An Empirical Study on Parameter-Efficient Fine-Tuning for MultiModal Large Language Models

Xiongtao Zhou[1*] **Jie He**[2*] **Yuhua Ke**[2] **Guangyao Zhu**[1]
Víctor Gutiérrez-Basulto[3] and **Jeff Z. Pan**[2†]
[1] Waseda University, Japan
[2] School of Informatics, University of Edinburgh, UK
[3] School of Computer Science and Informatics, Cardiff University, UK
alenai.tao@ruri.waseda.jp, j.he@ed.ac.uk
s2484588@ed.ac.uk, zhuzgy@akane.waseda.jp
gutierrezbasultov@cardiff.ac.uk, j.z.pan@ed.ac.uk

Abstract

Multimodal large language models (MLLMs) fine-tuned with multimodal instruction datasets have demonstrated remarkable capabilities in multimodal tasks. However, fine-tuning all parameters of MLLMs has become challenging as they usually contain billions of parameters. To address this issue, we study *parameter-efficient fine-tuning* (PEFT) methods for MLLMs. We aim to identify effective methods for enhancing the performance of MLLMs in scenarios where only a limited number of parameters are trained. This paper conducts empirical studies using four popular PEFT methods to fine-tune the LLM component of open-source MLLMs. We present a comprehensive analysis that encompasses various aspects, including the impact of PEFT methods on various models, parameters and location of the PEFT module, size of fine-tuning data, model stability based on PEFT methods, MLLM's generalization, and hallucination. We evaluated four PEFT methods on seven datasets from two different categories: unseen and seen datasets. Across all experiments, we show that the adapter is the best-performing PEFT method. At the same time, fine-tuning the connector layers leads to improved performance in most MLLMs. Code and data are available at https://github.com/alenai97/PEFT-MLLM.git

1 Introduction

In recent years, the landscape of multimodal learning has been transformed by the emergence of multimodal large language models (MLLMs), such as LLaVA (Liu et al., 2023b), MiniGPT4 (Zhu et al., 2024), and GPT4-Vision (OpenAI et al., 2023). MLLMs have showcased impressive competency across a spectrum of multimodal benchmarks (Fu et al., 2023; Liu et al., 2023c; Li et al., 2023b) thanks to the integrated architecture of pre-trained visual encoders, connector layers, and LLMs. This architecture is usually fine-tuned through multimodal instruction-following data (Xu et al., 2023). During fine-tuning, most existing MLLMs (Cha et al., 2023; Su et al., 2023; Lin et al., 2023) typically freeze the visual encoder, focusing solely on connector layers and the LLM component. Since LLMs (e.g. LLaMA (Touvron et al., 2023) and Vicuna-v1.5 (Chiang et al., 2023)) often contain hundreds of billions of parameters, *full fine-tuning* (FFT) (Wang et al., 2022) is unfeasible. Consequently, the *parameter-efficient fine-tuning* (PEFT) (Houlsby et al., 2019; Hu et al., 2021) approach (which leverages lightweight trainable parameters and keeps the majority of parameters frozen) has been widely employed in NLP for fine-tuning LLMs with instruction or task-specific datasets (Li et al., 2023c; You et al., 2023), as they allow for significant resource savings while achieving comparable performance or even surpassing FFT (Mangrulkar et al., 2022).

In contrast to standard LLMs, MLLMs introduce additional modules: visual encoder and connector layers. During the fine-tuning process, unimodal LLMs only receive text features while MLLMs get multimodal inputs, such that connector layers are also fine-tuned, not just fine-tuning the LLM. Therefore, it is crucial to reassess the performance of fine-tuning MLLMs using various PEFT methods, exploring the impact of connector fine-tuning on the model's performance in downstream tasks, and examining PEFT's effects on model stability, generalization, and hallucination. In this paper we address these issues by conducting comprehensive studies on three representative MLLMs containing connector layers: LLaVA-1.5 (7B, 13B) (Liu et al., 2023a), ShareGPTv4 (7B) (Chen et al., 2023), and Qwen-VL-Chat (7B) (Bai et al., 2023b). Our study looks at various issues related to PEFT methods. Specifically, we design our study to address the following questions: **(1)** Is it necessary to fine-tune

[*] Equal Contribution.
[†] Corresponding author

the connector when fine-tuning MLLMS via various PEFT methods on unseen and seen datasets? **(2)** How does the position of the PEFT module in the LLM affect the MLLM's performance? **(3)** Faced with different training data scales, what differences exist in the performance of different PEFT methods? **(4)** How do different PEFT approaches impact the stability of the model? Is there any relationship between trainable parameters and learning rate with stability?

Our **key findings** can be summarized as follows:

1. Fine-tuning the connector layers usually leads to performance improvement within MLLMs.
2. More trainable parameters results in better performance on unseen datasets, while fewer trainable parameters maintains the model's performance on seen datasets.
3. Generally, fine-tuning using large scale datasets leads to better performance. However, when resources are limited, one should consider medium-size datasets instead.
4. Adapters show the best overall performance in model generalization, stability, and hallucination.

Our contributions can be summarized as follows: (1) We have assembled a standardized evaluation suite that includes seven benchmarks from the vision-and-language research community. This suite encompasses five tasks in visual question answering, one in visual reasoning, and one in image caption, along with four PEFT methods. **(2)** We utilized these resources to conduct in-depth experiments investigating four crucial design dimensions (cf. Fig. 1, left): 1) data scaling, 2) stability of the training process, 3) overfitting and generalization, and 4) hallucination. (3) Our empirical findings show that Adapter outperforms other PEFT methods in all aspects, followed in second place by LoRA. Furthermore, we show that fine-tuning the connector layers frequently enhances performance within MLLMs.

2 Related Work

Multimodal Large Language Models. Flamingo (Alayrac et al., 2022) proposes a GATED XATTN-DENSE layer to align visual and textual features, connecting the visual module and language model. LLaMA-adapter (Zhang et al., 2024) applies a projection layer to connect a visual encoder and LLaMA. It proposes adding an adapter module on LLaMA, keeping only the adapter parameters updated during training. In contrast, LLaVA-1.5 (Liu et al., 2023a) employs two layers of MLP to connect the visual encoder and LLM. During fine-tuning, it only updates the parameters of the MLP and LLM. Subsequent works mostly build upon this approach, employing the connector layers to link a visual encoder and LLMs, and then fine-tune the model using multimodal instruction-following data (Li et al., 2023a; Hu et al., 2023a; Wang et al., 2023). Recently, there are also many work on multimodal large language models (Chen et al., 2024) from the perspective of knowledge computing (Pan et al., 2023).

Parameter-Efficient Fine-Tuning. Parameter-efficient fine-tuning emerges as an approach capable of achieving performance comparable to full fine-tuning while keeping the majority of parameters frozen. Prompt-based methods (Lester et al., 2021) incorporate soft prompts into the input prefixes, only updating these soft prompts. Another widely used family of methods is based on adapters (Pfeiffer et al., 2020; He et al., 2022; He and Fu, 2023), which insert adapter modules at specific positions within transformer layers and update only the parameters of these inserted modules during training. Also, in MLLMs, low-rank decomposition methods are commonly employed (Hu et al., 2021; Edalati et al., 2022). These methods involve training only the parameters in low-rank matrices, significantly reducing the number of trainable parameters.

3 PEFT Methods

Figure 1 illustrates the architecture of MLLMs and the location of various PEFT modules. In our experiments, all considered MLLMs consist of three components: a visual encoder, connector layers, and a LLM. The structure of the connector layers may vary depending on the specific MLLM. PEFT methods can be classified into three categories, from which we select four methods: (1) *reparametrization-based tuning*: LoRA, IA3 (2) *adapter-based tuning*: Adapter. (3) *prompt-based tuning*: Prefix-Tuning.

LoRA. We integrate the low-rank strategy proposed by Hu et al. (2021) to adjust the network weights, facilitating the model's handling of complex tasks with an efficient parameter footprint. The original pre-trained weight matrix $W_0 \in \mathbb{R}^{d\times k}$

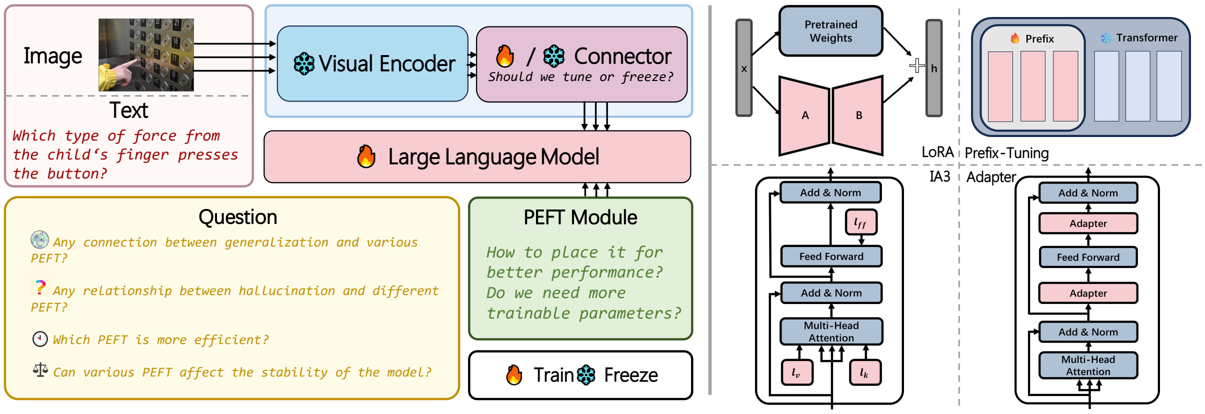


Image
Text
Which type of force from the child's finger presses the button?
Visual Encoder
/ Connector
Should we tune or freeze?
Large Language Model
Question
Any connection between generalization and various PEFT?
Any relationship between hallucination and different PEFT?
Which PEFT is more efficient?
Can various PEFT affect the stability of the model?
PEFT Module
How to place it for better performance?
Do we need more trainable parameters?
Train
Freeze
x
Pretrained Weights
A
B
h
LoRA
Prefix
Transformer
Prefix-Tuning
IA3
Add & Norm
l_{ff}
Feed Forward
Add & Norm
Multi-Head Attention
l_v
l_k
Adapter
Add & Norm
Adapter
Feed Forward
Adapter
Add & Norm
Multi-Head Attention

Figure 1: Left): Architecture of a Multimodal Large Language Model. Starting from 7 questions, we comprehensively explored the impact of PEFT methods and the connector on MLLMs, all of which are illustrated on the Left. Right): A detailed illustration of the PEFT module structure for the four PEFT methods.

is updated through low-rank decomposition using Equation 1, where $B \in \mathbb{R}^{d \times r}$ and $A \in \mathbb{R}^{r \times k}$.

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

This method ensures our model's adaptability is improved without a significant increase in the parameter space.

IA3. Following Liu et al. (2022), we integrate three vectors $v_k \in \mathbb{R}^{d_k}$, $v_v \in \mathbb{R}^{d_v}$, and $v_{ff} \in \mathbb{R}^{d_{ff}}$ into an attention mechanisms as:

$$\text{softmax}(\frac{Q(v_k \odot K^T)}{\sqrt{d_k}})(v_v \odot V) \tag{2}$$

where $\odot$ represents the element-wise multiplication, and $(v_{ff} \odot \gamma(W_1 x))W_2$ in the position-wise FFN layers, leveraging γ as the activation function. These formulas guide the model's attention to be fine-tuned to prioritize relevant features, optimizing performance without significantly increasing the model's complexity or number of parameters.

Adapter. We adopt the structure proposed by Houlsby et al. (2019), which adds adapter modules to the fully-connected networks after attention and the FFN layer within the transformer layers. This can be captured as follows:

$$h_i + f(W_{down}(h_i))W_{up} \rightarrow h_i \tag{3}$$

where h_i is the output of the previous layer, which is initially down-projected by $W_{\text{down}} \in \mathbb{R}^{d \times r}$ to a lower dimension r, and then up-projected back by $W_{\text{up}} \in \mathbb{R}^{r \times d}$ to the original dimension d, f is a non-linear layer.

