Multimodal Instruction Tuning with Conditional Mixture of LoRA

Anonymous ACL submission

Abstract

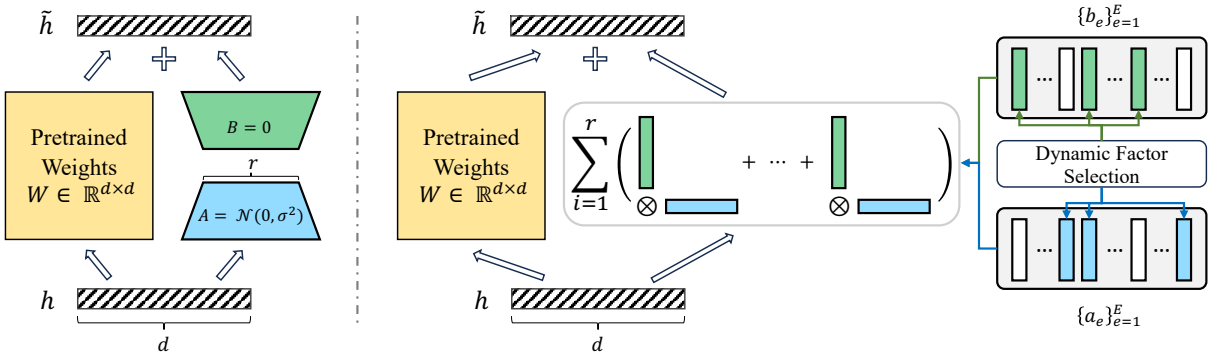
Multimodal Large Language Models (MLLMs) have demonstrated remarkable proficiency in diverse tasks across different domains, with an increasing focus on improving their zero-shot generalization capabilities for unseen multimodal tasks. Multimodal instruction tuning has emerged as a successful strategy for achieving zero-shot generalization by fine-tuning pre-trained models on diverse multimodal tasks through instructions. As MLLMs grow in complexity and size, the need for parameter-efficient fine-tuning methods like Low-Rank Adaption (LoRA), which fine-tunes with a minimal set of parameters, becomes essential. However, applying LoRA in multimodal instruction tuning presents the challenge of task interference, which leads to performance degradation, especially when dealing with a broad array of multimodal tasks. To address this, this paper introduces a novel approach that integrates multimodal instruction tuning with Conditional Mixture-of-LoRA (MixLoRA). It innovates upon LoRA by dynamically constructing low-rank adaptation matrices tailored to the unique demands of each input instance, aiming to mitigate task interference. Experimental results on various multimodal evaluation datasets indicate that MixLoRA not only outperforms the conventional LoRA with the same or even higher ranks, demonstrating its efficacy and adaptability in diverse multimodal tasks.

1 Introduction

The advent of Multimodal Large Language Models (MLLMs) (Li et al., 2023a; Liu et al., 2023; Driess et al., 2023; Dai et al., 2023) have revolutionized the field of artificial intelligence, demonstrating remarkable capabilities in processing and integrating information from various modalities, notably text and image. A key focus in advancing MLLMs is to enhance zero-shot generalization to novel multimodal tasks. In this pursuit, multimodal instruction tuning, which fine-tunes pre-trained models with diverse, instruction-based multimodal tasks, has demonstrated its efficacy in facilitating zero-shot generalization to unseen multimodal problems (Xu et al., 2023b; Liu et al., 2023; Ye et al., 2023).

Concurrently, the growing complexity and scale of MLLMs have spurred the development of various parameter-efficient fine-tuning (PEFT) techniques (Lee et al., 2019; Hu et al., 2021; Li and Liang, 2021; Karimi Mahabadi et al., 2021; Guo et al., 2021; Zaken et al., 2022). Among these, Low-Rank Adaption (LoRA) (Hu et al., 2021) has emerged as a powerful PEFT method that fine-tunes large pre-trained models by updating a small amount of injected adaption parameters. However, in multimodal instruction tuning, the effectiveness of conventional PEFT methods like LoRA diminishes due to their reliance on adjusting a limited portion of shared parameters to simultaneously accommodate diverse tasks, leading to task interference – a problem well-studied in multi-task learning (Yu et al., 2020; Liu et al., 2021; Navon et al., 2022), but insufficiently investigated in the context of parameter-efficient multimodal instruction tuning. The diverse nature of multimodal tasks significantly increases the risk of task interference. For instance, using the same limited set of adaptation parameters for distinct tasks like OCR and domain-specific classification can cause conflicting updates, potentially leading to suboptimal performance. Our research seeks to explore and address task interference in parameter-efficient multimodal instruction tuning. Specifically, we aim to answer two critical research questions: (1) Does task interference exist in parameter-efficient multimodal instruction tuning? (2) How can we effectively mitigate this issue for robust and versatile performance across various multimodal tasks?

To answer the first question, we investigate the task-interference issue in parameter-efficient multimodal instruction tuning from the perspective of gradient direction (Liu et al., 2021) in Section 3.2.



(a) LoRA (b) MixLoRA

Figure 1: **Comparative Overview of LoRA and MixLoRA.** *Left*: The conventional LoRA with static low-rank decomposition matrices BA. *Right*: MixLoRA treats the low-rank decomposition factors as experts that can be selectively combined through a Dynamic Factor Selection module, enabling the construction of varied low-rank decomposition matrices A and B tailored to varying input scenarios. The selected factors are visually distinguished by color coding: green for B and blue for A.

Our observations highlight notable task interference in this context, underscoring the necessity for more effective adaptation strategies to ensure robust and versatile performance across diverse multimodal tasks. In response to our second question, this paper proposes a novel multimodal instruction tuning framework – Conditional Mixture-of-LoRA (MixLoRA), designed to mitigate the task interference issue. As shown in Figure 1, unlike conventional LoRA which uses shared low-rank adaptation matrices A and B across all tasks and instances, MixLoRA dynamically constructs low-rank adaptation matrices A and B tailored to each input instance, by selecting their decomposition factors from two collections. MixLoRA introduces a dynamic factor selection mechanism, incorporating two Independent Factor Selection (IFS) routers and a Conditional Factor Selection (CFS) router. The two IFS routers independently select appropriate factors to dynamically construct LoRA A and B matrices tailored to each input. The CFS router further refines the selection for LoRA B based on the factors chosen for LoRA A, ensuring that the factors selections for LoRA A and B are not only tailed to input but also cohesively aligned.

