as a free lunch. In *ICML*, 2024. 3

[75] Raphael Yuster and Uri Zwick. Fast sparse matrix multiplication. *ACM Transactions On Algorithms (TALG)*, 1(1): 2–13, 2005. 2, 4

[76] Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid loss for language image pre-training. In *ICCV*, pages 11975–11986, 2023. 2

[77] Yuexiang Zhai, Shengbang Tong, Xiao Li, Mu Cai, Qing Qu, Yong Jae Lee, and Yi Ma. Investigating the catastrophic forgetting in multimodal large language model fine-tuning. pages 202–227. PMLR, 2024. 2

[78] Mingyang Zhang, Hao Chen, Chunhua Shen, Zhen Yang, Linlin Ou, Xinyi Yu, and Bohan Zhuang. Loraprune: Pruning meets low-rank parameter-efficient fine-tuning. *arXiv preprint arXiv:2305.18403*, 2023. 2, 3

[79] Qingru Zhang, Minshuo Chen, Alexander Bukharin, Pengcheng He, Yu Cheng, Weizhu Chen, and Tuo Zhao. Adaptive budget allocation for parameter-efficient fine-tuning. *arXiv preprint arXiv:2303.10512*, 2023. 1

[80] Wenxuan Zhang, Paul Janson, Rahaf Aljundi, and Mohamed Elhoseiny. Overcoming generic knowledge loss with selective parameter update. In *CVPR*, pages 24046–24056, 2024. 3

[81] Zhi Zhang, Srishti Yadav, Fengze Han, and Ekaterina Shutova. Cross-modal information flow in multimodal large language models. *arXiv preprint arXiv:2411.18620*, 2024. 8

[82] Shanshan Zhong, Shanghua Gao, Zhongzhan Huang, Wushao Wen, Marinka Zitnik, and Pan Zhou. Moextend: Tuning new experts for modality and task extension. *arXiv preprint arXiv:2408.03511*, 2024. 3

[83] Xiongtao Zhou, Jie He, Yuhua Ke, Guangyao Zhu, Víctor Gutiérrez-Basulto, and Jeff Z Pan. An empirical study on parameter-efficient fine-tuning for multimodal large language models. In *ACL*, 2024. 6

[84] Didi Zhu, Zhongyi Sun, Zexi Li, Tao Shen, Ke Yan, Shouhong Ding, Kun Kuang, and Chao Wu. Model tailor: Mitigating catastrophic forgetting in multi-modal large language models. In *ICML*, 2024. 1, 2, 3, 6

[85] Didi Zhu, Yibing Song, Tao Shen, Ziyu Zhao, Jinluan Yang, Min Zhang, and Chao Wu. Remedy: Recipe merging dynamics in large vision-language models. In *ICLR*, 2025. 8

LoRASculpt: Sculpting LoRA for Harmonizing General and Specialized Knowledge in Multimodal Large Language Models

Supplementary Material

A. Proof of Theorem 3.1

Theorem 3.1. *Let $B \in \mathbb{R}^{p\times r}$ and $A \in \mathbb{R}^{r\times q}$ be two low rank matrices in LoRA, then the expected sparsity of the product matrix $BA \in \mathbb{R}^{p\times q}$ is given by:*

$$\mathbb{E}\left[s_{BA}\right] = 1 - (1 - s_B s_A)^r. \tag{12}$$

Proof. We aim to determine the expected proportion of non-zero elements in the product matrix $BA \in \mathbb{R}^{p\times q}$. The element in the i-th row and j-th column of BA is given by

$$(BA)_{ij} = \sum_{k=1}^{r} B_{ik}A_{kj}. \tag{13}$$

We will prove a stronger conclusion: we assume that all elements in A and B are nonnegative. This assumption increases the number of nonzero elements in BA, making it more challenging to ensure the sparsity of BA.

Then an element $(BA)_{ij}$ is non-zero if and only if there exists at least one $k \in \{1, 2, \ldots, r\}$ such that both B_{ik} and A_{kj} are non-zero. For each k, the probability that B_{ik} is non-zero is s_B, and the probability that A_{kj} is non-zero is s_A. Since the positions of non-zero elements in B and A are independently and randomly distributed, the probability that both B_{ik} and A_{kj} are non-zero is

$$\mathbb{P}\left(B_{ik} \neq 0 \text{ and } A_{kj} \neq 0\right) = s_B s_A. \tag{14}$$