Prefix-Tuning. We follow the approach proposed by Li and Liang (2021) to employ prefix learning by appending task-specific vector "prefixes" to the input sequence fed into the pre-trained model. We initialize a trainable matrix P_θ with dimensions $|P_{idx}| \times \dim(y_i)$, where P_{idx} specifies the prefix length. This yields the following conditional formulation for each element y_i of the output sequence:

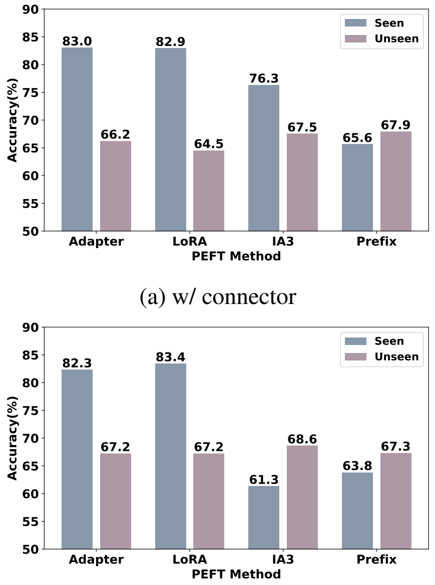
$$y_i = \begin{cases} P_\theta[i,:] & \text{if } i \in P_{idx}, \\ LM_\phi(z_i, y_{<i}) & \text{otherwise.} \end{cases} \tag{4}$$

If $i \in P_{idx}$, a bidirectional encoder computes the y_i. For $i \notin P_{idx}$, y_i is computed by an autoregressive neural LM as a function of y_i and the past activations in its left context.

4 Experiment Setup

4.1 Datasets

In the current era of large-scale models, dataset contamination is a significant concern as it is challenging to ensure that the data will be used for the next training process constitutes unseen data for large language models. Therefore, we categorize the datasets into two types: **Unseen datasets**, comprising datasets that have not been involved in the training of any of the considered models, including (1) the ScienceQA dataset (Lu et al., 2022); (2) the Vizwiz dataset (Gurari et al., 2018); (3) the IconQA dataset (Lu et al., 2021); and (4) the Flickr30k dataset (Young et al., 2014). **Seen datasets**, consisting of datasets used in the training of all considered models, including (1) the OKVQA dataset (Marino et al., 2019); (2) the OCRVQA dataset (Mishra et al., 2019); and (3) the VQAv2 dataset (Goyal et al., 2017). Details about datasets can be found in App. A.



(a) w/ connector

(b) w/o connector

Figure 2: The comparative performance of four PEFT methods on seen and unseen datasets, with and without the use of a connector.

4.2 Implementations

Models. We selected LLaVA-1.5 (7B, 13B), ShareGPTv4 (7B), and Qwen-VL-Chat (7B) as the base models for our experiments.

Hyperparameters. We conduct fine-tuning on the training set of each dataset separately and then test on their respective test or validation sets. All experiments were conducted with a global batch size of 128. We set the random seed of the experiment to 42. Additionally, each PEFT method was trained for three epochs on the fine-tuning dataset. For LoRA, we set its learning rate to 2e-4, the adapter's to 5e-5, IA3's to 2e-4, and Prefix-Tuning's to 1e-5. More information about model and hyperparameter settings is available in App. B

5 Experimental Results

5.1 Main Results

Should we tune or freeze the connector when considering unseen and seen datasets? Given the increasing availability of pretraining data for MLLMs, encountering contaminated data (i.e. training data contains information that is meant to be present only in the test set) is increasingly common. Additionally, most current MLLMs tune the connector layers during fine-tuning, yet the role of the connector remains unclear. Our main experiment thus focuses on investigating the performance of PEFT methods on both unseen and seen datasets. In our experiments, following existing work (Liu et al., 2023b; Bai et al., 2023b; Chen et al., 2023), we freeze the visual encoder and apply the PEFT module to fine-tune the LLM. For the connector layers, we experimented with both FFT and freezing. We investigate the model's performance on certain tasks during the task-specific fine-tuning when unfreezing the visual encoder, see App. C for details about unfreezing the visual encoder. We investigated the performance of LLaVA-1.5 (7B, 13B), ShareGPT4v, and Qwen-VL-Chat on the datasets mentioned in Section 4.1.

The obtained results are presented in Table 1. One can observe that LLaVA-1.5-13B with IA3 and fine-tuned connector layers achieved the best results across all unseen and seen datasets. Most IA3 models with fine-tuned connector achieved comparable performances to LoRA and Adapter on unseen datasets, while also maximizing the model's performance on seen datasets. Benefiting from the increased number of parameters in LLaVA-1.5-13B, we noticed that across various settings, the average performance on all datasets of LLaVA-1.5-13B surpasses that of LLaVA-1.5-7B. The average performance of ShareGPTv4 generally surpasses (except for the IA3 method with the frozen connector) that of LLaVA-1.5-7B, because it has been fine-tuned on the higher-quality multimodal instruction-following dataset (Chen et al., 2023). Under the setting of freezing the connector layers and fine-tuning the LLM with LoRA, Qwen-VL-Chat achieved the best performance. We found that choosing to tune the connector layers often leads to a significant deterioration in Qwen-VL-Chat's performance on seen datasets. Figure 2 illustrates the performance of various PEFT methods under the settings of tuning or freezing the connector layers. When fine-tuning the connector layers, LoRA, Prefix-Tuning, and IA3 all exhibit better performance than freezing the connector layers on unseen datasets. In this case, IA3 gets a 15.0% increase in the average result. On seen datasets, in most cases, the performance of freezing the connector layers and that of the remaining PEFT methods (except for a slight decrease in LoRA's performance) is similar. Note that whether the connector layers are fine-tuned or not, the performance of the Adapter remains relatively consistent on both seen and unseen datasets. Our **main findings** are the following:

Model	Method	SQA (img)	VizWiz	IconQA-txt	IconQA-blank	Flickr30k	OKVQA	OCRVQA	VQAv2	Avg
LLaVA-1.5-7B	Adapter	78.7	66.7	83.5	77.7	91.1	59.4	65.5	74.0	74.6
	-w/ connector	84.4	67.6	88.7	80.9	89.8	59.8	65.2	73.8	76.3
	LoRA	85.2	64.7	89.9	85.5	85.6	56.3	68.2	73.2	76.1
	-w/ connector	86.2	66.5	90.6	88.8	85.2	56.5	66.7	73.1	76.7
	IA3	69.1	56.2	55.2	41.9	77.2	59.8	66.9	78.1	63.1
	-w/ connector	82.7	61.9	89.2	82.2	91.9	60.5	67.1	75.2	76.3
	Prefix	68.2	59.0	73.0	46.8	91.5	61.1	68.6	76.9	68.1
	-w/ connector	69.7	60.8	76.7	50.9	91.9	61.3	68.5	77.0	69.6
LLaVA-1.5-13B	Adapter	82.4	66.6	88.9	84.2	94.0	59.4	67.4	74.7	77.2
	-w/ connector	83.7	66.8	90.6	85.8	93.1	59.6	67.2	74.5	77.7
	LoRA	86.3	66.3	90.9	90.3	87.9	59.1	70.8	74.4	78.3
	-w/ connector	**87.8**	66.1	91.6	90.4	84.1	59.9	68.6	73.6	77.8
	IA3	72.3	58.8	58.9	47.5	70.9	62.6	70.5	78.4	65.0
	-w/ connector	84.5	67.3	90.3	84.8	91.3	63.8	69.0	76.7	**78.5**
	Prefix	70.4	68.7	65.2	41.5	88.2	64.4	66.8	77.9	67.9
	-w/ connector	71.7	69.1	65.7	46.8	89.1	**64.7**	67.4	78.6	69.1
ShareGPT4V	Adapter	81.1	67.0	89.7	82.8	**95.6**	59.8	67.9	76.7	77.6
	-w/ connector	82.2	64.1	91.8	86.0	93.4	59.5	67.5	76.2	77.6
	LoRA	86.7	65.6	91.8	90.4	85.0	57.9	69.8	75.9	77.9
	-w/ connector	86.7	67.3	91.9	**90.6**	84.9	57.6	69.0	75.3	77.9
	IA3	69.0	61.1	58.7	47.7	57.5	60.7	69.1	**79.8**	63.0
	-w/ connector	82.0	60.9	90.9	84.1	93.8	61.4	68.7	77.3	77.4
	Prefix	67.9	63.6	73.8	45.2	91.6	62.4	68.9	78.7	69.0
	-w/ connector	68.4	65.2	81.3	53.2	92.4	62.3	67.7	78.8	71.2
Qwen-VL-Chat	Adapter	79.6	67.8	**92.4**	90.5	86.4	54.9	71.1	75.8	77.3
	-w/ connector	81.2	69.3	90.8	87.5	82.7	51.1	69.3	70.7	75.3
	LoRA	86.8	68.5	91.5	85.5	82.6	53.8	**71.4**	75.7	77.0
	-w/ connector	84.0	68.8	71.9	90.3	83.5	43.5	67.0	63.3	71.5
	IA3	70.0	66.9	71.0	41.8	73.6	50.7	68.3	77.8	65.0
	-w/ connector	67.3	69.8	57.3	28.7	65.1	50.5	62.1	77.5	59.8
	Prefix	52.2	**70.6**	52.4	33.2	52.2	50.1	61.3	70.6	55.3
	-w/ connector	51.9	70.4	52.5	31.8	52.9	49.8	61.5	77.4	56.0

Table 1: Main experimental results of various MLLMs with four PEFT methods. w/ connector: Tuning the connector.

- LoRA and Adapter have the best performance on all of the unseen datasets, while IA3 and Prefix-Tuning perform the best on the OKVQA and VQAv2. More trainable parameters allows the model to better adapt to unseen datasets, while fewer trainable parameters can maintain the model's performance on seen datasets.
- For the unseen datasets, tuning the connector layers often outperforms freezing the connector layers. For the seen datasets, freezing the connector layers yields the best performance.

5.2 Module Location

What is the best location for the PEFT module for MLLMs? Unlike LLMs, MLLMs include additional connector layers and visual encoders. Therefore, we can not straightforwardly transfer existing results for LLMs to MLLMs. With this in mind, we directly address this issue here. To this end, we selected all VQA datasets for the location study. We choose LLaVA-1.5-7B as the base model set the random seed to 42, freeze the visual encoder, and fine-tune the connector layers and LLM. We used this setting in subsequent experiments. For LoRA and IA3, we integrate them into the model's multi-head attention layer, MLP layer, or both. For adapters, we placed them in the same locations. The result of Qwen-VL-Chat can be found in App. D. Note that we do not consider Prefix-Tuning as the position is fixed. Table 2 presents the results on LLaVA-1.5-7B, which suggest that despite the additional modules in MLLMs compared to LLMs, the results of Hu et al. (2023b) for fine-tuning LLMs are also valid for MLLMs.

- We observe that for LoRA and IA3, the **Both** setting achieved the best results. As for Adapter, inserting it only into the **MLP** layer yielded the best performance.

5.3 Data Scale

In practical applications, MLLMs often require fine-tuning on downstream datasets (Li et al., 2023c; You et al., 2023), making PEFT an efficient choice. However, the sizes of these specific task datasets may vary, leading to the question: *How to select PEFT methods for datasets of different scales when training?* Therefore, we investigate the performance of PEFT methods on datasets of varying scales. We followed Chen et al. (2022) resource setting, and randomly sampled 1k, 5k, and 10k data points from the training set of each dataset. We categorize 1k data points as **Low-Resource**, 5k

Method	Location	SQA (img)	VizWiz	IconQA-txt	IconQA-blank	OKVQA	OCRVQA	VQAv2	Avg
				LLaVA-1.5$_{7B}$					
Adapter	Attn	81.3	67.9	89.2	80.3	58.8	66.2	75.0	74.1
	MLP	84.4	67.6	88.7	80.9	59.8	65.2	73.8	**74.3**
	Both	82.9	67.8	88.8	81.4	55.2	64.3	72.1	73.2
LoRA	Attn	84.1	**68.1**	90.5	83.8	58.4	67.0	73.5	75.1
	MLP	**85.6**	66.3	**90.8**	88.0	56.5	66.5	73.0	75.2
	Both	86.2	66.5	90.6	**88.8**	56.5	66.7	73.1	**75.5**
IA3	Attn	81.0	61.9	88.9	82.1	60.3	67.3	75.2	73.8
	MLP	82.0	62.1	88.7	82.6	**60.5**	**67.4**	**75.3**	**74.1**
	Both	82.7	61.9	89.2	82.2	**60.5**	67.1	75.2	**74.1**

Table 2: Average results of PEFT module location on LLaVA-1.5-7B. Attn: Placed on attention layer. MLP: Placed on MLP layer. Both: Placed both on attention layers and MLP layers.