To validate the effectiveness of MixLoRA, we conduct extensive experiments on MME (Fu et al., 2023), a comprehensive multimodal evaluation benchmark, and seven additional multimodal evaluation datasets that focus on various capabilities. Experimental results demonstrate that MixLoRA, with its dynamic factor selection approach, consistently outperforms LoRA across various multimodal tasks when using the same number of ranks and remains competitive or superior even against LoRA with a higher rank number. This effectiveness is attributed to the dynamic factor selection mechanism and its ability to generalize to unseen tasks through adaptive factor activation, underscoring the potential of MixLoRA to generalize and perform effectively on unseen multimodal tasks.

Our contributions are summarized as follows: (1) We empirically investigate and demonstrate the existence of task interference in parameter-efficient multimodal instruction tuning. (2) We propose the Conditional Mixture-of-LoRA (MixLoRA) framework, aimed at alleviating task interference by dynamically constructing low-rank adaptation matrices for various inputs. (3) Comprehensive experiments demonstrate the effectiveness of MixLoRA, outperforming LoRA across various unseen multimodal tasks at equal or even higher ranks.

2 Related Work

Multimodal Instruction tuning Instruction tuning (Wei et al., 2021) significantly improves the generalization of large language models to unseen tasks based on natural language instructions. With the advent of multimodal large language models, the scope of instruction tuning has expanded to encompass multimodal and vision tasks, facilitated by the development of diverse multimodal instruction datasets, including both machine-generated (Liu et al., 2023; Zhao et al., 2023; Zhu et al., 2023; Yin et al., 2023; Li et al., 2023b; Ye et al., 2023) and human-annotated (Xu et al., 2023b). Recently, Vision-Flan (Xu et al., 2023a) stands out as a comprehensive human-annotated visual instruction tuning dataset, covering a wide range of 187 tasks, making it ideal for our training.

Parameter-efficient fine-tuning (PEFT) Parameter-efficient fine-tuning (PEFT) (Lee et al., 2019; Hu et al., 2021; Li and Liang, 2021; Karimi Mahabadi et al., 2021; Guo et al., 2021; Zaken et al., 2022) strategies have become key in efficiently adapting large pre-trained models to various downstream tasks with minimal parameter adjustments. Among these, LoRA (Hu et al., 2021) demonstrates competitive trade-offs between performance and parameter efficiency, making it widely adopted. PEFT methods typically utilize shared adaptation parameters across diverse tasks or train task-specific adapters. However, when applied to multimodal instruction tuning, which requires simultaneous adaptation to diverse instruction tasks, PEFT can encounter task interference, highlighting the need for more adaptable and versatile PEFT methods to adeptly handle the complexities of multimodal instruction tuning.

Task Interference Task interference (Crawshaw, 2020) is a notable challenge in multi-task learning, where simultaneous training on multiple tasks can lead to performance decline due to conflicting gradients among tasks (Yu et al., 2020; Liu et al., 2021; Navon et al., 2022). To mitigate task interference in multi-task learning, researchers have explored various strategies, including dynamic adjustment of task loss contributions (Chen et al., 2018; Sener and Koltun, 2018; Liu et al., 2019) and parameter partitioning (Maninis et al., 2019; Bragman et al., 2019; Strezoski et al., 2019; Zhang et al., 2020). Despite the established understanding of task interference in multi-task learning, its presence and implications in instruction tuning, particularly in multimodal contexts, remain under-explored. Given the intrinsic complexity and diversity of multimodal instruction-based tasks, substantial task interference is likely to exist in multimodal instruction-tuning scenarios. Our research delves into this area, specifically investigating task interference within parameter-efficient multimodal instruction tuning.

3 Task Interference in Multimodal Instruction Tuning with LoRA

3.1 Background: Low-Rank Adaptation

Low-Rank Adaptation (LoRA) (Hu et al., 2021) is a parameter-efficient fine-tuning method that fine-tunes only the trainable rank decomposition matrices injected in each layer of the Transformer (Vaswani et al., 2017). As illustrated in Figure 1 (a), consider a linear layer, represented by $\tilde{h} = Wh$, where $W \in \mathbb{R}^{d_{out} \times d_{in}}$ denotes the pre-trained weight, with d_{in} and d_{out} being the input and output dimensions, respectively. LoRA modifies the model parameters by injecting low-rank decomposition matrices as the weight adjustment matrices, which can be expressed as:

$$\tilde{h} = Wh + \Delta Wh = Wh + \alpha \cdot BAh, \tag{1}$$

where $\Delta W = BA$ represents the trainable weight adjustment matrices formed by low-rank matrices $A \in \mathbb{R}^{r \times d_{in}}$ and $B \in \mathbb{R}^{d_{out} \times r}$, with the rank $r \ll \min(d_{in}, d_{out})$. The scalar $\alpha \geq 1$ controls the influence of the weight adjustment matrices. During fine-tuning, only these low-rank decomposition matrices, referred to as LoRA A and LoRA B throughout this paper, are updated, allowing for rapid, task-specific adaptation by training distinct LoRA A and B for each downstream task.

3.2 Investigating Task Interference in Multimodal Instruction Tuning

Our study delves into task interference in parameter-efficient multimodal instruction tuning by analyzing gradient direction conflicts between task pairs. For each task pair i and j, we first estimate the change in loss L_i of task i, when optimizing the shared parameters θ according to the loss L_j of task j, following (Zhu et al., 2022):

$$\begin{aligned}\Delta_j L_i(x_i) &= \mathbb{E}_{x_j}\left(L_i(x_i;\theta) - L_i(x_i;\theta - \lambda \frac{\nabla_\theta L_j(x_j)}{\|\nabla_\theta L_j(x_j)\|})\right) \\ &\approx \lambda \mathbb{E}_{x_j}\left(\frac{\nabla_\theta L_j(x_j)}{\|\nabla_\theta L_j(x_j)\|}^T \nabla_\theta L_i(x_i)\right)\end{aligned} \tag{2}$$