Therefore, the probability that $B_{ik}A_{kj} = 0$ is

$$\mathbb{P}\left(B_{ik}A_{kj} = 0\right) = 1 - s_B s_A. \tag{15}$$

Assuming independence across different k, the probability that all terms $B_{ik}A_{kj}$ are zero is

$$\begin{aligned}\mathbb{P}\left(\bigcap_{k=1}^{r}\{B_{ik}A_{kj} = 0\}\right) &= \prod_{k=1}^{r}\mathbb{P}\left(B_{ik}A_{kj} = 0\right) \\ &= \left(1 - s_B s_A\right)^r.\end{aligned} \tag{16}$$

Thus, the probability that $(BA)_{ij}$ is non-zero is

$$\begin{aligned}\mathbb{P}\left((BA)_{ij} \neq 0\right) &= 1 - \mathbb{P}\left((BA)_{ij} = 0\right) \\ &= 1 - \left(1 - s_B s_A\right)^r.\end{aligned} \tag{17}$$

Since there are $p \times q$ elements in BA, the expected number of non-zero elements is

$$\mathbb{E}\left[N_{BA}\right] = pq\left[1 - (1 - s_B s_A)^r\right], \tag{18}$$

where N_{BA} denotes the number of non-zero elements in BA.

The expected sparsity of BA is then

$$\mathbb{E}\left[s_{BA}\right] = \frac{\mathbb{E}\left[N_{BA}\right]}{pq} = 1 - \left(1 - s_B s_A\right)^r. \tag{19}$$

The proof of Theorem 3.1 is finished. □

B. Proof of Theorem 3.2

Theorem 3.2. *Let $B \in \mathbb{R}^{p\times r}$ and $A \in \mathbb{R}^{r\times q}$ be two low rank matrices in LoRA, where the sparsity of B is s_B and the sparsity of A is s_A. Define $C = BA$, with sparsity s_C. Then, for any $\delta > 0$:*

$$\mathbb{P}\left(|s_C - \mathbb{E}[s_C]| \geq \delta\right) \leq 2\exp\left(-\frac{2\delta^2 pq}{r(p+q)}\right), \tag{20}$$

where the expected sparsity $\mathbb{E}[s_C]$ is given by Theorem 3.1

Proof. We aim to apply McDiarmid's inequality to the total number of nonzero entries N in C.

McDiarmid's Inequality states that if $X_1, X_2, \ldots, X_n$ are independent random variables taking values in a set $\mathcal{X}$, and $f : \mathcal{X}^n \to \mathbb{R}$ satisfies the bounded differences condition: for all i and all $x_1, \ldots, x_n, x_i' \in \mathcal{X}$,

$$|f(x_1, \ldots, x_i, \ldots, x_n) - f(x_1, \ldots, x_i', \ldots, x_n)| \leq c_i,$$

then for all $\epsilon > 0$,

$$\mathbb{P}\left(f(X_1, \ldots, X_n) - \mathbb{E}[f] \geq \epsilon\right) \leq \exp\left(-\frac{2\epsilon^2}{\sum_{i=1}^{n} c_i^2}\right),$$

and similarly for $\mathbb{P}\left(\mathbb{E}[f] - f(X_1, \ldots, X_n) \geq \epsilon\right)$.

In our context, consider the function f representing the total number of nonzero entries in C:

$$N = \sum_{i=1}^{p}\sum_{j=1}^{q} X_{ij}, \tag{21}$$

where X_{ij} is the indicator variable:

$$X_{ij} = \begin{cases} 1, & \text{if } C_{ij} \neq 0, \\ 0, & \text{if } C_{ij} = 0. \end{cases} \tag{22}$$

Each C_{ij} depends on the random variables $\{B_{ik}, A_{kj}\}_{k=1}^{r}$. The variables B_{ik} and A_{kj} are independent and affect N through C_{ij}.

We have the bounded differences:

Effect of changing B_{ik}: Changing B_{ik} can affect all C_{ij} where $j = 1, \ldots, q$. The maximum change in N due to changing B_{ik} is $c_{B_{ik}} = q$.

Effect of changing A_{kj}: Changing A_{kj} can affect all C_{ij} where $i = 1, \ldots, p$. The maximum change in N due to changing A_{kj} is $c_{A_{kj}} = p$.

