Multimodal Large Language Model Fine-tuning

Eric Wang
University of Waterloo
e246wang@uwaterloo.ca

Shengye Chen
University of Waterloo
j57chen@uwaterloo.ca

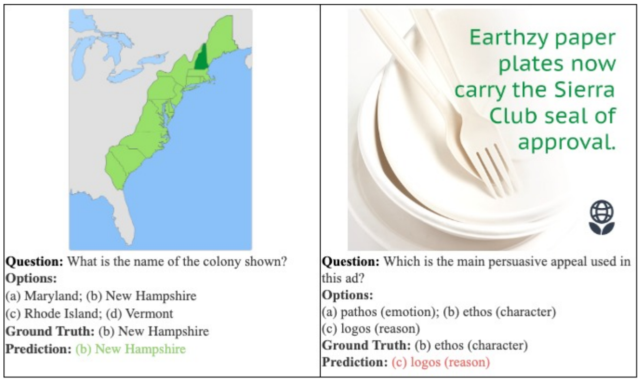
Abstract

Recent SOTA Multimodal Large Language Models (MLLMs) have seen outstanding success in multimodal content understanding and fusion, thanks to the combination of Vision-Language Models (VLMs) and Large Language Models (LLMs). However, despite the impressive zero-shot performance on some general Vision-Language (VL) tasks, their performance on the downstream tasks that require much domain knowledge is still imperfect. In this project, leveraging Low-Rank Adaptation of Large Language Models (LoRA) with limited number of learnable parameters, we fine-tuned InstructBLIP for ScienceQA and IconQA datasets to evaluate the model's capability for multimodal understanding, and achieved superb accuracy comparable to full fine-tuning with larger LLM backbone. We further investigated which component inside the model architecture is the most fine-tuning effective to the multimodal reasoning performance through a comprehensive ablation study by fine-tuning different components and freezing the backbone. We additionally proposed some empirical tricks for fine-tuning MLLMs using Parameter-Efficient Fine-Tuning (PEFT) methods to achieve better performance in terms of higher accuracy, faster training speed and lower memory utilization. We achieved the highest fine-tuning accuracy of 87.3 on the ScienceQA dataset and 75.3 on the IconQA dataset.

1 Introduction

Multimodal learning is an important branch of deep learning, aiming at using neural networks to process multiple modal data at the same time, including images and text understanding, image-to-text generation, and text-to-image generation. Particularly, the emergence of Transformer architecture has led to a great amount of pre-training methods, by jointly training visual and language models, to better learn cross modal representations and improve the performance on multimodal tasks. Recently, the state-of-the-art MLLMs have achieved outstanding success in multimodal content understanding and generation. However, despite the impressive zero-shot performance on some general VL tasks, their performance on some downstream tasks and dataset that require much domain knowledge about some specific fields is still imperfect. In this project, we fine-tuned a MLLM, InstructBLIP [4], for ScienceQA [21] and IconQA [20] datasets, to evaluate InstructBLIP's capability that utilizes the information available across different modalities to synthesize consistent and complete answers. We further investigated which component inside the model architecture contributes the most to the multimodal reasoning performance through a comprehensive ablation study by fine-tuning different components and freezing the backbone. We additionally proposed some empirical tricks for fine-tuning MLLMs using LoRA [8] to achieve better performance in terms of higher accuracy, faster training speed and lower memory utilization. Finally we achieved superb accuracy comparable to full fine-tuning with larger LLM backbone. We believe this project is an excellent case study of fine-tuning MLLMs for downstream task and can be an appropriate outset for those who intend to perform multimodal tasks that require specific domain knowledge.

In the field of multimodal information fusion, some early VLMs, such as ViLBERT [19] and Uniter [2], proposed the introduction of Object-Text module into the model to improve the model's ability to


Earthzy paper plates now carry the Sierra Club seal of approval.

Question: What is the name of the colony shown?
Options:
(a) Maryland; (b) New Hampshire
(c) Rhode Island; (d) Vermont
Ground Truth: (b) New Hampshire
Prediction: (b) New Hampshire

Question: Which is the main persuasive appeal used in this ad?
Options:
(a) pathos (emotion); (b) ethos (character)
(c) logos (reason)
Ground Truth: (b) ethos (character)
Prediction: (c) logos (reason)

Figure 1: **Two test question examples.** The best performing model answers the left one correctly and fails at the right one.

understand visual information. The introduction of the Object module inevitably requires a bulky detector to detect objects with bounding boxes, making the image modality of the multimodal model clumsy and difficult to adjust. In these methods, multimodal information processing leans more towards the understanding of images while the understanding of text is more of an auxiliary for image understanding tasks. To better unify multimodal semantics between visual representation and text representation, Oscar [16] hires anchor point to bridge visual information and text information and VinVL [28] further employs finer object detection model trained with dataset including additional attribute data to better obtain visual information descriptions, hence increasing the performance on a variety of downstream tasks despite the existence of the frozen detection model.