	Method	SQA (img)	VizWiz	IconQA-txt	IconQA-blank	Flickr30k	OKVQA	OCRVQA	VQAv2	Avg
Low-Resource	Adapter	63.0	62.5	52.4	35.3	87.6	57.5	61.1	73.4	61.6
	LoRA	**68.8**	**62.7**	**62.9**	38.2	**89.4**	56.5	64.2	74.5	**64.7**
	IA3	67.0	50.3	60.6	**40.9**	86.0	**59.2**	64.0	**75.4**	62.9
	Prefix	49.8	56.3	51.3	20.6	81.6	51.6	**65.6**	53.1	53.7
Medium-Resource	Adapter	74.9	63.5	72.9	66.5	85.4	58.1	64.1	73.3	69.8
	LoRA	**80.0**	**66.4**	**78.9**	**74.8**	78.3	54.7	65.2	72.3	71.3
	IA3	77.4	55.1	77.8	76.4	88.5	59.3	65.6	**75.0**	**71.9**
	Prefix	56.5	52.2	63.1	38.8	**88.9**	**60.0**	**65.9**	74.7	62.5
High-Resource	Adapter	79.8	**66.0**	81.3	80.2	**91.9**	59.8	64.2	73.3	74.6
	LoRA	**84.7**	64.5	**84.9**	**87.1**	83.5	55.9	65.8	72.4	**74.9**
	IA3	80.9	58.3	84.0	85.0	88.9	60.5	65.8	74.7	74.8
	Prefix	67.5	55.7	70.3	52.1	91.0	**61.3**	**67.6**	**76.3**	67.7

Table 3: Fine-tuned average results with all PEFT methods on datasets of different sizes.

data points as **Medium-Resource**, and 10k data points as **High-Resource**. Note that since the training set of OKVQA contains only 9k samples, we considered the full data as high-resource. Table 3 presents the results.

Our main findings are the following:

- *High-Resource will make the MLLM more powerful, while Medium-Resource will be more efficient.* The performance of the four PEFT methods improves as the scale of resources grows, i.e. all achieve their best performance with high-resource. Thus, when resources are sufficient, fine-tuning on high-resource datasets will yield better performance. Average performance improvement is shown in App. E.

- *The unseen datasets tend to favor more resources.* When fine-tuning on an unseen dataset with more data, all PEFT methods show a significant performance improvement. In contrast, as the resources of the dataset increase, we did not observe significant performance improvement on seen datasets.

5.4 Stability Analysis

He et al. (2021) and Chen et al. (2022) carried out experiments with different random seeds to investigate the instability of fine-tuning LLMs using PEFT methods. Analogously, we look at such instability for MLLMs. We concentrate on the SQA (img) from the unseen datasets and OKVQA from seen datasets and select three random seeds: **[seed21, seed42, seed63]**. We present a stability analysis for LLaVA-1.5-7B, more analysis can be found in App. F.

The number of trainable parameters plays a crucial role in the fine-tuning process of a model. However, in the multimodal setting, the relationship between the number of trainable parameters and stability when fine-tuning with PEFT is not yet clear. With this in mind, we look at the following question: *Does fewer trainable parameters lead to higher stability?* We conducted an experiment under different trainable parameter conditions: on seed 21, seed 42, and seed 63, and varied the Lora Rank, Adapter Bottleneck Size, and Virtual Tokens to control the number of trainable parameters. Table 6 presents the performance of various PEFT methods and their standard deviations under different numbers of trainable parameters. IA3 is not tested since its trainable parameters cannot be modified. We draw the following conclusions:

- *Adapter and LoRA exhibit drastically different levels of stability on the unseen and seen*

	Source domain	Target domain	overfitting epoch 1	overfitting epoch 2	overfitting epoch 3	overfitting epoch 4
Adapter	IconQA-txt	SQA (img)	56.0	**56.4**	56.0	56.0
		VizWiz	58.7	**58.9**	**58.9**	58.2
	SQA (img)	IconQA-txt	36.7	36.9	**37.3**	**37.3**
		VizWiz	**58.3**	58.0	57.8	57.7
	VizWiz	SQA (img)	**56.9**	56.7	56.2	56.1
		IconQA-txt	**47.6**	45.9	44.9	44.4
	Avg	-	**52.4**	52.1	51.9	51.6
LoRA	IconQA-txt	SQA (img)	50.8	51.8	**52.5**	52.4
		VizWiz	**58.5**	57.6	57.8	57.6
	SQA (img)	IconQA-txt	33.8	**33.9**	33.5	33.4
		VizWiz	56.8	**57.2**	56.9	56.8
	VizWiz	SQA (img)	**61.3**	59.9	59.1	59.0
		IconQA-txt	42.2	44.3	**46.9**	40.9
	Avg	-	50.6	50.8	**51.1**	50.0
IA3	IconQA-txt	SQA (img)	61.1	**61.5**	61.0	61.2
		VizWiz	43.4	42.3	**45.2**	43.5
	SQA (img)	IconQA-txt	**45.2**	44.7	44.0	43.5
		VizWiz	**53.5**	53.1	53.1	52.9
	VizWiz	SQA (img)	**61.4**	60.2	60.4	60.0
		IconQA-txt	42.6	**43.3**	41.1	41.8
	Avg	-	**51.2**	50.9	50.8	50.5
Prefix	IconQA-txt	SQA (img)	**41.1**	37.4	34.7	33.9
		VizWiz	**47.2**	43.6	41.8	41.3
	SQA (img)	IconQA-txt	28.3	32.6	**33.1**	32.8
		VizWiz	**54.0**	53.7	53.9	53.5
	VizWiz	SQA (img)	**47.1**	39.9	41.4	46.3
		IconQA-txt	40.3	**44.5**	40.0	40.6
	Avg	-	**43.0**	42.0	40.8	41.4

Table 4: Performance on target domain with different PEFT methods. For each target domain and PEFT method, four epochs closest to the optimal point of overfitting were selected to test on the target domain. Avg: The average results of target domain at each epoch.

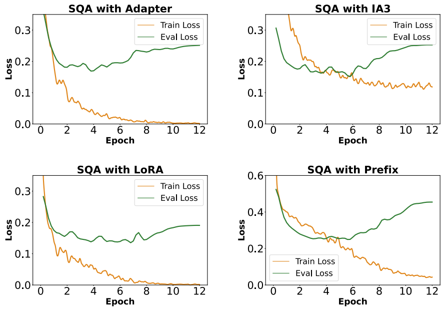


Figure 3: Train-Eval loss of all PEFT methods on SQA (img). The orange line shows Train Loss. Eval loss is colored with green.

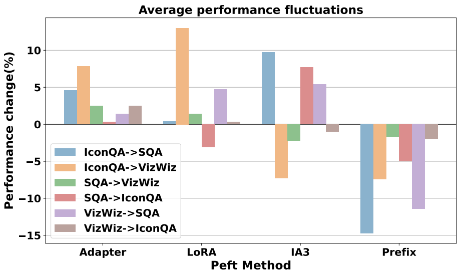


Figure 4: Average performance fluctuation of four epochs on each source-target domain. We calculate the mean of four PEFT methods on each source-target domain and display the average performance fluctuation of all PEFT methods on those domain-pair.

datasets. Prefix-Tuning shows a strong instability on the unseen datasets. Adapter gradually stabilizes with decreasing parameters on OKVQA, but becomes unstable with fewer parameters on SQA (img). Conversely, LoRA becomes unstable with decreasing parameters on OKVQA, but stabilizes with fewer parameters on SQA (img). Prefix-Tuning exhibits stability on OKVQA, and shows a relatively stable performance with fewer parameters on SQA (img).

5.5 Overfitting and Generalization

How robust different PEFT methods are relative to overfitting? To address this question, we considered three datasets from the unseen datasets: SQA (img), IconQA-txt, and Vizwiz. We choose one dataset from these three datasets as the source domain, and fine-tuned LLaVA-1.5-7B with PEFT methods on each source domain for 12 epochs.

Adapter and LoRA exhibit stronger robustness. Figure 3 (and Figure 8 in App. G) shows the evaluation loss of various PEFT methods on SQA as the number of training epochs changes. We observe that when overfitting occurs, there are differences in the robustness exhibited by each PEFT method on each dataset. On SQA, LoRA, IA3, and Adapter a relatively strong robustness is demonstrated, with LoRA performing the best. Prefix-Tuning shows

Fine-tuning Task	Method	Overall Score↑	Hallucination Rate↓	Attribute	Adversarial	Comparison	Counting	Relation	Environment	Holistic	Other
IconQA-txt	Adapter	**1.08**	**0.70**	0.58	1.17	1.42	0.25	2.17	1.33	0.00	1.75
	LoRA	0.69	0.83	0.58	0.00	1.42	0.67	1.17	1.00	0.00	0.67
	IA3	0.76	0.83	0.42	0.42	1.67	1.00	0.50	0.75	0.00	1.33
	Prefix	1.00	**0.70**	2.33	0.75	0.25	1.00	1.00	1.25	0.00	1.42
Flickr30k	Adapter	**0.73**	**0.81**	0.08	0.00	1.92	0.75	0.33	1.08	0.27	0.92
	LoRA	0.70	0.83	0.75	0.33	1.92	0.58	0.42	0.33	0.00	1.25
	IA3	0.70	0.84	0.25	0.42	1.33	1.17	0.75	0.67	0.00	1.00
	Prefix	0.59	0.82	0.25	0.00	0.50	0.33	0.58	1.50	0.00	1.33

Table 5: Evaluation results of LLaVA-1.5-7B with different PEFT methods on MMHAL-Bench.

		OKVQA	SQA (img)
Adapter	Bottleneck Size=32	$62.9_{\pm 0.21}$	$80.4_{\pm 3.12}$
	Bottleneck Size=64	$62.7_{\pm 0.60}$	$81.2_{\pm 2.75}$
	Bottleneck Size=128	$61.4_{\pm 0.20}$	$81.3_{\pm 1.80}$
	Bottleneck Size=256	$58.8_{\pm 0.84}$	$82.2_{\pm 1.91}$
LoRA	LoRA Rank=16	$56.1_{\pm 0.53}$	$85.7_{\pm 0.32}$
	LoRA Rank=32	$56.1_{\pm 0.27}$	$85.3_{\pm 0.92}$
	LoRA Rank=64	$56.4_{\pm 0.20}$	$85.0_{\pm 0.85}$
	LoRA Rank=128	$56.6_{\pm 0.12}$	$85.4_{\pm 0.85}$
Prefix	Virtual Tokens=10	$62.2_{\pm 0.10}$	$73.4_{\pm 2.62}$
	Virtual Tokens=20	$61.5_{\pm 0.20}$	$72.2_{\pm 1.11}$
	Virtual Tokens=30	$61.2_{\pm 0.06}$	$67.7_{\pm 0.78}$
	Virtual Tokens=40	$61.2_{\pm 0.27}$	$56.2_{\pm 19.20}$

Table 6: Performance on three PEFT methods with different hyperparameter settings. Reported results are averages across three runs with different random seeds.

Method	Epoch 3	Epoch 6	Epoch 9	Epoch 12	Avg
Adapter	17	12	14	10	13.3
LoRA	15	14	18	20	16.8
IA3	14	17	17	16	16.0
Prefix	24	18	27	31	25.0

Table 7: Hallucinations statistic of PEFT methods on four epochs. We selected 100 hallucination-free examples from 1k random sampled data from the outputs of LLaVA-1.5-7B. We examined the outputs of LLaVA-1.5-7B with four PEFT methods on those examples, table presents the number of outputs with hallucination.

poor robustness on SQA. App. G provides further analysis on IconQA-txt and Vizwiz.