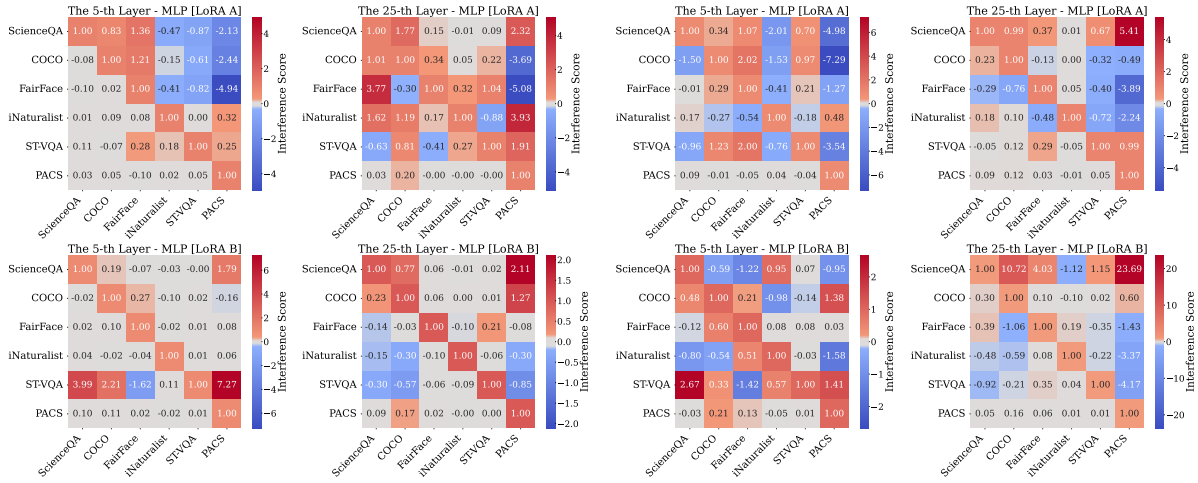
where x_i and x_j are sampled training batches for tasks i and j, and λ is the learning rate.

The interference of task j on task i is then quantified as follows:

$$\mathcal{I}_{i,j} = \mathbb{E}_{x_i}\left(\frac{\Delta_j L_i(x_i)}{\Delta_i L_i(x_i)}\right) \tag{3}$$

Here, a positive $\mathcal{I}_{i,j}$ suggests aligned gradient directions between tasks i and j, while a negative value implies divergent gradient directions, indicating that task j adversely impacts task i.

We conduct experiments on the fine-tuned LLaVa (Liu et al., 2023) model using LoRA with a rank of 4, computing the task interference among six diverse tasks from Vision-Flan (Xu et al., 2023a), including "ScienceQA" (Lu et al., 2022) (for "Complex Reasoning"),"COCO" (Lin et al.,



(a) MLP (b) Self-Attention

Figure 2: **The Task Interference Score $\mathcal{I}$ for LoRA decomposition matrices A and B.** Each cell in the heatmap corresponds to the average interference score $\mathcal{I}_{i,j}$ of task j (column) on the task i (row). A blue hue indicates a negative impact of task j on task i, whereas a red hue signifies a positive impact.

2014) (for "Coarse-grained Perception"), "FairFace" (Karkkainen and Joo, 2021) (for "Fine-grained Perception"), "iNaturalist" (Van Horn et al., 2018) (for "Knowledge Intensive"), "ST-VQA" (Biten et al., 2019) (for "OCR"), and "PACS" (Li et al., 2017) (for "Domain specific"). We compute the average task interference matrix $\mathcal{I}$ based on the gradients concerning LoRA A and B, across various layers. Figure 2 shows the task interference score for LoRA A and B at the 5-th and 25-th Transformer Layer for both MLP (Figure 2a) and Self-Attention (Figure 2b).

Our results reveal notable task interference at both shallow and deep Transformer layers for LoRA A and B. For instance, as shown in Figure 2b, at the 5-th layer for LoRA A, the domain-specific classification task "PACS" negatively impacts "COCO", a coarse-grained perception task, with a negative interference score of -7.3. Meanwhile, positive influences are also observed. For example, Figure 2a shows that at the 5-th layer for LoRA B, "PACS" positively affects the OCR task "ST-VQA". The presence of both positive and negative interference suggests complex dynamics among instruction tasks: positive scores (in red), suggest that the learning of one task can enhance the performance of another, while negative scores (in blue), imply that one task's learning can hinder another. These findings highlight notable task interference in parameter-efficient multimodal instruction tuning and reinforce the need for effective adaption methods to ensure robust and versatile performance across diverse multimodal tasks.

4 Conditional Mixture-of-LoRA

Inspired by the concept of Mixture-of-Experts (Shazeer et al., 2016), we propose Conditional Mixture-of-LoRA (MixLoRA) which leverages low-rank decomposition factors as dynamically chosen experts to construct tailored decomposition matrices A and B for specific input instances. MixLoRA facilitates dynamic processing pathways for varying input instances, thereby enhancing the efficacy in handling diverse and complex multimodal instruction tasks.

The core of Conditional Mixture-of-LoRA lies in representing the weight adjustment matrices ΔW from Equation 1 via tensor decomposition:

$$\Delta W = BA = \sum_{i=1}^{r} b_i \otimes a_i, \tag{4}$$

where $\{a_i, b_i\}_{i=1}^{r}, a_i \in \mathbb{R}^{d_{in} \times 1}, b_i \in \mathbb{R}^{d_{out} \times 1}$ are the rank r decomposition factors of ΔW.

Leveraging the concept that ΔW can be expressed as sum of outer products of low-rank decomposition factors a_i and b_i, MixLoRA introduces a **Dynamic Factor Selection** module. This module dynamically constructs unique ΔW for specific inputs by selecting r appropriate factors from an expanded pool of decomposition factors $\{a_e\}_{e=1}^{E}, \{b_e\}_{e=1}^{E}, E > r$, as shown in Fig. 1 (b).

4.1 Dynamic Factor Selection

The Dynamic Factor Selection module uses two main components to dynamically constructs LoRA A and B. First, two **Independent Factor Selection (IFS)** routers (Section 4.1.1), independently

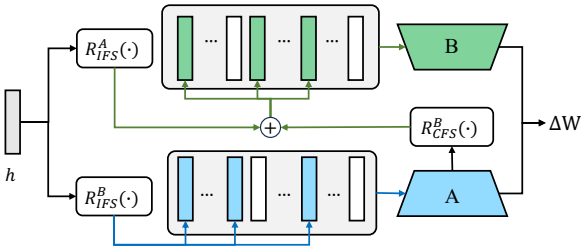


Figure 3: **Dynamic Factor Selection in MixLoRA.** MixLoRA treats low-rank decomposition factors as experts and dynamically constructs the LoRA A and B through two independent routers $R^A_{\text{IFS}}(\cdot)$ and $R^B_{\text{IFS}}(\cdot)$, complemented by a conditional router $R^B_{\text{CFS}}(\cdot)$.

select r relevant factors to form adaptation matrices LoRA A and B, ensuring precise, instance-specific adaptations. Second, a **Conditional Factor Selection (CFS)** router (Section 4.1.2) further refines the selection for LoRA B by conditioning the selection for B also on the factors chosen for LoRA A, promoting a coherent adaptation process.