CLIP [24] firstly demonstrated that solely training both image encoder and text encoder with contrastive loss on large scale dataset collected from the Internet can achieve outstanding performance on many VL tasks. ALIGN [9] further showed that the scale of the training corpus can make up for its noise and leads to state-of-the-art representations even with simple contrastive learning. Thus mainstream VLMs, including ViLT [10], ALBEF [12], VLMo [1] and BLIP [14], tend to adopt larger network architecture without object detectors and perform training with raw data without human annotations, which is mainly attributed to the success of ViT [5] in Computer Vision (CV) field, making it possible to directly fuse visual representation and text representation through Transformer in the fashion of two towers to exact visual and test representations respectively. In these works text modality and visual modality are almost equally important and the performance of these methods on various downstream tasks, including Visual Question Answering (VQA), Visual Grounding (VG) and Image-Text Retrieval (ITR), is considerably desired. Given the breakthrough progress in the cross modal representation fusion, the community is now focusing more on developing VLMs' complicated reasoning capability and the open-ended understanding and generation capability for cross modal information by introducing Large Language Models (LLMs) into VLMs to build MLLMs.

ImageBind [6], LanguageBind [29] and NExT-GPT [26] are state-of-the-art MLLMs that have done outstanding work in multimodal content understanding and generation, so they should have been our first choices for the backbone model of this project. However, given the limited time and compute, we initialized this work with a smaller model, InstructBLIP, to fine-tune and perform multimodal downstream tasks. Empirically, InstructBLIP is excellent work that transfers instruction tuning technique, widely used in LLMs, to MLLM, so we would naturally expect InstructBLIP's superior performance on the harder downstream tasks, such as ScienceQA, compared to previous VLMs and MLLMs like BLIP [14] and BLIP-2 [13], as the effectiveness of the instruction-aware query transformer architecture makes up for the shortcomings of previous work in visual feature extraction. Furthermore, speaking of coding, the BLIP family has a clear development path from BLIP's multi-task alignment of image and text, BLIP-2's combination of LLMs to InstructBLIP's integrated instruction tuning technique. The LAVIS library [11] has well encapsulated the three models and reused much common modules, so the source code is highly readable and effective for us to work on the fine-tuning for downstream tasks. Our code is adapted from [11] [25] [22] [27].

In terms of downstream tasks, we chose ScienceQA and IconQA, which are much more difficult than the traditional VQA, VG and ITR tasks, to evaluate InstrcutBLIP through fine tuning. For example, ScienceQA contains multiple-choice questions from grade 1 to grade 12 courses, covering natural science, language science and social science. A typical question involves multimodal context, correct options, general background knowledge and specific explanation [21]. Full fine-tuning an MLLM as large as InstructBLIP requires excessive time and compute, while LoRA shows the ability to achieve better fine-tuning performance with fewer parameters than other PEFT methods, such as Adapter Tuning [7], Prefix-Tuning [15] or P-tuning [17]. Hence we hired LoRA as the fine-tuning method for ScienceQA and IconQA tasks. The detailed analysis into the increased performance brought by fune-tuning using LoRA can be found in Sections 4 and 5. Our code is available at `https://github.com/jacky1c/CS886FoundationModels`.

2 Problem Definition

Table 1: **Experiment results of InstructBLIP on zero-shot VL tasks [4].** Only a subset of results is shown in this table to show the limitation of InstructBLIP on specific domain tasks. Refer to [4] for complete results.

	NoCaps	Flickr 30 K	GQA	VSR	IconQA	SciQA image
InstructBLIP (FlanT5$_{\mathrm{XL}}$)	119.9	84.5	48.4	64.8	50.0	70.4

Given an image, MLLMs tend to extract important visual representation from a frozen image encoder and then hire an LLM as the generative language module to generate natural language signals based on the visual representation. Since the LLM used in MLLMs is usually frozen, in spite of having been pre-trained with large scale corpus, the limited performance of the used LLM on specific domain tasks and dataset is always reflected on the performance of MLLMs on similar tasks. As shown in Table 1, the experiment results of InstructBLIP with FlanT5$_{\mathrm{XL}}$ [3] as the backbone LLM on zero-shot ScienceQA and IconQA are 70.4 and 50.0 respectively, which necessitates the prior learning for the specific domain knowledge when MLLMs deal with these finer tasks.