When facing overfitting an important question is *How do various PEFT methods perform in terms of generalization? And how to achieve the best generalization performance during training?* With these questions in mind, we conducted the next experiment. Based on Figure 3, we identified the training step with the minimum evaluation loss for each PEFT method. We selected four overfitting points which are the closest to the minimum evaluation loss point on the source domain. Subsequently, we tested the performance of each epoch on the other two target domains, yielding the results shown in Table 4. We draw the following conclusions.

- *Adapter exhibits the strongest in generalization.* Figure 4 shows that when a model is fine-tuned using Prefix-Tuning its generalization performance is quite poor. Models using Adapter consistently exhibit a good generalization performance regardless of the situation, while the generalization performance of a model fine-tuned with Prefix-Tuning is consistently negative. Models using LoRA and IA3 show fluctuation in generalization performance.
- *IA3, Adapter, and Prefix-Tuning show the best generalization performance at the first overfitting epoch.* From the results of Table 4, we can find that Adapters, IA3, and Prefix-Tuning, all achieve the best average generalization performance at the first overfitting epoch. In general, the model's generalization weakens as overfitting intensifies. However, LoRA achieves the best model generalization performance at the third overfitting epoch, indicating that models fine-tuned with LoRA exhibit the best generalization when overfitting reaches a certain level, gradually weakening afterwards.

5.6 Hallucination

The hallucination problem in LLMs has been widely acknowledged (Ji et al., 2023; Gudibande et al., 2024). Since MLLMs are built upon LLMs, this problem is also present in them. Zhai et al. (2023) found that further fine-tuning with multimodal instruction-following data leads to hallucinations. Therefore, we aim to investigate the following question: *Which PEFT method results in fewer hallucinations during fine-tuning?* We select IconQA-txt as the source domain and the Flickr30k dataset as the target domain to assess the out-of-domain hallucinations of models fine-tuned with various PEFT methods. App. I.1 elaborates on how we evaluated the model's hallucination.

MMHAL-Bench (Sun et al., 2023) is used to evaluate the hallucinations induced by fine-tuning LLaVA-1.5-7B with four PEFT methods on Flickr30k and IconQA-txt, yielding the results presented in Table 5. The results show that Adapter consistently achieved the highest Avg Score and the lowest Hallucination Rate across both fine-tuning tasks. This is consistent with our manual evaluation results.

Adapter demonstrates potential for addressing hallucinations in MLLMs. The results are illustrated in Table 7. We observe that Adapter achieves the lowest average hallucination rate across four epochs, at only 13.3%. It can also be found that other PEFT methods tend to produce more hallucinations after further fine-tuning, especially Prefix-Tuning, which generates an additional 24% of hallucinations from epoch 3 to epoch 12. In contrast, with further fine-tuning, Adapter reduced the number of hallucinations produced. In line with previous studies (Wang et al., 2023), we attribute this phenomenon to the new parameters in the Adapter method, which provides a new module to adapt to downstream datasets while keeping the base model's original weights.

6 Conclusion

We conducted an extensive investigation on four PEFT methods applied to MLLMs across different multimodal tasks. By fine-tuning different MLLMs in a uniform way and conducting thorough hyperparameter optimization, we benchmarked the performance of these methods. Our findings indicate that Adapter excels in accuracy, stability, generalization, and producing fewer hallucinations. Additionally, we found that fine-tuning the connector layers of MLLMs simultaneously does not always yield better results. Finally, comprehensive ablation studies were performed to understand the contributions of the location of PEFT modules, learning rate settings, and the size of training data on PEFT performance.

Limitations

All our experiments were conducted within the defined framework, which involves connector layers serving as the bridge between the visual encoder and LLM, and no additional modules were inserted on the LLM. Due to the limitation of computational resources, we have currently employed only a subset of datasets to conduct our analysis. Additionally, our choice of MLLMs on the analysis experiments is limited to LLaVA-1.5-7B or Qwen-VL-Chat. In the future, we plan to conduct an analysis on more datasets and MLLMs.

References

Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, et al. 2022. Flamingo: a visual language model for few-shot learning. *Advances in Neural Information Processing Systems*, 35:23716–23736.

Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, Binyuan Hui, Luo Ji, Mei Li, Junyang Lin, Runji Lin, Dayiheng Liu, Gao Liu, Chengqiang Lu, Keming Lu, Jianxin Ma, Rui Men, Xingzhang Ren, Xuancheng Ren, Chuanqi Tan, Sinan Tan, Jianhong Tu, Peng Wang, Shijie Wang, Wei Wang, Shengguang Wu, Benfeng Xu, Jin Xu, An Yang, Hao Yang, Jian Yang, Shusheng Yang, Yang Yao, Bowen Yu, Hongyi Yuan, Zheng Yuan, Jianwei Zhang, Xingxuan Zhang, Yichang Zhang, Zhenru Zhang, Chang Zhou, Jingren Zhou, Xiaohuan Zhou, and Tianhang Zhu. 2023a. Qwen technical report.

Jinze Bai, Shuai Bai, Shusheng Yang, Shijie Wang, Sinan Tan, Peng Wang, Junyang Lin, Chang Zhou, and Jingren Zhou. 2023b. Qwen-vl: A versatile vision-language model for understanding, localization, text reading, and beyond.

Junbum Cha, Wooyoung Kang, Jonghwan Mun, and Byungseok Roh. 2023. Honeybee: Locality-enhanced projector for multimodal llm.

Guanzheng Chen, Fangyu Liu, Zaiqiao Meng, and Shangsong Liang. 2022. Revisiting parameter-efficient tuning: Are we really there yet? In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 2612–2626, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Lin Chen, Jinsong Li, Xiaoyi Dong, Pan Zhang, Conghui He, Jiaqi Wang, Feng Zhao, and Dahua Lin. 2023. Sharegpt4v: Improving large multi-modal models with better captions.

Zhuo Chen, Yichi Zhang, Yin Fang, Yuxia Geng, Lingbing Guo, Xiang Chen, Qian Li, Wen Zhang, Jiaoyan Chen, Yushan Zhu, Jiaqi Li, Xiaoze Liu, Jeff Z. Pan, Ningyu Zhang, and Huajun Chen. 2024. Knowledge Graphs Meet Multi-Modal Learning: A Comprehensive Survey. In *arxiv*.

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.

Ali Edalati, Marzieh Tahaei, Ivan Kobyzev, Vahid Partovi Nia, James J. Clark, and Mehdi Rezagholizadeh. 2022. Krona: Parameter efficient tuning with kronecker adapter.

Chaoyou Fu, Peixian Chen, Yunhang Shen, Yulei Qin, Mengdan Zhang, Xu Lin, Jinrui Yang, Xiawu Zheng, Ke Li, Xing Sun, Yunsheng Wu, and Rongrong Ji. 2023. Mme: A comprehensive evaluation benchmark for multimodal large language models.

Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. 2017. Making the V in VQA matter: Elevating the role of image understanding in Visual Question Answering. In *Conference on Computer Vision and Pattern Recognition (CVPR)*.

Arnav Gudibande, Eric Wallace, Charlie Victor Snell, Xinyang Geng, Hao Liu, Pieter Abbeel, Sergey Levine, and Dawn Song. 2024. The false promise of imitating proprietary language models. In *The Twelfth International Conference on Learning Representations*.

Danna Gurari, Qing Li, Abigale J. Stangl, Anhong Guo, Chi Lin, Kristen Grauman, Jiebo Luo, and Jeffrey P. Bigham. 2018. Vizwiz grand challenge: Answering visual questions from blind people. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Jie He and Yu Fu. 2023. Metaxcr: Reinforcement-based meta-transfer learning for cross-lingual commonsense reasoning. In *Proceedings of The 1st Transfer Learning for Natural Language Processing Workshop*, volume 203 of *Proceedings of Machine Learning Research*, pages 74–87. PMLR.

Junxian He, Chunting Zhou, Xuezhe Ma, Taylor Berg-Kirkpatrick, and Graham Neubig. 2022. Towards a unified view of parameter-efficient transfer learning. In *Proceedings of the 10th International Conference on Learning Representations (ICLR-2022)*.

Ruidan He, Linlin Liu, Hai Ye, Qingyu Tan, Bosheng Ding, Liying Cheng, Jiawei Low, Lidong Bing, and Luo Si. 2021. On the effectiveness of adapter-based tuning for pretrained language model adaptation. 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 2208–2222, Online. Association for Computational Linguistics.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. 2019. Parameter-efficient transfer learning for NLP. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 2790–2799. PMLR.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models.

Wenbo Hu, Yifan Xu, Yi Li, Weiyue Li, Zeyuan Chen, and Zhuowen Tu. 2023a. Bliva: A simple multimodal llm for better handling of text-rich visual questions.

Zhiqiang Hu, Lei Wang, Yihuai Lan, Wanyu Xu, Ee-Peng Lim, Lidong Bing, Xing Xu, Soujanya Poria, and Roy Lee. 2023b. LLM-adapters: An adapter family for parameter-efficient fine-tuning of large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 5254–5276, Singapore. Association for Computational Linguistics.

Gabriel Ilharco, Mitchell Wortsman, Ross Wightman, Cade Gordon, Nicholas Carlini, Rohan Taori, Achal Dave, Vaishaal Shankar, Hongseok Namkoong, John Miller, Hannaneh Hajishirzi, Ali Farhadi, and Ludwig Schmidt. 2021. Openclip. If you use this software, please cite it as below.

Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Ye Jin Bang, Andrea Madotto, and Pascale Fung. 2023. Survey of hallucination in natural language generation. *ACM Comput. Surv.*, 55(12).

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B. Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. 2020. Scaling laws for neural language models.

Brian Lester, Rami Al-Rfou, and Noah Constant. 2021. The power of scale for parameter-efficient prompt tuning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3045–3059, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Bo Li, Yuanhan Zhang, Liangyu Chen, Jinghao Wang, Jingkang Yang, and Ziwei Liu. 2023a. Otter: A multi-modal model with in-context instruction tuning. *arXiv preprint arXiv:2305.03726*.

Bohao Li, Rui Wang, Guangzhi Wang, Yuying Ge, Yixiao Ge, and Ying Shan. 2023b. Seed-bench: Benchmarking multimodal llms with generative comprehension. *ArXiv*, abs/2307.16125.

Chunyuan Li, Cliff Wong, Sheng Zhang, Naoto Usuyama, Haotian Liu, Jianwei Yang, Tristan Naumann, Hoifung Poon, and Jianfeng Gao. 2023c. LLaVA-med: Training a large language-and-vision assistant for biomedicine in one day. In *Thirty-seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

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, Online. Association for Computational Linguistics.

Ziyi Lin, Chris Liu, Renrui Zhang, Peng Gao, Longtian Qiu, Han Xiao, Han Qiu, Chen Lin, Wenqi Shao, Keqin Chen, Jiaming Han, Siyuan Huang, Yichi Zhang, Xuming He, Hongsheng Li, and Yu Qiao.

2023. Sphinx: The joint mixing of weights, tasks, and visual embeddings for multi-modal large language models.

Haokun Liu, Derek Tam, Mohammed Muqeeth, Jay Mohta, Tenghao Huang, Mohit Bansal, and Colin Raffel. 2022. Few-shot parameter-efficient fine-tuning is better and cheaper than in-context learning.

Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. 2023a. Improved baselines with visual instruction tuning.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. 2023b. Visual instruction tuning. *ArXiv*, abs/2304.08485.

Yuan Liu, Haodong Duan, Yuanhan Zhang, Bo Li, Songyang Zhang, Wangbo Zhao, Yike Yuan, Jiaqi Wang, Conghui He, Ziwei Liu, Kai Chen, and Dahua Lin. 2023c. Mmbench: Is your multi-modal model an all-around player?

Pan Lu, Swaroop Mishra, Tony 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. In *NeurIPS*.

Pan Lu, Liang Qiu, Jiaqi Chen, Tony Xia, Yizhou Zhao, Wei Zhang, Zhou Yu, Xiaodan Liang, and Song-Chun Zhu. 2021. IconQA: A new benchmark for abstract diagram understanding and visual language reasoning. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*.

Sourab Mangrulkar, Sylvain Gugger, Lysandre Debut, Younes Belkada, Sayak Paul, and Benjamin Bossan. 2022. Peft: State-of-the-art parameter-efficient fine-tuning methods. https://github.com/huggingface/peft.