4.1.1 Independent Factor Selection

MixLoRA employs two Independent Factor Selection (IFS) routers, $R^A_{\text{IFS}}(\cdot)$ and $R^B_{\text{IFS}}(\cdot)$, to select r relevant factors for LoRA A and B, respectively, as shown in Figure 3. IFS routers employ an instance-based routing method, which is more memory-efficient than conventional input-token-based routing, for selecting r decomposition factors. The routing strategy can be expressed as:

$$R^A_{\text{IFS}}(h) = \text{Avg}(h), \tag{5}$$

where $\text{Avg}(\cdot)$ averages across the sequence dimension of the hidden states $h \in \mathbb{R}^{seq \times d_{in}}$ from the preceding layer.

Factor Selection Process The factor selection process involves calculating vectors $g_A \in \mathbb{R}^E$ and $g_B \in \mathbb{R}^E$ to selectively identify specific subsets of decomposition factors from the set $\{a_e\}_{e=1}^E$ and $\{b_e\}_{e=1}^E$, respectively. To compute g_A, input $R^A_{\text{IFS}}(h) \in \mathbb{R}^{d_{in}}$ is processed through a dense layer with weights $W_A \in \mathbb{R}^{E \times d_{in}}$, followed by a softmax normalization and top-k selection:

$$g_A = \text{top}_r(\text{softmax}(W_A \cdot R^A_{\text{IFS}}(h))). \tag{6}$$

This procedure ensures the selection of r factors for LoRA A, with $g_A[i] = 1$ indicating the selection of factor i. The same process is applied to determine g_B for LoRA B.

4.1.2 Conditional Factor Selection

While the factors for LoRA A and B have been independently selected so far, we hypothesize that an interdependence exists between the selections for LoRA A and B, which can be harnessed to improve the model's overall adaptability and performance. To leverage this relationship, we propose a Conditional Factor Selection (CFS) strategy, wherein the selection of factors for the projection-up weight of LoRA B is also influenced by the factors chosen for the projection-down weight of LoRA A.

With the IFS router, LoRA A is assembled from chosen decomposition factors, denoted as $A = [a_1, \cdots, a_r]^T$, where $A \in \mathbb{R}^{r \times d_{in}}$. Following this, the CFS router employs a weight tensor $W_{AB} \in \mathbb{R}^{r \times d_{in} \times E}$ to map each factor $A[i] \in \mathbb{R}^{1 \times d_{in}}$ in A to an expert dimension E. The mapping process for each factor $A[i]$, normalized via softmax and aggregated across r factors, is given by:

$$R^B_{\text{CFS}}(A) = \sum_{i=1}^{r} \text{softmax}(A[i] \cdot W_{AB}[i]), \tag{7}$$

where $W_{AB}[i] \in \mathbb{R}^{d_{\text{in}} \times E}$ is the mapping matrix associated with $A[i]$.

The factors selection for LoRA B integrates outputs from both the IFS $R^B_{\text{IFS}}(\cdot)$ and CFS $R^B_{\text{CFS}}(\cdot)$ routers via a late fusion strategy, forming the selection vector g_B as follows:

$$\begin{aligned} p^B_{\text{IFS}} &= \text{softmax}(W^B_{\text{IFS}} \cdot R^B_{\text{IFS}}(h)) \\ p^B_{\text{CFS}} &= \text{softmax}(R^B_{\text{CFS}}(A)) \\ g_B &= \text{top}_r(p^B_{\text{IFS}} + p^B_{\text{CFS}}). \end{aligned} \tag{8}$$

The final selection vector g_B is determined by combining the probability distributions p^B_{IFS} and p^B_{CFS} from the IFS and CFS routers. This CFS strategy enables the selection for LoRA B to be informed by factors selected for LoRA A, fostering a more cohesive selection process.

4.1.3 Reconstruction of Dynamic Adaptation Matrices

Finally, MixLoRA constructs dynamic adaptation matrices by leveraging the factor selection vectors g_A and g_B, gathering the chosen factors $a_k, b_k k \in K, |K| = r$, to assemble the final matrices for LoRA A and B. Consequently, in each forward pass, the weight adjustment matrix $\Delta W \in \mathbb{R}^{d_{out} \times d_{in}}$ is dynamically calculated based on these selected factors, formulated as:

$$\Delta W = BA = [b_1, \cdots, b_r][a_1, \cdots, a_r]^T \tag{9}$$

Model	Factors	Rank	MME	Text-VQA	VSR	SNLI-VE	CIFAR-10	CIFAR-100	MNIST	Pope	MMAvg
$\text{LLaVA}_{\text{Align}}$	-	-	1110.82	32.62	50.16	34.51	80.00	58.04	52.79	59.10	52.46
LLaVA_{FT}	-	-	1587.26	37.26	53.76	43.35	92.97	63.73	94.27	80.82	66.59
LoRA	-	2	1291.20	**39.86**	51.88	31.80	85.51	**49.23**	79.22	76.72	59.17
LoRA	-	4	1345.86	39.44	53.19	33.08	86.62	47.36	80.89	**76.89**	59.64
LoRA	-	8	1312.87	39.20	53.27	36.36	88.92	46.88	82.95	75.48	60.44
LoRA	-	16	1381.23	39.22	**53.60**	36.11	87.31	45.60	**85.92**	75.16	60.42
LoRA	-	32	**1393.67**	39.20	52.95	**44.56**	**90.10**	45.90	83.42	72.33	**61.21**
MixLoRA	16	2	1417.83	39.82	52.13	**35.38**	90.14	**58.05**	**85.98**	73.86	62.19
MixLoRA	32	2	**1459.15**	**40.46**	**52.62**	35.04	**91.02**	57.95	85.26	**78.31**	**62.95**
MixLoRA	16	4	1443.82	**40.66**	**52.70**	**43.10**	**91.59**	57.28	85.25	78.13	**64.10**
MixLoRA	32	4	**1509.61**	40.42	49.18	36.69	91.40	**59.27**	**87.68**	**78.48**	63.30
MixLoRA	16	8	1485.26	39.92	**52.70**	**40.74**	**92.85**	53.96	82.95	75.31	62.63
MixLoRA	32	8	**1485.48**	**40.02**	51.15	37.77	91.12	**60.25**	**86.64**	**78.87**	**63.69**

Table 1: **Zero-shot Multi-modal Evaluation.** $\text{LLaVA}_{\text{Align}}$ indicates the stage-one LLaVA-v1 with only feature alignment but not visual instruction tuning, and LLaVA_{FT} is the fully fine-tuned LLaVA using the same Vision-Flan dataset. The **MMAvg** column denotes the average performance across seven multimodal datasets, except for MME. The best performance is in **bold**.