Furthermore, fine-tuning an MLLMs for a specific downstream task is expected to increase the model's performance considerably, but full fine-tuning is a laborious work and the fine-tuned model cannot be transferred to other tasks, which makes it valuable to quickly fine-tune certain components of a model to achieve optimal performance on the task. However, Transformer is a complex architecture with many neural network layers each with a large number of learnable parameters. For example, just within the image tower of InstructBLIP Q-Former there exist 12 stacked encoders interacting with the text tower and the frozen image encoder, which makes it even harder to figure out which components are the most fine-tuning effective to the performance enhancement, not to mention the LLM module. Nevertheless, the reasoning capability of MLLMs is generally stemmed from the LLM side, therefore fine-tuning LLM is an essential step for the MLLM to perform optimally on downstream tasks.

Targeting to solve this intricate problem, our aim lies to propose an effectively and efficiently scalable fine-tuning strategy for MLLMs to quickly adapt to downstream tasks with higher accuracy, faster training speed and lower memory utilization. The detailed method is described in Section 3.2.

3 Method

In this section, we briely introduce the architecture and pre-trained methods of InstructBLIP and enunciate our strategy to fine-tune InstructBLIP.

3.1 InstructBLIP

BLIP-2, the previous version and integral part of InstructBLIP, is a typical work that introduces an LLM into the model architecture to shape MLLM the complicated image-based reasoning capability by leveraging the emergent ability of the used LLM. There are two different stages in BLIP-2 pre-training. The first one is vision-language representation learning stage. Specifically, in the Q-Former module with 12 stacked encoder, there are several learnable query embeddings as input to the image transformer to extract visual representation from the frozen image encoder. The learnable queries interact with each other and the text embeddings from text transformer through self-attention block,

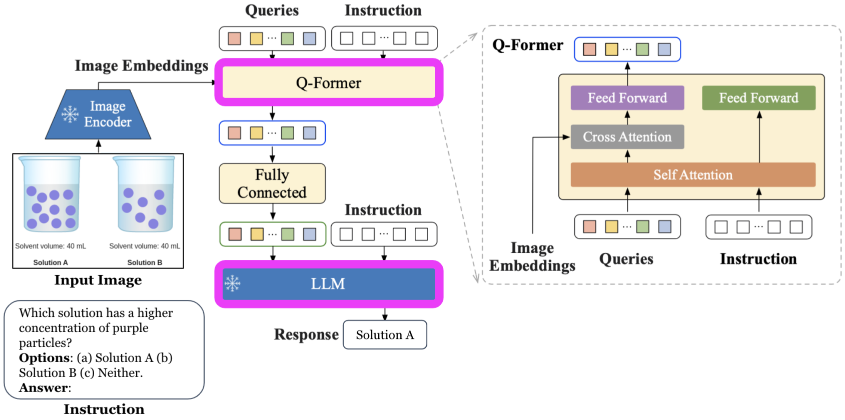


Figure 2: Model architecture of InstructBLIP [4].

and interact with the image features through cross-attention block. The second pre-training stage of BLIP-2 is vision-to-language generative learning stage where Q-Former is connected to the frozen LLM to utilize its generative language capability. Essentially, BLIP-2 employs Q-Former to fuse visual semantics and the generation capabilities of LLM. The only difference between InstructBLIP and BLIP-2 is that InstructBLIP has an additional pre-training stage 3, instruction tuning. After BLIP-2 pre-training, the instruction text is added to the Q-Former input and LLM input respectively to train the Q-Former parameters again so that Q-Former extracts instruction-aware visual representations and feeds the more flexible visual representations as soft prompt input to the LLM, as shown in Figure 2. Briefly, the visual representations of BLIP-2 are static, directly output from the image encoder, while, given different instructions, InstructBLIP can extract different representations from the same image for different vision-language tasks.

3.2 Fine-Tuning

We use LoRA to fine-tune different components of InstructBLIP for downstream tasks. LoRA optimizes the rank decomposition matrices of some certain dense layers during fine-tuning to indirectly train these dense layers in the neural network while keeping the pre-trained weights unchanged. Particularly, the image encoder is believed to have been well-trained to encode visual features given the impressive performance of InstructBLIP on general VL tasks so we leave the image encoder unchanged in our fine-tuning strategy.