Therefore, the sum of the squares of the bounded differences is:

$$\sum_{i,k} c_{B_{ik}}^2 + \sum_{k,j} c_{A_{kj}}^2 = pr \cdot q^2 + rq \cdot p^2 = rpq(p+q). \tag{23}$$

Applying McDiarmid's inequality, for any $\epsilon > 0$:

$$\mathbb{P}\left(N - \mathbb{E}[N] \geq \epsilon\right) \leq \exp\left(-\frac{2\epsilon^2}{rpq(p+q)}\right), \tag{24}$$

and similarly for $\mathbb{P}\left(\mathbb{E}[N] - N \geq \epsilon\right)$. Therefore,

$$\mathbb{P}\left(|N - \mathbb{E}[N]| \geq \epsilon\right) \leq 2\exp\left(-\frac{2\epsilon^2}{rpq(p+q)}\right). \tag{25}$$

Since $s_C = \dfrac{N}{pq}$, we have:

$$|s_C - \mathbb{E}[s_C]| = \frac{|N - \mathbb{E}[N]|}{pq}. \tag{26}$$

Let $\delta = \dfrac{\epsilon}{pq}$, so $\epsilon = \delta pq$. Substituting back into the inequality:

$$\begin{aligned}\mathbb{P}\left(|s_C - \mathbb{E}[s_C]| \geq \delta\right) &\leq 2\exp\left(-\frac{2(\delta pq)^2}{rpq(p+q)}\right) \\ &= 2\exp\left(-\frac{2\delta^2 pq}{r(p+q)}\right).\end{aligned} \tag{27}$$

The proof of Theorem 3.2 is finished.

□

C. Proof of Theorem D.1

Theorem D.1. *Consider matrices $A \in \mathbb{R}^{r\times q}$ and $B \in \mathbb{R}^{p\times r}$, where each row of B and each column of A exhibit uniform sparsity internally but vary across rows and columns, respectively, with average sparsities s_A and s_B. Then, the expected proportion $\mathbb{E}[s_C]$ of nonzero entries in the product matrix $C = BA$ satisfies:*

$$\mathbb{E}[s_C] \leq 1 - (1 - s_A s_B)^r. \tag{28}$$

Proof. Consider any entry c_{ij} of the matrix $C = BA$, which is computed as:

$$c_{ij} = \sum_{k=1}^{r} b_{ik} a_{kj}. \tag{29}$$

To determine the probability that c_{ij} is nonzero, we analyze the sparsity of b_{ik} and a_{kj}.

For fixed i and j, we define: s_{B_i} is the sparsity of the i-th row of B; s_{A_j} is the sparsity of the j-th column of A,.

For b_{ik} and a_{kj}, we have:

$$\mathbb{P}(b_{ik} \neq 0) = s_{B_i}, \quad \mathbb{P}(a_{kj} \neq 0) = s_{A_j}. \tag{30}$$

Since the positions of nonzero elements within the i-th row of B and the j-th column of A are independently and uniformly distributed, the events that b_{ik} and a_{kj} are nonzero are independent for each k. Therefore, the probability that both b_{ik} and a_{kj} are nonzero is:

$$\mathbb{P}(b_{ik} \neq 0 \text{ and } a_{kj} \neq 0) = s_{B_i} s_{A_j}. \tag{31}$$

Same as the proof for Theorem 3.1, we prove a stronger conclusion by assuming that all elements in A and B are nonnegative. Thus, for $c_{ij} = 0$, it must hold that for all $k = 1, 2, \ldots, r$, either $b_{ik} = 0$ or $a_{kj} = 0$. Consequently, the probability that $c_{ij} = 0$ is:

$$\begin{aligned}\mathbb{P}(c_{ij} = 0) &= \prod_{k=1}^{r} \left[1 - \mathbb{P}(b_{ik} \neq 0 \text{ and } a_{kj} \neq 0)\right] \\ &= \left(1 - s_{B_i} s_{A_j}\right)^r.\end{aligned} \tag{32}$$

Thus, the probability that c_{ij} is nonzero is:

$$\begin{aligned}\mathbb{P}c_{ij} \neq 0) &= 1 - \mathbb{P}(c_{ij} = 0) \\ &= 1 - \left(1 - s_{B_i} s_{A_j}\right)^r.\end{aligned} \tag{33}$$

Therefore, the expected proportion of nonzero entries in C is:

$$\mathbb{E}[s_C] = \frac{1}{pq} \sum_{i=1}^{p} \sum_{j=1}^{q} \left[1 - \left(1 - s_{B_i} s_{A_j}\right)^r\right]. \tag{34}$$

Note that for $x \in [0, 1]$ and $r \geq 1$, the function $f(x) = (1 - x)^r$ is convex. According to Jensen's Inequality, for a convex function f and a random variable X, we have:

$$\mathbb{E}[f(X)] \geq f(\mathbb{E}[X]). \tag{35}$$

In our case, let the random variables be $X_{ij} = s_{B_i} s_{A_j}$, then:

$$\begin{aligned}\mathbb{E}[X] &= \frac{1}{pq} \sum_{i=1}^{p} \sum_{j=1}^{q} X_{ij} \\ &= \left(\frac{1}{p} \sum_{i=1}^{p} s_{B_i}\right) \left(\frac{1}{q} \sum_{j=1}^{q} s_{A_j}\right) \\ &= s_B s_A.\end{aligned} \tag{36}$$