5 Experimental Methodology

5.1 Datasets

Training Datasets We perform instruction tuning on **Vision-Flan** (Xu et al., 2023a), a human-annotated multimodal instruction tuning dataset with 187 diverse tasks. Its diversity in visual instruction tasks makes it ideal for investigating task interference. To minimize computational cost, we utilize a scaled-down version with up to 1,000 instances per task, totaling 191,105 instances.

Evaluation Datasets We evaluate our method on **MME** (Fu et al., 2023), a comprehensive multimodal evaluation benchmark measuring both perception and cognition abilities across 14 subtasks. Alongside MME, we further probe the model's various capabilities using 7 multimodal datasets. For Optical Character Recognition, we utilize **Text-VQA** (Singh et al., 2019), and for reasoning, we employ the Visual Spatial Reasoning (**VSR**) dataset (Liu et al., 2022). Perception capability is tested on **CIFAR-10/100** (Krizhevsky et al., 2009) and **MNIST** (LeCun, 1998), following the guidance of (Zhai et al., 2023). The **SNLI-VE** dataset (Xie et al., 2019) evaluates the Visual Entailment capabilities, and **POPE** (Li et al., 2023c) examines the tendency to objects hallucination.

5.2 Evaluation Metrics

For MME scores, we employ the official evaluation tool[1], aggregating the Perception and Cognition metrics. For other multimodal datasets, we leverage Vicuna 1.5 13B (Chiang et al., 2023), the state-of-the-art open-source LLM to assess the accuracy of each prediction compared with ground-truth target output. More details are in Appendix C.

[1]https://github.com/BradyFU/Awesome-Multimodal-Large-Language-Models/tree/Evaluation

6 Results and Discussion

Comparison with LoRA We first present a detailed comparison between MixLoRA and the conventional LoRA, focusing on their performance in MME and 7 other multimodal tasks, as detailed in Table 1. We observe that MixLoRA consistently surpasses LoRA when both models operate at the same ranks on both MME and the additional multimodal tasks, and even demonstrate superior performance when compared to LoRA with a higher rank number. For instance, MixLoRA (with rank r=2 and factors E=16) outperforms LoRA (rank r=32) by 1.7% in MME and 1.6% on average across other multimodal evaluations.

Increase the Number of Rank We investigate the impact of increasing the rank number while keeping the number of factors constant. As shown in Table 1, MixLoRA exhibited a notable performance enhancement as the rank number increased from 2 to 4, when the factor number was fixed. Specifically, increasing the rank r from 2 to 4 leads to a performance uplift of 1.8% in MME and 3.1% in MMAvg with $E = 16$ factors, and a 3.5% improvement in MME and a 0.6% increase in MMAvg with $E = 32$ factors. However, further increasing the rank to 8 shows diminishing returns in performance gains. We hypothesize this decline might potentially be due to the expanded combination pool for constructing the adaptation matrices.

Model	Routing	Factors	Rank	MME	MMAvg
MixLoRA	Random	32	4	1007.40	49.12
MixLoRA	Instance	32	4	**1509.61**	**63.30**
MixLoRA	Task	32	4	1381.87	61.75

Table 2: **Comparison between Various Routing Strategies.** The **MMAvg** column denotes the average performance across seven multimodal datasets.

Increasing the Number of Factors In scenarios where the rank number is held constant, our findings reveal a general trend of performance improvement for MixLoRA, as shown in Table 1. This improvement can be attributed to the model's increased capacity for providing a richer set of factors to tailor the model to specific multimodal tasks.

The Effect of Routing Strageties In this experiment, we examine different routing strategies for the IFS router. In particular, we implement the Task-Specific Routing paradigm which leverages the definition of each multimodal instruction task to inform the selection of decomposition factors (details can be found in Appendix A). Table 2 shows that Instance-based Routing significantly outperforms Task-specific routing, achieving a higher MME score and average performance across the additional multimodal tasks. The superior performance of Instance-based Routing likely stems from its inherent flexibility. Unlike Task-specific Routing, which has the same selection of factors at different layers for inputs from the same task, Instance-based Routing adapts its selection based on the varying hidden states from previous layers, leading to a more flexible routing mechanism.

Furthermore, we investigate whether the superior performance is due to the introduction of extra expert parameters and not the routing mechanism. Table 2 reports the comparison with a random routing baseline, which randomly selects r factors. Our observations reveal that both Instance-based Routing and Task-specific routing surpass the random baseline, suggesting that the routing mechanism, rather than the inclusion of additional expert parameters, is responsible for the performance enhancements.

Impact of Conditional Factor Selection We assess the impact of Conditional Factor Selection (CFS) through an ablation analysis, comparing MixLoRA's averaged performance with and without the CFS across seven multimodal datasets. The comparative results, as shown in Figure 4 demonstrate that incorporating the CFS router in general consistently improves the performance across different factor and rank settings. This enhancement is hypothesized to stem from the CFS's role in strengthening the interdependency between the factor selections of LoRA A and B.

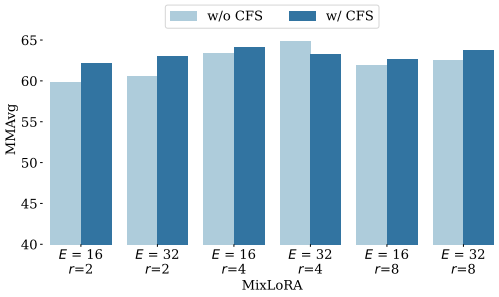


Figure 4: **Effect of Conditional Factor Selection**

Factor Selection Pattern on Unseen Tasks Our analysis delves into the factor selection patterns of LoRA A for unseen multimodal tasks. We randomly sample 300 instances from each of seven unseen multimodal tasks and visualize the factor selection within the MLP layer using t-SNE (Van der Maaten and Hinton, 2008), as shown in Figure 5. We observe that instances from identical tasks tend to cluster, indicating the effectiveness of an instance-based routing strategy in assigning diverse factor sets across tasks.