We primarily focus on fine-tuning the Q-Former module and the LLM module of InstructBLIP as shown in the purple boxes of Figure 2. The former extracts and passes the instruction-aware visual representation to the LLM and the later keeps frozen in the entire pre-training stage, therefore we would expect that fine-tuning these two modules will achieve superb performance increase once their parameters are adapted to the domain knowledge of the downstream tasks. We further separate the Q-Former and LLM as different sub-component, like Q-Former's own self-attention block, cross-attention block, feedforward netwwork (FFN), etc., to perform experiments with finer granularity and different rank values for the comprehensive ablation study, which is described in Section 4.3. For example, if we fine-tune the FFN of Q-Former, the parameters of FFN within each out of 12 Transformers will be trained and all other layers and parameters of InstructBLIP are frozen. In this way, the experiment result will demonstrate an obvious signal as to if this specific component is fine-tuning effective to the performance increase, without contribution from other components at all.

4 Experiments

4.1 Datasets

We chose publicly available ScienceQA and IconQA dataset to fine-tune InstructBLIP. These dataset were not included in the pre-training process for InstructBLIP, and, more importantly, pre-training dataset of InstructBLIP were selected carefully so that there is no data contamination. In addition, initial tests on these two dataset revealed poor performance in zero-shot settings. Employing the FlanT5$_{\text{XXL}}$ backbone, InstructBLIP achieves the highest zero-shot test accuracy on ScienceQA (70.6) and IconQA (51.2), while the FlanT5$_{\text{XL}}$ backbone demonstrates comparable performance (70.4 on ScienceQA and 50.0 on IconQA) as shown in Table 1. Moreover, we utilized the ScienceQA dataset to measure the effectiveness of fine-tuning InstructBLIP models on domain-specific data. Meanwhile, the IconQA dataset requires InstructBLIP to exhibit diverse cognitive reasoning abilities, including geometric, commonsense, and arithmetic reasoning, especially when dealing with abstract diagrams with rich semantics. For simplicity and consistency in evaluation, we exclusively employ the multi-text-choice questions from both dataset. We fine-tuned and evaluated identical experimental setups across these two datasets.

4.2 Model Architecture

Since the FlanT5$_{\text{XXL}}$(11B) backbone achieves the highest zero-shot accuracy, we should have adopted this backbone as our fine-tuning model. However, the FlanT5$_{\text{XL}}$(3B) backbone produces on-par performance and requires significantly less GPU memory usage, so it was chosen as the backbone in our experimental setups.

We did not consider using Vicuna as the backbone because Dai et al. [4] have shown that, during fine-tuning on various down-stream tasks, Vicuna-based InstructBLIP is generally worse at multi-choice tasks and better at open-ended generation tasks compared to FlanT5-based models. This disparity occurs because FlanT5 is mainly fine-tuned on datasets containing many multi-choice QA and classification tasks, while Vicuna is finetuned on open-ended instruction-following data. Using a full fine-tuned InstructBLIP (FlanT5$_{\text{XXL}}$) model, Dai et al. improved ScienceQA test accuracy from 70.6 to 90.7, an ambitious goal for us to achieve considering that we use a smaller backbone and even less trainable parameters with LoRA.

To align with [4], we also keep the visual encoder frozen during fine-tuning as this significantly reduces the number of trainable parameters and thus greatly improves fine-tuning efficiency.

Table 2: **Experiment space of trainable components.** The number of total trainable parameters is $Nr \sum_{W_i \in \mathbb{W}} k_i + d_i$, where $\mathbb{W}$ is the set of weight matrices per layer of a certain trainable component.

Trainable Component	Layers N	Weight Matrices W_i	LoRA A k	LoRA B d	Trainable Parameters ($r = 1$)
Q-Former Self-Attn	12	W_Q, W_V	768	768	36,864
Q-Former Cross-Attn	6	W_Q, W_{out} W_K, W_V	768 1,408	768 768	44,544
Q-Former FFN	12	$W_0^{instruct}$, W_0^{query} $W_1^{instruct}$, W_1^{query}	768 3,072	3,072 768	184,320
LLM Attn	24	W_Q, W_V (encoder) W_Q, W_V (decoder) W_Q, W_V (enc-dec)	2,048	2,048	589,824
LLM FFN	24	W_0, W_1 (encoder) W_0, W_1 (decoder) W_{out} (encoder, decoder)	2,048 2,048 5,120	5,120 5,120 2,048	1,032,192

4.3 Trainable Components

To investigate which module of InstructBlIP is the most fine-tuning effective, we fine-tuned different components of InstructBLIP while keeping other components frozen. More specifically, our goal is to compare the performance of tuning LLM and Q-Former separately with tuning both LLM and Q-Former. To further identify which component contributes the most to performance improvement, we applied fine-tuning to each component in LLM and Q-Former, including self-attention, cross-attention and FFN.