Applying Jensen's Inequality, we obtain:

$$\frac{1}{pq} \sum_{i=1}^{p} \sum_{j=1}^{q} \left(1 - s_{B_i} s_{A_j}\right)^r \geq \left(1 - s_B s_A\right)^r. \tag{37}$$

That is:

$$\begin{aligned}1 - \mathbb{E}[s_C] &= \frac{1}{pq} \sum_{i=1}^{p} \sum_{j=1}^{q} \left(1 - s_{B_i} s_{A_j}\right)^r \\ &\geq \left(1 - s_B s_A\right)^r.\end{aligned} \tag{38}$$

From the inequality above, we have:

$$\mathbb{E}[s_C] \leq 1 - \left(1 - s_B s_A\right)^r. \tag{39}$$

The proof of Theorem 3.3 is finished.

□

D. Proof of LoRASculpt Sparsity Guarantee

We first demonstrate that incorporating Knowledge-Guided Regularization impacts the sparsity structure of the low-rank LoRA matrices. Specifically, each row of B maintains uniform sparsity, though different rows have varied sparsity levels; similarly, each column of A has consistent sparsity within itself, while sparsity varies across columns. The overall sparsity of the two low-rank matrices remains at s_B and s_A following one-shot pruning. For the partial derivative of the (i, j)-th element of the delta weight BA, we have

the following expression:

$$\frac{\partial \mathcal{L}_{CMR}^2}{\partial (BA)_{ij}} = 2 \cdot M_{ij}^2 \cdot \sum_k B_{ik} A_{kj}, \tag{40}$$

This indicates that the penalty on the (i, j)-th position in the delta weight BA affects the i-th row of B and the j-th column of A . Consequently, B is constrained by rows, and A by columns, resulting in varying sparsity across the rows of B and the columns of A. Under this condition, the following theorem holds:

Theorem D.1. *Consider matrices $A \in \mathbb{R}^{r \times q}$ and $B \in \mathbb{R}^{p \times r}$, where each row of B and each column of A exhibit uniform sparsity internally but vary across rows and columns, respectively, with average sparsities s_A and s_B. Then, the expected proportion $\mathbb{E}[s_C]$ of nonzero entries in the product matrix $C = BA$ satisfies:*

$$\mathbb{E}[s_C] \leq 1 - (1 - s_A s_B)^r. \tag{41}$$

Proof. See Appendix C. □

Despite the non-uniform sparsity of matrices B and A across rows and columns, where different rows of B and different columns of A exhibit varied distributions, we can still assume the independence of updates across all elements. This does not hinder the application of McDiarmid's inequality, thereby allowing us to obtain the previously established error bounds in Theorem 3.2. Thus, we have established the sparsity guarantees of LoRASculpt.

E. Algorithm of LoRASculpt

The algorithm is outlined in Algorithm 1. Please refer to Sec. 3.2 for more details.

Algorithm 1: LoRASculpt

Input: Training Steps T, Warmup Steps T_{warmup}, Training data $\mathcal{D}_{\text{tr}}$, Sparsity Ratio s_A, s_B , Number of Layer in LLM and Connector L_{LLM}, L_{Con}, Option of whether training Connector with LoRA Flag_{Con}.
Output: Final LoRA weights.