Kenneth Marino, Mohammad Rastegari, Ali Farhadi, and Roozbeh Mottaghi. 2019. Ok-vqa: A visual question answering benchmark requiring external knowledge. In *Conference on Computer Vision and Pattern Recognition (CVPR)*.

Anand Mishra, Shashank Shekhar, Ajeet Kumar Singh, and Anirban Chakraborty. 2019. Ocr-vqa: Visual question answering by reading text in images. In *ICDAR*.

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

Jeff Z. Pan, Simon Razniewski, Jan-Christoph Kalo, Sneha Singhania, Jiaoyan Chen, Stefan Dietze, Hajira Jabeen, Janna Omeliyanenko, Wen Zhang, Matteo Lissandrini, ussa Biswas, Gerard de Melo, Angela Bonifati, Edlira Vakaj, Mauro Dragoni, and amien Graux. 2023. Large language models and knowledge graphs: Opportunities and challenges. *Transactions on Graph Data and Knowledge*.

Jonas Pfeiffer, Ivan Vulić, Iryna Gurevych, and Sebastian Ruder. 2020. MAD-X: An Adapter-Based Framework for Multi-Task Cross-Lingual Transfer. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7654–7673, Online. Association for Computational Linguistics.

Yixuan Su, Tian Lan, Huayang Li, Jialu Xu, Yan Wang, and Deng Cai. 2023. Pandagpt: One model to instruction-follow them all. *arXiv preprint arXiv:2305.16355*.

Zhiqing Sun, Sheng Shen, Shengcao Cao, Haotian Liu, Chunyuan Li, Yikang Shen, Chuang Gan, Liang-Yan Gui, Yu-Xiong Wang, Yiming Yang, Kurt Keutzer, and Trevor Darrell. 2023. Aligning large multimodal models with factually augmented rlhf.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. Llama: Open and efficient foundation language models.

Weihan Wang, Qingsong Lv, Wenmeng Yu, Wenyi Hong, Ji Qi, Yan Wang, Junhui Ji, Zhuoyi Yang, Lei Zhao, Xixuan Song, Jiazheng Xu, Bin Xu, Juanzi Li, Yuxiao Dong, Ming Ding, and Jie Tang. 2023. Cogvlm: Visual expert for pretrained language models.

Yaqing Wang, Sahaj Agarwal, Subhabrata Mukherjee, Xiaodong Liu, Jing Gao, Ahmed Hassan Awadallah, and Jianfeng Gao. 2022. AdaMix: Mixture-of-adaptations for parameter-efficient model tuning. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 5744–5760, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Zhiyang Xu, Ying Shen, and Lifu Huang. 2023. 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.

Haoxuan You, Haotian Zhang, Zhe Gan, Xianzhi Du, Bowen Zhang, Zirui Wang, Liangliang Cao, Shih-Fu Chang, and Yinfei Yang. 2023. Ferret: Refer and ground anything anywhere at any granularity. *arXiv preprint arXiv:2310.07704*.

Peter Young, Alice Lai, Micah Hodosh, and Julia Hockenmaier. 2014. From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions. *Transactions of the Association for Computational Linguistics*, 2:67–78.

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 model fine-tuning. In *Conference on Parsimony and Learning (Proceedings Track)*.

Renrui Zhang, Jiaming Han, Chris Liu, Aojun Zhou, Pan Lu, Hongsheng Li, Peng Gao, and Yu Qiao. 2024. LLaMA-adapter: Efficient fine-tuning of large language models with zero-initialized attention. In *The Twelfth International Conference on Learning Representations*.

Deyao Zhu, Jun Chen, Xiaoqian Shen, Xiang Li, and Mohamed Elhoseiny. 2024. MiniGPT-4: Enhancing vision-language understanding with advanced large language models. In *The Twelfth International Conference on Learning Representations*.

A Datasets Setup

A.1 Detailed Description of Datasets

Since OCRVQA and VQAv2 are very large, we randomly extract 20k samples from their training sets as the new training sets and another 5k samples from the test sets to create the new test sets. We utilized a variety of multimodal datasets for fine-tuning and evaluating. Detailed information for each dataset is provided in Table 8 below.

A.2 Seen Datasets for All MLLMs

In our experiments, we divided the datasets into unseen datasets and seen datasets. In Table 9, we present the training datasets used by each MLLM in our experiments. It is worth noting that, due to the data used in both pre-training and fine-tuning stages being filtered or sampled from the datasets, to ensure a fair comparison, it is imperative that each seen dataset is fully seen by the model. We composed a mixed dataset consisting of three seen datasets: OKVQA, VQAv2, and OCRVQA. Subsequently, we kept the visual encoder frozen and conducted a Full Fine-Tuning on each model using this mixed dataset. During this process, we set the learning rate to 2e-5 and the global batch size to 128. We maintained all other settings the same as those used in the original paper for Full Fine-Tuning (Liu et al., 2023a; Bai et al., 2023b).

A.3 Instruction-following Data Template

Multimodal Instruction-Following Tuning is a crucial component for the success of MLLMs. The MLLMs used in this experiment follow the same approach in data processing as the original models. Therefore, there are slight differences in the processing of LLaVA-1.5, ShareGPT4v, and Qwen-VL-Chat. They employ different image annotations, but the data instruction format remains consistent. Table 10 shows the template of all dataset types used in the experiments.

B Models and Hyperparameters

B.1 Models

LLaVA-1.5 consists of CLIP-ViT-L/14 (Ilharco et al., 2021), Vicuna-v1.5 (Chiang et al., 2023), and an MLP serving as the connector layer. During fine-tuning with multimodal instruction-following data, LLaVA-1.5 updates only the parameters of the MLP connector and the LLM, while keeping the parameters of the visual encoder frozen. ShareGPTv4 is obtained by fine-tuning LLaVA-1.5 using the ShareGPT4v dataset (Chen et al., 2023), which comprises 100K high-quality captions from various images generated by GPT4-vision. Qwen-VL-Chat comprises ViT-G/16 (Ilharco et al., 2021), Qwen-7B (Bai et al., 2023a), and a cross-attention module serving as the connector. During its vision instruction tuning phase, Qwen-VL-Chat updates only the parameters of the connector and the LLM.

B.2 HyperParameters

Following Hu et al. (2023b), we conducted the following parameter selection experiments. Due to computational constraints, we concentrate on the SQA (all) dataset, which has a test set that contains both multimodal and text-only data. So, our goal is to enhance the model's multimodal performance while maximizing its performance on text-only datasets.

We choose LLaVA-1.5-7B as the base model set the random seed to 42, freeze the visual encoder, and fine-tune the connector layers and LLM. Note that we do not consider IA3 in this experiment as it cannot change the number of trainable parameters. We look at different parameter settings: LoRA rank of {16, 32, 64, 128}, Adapter bottleneck size of {32, 64, 128, 256}, and virtual tokens of {10, 20, 30, 40} in Prefix-Tuning. Figure 5 shows the results with various parameter settings on SQA (all). We observe that for LoRA, a rank of 128 yielded an accuracy of 89.3%. Setting the adapter's bottleneck size to 256 resulted in an accuracy of 87.4%. In the case of Prefix-Tuning, setting the virtual tokens to 20 achieved the best accuracy of 68.0%.

Based on our findings, for all remaining experiments, we use the following PEFT parameters: **LoRA Rank=128**, **Adapter Bottleneck Size=256**, and **Prefix Virtual Token=20**. More detailed hyperparameter settings are presented in Table 11. We utilized two NVIDIA A100 80GB GPUs and DeepSpeed for distributed training.

C Training of Visual Encoder Analysis

Qwen-vl-chat unfroze the visual encoder during pre-training, which improved the model's performance. However, most current MLLMs maintain the visual encoder frozen during task-specific fine-tuning. We fine-tuned the visual encoder on SQA and VizWiz for unseen tasks, and on OKVQA and OCRVQA for seen tasks. The results are shown in the Table 12. We found that although unfreezing the visual encoder does not significantly increase training resource consumption, the improvement in

Dataset	Task	Split	Metric	Answer type	Description	Dataset Type	# Train	# Test (Val)
Flickr30K	Image Caption	train & test	CIDEr (↑)	Caption	Image dataset with captions for natural scenes.	Unseen	31k	4k
IconQA-blank	Visual Reasoning	train & test	Accuracy (↑)	Word	Visual reasoning with abstract icons, no text.	Unseen	11k	4k
IconQA-txt	Visual Reasoning	train & test	Accuracy (↑)	Word	Abstract icon reasoning with textual hints.	Unseen	19k	6k
OKVQA	Knowledge Grounded VQA	train & test	VQA-Score (↑)	Phrase	VQA requiring external knowledge.	Seen	9k	5k
SQA(img)	Knowledge Grounded VQA	train & test	Accuracy (↑)	Option	Science-focused multiple-choice VQA	Unseen	13k	4k
OCRVQA	Reading Comprehension VQA	train & test	Accuracy (↑)	Phrase	VQA with text recognition in images.	seen	20k	5k
VQAv2	General VQA	train & test	VQA-Score (↑)	Phrase	Diverse open-ended visual question answering.	Unseen	20k	5k
VizWiz	General VQA	train & val	VQA-Score (↑)	Phrase	VQA sourced from visually impaired users' photos.	seen	20k	4k

Table 8: Detailed description for the datasets we used, including task types, training and test split, evaluation metric, statistic, dataset type, and the type of answer. To be specific, "Seen" means that the dataset has been used as a pre-training dataset in the model being evaluated. "Unseen" refers to datasets that have not been encountered by the model.

model	Phrase	Seen datasets
LLaVA	Pretrained	CC-595K, LLaVA-Instruct-158K
	Fine-tuning	**VQAv2**, **OKVQA**, **OCRVQA**, GQA, A-OKVQA, TextCaps, RefCOCO, VG
Qwen-VL-Chat	Pretrained	LAION-en, LAION-COCO, DataComp, Coyo, CC12M, CC3M, SBU, COCO Caption, LAION-zh, GQA, VGQA, VQAv2, DVQA, OCRVQA, DocVQA. TextVQA, ChartQA, AI2D, GRIT, VG, RefCOCO+, RefCOCOg, SynthDoG-en & zh, Common Crawl of pdf & HTML, In-house Data
	Fine-tuning	**OKVQA**, **OCRVQA**, **VQAv2**
ShareGPT4v	Pretrained	CC-595K, LLaVA-Instruct-158K, ShareGPT4V-PT
	Fine-tuning	**VQAv2**, **OKVQA**, **OCRVQA**, A-OKVQA, GQA, TextCaps, RefCOCO, VG, ShareGPT4V

Table 9: The datasets used during the pretraining and further fine-tuning processes of MLLMs.

model performance is limited and, in most cases, can even lead to performance degradation. Therefore, in our experiments, we adhered to the mainstream setting and kept the visual encoder frozen.

D Location Analysis

We also conducted the Module Location experiment on Qwen-VL-Chat. Table 13 shows the results. We observe that for LoRA and IA3, Both settings achieved the best results. As for Adapter, inserting it only into the MLP layer yielded the best performance. It reveals that the results on Qwen-VL-Chat are consistent with those on LLaVA-1.5-7B.

E Data Scale Analysis

Figure 6 shows the improvement in the average performance of the four PEFT methods as resources transition from low to high. When datasets transition from low to medium resources, all four PEFT methods achieve performance improvements of over 10%, higher than from medium to high resources. Thus, when computational resources are limited, fine-tuning on medium-resource datasets is more efficient.

F Stability Analysis

Figure 7 presents the training loss, showing that the stability of PEFT varies across different datasets. We observe that Prefix-Tuning and Adapter exhibit larger fluctuations in training loss at each step when trained with different seeds, followed by LoRA, while IA3 shows relatively smaller fluctuations.

We also investigate whether the learning rate correlates with stability. We conducted experiments with learning rates of {2e-4, 5e-5, 1e-5, 5e-6}. The results are shown in Table 14. It can be observed that IA3 and Prefix-Tuning demonstrate more stable performance at smaller learning rates, Prefix-Tuning tends to stabilize gradually as the learning rate decreases.