We followed the findings in [8] and applied LoRA to both W_Q and W_V in transformer blocks. We also applied W_K to the cross attention blocks in Q-Former and observed no enhancement in test accuracy compared to scenarios without W_K. Our experiments covered the same spectrum of rank values, r, as detailed in [8], with the exception of $r = 64$, which failed to outperform other alternatives across any of their experimental setups. The trainable components along with their configurations is presented in Table. 2.

4.4 Hyperparameters and Hardware

Our experiments were fine-tuned with 224×224 image resolution on both datasets. We trained all models with AdamW [18]. We used an initial learning rate of 2×10^{-5}, a weight decay of 0.05, and a batch size of 4. Training hyperparameters were set to identical across all experiments and all experiments were trained using a single NVIDIA A100 GPU with 32G RAM [23].

5 Evaluation and Analysis

5.1 Qualitative Analysis

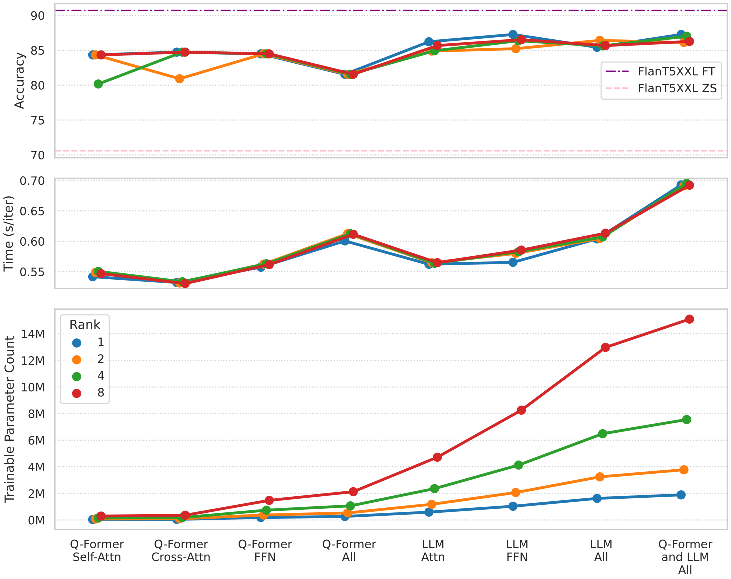


Figure 3: **Evaluation of finetuning InstructBLIP (FlanT5$_{\text{XL}}$) models using LoRA on ScienceQA dataset.** Utilizing a rank as small as one can not only yield significant enhancements over the zero-shot (ZS) configuration but also achieve a test accuracy that is competitive with that of a fully fine-tuned (FT) model employing a considerably larger backbone, such as FlanT5$_{\text{XXL}}$.

In this subsection we outline some qualitative advantages of our strategy using LoRA. Aside from accuracy, LoRA occupies less memory and trains faster. Firstly, in terms of memory utilization, actually the weights of the backbone model still are stored in memory and cannot be saved even using LoRA. Additionally, since the gradient of LoRA model also depends on that of the backbone model, the gradient of the backbone model also needs to be calculated even if the backbone model is not optimized. However, reduced memory utilization can be achieved by the fact that the optimizer status corresponding to the backbone model do not need to be stored due to no need to optimize the backbone model. Though the weights, gradients and optimizer status of the LoRA model all need to be stored, one can use low-precision data types to store the backbone model to reduce memory occupation, which is also an advantage of mixed-precision training. LoRA always takes 25% less memory than full fine-tuning in our experiments.

Secondly, in terms of training speed, the calculation amount of using LoRA is basically the same as that of full fine-tuning as it is still necessary to calculate the gradient for the backbone model. However, low-precision quantization of the backbone model can reduce the time for forward propagation and backpropagation of the backbone model, and the resulting high training efficiency is significant especially when we only use LoRA to fine-tune some certain components of the model with the parameters of the backbone model still remaining dominating. Especially when hiring data parallelism fine-tuning, only the gradient of the LoRA model part being synchronized greatly reduces the overhead for inter-device communication to increase the overall training efficiency. We leave the study of fine-tuning efficiency improvement under data parallelization training as our future work.