Furthermore, we visualize the factor selection patterns for similar seen and unseen tasks. We pair five distinct unseen tasks, each probing a different capability, with five similar seen tasks from the training set: SNLI-VE (unseen) with Image-Text (seen) for assessing visual entailment, Text-VQA (unseen) with InfoGraphicVQA (seen) for OCR capabilities, VSR (unseen) with GQA (seen) for reasoning, Pope (unseen) with VQA-Object-Presence (seen) for hallucination detection, and CIFAR-10 (unseen) with ExDark (seen) for perception capabilities. The t-SNE visualization shown in Figure 6 depicts the distribution of factor selection across MLP layers, with the first row in the legend indicating the seen tasks, and the second row denoting the corresponding unseen tasks. Similar color schemes are used for each pair of similar seen and unseen tasks for clarity. Our observations reveal that MixLoRA effectively activates factors analogous to those employed in similar training tasks. This finding suggests that the model can adapt its factor selection strategies to new, unseen tasks based on its training on similar seen tasks.

Analysis of Task Interference To assess MixLoRA's efficacy in mitigating task interference,

Model	Factors	Rank	ScienceQA	COCO	FairFace	iNaturalist	ST-VQA	PACS	AVG
$\text{LoRA}_{\text{Specialist}}$	-	4	64.33	77.67	54.67	58.67	44.67	99.00	66.50
$\text{LoRA}_{\text{Specialist}}$	-	16	67.33	76.33	59.00	60.00	46.33	99.00	68.00
LoRA	-	4	57.67	76.33	59.67	57.00	42.33	99.33	65.39
LoRA	-	16	59.67	73.00	59.33	58.33	43.67	99.00	65.50
MixLoRA	16	4	60.67	78.67	59.00	61.00	44.33	99.33	67.17

Table 3: **Multi-modal Evaluation on Seen Tasks.** $\text{LoRA}_{\text{Specialist}}$ represents the specialist LoRA model fine-tuned for each seen task individually. The **AVG** column denotes the average performance across six seen tasks.

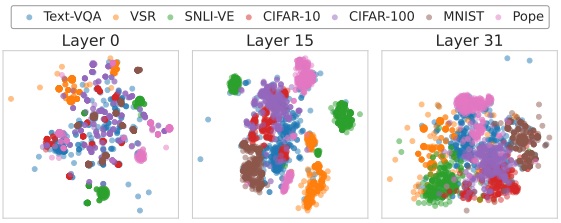


Figure 5: **T-SNE Visualization of Factor Selection Distribution for MixLoRA (E = 32, r = 8)**. Instances are represented as points, where instances from the same task share a common color.

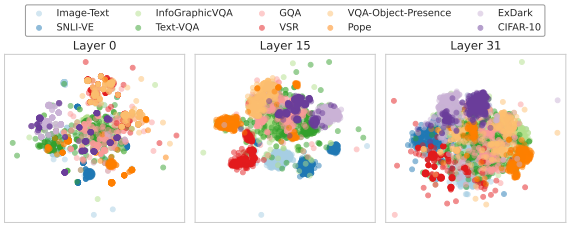


Figure 6: **T-SNE Visualization of Factor Selection in MixLoRA (E = 32, r = 8) for Seen and Unseen Tasks**. Seen tasks (Image-Text, InfoGraphicVQA, GQA, VQA-Object-Presence, CIFAR-10) in the first row are color-matched with their unseen counterparts (SNLI-VE, Text-VQA, VSR, Pope, ExDark) in the second row.

we test it on the same six training tasks: "ScienceQA", "COCO", "FairFace", "iNaturalist", "ST-VQA", and "PACS", discussed in Section 3.2. For each task, we randomly sample 300 instances not included in the instruction-tuning phase for evaluation. We compare MixLoRA against both the conventional LoRA and task-specialized LoRA models ($\text{LoRA}_{\text{Specialist}}$) that are fine-tuned with task-specific adaptation parameters for each task. Table 3 shows that conventional LoRA models exhibit varying degrees of performance degradation across tasks when compared to $\text{LoRA}_{\text{Specialist}}$. In contrast, MixLoRA suffers less from performance degradation and demonstrates more consistent and robust performance across different tasks, suggesting its effectiveness in reducing task interference.

Moreover, we visualize the task interference scores using Equation 2 and 3. Given that MixLoRA dynamically selects a subset of factors

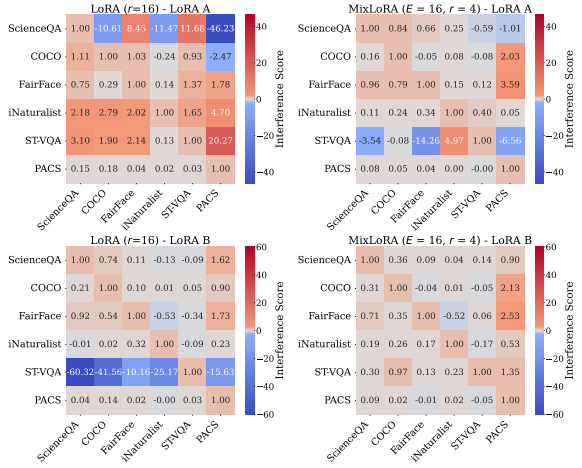


Figure 7: **The Comparison of Task Interference Score $\mathcal{I}$ between LoRA (r=16) and MixLoRA (E = 16, r = 4).** Each cell in the heatmap corresponds to the average interference score $\mathcal{I}_{i,j}$ of task j (column) on the task i (row) averaged across all adaption layers.

(r out of E) for different instances, we record gradients concerning all E factors and compare the task interference scores between standard LoRA models (with r=16) and MixLoRA (with E = 16 and r = 4). Figure 7 visualizes the interference scores for both LoRA A and LoRA B aggregated across all adaptation layers, including MLP and self-attention layers. The analysis reveals that MixLoRA (E=16, r=4) exhibits lower negative interference scores compared to the standard LoRA (r=16), underscoring MixLoRA's efficacy in reducing task interference.

7 Conclusion

We introduce Conditional Mixture-of-LoRA, an innovative strategy that dynamically constructs low-rank adaptation matrices specific to individual inputs, to mitigate task interference during parameter-efficient multimodal instruction tuning. Comprehensive experiments across a variety of multimodal datasets have demonstrated the efficacy of MixLoRA, showcasing an enhanced performance on unseen multimodal tasks compared to conventional LoRA and demonstrating its effectiveness in mitigating task interference.