G Overftitting and Generalization Analysis

Overfitting and Generalization experiments were conducted on three unseen VQA datasets. The train-loss curves for IconQA-txt and VizWiz are depicted in Figure 8. For IconQA-txt, all four PEFT

Model	Image Annotation[IMAGE]
LLaVA Image	<Image>
Qwen Image	<img></img>
Task	Instruction Template
Image Caption	[IMAGE] Share a concise interpretation of the image provided. [IMAGE] Render a clear and concise summary of the photo. [IMAGE] Write a terse but informative summary of the picture. [IMAGE] Offer a succinct explanation of the picture presented. [IMAGE] Describe the image concisely. [IMAGE] Provide a brief description of the given image. [IMAGE] Create a compact narrative representing the image presented. [IMAGE] Relay a brief, clear account of the picture shown. [IMAGE] Summarize the visual content of the image.
Knowledge Grounded VQA	[IMAGE] Give a short and clear explanation of the subsequent image [IMAGE] {Question} Answer the question using a single word or phrase. [IMAGE] {Question} A.choice 1, B.choice 2, C.choice 3, ...
Visual Reasoning	[IMAGE] {Question} Fill in the blanks in (_) or answer this question. [IMAGE] {Question} Choices: choice 1, choice 2, choice 3,... Choose an option from the choices to answer the question.
Reading Comprehension VQA	[IMAGE] {Question} Answer the question using a single word or phrase.
General VQA	[IMAGE] Question: {Question} [IMAGE] Question: {Question} When the information is insufficient, respond with "Unanswerable". Answer the question using a single word or phrase. [IMAGE] {Question} Answer the question using a single word or phrase.

Table 10: The instruction format of different tasks when we fine-tune MLLMs.

methods exhibit strong robustness, with LoRA performing the best. On Vizwiz, Prefix-Tuning shows the strongest robustness compared to the other three PEFT methods.

H Efficiency

We investigate the number of trainable parameters, the training and inference Flops for various MLLMs when fine-tuned with different PEFT methods in this section. Table 15 shows the results. We derive the training and inference FLOPs in accordance with the methodology outlined in Kaplan et al. (2020). Our analysis shows that models perform better when the connector is not frozen, which suggests that having more trainable parameters improves the performance, even though the overall parameter efficiency might decrease. Within the 7B model, the Adapter method without freezing connector is remarkably efficient, utilizing only 3.060% of trainable parameters and yet securing a high performance rate of 76.3%, showcasing an optimal balance between parameter efficiency and model efficacy.

Additionally, using the IA3 method without freezing the connector can reduce the computing effort needed for training. With more trainable parameters, the model becomes more efficient, producing shorter and more accurate texts and thus requiring less computing power, even as the number of trainable parameters grows. In the case of the 13B model, the increase in the total number of parameters is not necessarily reflected in an increase in the percentage of trainable parameters to reach a good performance. According to Table 15, the IA3 method without a frozen connector yields significantly fewer trainable parameters compared to the Adapter and LoRA methods, but achieves the highest performance.

The details of the flops calculation are :

Training Flops Since the computational cost of the backward pass is approximately twice as the forward pass, we modify the formula as:

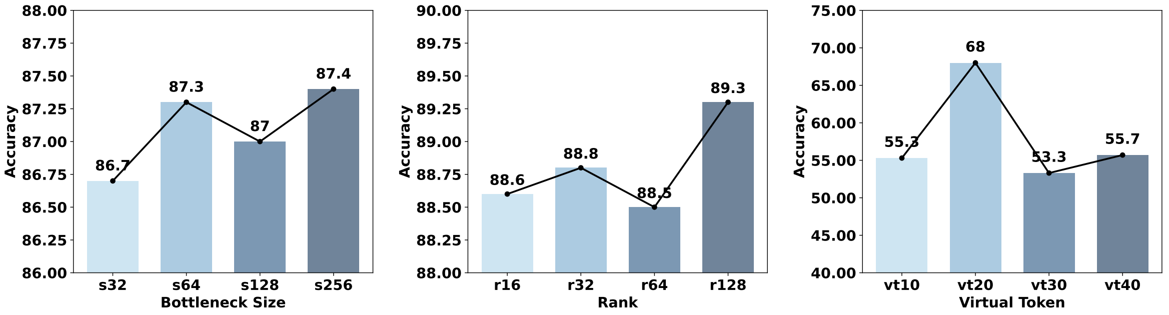
$$\text{Train Flops} = (2P_f + 4P_t) \times N_t \tag{5}$$

where P_f and P_t represent the number of frozen and trainable parameters respectively, N_t is the number of input and model-generated tokens.

Inference Flops We calculate the inference flops based on the following equation:

$$\text{Inference Flops} = 2\times(P_g + N_{layer} d_{model} N_t) \tag{6}$$

where P_g indicates *non-embedding* parameters,



(a) Adapter

(b) LoRA

(c) Prefix

Figure 5: Average accuracy of Results of various PEFT parameters on SQA (all). s: Bottleneck Size. r: LoRA Rank. vt: Virtual Token

Configuration	LLaVA-7B	LLaVA-13B	Qwen-VL-Chat	ShareGPT4v
ViT	Vicuna-v1.5-7B	Vicuna-v1.5-7B	Qwen-7B	Vicuna-v1.5-7B
LLM	CLIP-ViT-L/14	CLIP-ViT-L/14	ViT-G/16	CLIP-ViT-L/14
Connector	MLP	MLP	CrossAttn	MLP
Optimizer	AdamW			
Connector learning rate	2e-5	2e-5	1e-5	2e-5
Learning rate schedule	cosine decay			
Warm-up ratio	0.03	0.03	0.01	0.03
Weight decay	0.0	0.0	0.1	0.0
Global batch size	128	128	128	128
Gradient Acc	1	1	1	1
Training epoch	3			
Numerical precision	bfloat16			

Table 11: Training hyperparameters when we use PEFT methods to fine-tune those models.

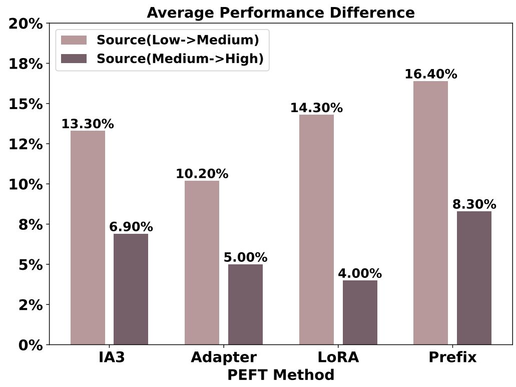


Figure 6: Average performance difference for different PEFT methods in various data-scaling settings. Source (Low->Medium): fine-tuned dataset scaling change from low-resource to high-resource. Source (Medium->High): fine-tuned dataset scaling change from medium-resource to high-resource.

N_{layer} is the number of model's layers, and d_{model} represents the dimension of the residual stream.

I Case Study

I.1 Hallucination Analysis

In this section, we explain how we tested the model's hallucination. As the first step, we employed LLaVA-1.5-7B to generate captions for all images in the Flickr30k test set in a zero-shot manner. Subsequently, we randomly sampled 1k captions and manually curated 100 correct captions without hallucinations. Then, we utilized the LLaVA-1.5-7B model fine-tuned with four PEFT methods on IconQA-txt to generate captions for these 100 samples. Thereafter, we manually annotate the fine-tuned model-generated outputs and count the number of hallucination samples.

One sample is selected, as illustrated in Figure 9. In this example, the original LLaVA model delivers a hallucination-free description, accurately identifying the color and actions depicted in the image. On the other hand, the Adapter model incorrectly identifies the girl's face as red. The IA3 model inaccurately attributes a mustache to the girl and misidentifies her hair color as red. The LoRA model also fails to recognize the girl's hand and refers to a red substance, which is not present. As for the Prefix model, it attempts to provide a detailed description of the picture, including the mention of a ponytail and attributing emotions such as a "funny or playful gesture" to the subject. How-

Model	Method	SQA (img)	VizWiz	OKVQA	OCRVQA	Avg
LLaVA-1.5-7B	Adapter	84.4	**67.6**	59.8	65.2	**69.3**
	-w/ Visual Encoder	84.2	67.1	60.5	65.1	69.2
	LoRA	**86.2**	66.5	56.5	66.7	69.0
	-w/ Visual Encoder	85.9	66.7	55.9	66.8	68.8
	IA3	82.7	61.9	60.5	67.1	68.1
	-w/ Visual Encoder	83.4	62.2	60.3	66.8	68.2
	Prefix	68.2	60.8	**61.3**	68.5	64.7
	-w/ Visual Encoder	65.9	62.1	60.8	**68.7**	64.4

Table 12: Results of tuning visual encoder on LLaVA-1.5-7B with four PEFT methods. w/ Visual Encoder: Tuning the Visual Encoder.

Method	Location	SQA (img)	VizWiz	IconQA-txt	IconQA-blank	OKVQA	OCRVQA	VQAv2	Avg
Qwen-VL-Chat									
Adapter	Attn	85.2	64.5	**92.2**	88.3	49.3	**71.7**	63.9	73.6
	MLP	81.2	69.3	90.8	87.5	51.1	69.3	70.7	**74.3**
	Both	**85.6**	67.1	91.8	**90.5**	50.7	55.9	69.2	73.0
LoRA	Attn	80.2	67.3	80.5	87.1	43.3	69.6	39.1	66.7
	MLP	74.5	60.7	78.5	88.5	43.6	67.8	60.1	67.7
	Both	84.0	68.8	71.9	83.5	43.5	67.0	63.3	**68.9**
IA3	Attn	66.4	69.1	58.2	21.7	50.8	63.3	77.3	58.1
	MLP	62.8	68.3	59.8	23.1	**51.3**	62.7	**77.6**	57.9
	Both	67.3	**69.8**	57.3	28.7	50.5	62.1	77.5	**59.0**

Table 13: Average results of PEFT module location on Qwen-VL-Chat.

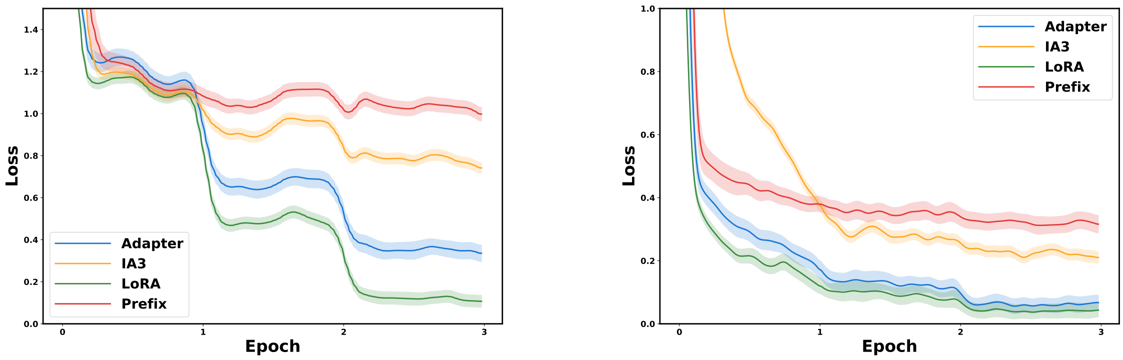
		OKVQA	SQA (img)
Adapter	learning rate=2e-4	$54.4_{\pm 0.44}$	$81.2_{\pm 1.27}$
	learning rate=5e-5	$58.9_{\pm 0.81}$	$81.3_{\pm 1.85}$
	learning rate=1e-5	$60.3_{\pm 0.10}$	$72.4_{\pm 0.25}$
	learning rate=5e-6	$58.5_{\pm 0.35}$	$82.2_{\pm 1.31}$
LoRA	learning rate=2e-4	$56.7_{\pm 0.40}$	$86.0_{\pm 0.35}$
	learning rate=5e-5	$59.8_{\pm 0.10}$	$83.1_{\pm 0.45}$
	learning rate=1e-5	$62.9_{\pm 0.06}$	$74.0_{\pm 2.71}$
	learning rate=5e-6	$61.6_{\pm 0.06}$	$71.2_{\pm 1.64}$
IA3	learning rate=2e-4	$62.7_{\pm 0.98}$	$81.7_{\pm 0.87}$
	learning rate=5e-5	$61.4_{\pm 4.89}$	$77.1_{\pm 2.54}$
	learning rate=1e-5	$62.9_{\pm 0.20}$	$72.2_{\pm 1.28}$
	learning rate=5e-6	$58.8_{\pm 0.06}$	$70.8_{\pm 0.64}$
Prefix	learning rate=2e-4	$60.9_{\pm 0.21}$	$35.3_{\pm 0.85}$
	learning rate=5e-5	$59.6_{\pm 0.06}$	$64.6_{\pm 0.01}$
	learning rate=1e-5	$61.5_{\pm 0.06}$	$72.3_{\pm 0.21}$
	learning rate=5e-6	$60.8_{\pm 0.01}$	$70.2_{\pm 0.01}$

Table 14: Performance on OKVQA and SQA (img) datasets with different training learning rate of PEFT module.

ever, these emotions cannot be confirmed simply by viewing the picture. Additionally, the Prefix model presents more severe hallucinations compared to the others, as it "imagines" another hand that is not visible in the image. More examples at several epochs are illustrated in this section.