5.2 The Optimal Rank for LoRA

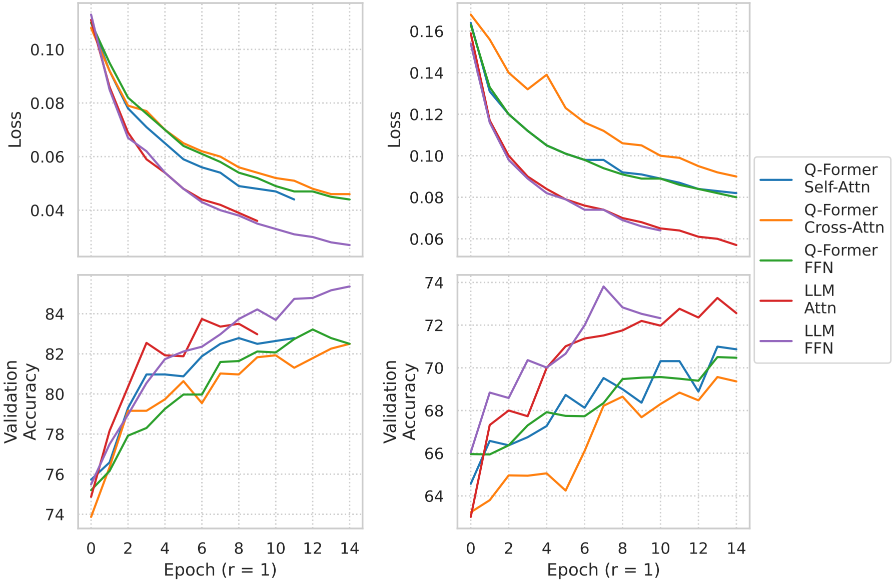


Figure 4: **Training loss and validation accuracy of ScienceQA (left) and IconQA (right).** Tuning LLM component yields lower loss value than tuning Q-Former components. Tuning the FFN layers in LLM yield the highest validation accuracy for both datasets. Training processes are stopped early if there is not accuracy improvement for three consecutive epochs.

The experimental results are quantitatively evaluated in terms of accuracy, training time, and total number of trainable parameters as we want to investigate which component is the key to accuracy improvements with the consideration of cost-effective trade-offs. Considering N layers composed of a set of weight matrices $\mathbb{W}$ per layer, with each weight matrix $W_i \in \mathbb{R}^{d_i \times k_i}$ and having a rank r, the total number of trainable parameters is $Nr \sum_{W_i \in \mathbb{W}} k_i + d_i$. The GPU memory usage grows linearly with respect to r, representing a significant hardware constraint for many enterprises and individuals. Hence, our goal is to identify a relatively modest r value that yields competitive results compared to greater values.

The second subplot of Fig. 3 and 6 shows that, regardless of the value of r, training time remains unaffected for the identical trainable component, even as the number of trainable parameters escalates to a maximum of eightfold. This is due to the model's requirement to calculate the gradient of the entire weight matrices in order to compute gradients for a limited number of trainable parameters.

In terms of accuracy, r as small as 1 and 2 generally yields more accurate results than higher values. This is consistent with the findings in [8] that the update matrix ΔW has a very small intrinsic rank for most of the downstream tasks. However, Hu et al. [8] have also pointed out that a small r may not work all the times, such as a different language in downstream tasks.

5.3 The Key Component of InstructBLIP

Fig. 4 indicates that tuning the LLM component is more effective than tuning the Q-Former component for ScienceQA and IconQA datasets because the LLM component has a larger number of parameters and captures a broader context of language. This allows the model to better understand and generate natural language. Additionally, fine-tuning the LLM component enables the model to adapt its language understanding capabilities to specific downstream tasks, such as multi-choice questions, resulting in improved performance on downstream tasks. We also observe that tuning the FFN layers in LLM yields the highest validation accuracy, and this phenomenon holds for other r values.

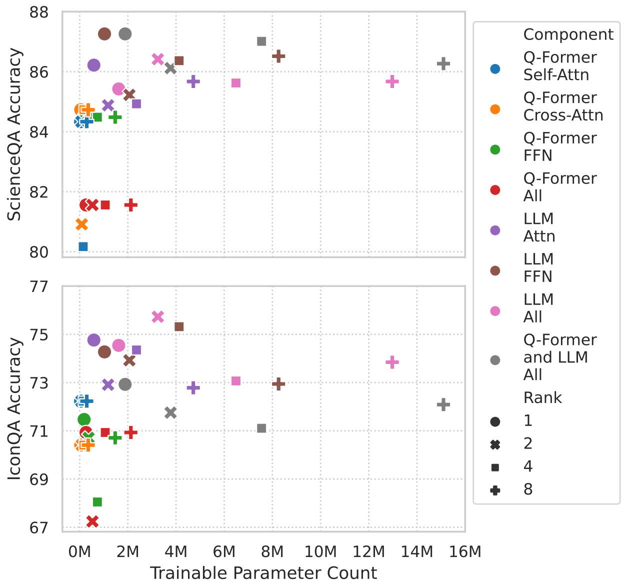


Figure 5: **The relationship between test accuracy and the total number of trainable parameters on ScienceQA and IconQA datasets.** Tuning LLM components yields higher accuracy than tuning Q-Former components on both datasets. There is no correlation between accuracy and the number of trainable parameters.