8 Limitations

Our study focuses on task interference within parameter-efficient multimodal instruction tuning, specifically for image and text modalities, leaving the integration of other modalities like sound and 3D point clouds as an avenue for future work. Moreover, due to the cost of training large models, our experimentation was conducted on a scaled-down version of Vision-Flan. Future studies could benefit from evaluating the effectiveness of MixLoRA when applied to more extensive multimodal instruction-tuning datasets. Additionally, our method introduces extra training overhead compared to standard LoRA of the same rank.

References

Ali Furkan Biten, Ruben Tito, Andres Mafla, Lluis Gomez, Marçal Rusinol, Ernest Valveny, CV Jawahar, and Dimosthenis Karatzas. 2019. Scene text visual question answering. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 4291–4301.

Felix JS Bragman, Ryutaro Tanno, Sebastien Ourselin, Daniel C Alexander, and Jorge Cardoso. 2019. Stochastic filter groups for multi-task cnns: Learning specialist and generalist convolution kernels. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1385–1394.

Zhao Chen, Vijay Badrinarayanan, Chen-Yu Lee, and Andrew Rabinovich. 2018. Gradnorm: Gradient normalization for adaptive loss balancing in deep multitask networks. In *International conference on machine learning*, pages 794–803. PMLR.

Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E. Gonzalez, Ion Stoica, and Eric P. Xing. 2023. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality.

Michael Crawshaw. 2020. Multi-task learning with deep neural networks: A survey. *arXiv preprint arXiv:2009.09796*.

Wenliang Dai, Junnan Li, Dongxu Li, Anthony Meng Huat Tiong, Junqi Zhao, Weisheng Wang, Boyang Li, Pascale Fung, and Steven Hoi. 2023. Instructblip: Towards general-purpose vision-language models with instruction tuning.

Danny Driess, Fei Xia, Mehdi S. M. Sajjadi, Corey Lynch, Aakanksha Chowdhery, Brian Ichter, Ayzaan Wahid, Jonathan Tompson, Quan Vuong, Tianhe Yu, Wenlong Huang, Yevgen Chebotar, Pierre Sermanet, Daniel Duckworth, Sergey Levine, Vincent Vanhoucke, Karol Hausman, Marc Toussaint, Klaus Greff, Andy Zeng, Igor Mordatch, and Pete Florence. 2023. PaLM-e: An embodied multimodal language model. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 8469–8488. PMLR.

Chaoyou Fu, Peixian Chen, Yunhang Shen, Yulei Qin, Mengdan Zhang, Xu Lin, Zhenyu Qiu, Wei Lin, Jinrui Yang, Xiawu Zheng, et al. 2023. Mme: A comprehensive evaluation benchmark for multimodal large language models. *arXiv preprint arXiv:2306.13394*.

Demi Guo, Alexander M Rush, and Yoon Kim. 2021. Parameter-efficient transfer learning with diff pruning. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4884–4896.

Edward J Hu, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. 2021. Lora: Low-rank adaptation of large language models. In *International Conference on Learning Representations*.

Rabeeh Karimi Mahabadi, James Henderson, and Sebastian Ruder. 2021. Compacter: Efficient low-rank hypercomplex adapter layers. *Advances in Neural Information Processing Systems*, 34:1022–1035.

Kimmo Karkkainen and Jungseock Joo. 2021. Fairface: Face attribute dataset for balanced race, gender, and age for bias measurement and mitigation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1548–1558.

Alex Krizhevsky, Geoffrey Hinton, et al. 2009. Learning multiple layers of features from tiny images.

Yann LeCun. 1998. The mnist database of handwritten digits. *http://yann. lecun. com/exdb/mnist/*.

Jaejun Lee, Raphael Tang, and Jimmy Lin. 2019. What would elsa do? freezing layers during transformer fine-tuning. *arXiv preprint arXiv:1911.03090*.

Da Li, Yongxin Yang, Yi-Zhe Song, and Timothy M Hospedales. 2017. Deeper, broader and artier domain generalization. In *Proceedings of the IEEE international conference on computer vision*, pages 5542–5550.

Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. 2023a. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. *arXiv preprint arXiv:2301.12597*.

Xiang Lisa Li and Percy Liang. 2021. Prefix-tuning: Optimizing continuous prompts for generation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4582–4597.

Yanda Li, Chi Zhang, Gang Yu, Zhibin Wang, Bin Fu, Guosheng Lin, Chunhua Shen, Ling Chen, and Yunchao Wei. 2023b. Stablellava: Enhanced visual instruction tuning with synthesized image-dialogue data. *arXiv preprint arXiv:2308.10253*.

Yifan Li, Yifan Du, Kun Zhou, Jinpeng Wang, Wayne Xin Zhao, and Ji-Rong Wen. 2023c. Evaluating object hallucination in large vision-language models. *arXiv preprint arXiv:2305.10355*.

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. 2014. Microsoft coco: Common objects in context. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, pages 740–755. Springer.

Bo Liu, Xingchao Liu, Xiaojie Jin, Peter Stone, and Qiang Liu. 2021. Conflict-averse gradient descent for multi-task learning. *Advances in Neural Information Processing Systems*, 34:18878–18890.

Fangyu Liu, Guy Emerson, and Nigel Collier. 2022. Visual spatial reasoning. *CoRR*, abs/2205.00363.

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

Shikun Liu, Edward Johns, and Andrew J Davison. 2019. End-to-end multi-task learning with attention. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1871–1880.

Pan Lu, Swaroop Mishra, Tanglin Xia, Liang Qiu, Kai-Wei Chang, Song-Chun Zhu, Oyvind Tafjord, Peter Clark, and Ashwin Kalyan. 2022. Learn to explain: Multimodal reasoning via thought chains for science question answering. *Advances in Neural Information Processing Systems*, 35:2507–2521.

Kevis-Kokitsi Maninis, Ilija Radosavovic, and Iasonas Kokkinos. 2019. Attentive single-tasking of multiple tasks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1851–1860.

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

Aviv Navon, Aviv Shamsian, Idan Achituve, Haggai Maron, Kenji Kawaguchi, Gal Chechik, and Ethan Fetaya. 2022. Multi-task learning as a bargaining game. *arXiv preprint arXiv:2202.01017*.

Ozan Sener and Vladlen Koltun. 2018. Multi-task learning as multi-objective optimization. *Advances in neural information processing systems*, 31.

Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. 2016. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. In *International Conference on Learning Representations*.

Amanpreet Singh, Vivek Natarajan, Meet Shah, Yu Jiang, Xinlei Chen, Dhruv Batra, Devi Parikh, and Marcus Rohrbach. 2019. Towards VQA models that can read. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pages 8317–8326. Computer Vision Foundation / IEEE.