I.2 Qualitative Illustrations

In this section, We randomly sampled several examples from each dataset and provided the original labels along with the outputs of various PEFT models. See Figures 14 to 23.



(a) Train Loss of OKVQA

(b) Train Loss of SQA

Figure 7: Train loss reported across three runs with different random seeds. The line plotted the mean of three seeds, where the shaded region represents its 95% confidence interval.

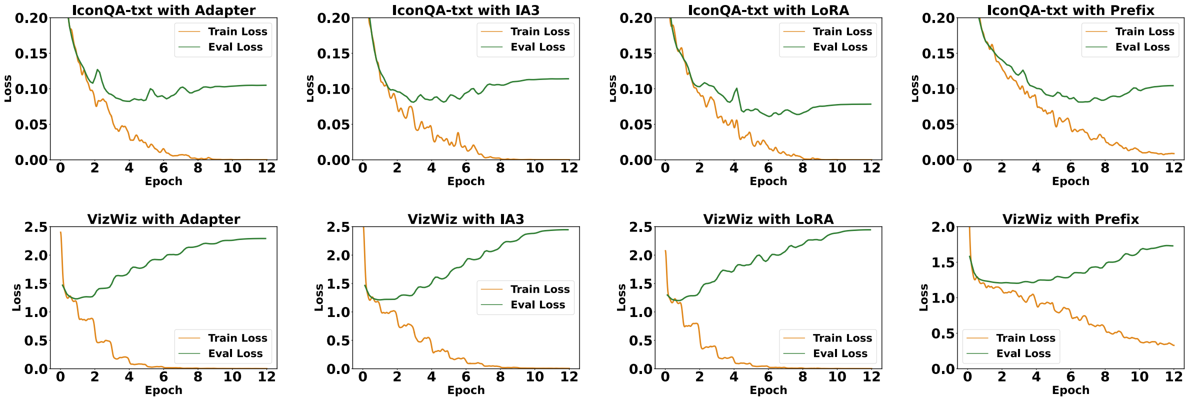


Figure 8: Train-Loss curve on IconQA-txt and Vizwiz. The orange line shows Train Loss. Eval loss is colored with green.

Model	Method	Trainable parameters	Train flops	Inference flops	Performance
LLaVA-v1.5$_{7B}$	Adapter	2.771%	1.526e+15	4.105e+10	74.6
	-w/ connector	3.060%	1.532e+15	4.109e+10	76.3
	LoRA	4.332%	1.552e+15	4.092e+10	76.1
	-w/ connector	4.616%	1.561e+15	4.099e+10	76.7
	IA3	0.009%	1.424e+15	4.028e+10	63.1
	-w/ connector	0.306%	1.248e+15	3.695e+10	76.3
	Prefix	0.074%	1.496e+15	4.159e+10	68.1
	-w/ connector	0.371%	1.498e+15	4.154e+10	69.6
LLaVA-v1.5$_{13B}$	Adapter	2.301%	2.898e+15	6.946e+10	77.2
	-w/ connector	2.532%	2.908e+15	6.951e+10	77.7
	LoRA	3.615%	2.947e+15	6.942e+10	78.3
	-w/ connector	3.842%	3.083e+15	7.127e+10	77.8
	IA3	0.007%	2.660e+15	6.716e+10	65.0
	-w/ connector	0.243%	2.634e+15	6.667e+10	**78.5**
	Prefix	0.061%	2.741e+15	6.838e+10	67.9
	-w/ connector	0.297%	2.749e+15	6.840e+10	69.1

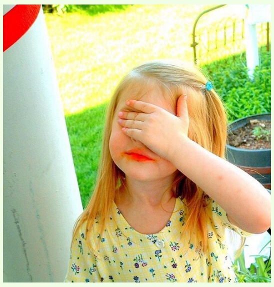
Table 15: Efficient comparison across all types of settings. The number of trainable parameters, PEFT Flops, and Inference Flops are analyzed, considering whether to tune the connector with different PEFT methods on two models with different parameters. w/ connector: Tuning the connector.

Prompt:

Render a clear and concise summary of the photo.

Label:

A blond girl in a yellow shirt with blue flowers is covering her face.
A blond girl covering her eyes with her hand in yellow dress.
A little girl in a yellow shirt is standing covering her eyes.
A small girl covering her eyes in a yellow shirt , outdoors.
Little girl covering her eyes!



LLaVA-1.5-7B:

A blond girl in a yellow shirt is covering her face.

LLaVA-1.5-7B-Adapter:

A little girl with a red face is covering her eyes with her hand.

LLaVA-1.5-7B-IA3:

A girl with red hair and a red mustache.

LLaVA-1.5-7B-LoRA:

A young girl with a red substance on her face, covering her eyes.

LLaVA-1.5-7B-Prefix:

A little girl with a ponytail is playfully hiding her face with one hand while the other hand is holding her nose, possibly making a funny or playful gesture.

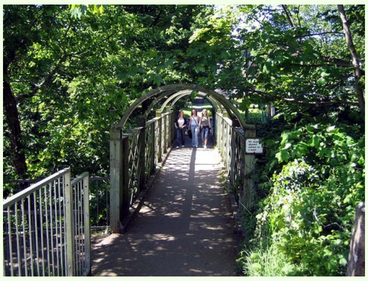
Figure 9: A qualitative hallucination example generated by LLaVA-1.5-7B and LLaVA-1.5-7B with various PEFT methods. LLaVA-1.5-7B generated hallucination-free text. After fine-tuning, all models generated hallucinations.

Prompt:

Render a clear and concise summary of the photo.

Label:

Three girls are walking across a bridge covered with wood and surrounded by trees.
Three girls walk under an arbor on a path surrounded by trees.
Three women walk through a plant covered atrium hallway.
Three girls walking along an arched garden pathway.
Three women walking through a tunnel in a park.



LLaVA-1.5-7B:

Three girls walking along an arched garden pathway.

LLaVA-1.5-7B-Adapter:

Three people are standing under a bridge, enjoying the shade it provides.

LLaVA-1.5-7B-IA3:

Three people standing under a bridge.

LLaVA-1.5-7B-LoRA:

Three people standing in a circular walkway.

LLaVA-1.5-7B-Prefix:

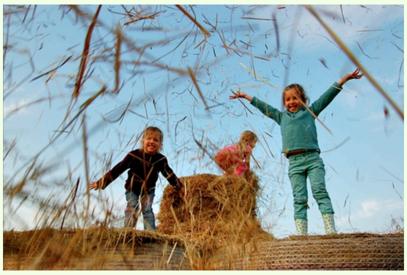
Three people walking on a bridge.

Prompt:

Offer a succinct explanation of the picture presented.

Label:

Three children playing with hay outdoors, throwing some at camera lens.
Three young children play in the hay, throwing some at the camera.
Two young children are throwing hay.
Three children play on corn stacks.
Three children playing in hay.



LLaVA-1.5-7B:

Three young children play in the hay , throwing some at the camera.

LLaVA-1.5-7B-Adapter:

Two young children playing in the hay

LLaVA-1.5-7B-IA3:

Two children playing in hay, one girl is jumping, the other is smiling.

LLaVA-1.5-7B-LoRA:

Two children are standing on a pile of hay, one of them is raising her arm in the air.

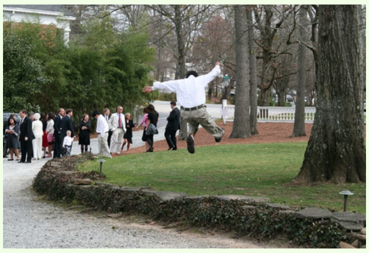
LLaVA-1.5-7B-Prefix:

The image shows two little girls standing on a pile of hay. They seem to be enjoying their time together, possibly playing or celebrating.

Figure 10: A randomly sampled hallucination results (from Flickr30K) of various PEFT methods' models trained on the IconQA-txt dataset for 3 epochs.

Prompt:

Write a terse but informative summary of the picture.



Label:

A group of people in formal attire are in the background and a man in a white shirt with khaki pants jumps from the dirt road to the green grass.
Someone is getting some serious air while the more formally dressed group gathers in the background.
A guy is jumping in the air while a group of people stand around and ignore him.
Man started jumping because the sprinklers were about to go off.
Man jumping over a small stone wall in front of a group of people.

LLaVA-1.5-7B:

Man started jumping because the sprinklers were about to go off.

LLaVA-1.5-7B-Adapter:

A man jumps in the air in a park.

LLaVA-1.5-7B-IA3:

A man jumping in air.

LLaVA-1.5-7B-LoRA:

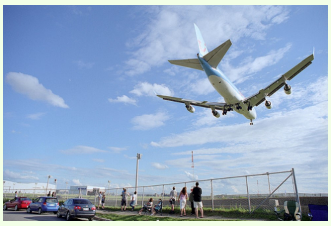
A man is jumping in the air, while a group of people watches.

LLaVA-1.5-7B-Prefix:

A man is jumping in the air in front of a group of people who are watching him.

Prompt:

Render a clear and concise summary of the photo.



Label:

Silver plane in a blue sky , ready to land with its wheels down , while spectators watch behind a high fence.
An airplane flying over a fence with a few bystanders watching it as it approaches its landing.
Group of people stand near a fence watching a plane fly over them.
People watch as a plane flies over their heads at an airfield.
A group of people are watching a plane flying low.

LLaVA-1.5-7B:

An airplane flying over a fence with a few bystanders watching it as it approaches its landing.

LLaVA-1.5-7B-Adapter:

A large commercial airplane is flying in the air above a fence with people sitting on the ground.

LLaVA-1.5-7B-IA3:

A plane flying in the sky.

LLaVA-1.5-7B-LoRA:

A plane is flying in the air, with people watching from below.

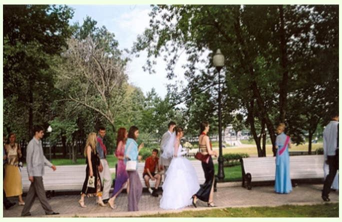
LLaVA-1.5-7B-Prefix:

A large commercial airplane is flying in the sky with its landing gear down, as people watch it from the ground.

Figure 11: A randomly sampled hallucination results (from Flickr30K) of various PEFT methods' models trained on the IconQA-txt dataset for 6 epochs.

Prompt:

Provide a brief description of the given image.



Label:

A group of people , a wedding party including a bride and groom, are walking down a path lined by benches and surrounded by grass and trees.
Several members of a wedding party , including the bride , are walking past a lamp and some park benches.
A bride and a groom walk down a sidewalk with other guests surrounding them.
The people are following the Bride and the Groom at a wedding.
A wedding day with the bride in a bright white beautiful dress.

LLaVA-1.5-7B:

The people are following the Bride and the Groom at a wedding.

LLaVA-1.5-7B-Adapter:

A group of people are walking down a sidewalk, with some of them in wedding dresses.

LLaVA-1.5-7B-IA3:

A group of people walking down a sidewalk.

LLaVA-1.5-7B-LoRA:

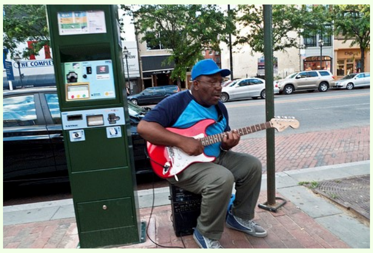
A group of people walking down a path, with a bride in a white dress and a man in a blue suit.