Fig.5 demonstrates that the test accuracy exhibits a weak correlation with the total number of trainable parameters. For instance, fine-tuning the FFN layers in LLM with $r = 1$ utilizes half the number of parameters compared to fine-tuning self-attention, cross-attention, and FFN layers of Q-Former using $r = 8$, yet it achieves superior performance. For ScienceQA dataset, tuning the FFN layer in LLM alone yields the best accuracy; and for IconQA dataset, 3-times larger than ScienceQA, tuning all layers LLM yields the best test accuracy. Furthermore, focusing solely on tuning the LLM has its constraints. We observe that a significant portion of questions where models underperform require optical character recognition (OCR) of images. However, as the vision encoder remains frozen, it poses challenges for the Q-Former in effectively integrating language and visual data. An illustrative example of a question that the top-performing model answers incorrectly is depicted in Fig. 1

6 Conclusion

We conducted an extensive ablation study on InstructBLIP, fine-tuning various components with LoRA while freezing the remainder. Our findings reveal that fine-tuning the LLM component yields superior results compared to fine-tuning the Q-Former component for multimodal reasoning tasks, particularly those formulated as multiple-choice questions. Furthermore, even fine-tuning with a rank as low as one or two proves adequate to achieve competitive performance compared to fully fine-tuned InstructBLIP models utilizing larger backbones. With FlanT5$_{\text{XL}}$ backbone, this approach resulted in achieving the highest fine-tuning accuracy of 75.3 on the IconQA dataset and 87.3 on the ScienceQA dataset, competitive to the full fine-tuning accuracy of a FlanT5$_{\text{XXL}}$ InstructBLIP model on ScienceQA (90.7).

References

[1] Hangbo Bao et al. “Vlmo: Unified vision-language pre-training with mixture-of-modality-experts”. In: *Advances in Neural Information Processing Systems* 35 (2022), pp. 32897–32912.

[2] Yen-Chun Chen et al. “Uniter: Universal image-text representation learning”. In: *European conference on computer vision*. Springer. 2020, pp. 104–120.

[3] Hyung Won Chung et al. *Scaling Instruction-Finetuned Language Models*. 2022. arXiv: 2210.11416 [cs.LG].

[4] Wenliang Dai et al. *InstructBLIP: Towards General-purpose Vision-Language Models with Instruction Tuning*. 2023. arXiv: 2305.06500 [cs.CV].

[5] Alexey Dosovitskiy et al. “An image is worth 16x16 words: Transformers for image recognition at scale”. In: *arXiv preprint arXiv:2010.11929* (2020).

[6] Rohit Girdhar et al. “Imagebind: One embedding space to bind them all”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2023, pp. 15180–15190.

[7] Neil Houlsby et al. “Parameter-efficient transfer learning for NLP”. In: *International conference on machine learning*. PMLR. 2019, pp. 2790–2799.

[8] Edward J Hu et al. “Lora: Low-rank adaptation of large language models”. In: *arXiv preprint arXiv:2106.09685* (2021).

[9] Chao Jia et al. “Scaling up visual and vision-language representation learning with noisy text supervision”. In: *International conference on machine learning*. PMLR. 2021, pp. 4904–4916.

[10] Wonjae Kim, Bokyung Son, and Ildoo Kim. “Vilt: Vision-and-language transformer without convolution or region supervision”. In: *International conference on machine learning*. PMLR. 2021, pp. 5583–5594.

[11] Dongxu Li et al. “LAVIS: A One-stop Library for Language-Vision Intelligence”. In: *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 3: System Demonstrations)*. Toronto, Canada: Association for Computational Linguistics, July 2023, pp. 31–41. URL: https://aclanthology.org/2023.acl-demo.3.

[12] Junnan Li et al. “Align before fuse: Vision and language representation learning with momentum distillation”. In: *Advances in neural information processing systems* 34 (2021), pp. 9694–9705.

[13] Junnan Li et al. “Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models”. In: *arXiv preprint arXiv:2301.12597* (2023).

[14] Junnan Li et al. “Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation”. In: *International Conference on Machine Learning*. PMLR. 2022, pp. 12888–12900.