Gjorgji Strezoski, Nanne van Noord, and Marcel Worring. 2019. Many task learning with task routing. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1375–1384.

Laurens Van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-sne. *Journal of machine learning research*, 9(11).

Grant Van Horn, Oisin Mac Aodha, Yang Song, Yin Cui, Chen Sun, Alex Shepard, Hartwig Adam, Pietro Perona, and Serge Belongie. 2018. The inaturalist species classification and detection dataset. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8769–8778.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Jason Wei, Maarten Bosma, Vincent Y. Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M. Dai, and Quoc V. Le. 2021. Finetuned language models are zero-shot learners. *CoRR*, abs/2109.01652.

Ning Xie, Farley Lai, Derek Doran, and Asim Kadav. 2019. Visual entailment: A novel task for fine-grained image understanding. *arXiv preprint arXiv:1901.06706*.

Zhiyang Xu, Trevor Ashby, Chao Feng, Rulin Shao, Ying Shen, Di Jin, Qifan Wang, and Lifu Huang. 2023a. Vision-flan: Scaling visual instruction tuning.

Zhiyang Xu, Ying Shen, and Lifu Huang. 2023b. Multi-Instruct: Improving multi-modal zero-shot learning via instruction tuning. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 11445–11465, Toronto, Canada. Association for Computational Linguistics.

Qinghao Ye, Haiyang Xu, Guohai Xu, Jiabo Ye, Ming Yan, Yiyang Zhou, Junyang Wang, Anwen Hu, Pengcheng Shi, Yaya Shi, et al. 2023. mplug-owl: Modularization empowers large language models with multimodality. *arXiv preprint arXiv:2304.14178*.

Zhenfei Yin, Jiong Wang, Jianjian Cao, Zhelun Shi, Dingning Liu, Mukai Li, Lu Sheng, Lei Bai, Xiaoshui Huang, Zhiyong Wang, et al. 2023. Lamm: Language-assisted multi-modal instruction-tuning dataset, framework, and benchmark. *arXiv preprint arXiv:2306.06687*.

Tianhe Yu, Saurabh Kumar, Abhishek Gupta, Sergey Levine, Karol Hausman, and Chelsea Finn. 2020. Gradient surgery for multi-task learning. *Advances in Neural Information Processing Systems*, 33:5824–5836.

Elad Ben Zaken, Yoav Goldberg, and Shauli Ravfogel. 2022. Bitfit: Simple parameter-efficient fine-tuning for transformer-based masked language-models. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 1–9.

Yuexiang Zhai, Shengbang Tong, Xiao Li, Mu Cai, Qing Qu, Yong Jae Lee, and Yi Ma. 2023. Investigating the catastrophic forgetting in multimodal large language models. *arXiv preprint arXiv:2309.10313*.

Biao Zhang, Ankur Bapna, Rico Sennrich, and Orhan Firat. 2020. Share or not? learning to schedule language-specific capacity for multilingual translation. In *International Conference on Learning Representations*.

Bo Zhao, Boya Wu, and Tiejun Huang. 2023. Svit: Scaling up visual instruction tuning. *arXiv preprint arXiv:2307.04087*.

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

Jinguo Zhu, Xizhou Zhu, Wenhai Wang, Xiaohua Wang, Hongsheng Li, Xiaogang Wang, and Jifeng Dai. 2022. Uni-perceiver-moe: Learning sparse generalist models with conditional moes. *Advances in Neural Information Processing Systems*, 35:2664–2678.

A Task-Specific Routing

The Task-Specific Routing paradigm leverages the distinct characteristics of each multimodal instruction task to inform the selection of decomposition factors. This strategy utilizes the detailed task definition, which includes a comprehensive description of the task's requirements and the specific skills or modalities needed to successfully perform the task. For instance, consider the task "OK-VQA" (Marino et al., 2019), the task definition is: "*Answer the question in natural language based on the content of the image. The questions require external knowledge to answer.*" The task-specific routing strategy is formulated as:

$$R^{A}_{\text{IFS}}(z) = \text{Avg}(f_{\phi}(z)), \tag{10}$$

where $f_{\phi}(\cdot)$ denotes a pre-trained Large Language Model (LLM) parameterized by ϕ, responsible for encoding the task definition z.

B Implementation Details

We leverage the stage-one LLaVA-v1 [2] (before the visual instruction tuning stage) as our pre-trained large multimodal models, specifically employing LLaVA with Vicunna-7B v1.3. For all model variants, we fine-tune this stage-one LLaVa on the scale-down version of Vision-Flan for three epochs, using a total batch size of 128 and a learning rate of $4e-5$. The fine-tuning process for MixLoRA (E=16, r=4) takes approximately 20 hours on 4 A100 GPUs, with an effective batch size of 8 per GPU and a gradient accumulation step of 4. For LoRA, we set the hyper-paramter α in Equation 1 to be 2 $\times$ rank r and for MixLoRA, we define α as 2 $\times$ factors $|E|$. For the other configuration, we adopt LLaVA's default setting for LoRA fine-tuning, as provided in its codebase. For the task-specific routing, we adopt the Vicunna (Chiang et al., 2023) as our pre-trained large language model $f_{\phi}(\cdot)$ for encoding task definition. Notably, Vicuna also serves as the language backbone of the LLaVA model. Following a similar approach to LoRA, for the LLaVA model with 32 Transformer layers, we insert MixLoRA into all linear layers within the Transformer layers. During training, all parameters in the MixLoRA module are updated, while the rest of LLaVA's parameters remain frozen.

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

C Evaluation Metrics

To evaluate the model performance on unseen multimodal datasets, we leverage Vicuna 1.5 13B (Chiang et al., 2023), the state-of-the-art open-source LLM to perform the evaluation. Specifically, we craft a prompt template that directs Vicuna to assess the accuracy of each prediction, considering the given task instructions and ground-truth target output. The prompt template used is as follows: " *A chat between a curious user and an artificial intelligence assistant. The assistant gives helpful, detailed, and polite answers to the user's questions. USER: Decide if the prediction is correct given the question and the answer. Questions: {Question} Answer: {Ground-truth Answer} Prediction: {Prediction} Your response should only be Yes or No. ASSISTANT:*" In this template, placeholders such as "{Question}", "{Ground-truth Answer}", and "{Prediction}" will be substituted with the specific details of each test instance. If Vicuna determines the prediction is correct, it outputs "Yes", and "No" otherwise. As all tasks are classification tasks, we compute accuracy based on Vicuna's judgments.