LLaVA-1.5-7B-Prefix:

A group of people walking down a sidewalk.

Prompt:

Render a clear and concise summary of the photo



Label:

An older African-American male is playing an electric guitar in the public street.
An older black man plays an electric guitar on a city street.
An older black man plays an electric guitar on the sidewalk.
An older man is playing a red guitar on the street.
A man trying to play guitar for entertainment.

LLaVA-1.5-7B:

An older man is playing red guitar on the street.

LLaVA-1.5-7B-Adapter:

A man is playing red guitar.

LLaVA-1.5-7B-IA3:

An older man is playing guitar.

LLaVA-1.5-7B-LoRA:

A man is sitting on a street corner playing a guitar.

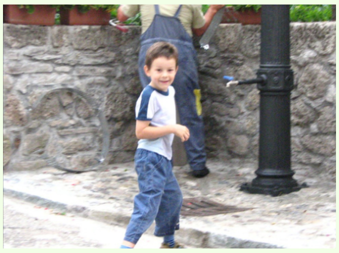
LLaVA-1.5-7B-Prefix:

A man is sitting on a stool on a city street corner, playing a guitar.

Figure 12: A randomly sampled hallucination results (from Flickr30K) of various PEFT methods' models trained on the IconQA-txt dataset for 9 epochs.

Prompt:

Offer a succinct explanation of the picture presented.



Label:

Smiling boy in white shirt and blue jeans in front of rock wall with man in overalls behind him.
A little boy is standing on the street while a man in overalls is working on a stone wall.
A young child is walking on a stone paved street with a metal pole and a man behind him.
A boy smiles in front of a stony wall in a city.
A young boy runs aross the street.

LLaVA-1.5-7B:

A little boy is standing on the street while a man in overalls is working on a stone wall.

LLaVA-1.5-7B-Adapter:

A young boy is walking on a sidewalk, smiling at the camera.

LLaVA-1.5-7B-IA3:

A boy in a white shirt running.

LLaVA-1.5-7B-LoRA:

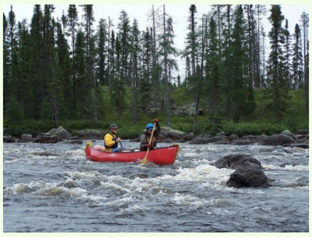
A young boy is walking down the street with a man in a blue shirt and a black lamp post.

LLaVA-1.5-7B-Prefix:

A young boy in a white shirt and blue jeans walking down the street.

Prompt:

Render a clear and concise summary of the photo.



Label:

Two paddlers in a red canoe approach a couple of large rocks in the rushing river with conifers in the background.
Two people paddle a canoe down a rough stretch of river near a pine forest.
A couple in a red canoe paddle down a rough river lined with trees.
Two people canoeing down a river in a red canoe.
Two people are in a canoe.

LLaVA-1.5-7B:

Two people paddle a canoe down a rough stretch of river near a pine forest.

LLaVA-1.5-7B-Adapter:

Two people in a red boat on a river.

LLaVA-1.5-7B-IA3:

Two people in a red and yellow boat on a river.

LLaVA-1.5-7B-LoRA:

A man in a red canoe is navigating through rough waters near rocks.

LLaVA-1.5-7B-Prefix:

A man and a woman in a red canoe.

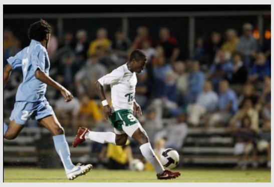
Figure 13: A randomly sampled hallucination results (from Flickr30K) of various PEFT methods' models trained on the IconQA-txt dataset for 12 epochs.

Prompt:

Relay a brief, clear account of the picture shown.

Label:

A football player dressed in green and white advances the ball while watched by a defender in blue .



LLaVA-1.5-13B-Adapter:

A soccer player in a blue uniform is chasing a player in a white uniform who is kicking the soccer ball.

- w/ connector:

Two soccer players, one in blue and one in white, are running after the ball

LLaVA-1.5-13B-IA3:

A soccer player in a green and white uniform kicks the ball.

- w/ connector:

Two soccer players are playing a game in front of a crowd

LLaVA-1.5-13B-LoRA:

A soccer player in a green and white uniform kicks the ball while a player in a light blue uniform tries to stop him.

- w/ connector:

Two soccer players, one in light blue and one in white, playing soccer in front of a crowd.

LLaVA-1.5-13B-Prefix:

A soccer player in a white shirt and green shorts kicks the ball.

- w/ connector:

A soccer player in a white shirt and green shorts kicks the ball

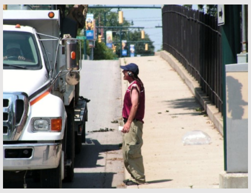
Figure 14: An example randomly chosen from the Flickr30K dataset. The outcomes produced by LLaVA-1.5-13B using different PEFT methods, each with the connector fine-tuned and frozen.

Prompt:

What company does this truck work for?

Label:

1.waste management, 2.waste management, 3.waste management, 4.waste management, 5.garbate, 6.garbate, 7.tow company, 8.tow company, 9.up, 10.up



LLaVA-1.5-13B-Adapter:

construction

- w/ connector:

construction

LLaVA-1.5-13B-IA3:

12

- w/ connector:

Tow company

LLaVA-1.5-13B-LoRA:

garbage

- w/ connector:

city

LLaVA-1.5-13B-Prefix:

waste management

- w/ connector:

city

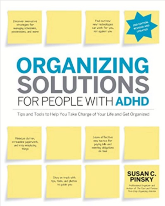
Figure 15: An example randomly chosen from the OKVQA dataset. The outcomes produced by LLaVA-1.5-13B using different PEFT methods, each with the connector fine-tuned and frozen.

Prompt:

What is the genre of this book?

Label:

Parenting & Relationships



LLaVA-1.5-13B-Adapter:

Health, Fitness & Dieting

- w/ connector:

Health, Fitness & Dieting

LLaVA-1.5-13B-IA3:

Health, Fitness & Dieting

- w/ connector:

Health, Fitness & Dieting

LLaVA-1.5-13B-LoRA:

Health, Fitness & Dieting

- w/ connector:

Health, Fitness & Dieting

LLaVA-1.5-13B-Prefix:

Parenting & Relationships

- w/ connector:

Parenting & Relationships

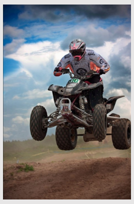
Figure 16: An example randomly chosen from the OCRVQA dataset. The outcomes produced by LLaVA-1.5-13B using different PEFT methods, each with the connector fine-tuned and frozen.

Prompt:

Render a clear and concise summary of the photo.

Label:

The man does a trick on the fourwheeler..



LLaVA-1.5-13B-Adapter:

A person is riding a four wheeler vehicle.

- w/ connector:

A person is riding a four wheeler vehicle.

LLaVA-1.5-13B-IA3:

A man on a dirt bike.

- w/ connector:

A man is riding a dirt bike

LLaVA-1.5-13B-LoRA:

A person on a four wheeler

- w/ connector:

A person jumping on a four wheeler.

LLaVA-1.5-13B-Prefix:

A man on a dirt bike jumping in the air.

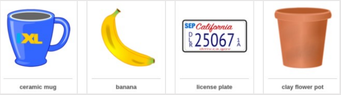
- w/ connector:

A man on a dirt bike jumping in the air.

Figure 17: An example randomly chosen from the Flickr30K dataset. The outcomes produced by LLaVA-1.5-13B using different PEFT methods, each with the connector fine-tuned and frozen.

Prompt:

Which property do these four objects have in common?
A. Opaque B. Salty C. Slippery



Label:

A

LLaVA-1.5-13B-Adapter:

A

- w/ connector:

A

LLaVA-1.5-13B-IA3:

A

- w/ connector:

A

LLaVA-1.5-13B-LoRA:

A

- w/ connector:

A

LLaVA-1.5-13B-Prefix:

A

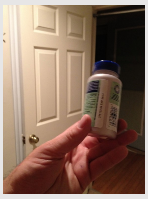
- w/ connector:

A

Figure 18: An example randomly chosen from the SQA (img) dataset. The outcomes produced by LLaVA-1.5-13B using different PEFT methods, each with the connector fine-tuned and frozen.

Prompt:

Can you tell me what pill this is? When the provided information is insufficient, respond with 'Unanswerable'.



Label:

unanswerable

LLaVA-1.5-13B-Adapter:

unanswerable

- w/ connector:

unanswerable

LLaVA-1.5-13B-IA3:

unanswerable

- w/ connector:

unanswerable

LLaVA-1.5-13B-LoRA:

aspirin

- w/ connector:

unanswerable

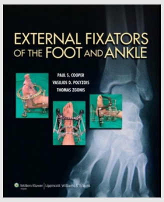
LLaVA-1.5-13B-Prefix:

unanswerable

- w/ connector:

unanswerable

Figure 19: An example randomly chosen from the VizWiz dataset. The outcomes produced by LLaVA-1.5-13B using different PEFT methods, each with the connector fine-tuned and frozen.



Prompt:
What is the title of this book? Answer the question using a single word or phrase.

Label:
External Fixators of the Foot and Ankle

LLaVA-1.5-13B-Adapter:
External Fixators of the Foot and Ankle

- w/ connector:
External Fixators of the Foot and Ankle

LLaVA-1.5-13B-IA3:
External Fixators of the Foot and Ankle

- w/ connector:
External Fixators of the Foot and Ankle

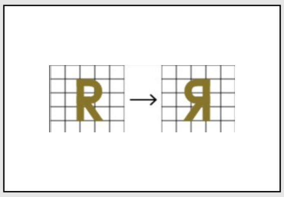
LLaVA-1.5-13B-LoRA:
External Fixators of the Foot and Ankle

- w/ connector:
External Fixators of the Foot and Ankle

LLaVA-1.5-13B-Prefix:
External Fixators of the Foot and Ankle

- w/ connector:
External Fixators of the Foot and Ankle

Figure 20: An example randomly chosen from the OCRVQA dataset. The outcomes produced by LLaVA-1.5-13B using different PEFT methods, each with the connector fine-tuned and frozen.



Prompt:
What has been done to this letter?
Choices: turn, slide, flip.
Choose an option from choices to answer the question.

Label:
flip

LLaVA-1.5-13B-Adapter:
flip

- w/ connector:
flip

LLaVA-1.5-13B-IA3:
slide

- w/ connector:
flip

LLaVA-1.5-13B-LoRA:
turn

- w/ connector:
flip

LLaVA-1.5-13B-Prefix:
flip

- w/ connector:
flip

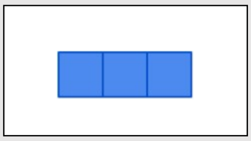
Figure 21: An example randomly chosen from the IconQA-txt dataset. The outcomes produced by LLaVA-1.5-13B using different PEFT methods, each with the connector fine-tuned and frozen.

Prompt:

The shape is made of unit squares. What is the area of the shape? Fill in the blanks in (_) or answer this question.

Label:

3



LLaVA-1.5-13B-Adapter:

3

- w/ connector:

3

LLaVA-1.5-13B-IA3:

12

- w/ connector:

3

LLaVA-1.5-13B-LoRA:

3

- w/ connector:

4

LLaVA-1.5-13B-Prefix:

4

- w/ connector:

4

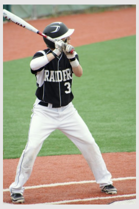
Figure 22: An example randomly chosen from the IconQA-blank dataset. The outcomes produced by LLaVA-1.5-13B using different PEFT methods, each with the connector fine-tuned and frozen.

Prompt:

Is the player right-handed? Answer the question using a single word or phrase.

Label:

1.yes, 2. yes, 3. no, 4. yes, 5. yes,
6. yes, 7. yes, 8. yes, 9. yes, 10. yes



LLaVA-1.5-13B-Adapter:

no

- w/ connector:

no

LLaVA-1.5-13B-IA3:

yes

- w/ connector:

yes

LLaVA-1.5-13B-LoRA:

no

- w/ connector:

yes

LLaVA-1.5-13B-Prefix:

yes

- w/ connector:

yes

Figure 23: An example randomly chosen from the VQAv2 dataset. The outcomes produced by LLaVA-1.5-13B using different PEFT methods, each with the connector fine-tuned and frozen.