[15] Xiang Lisa Li and Percy Liang. “Prefix-tuning: Optimizing continuous prompts for generation”. In: *arXiv preprint arXiv:2101.00190* (2021).

[16] Xiujun Li et al. “Oscar: Object-semantics aligned pre-training for vision-language tasks”. In: *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXX 16*. Springer. 2020, pp. 121–137.

[17] Xiao Liu et al. “P-tuning v2: Prompt tuning can be comparable to fine-tuning universally across scales and tasks”. In: *arXiv preprint arXiv:2110.07602* (2021).

[18] Ilya Loshchilov and Frank Hutter. “Decoupled weight decay regularization”. In: *arXiv preprint arXiv:1711.05101* (2017).

[19] Jiasen Lu et al. “Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks”. In: *Advances in neural information processing systems* 32 (2019).

[20] Pan Lu et al. “IconQA: A New Benchmark for Abstract Diagram Understanding and Visual Language Reasoning”. In: *The 35th Conference on Neural Information Processing Systems (NeurIPS) Track on Datasets and Benchmarks*. 2021.

[21] Pan Lu et al. “Learn to Explain: Multimodal Reasoning via Thought Chains for Science Question Answering”. In: *The 36th Conference on Neural Information Processing Systems (NeurIPS)*. 2022.

[22] Sourab Mangrulkar et al. *PEFT: State-of-the-art Parameter-Efficient Fine-Tuning methods*. https://github.com/huggingface/peft. 2022.

[23] NVIDIA. *NVIDIA A100 Tensor Core GPU*. Available at https://www.nvidia.com/en-us/data-center/a100/ (2024/04/07).

[24] Alec Radford et al. “Learning transferable visual models from natural language supervision”. In: *International conference on machine learning*. PMLR. 2021, pp. 8748–8763.

[25] Thomas Wolf et al. “Transformers: State-of-the-Art Natural Language Processing”. In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*. Online: Association for Computational Linguistics, Oct. 2020, pp. 38–45. URL: `https://www.aclweb.org/anthology/2020.emnlp-demos.6`.

[26] Shengqiong Wu et al. “Next-gpt: Any-to-any multimodal llm”. In: *arXiv preprint arXiv:2309.05519* (2023).

[27] XAttention. *peft*$_{A}ttentionX$. URL: `https://github.com/AttentionX/InstructBLIP_PEFT?tab=readme-ov-file#citation`.

[28] Pengchuan Zhang et al. “Vinvl: Revisiting visual representations in vision-language models”. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2021, pp. 5579–5588.

[29] Bin Zhu et al. “Languagebind: Extending video-language pretraining to n-modality by language-based semantic alignment”. In: *arXiv preprint arXiv:2310.01852* (2023).

7 Appendix

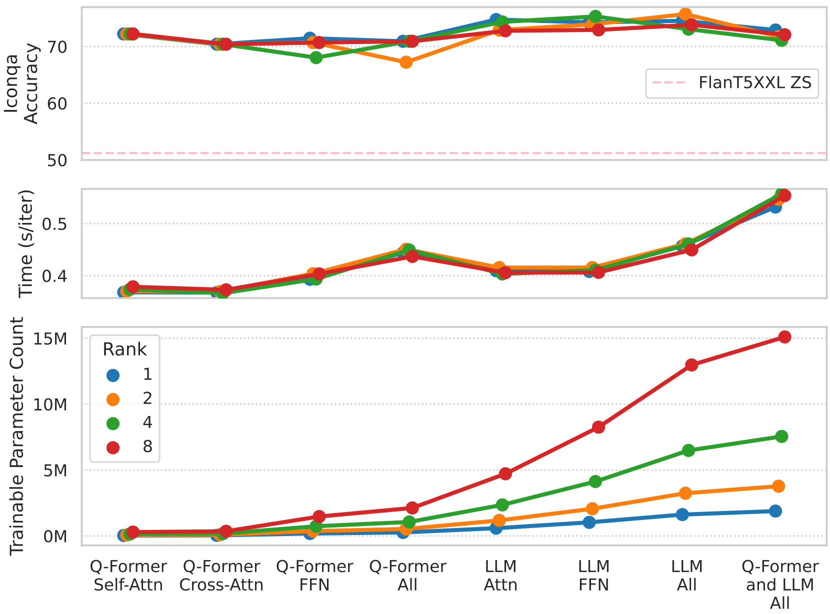


Figure 6: **Evaluation of finetuning InstructBLIP (FlanT5$_{\mathbf{XL}}$) models using LoRA on IconQA dataset.** Fine-tuning accuracy is only compared with zero-shot accuracy because InstructBLIP was not fine-tuned on IconQA in [4].