$\mathcal{L}_{CMR}^{\text{LLM}} \leftarrow 0, \quad \mathcal{L}_{CMR}^{\text{Con}} \leftarrow 0$;
$S \leftarrow \psi(W) = \left| 1/\log\left(\frac{|W|}{\|W\|_2} + \epsilon\right)\right|$; ▷ Eq. (6)
$M \leftarrow \tanh(\omega \odot S)$; ▷ Eq. (7)
for $t = 1, 2, \ldots, T$ **do**
  Sample a batch $(x^{\text{vision}}, x^{\text{text}}, y)$ in $\mathcal{D}_{\text{tr}}$;
  if $t \geq T_{\text{warmup}}$ **then**
    if $t = T_{\text{warmup}}$ **then**
      $M_A \leftarrow \text{Mask}(A, s_A)$;
      $M_B \leftarrow \text{Mask}(B, s_B)$; ▷ Eq. (3)
    end
    $A \leftarrow M_A \odot A$;
    $B \leftarrow M_B \odot B$; ▷ Eq. (2)
  end
  $h^{\text{vision}} = \varphi_{Con} \circ \varphi_{Vis}(x^{\text{vision}}), h^{\text{text}} = \text{Tokenize}(x^{\text{text}})$;
  $\mathcal{L}_{Task} \leftarrow \mathcal{L}_{CE}\Big(\Phi\big[h^{\text{vision}}, h^{\text{text}}\big], y\Big)$;
  for $l = 1, 2, \ldots, L_{LLM}$ **do**
    $\mathcal{L}_{CMR}^{\text{LLM}} \leftarrow \mathcal{L}_{CMR}^{\text{LLM}} + \|M_l \odot (B_l A_l)\|_F$; ▷ Eq. (8)
  end
  if $\text{Flag}_{Con} = \text{True}$ **then**
    for $\tilde{l} = 1, 2, \ldots, L_{Con}$ **do**
      $\mathcal{L}_{CMR}^{\text{Con}} \leftarrow \mathcal{L}_{CMR}^{\text{Con}} + \|M_{\tilde{l}} \odot (B_{\tilde{l}} A_{\tilde{l}})\|_1$; ▷ Eq. (10)
    end
  end
  $\mathcal{L} = \mathcal{L}_{Task} + \alpha \cdot \mathcal{L}_{CMR}^{\text{LLM}} + \beta \cdot \mathcal{L}_{CMR}^{\text{Con}}$; ▷ Eq. (11)
  Update low-rank adapters to minimize $\mathcal{L}$;
end
return Fine-tuned LoRA in Φ (and φ_{Con})

F. Addition Evaluation Details

Details of Compared Baselines

(a) LoRA [ICLR'22] [17]: Introduces low-rank adapters to efficiently fine-tune large models.
(b) DoRA [ICML'24] [39]: Enhances the learning capacity and training stability of LoRA by decomposing weights into magnitude and direction.
(c) Orth-Reg [ECCV'24] [18]: Adds an orthogonal regularization with a hyperparameter (*i.e.*, 1e-3) to LoRA weights, encouraging fine-tuned features to be orthogonal to pretrained features to preserve model generalization. For fair comparison and due to resource constraints, the component that involves multiple LoRA modules is excluded.
(d) L2-Regularization [PNAS'17] [28]: Apply L_2 regularization with a hyperparameter (*i.e.*, 1e-3) to the LoRA weights, guiding the fine-tuned model closer to the pretrained model thus reducing forgetting.
(e) DARE [ICML'24] [73]: Parameters from the fine-tuned LoRA weights are randomly selected and re-scaled to mitigate knowledge conflict of the target task and other tasks.
(f) Model Tailor [ICML'24] [84]: Retains pretrained parameters while selectively replacing a small portion (*i.e.*, 10%) of fine-tuned parameters, guided by salience and sensitivity analysis.

Evaluation Metric.

To evaluate the performance of MLLMs in general and specialized knowledge, we compute the source performance (denotes by *Source*) and target performance (denotes by *Target*):

$$\textit{Source} = \frac{1}{|\mathcal{D}|} \sum_i^{|\mathcal{D}|} \text{Score}(\mathcal{D}_i), \quad \textit{Target} = \text{Score}(\mathcal{T}). \tag{42}$$

where $\text{Score}(\cdot)$ denotes the evaluation metric for different datasets, which is set to Accuracy and CIDEr for VQA and Captioning tasks, respectively. Here, $\mathcal{D} = \{\mathcal{D}_i\}_{i=1}^{|\mathcal{D}|}$ represents the datasets used to evaluate general knowledge, and $\mathcal{T}$ denotes the downstream task dataset. We use the average

score of *Source* and *Target*, denoted as *Avg* to measure the overall capability of the MLLM.

G. Ablation Study of β

β controls the sparsity strength for MLLM connector in Eq. (11). Since the connector plays a crucial role in modality alignment, adopting a high sparsity level could lead to performance degradation on downstream tasks (denoted by *Target* in Tab. I). Selecting an appropriate β to sparsify the connector can achieve a balance between *Source* and *Target*.

β	10^{-2}	10^{-3}	10^{-4}	10^{-5}	10^{-6}
Source	60.19	58.58	59.73	59.55	59.13
Target	80.10	79.99	84.02	85.34	85.01
Avg	70.15	69.29	71.87	**72.45**	72.07

Table I. **Ablation Study of** β, which represents the intensity of sparsity applied to the MLLM connector. When set to 10^{-5}, the optimal *Avg* is achieved.