Parameter-Efficient Fine-tuning of InstructBLIP for Visual Reasoning Tasks

Sungkyung Kim1* Adam Lee2* Junyoung Park1 Sounho Chung1 Jusang Oh1 Jayyoon Lee3†

1Seoul National University 2UC Berkeley 3Graduate School of Data Science, Seoul National University

{sk0428, jyp0314, aschung01, dhwntkd412, lee.jayyoon}@snu.ac.kr alee00@berkeley.edu

Abstract

Visual language models have recently demonstrated enhanced capabilities in visual reasoning tasks by employing external modules upon language models for visual language alignment. InstructBLIP uses a Q-Former and a projection layer to convert input image embeddings into soft visual prompts to enhance the instruction-following capabilities of large language models (LLMs). Although fine-tuning InstructBLIP has shown great results in downstream tasks, previous works have been restrictive, only full fine-tuning the Q-Former, while freezing the LLM. In this work, we investigate the performance of the PEFT method, LoRA, on both the Q-Former and the base LLMs, specifically Flan-T5-XL and Vicuna-7B, using visual reasoning benchmarks ScienceQA and IconQA. We observe that, when the LLM is frozen, training the Q-Former with LoRA achieves comparable performance to full fine-tuning using under 2% of the trainable parameters. Furthermore, fine-tuning the LLM consistently result in better performances, regardless of how the Q-Former is fine-tuned. Lastly, applying LoRA to both the LLM and the Q-Former surpasses the performance of only full fine-tuning the Q-Former while using less than 10% of the trainable parameters. These results highlight the effectiveness of applying PEFT to visual language models for visual reasoning tasks. The code is available at https://github.com/AttentionX/InstructBLIP_PEFT.

1 Introduction

Pre-trained large language models can be fine-tuned to achieve high performance on many tasks. For instance, instruction tuning has been proposed to align the model’s responses more closely with human intentions [1,2]. This approach is also applicable in multi-modal settings with visual instruction tuning, that enhances the model’s capabilities to follow instructions for visual question answering and visual reasoning tasks. LLaVA [3] uses a projection layer to convert the CLIP [5] image embeddings to the word embedding space of language models, and trains both the projection layer and the language model. BLIP-2 [6] and InstructBLIP [7] use a Q-Former for visual-language alignment, similarly to Perceiver IO [8], which extract visual features in a fixed number of learnable embeddings (32 in BLIP-2). InstructBLIP only fine-tunes the Q-Former and a fully connected projection layer while freezing the LLM. The Q-Former is especially significant for its role aligning several modalities with cross-attention and encoding the information in a small number of learnable embeddings. This multimodal alignment approach has been adopted in several recent studies, including Video-LLaMA [9] and Qwen-VL [10].

*Equal contribution.
†Corresponding author.
While InstructBLIP and LLaVA achieve competitive performance on several downstream visual reasoning benchmarks [3, 7], each visual language model has its own limitations. In the case of LLaVA, full fine-tuning the large language model may be costly, due to the computational memory required to update billions of parameters. For InstructBLIP, freezing the LLM model may hinder the model from learning task-specific language understanding and generation abilities. Parameter efficient fine-tuning (PEFT) can mitigate both problems by fine-tuning large models with much less computational memory while still maintaining competitive performance [11, 12, 13, 14, 15, 16, 17, 18]. Although various PEFT methods perform competitively on downstream tasks [13, 19, 20], the efficacy of PEFT methods, both for visual reasoning tasks and for visual-language alignment models like the Q-Former, remains under-explored.

In this work, we evaluate the performance of PEFT on InstructBLIP in two benchmarks, ScienceQA [21] and IconQA [22], that respectively test knowledge-grounded visual reasoning and abstract visual reasoning capabilities. Specifically we apply LoRA to the Q-Former and base LLMs, Vicuna-7B [23] and Flan-T5-XL [24], in InstructBLIP and test with 3 different settings: applying LoRA only to the LLM, applying LoRA only to the Q-Former, and applying LoRA to both the LLM and the Q-Former (Figure 1). We also comprehensively test the performance of LoRA applied to different sublayers in the transformer with different ranks. To the best of our knowledge, we are the first to inspect the effectiveness of PEFT methods on the Q-Former for visual reasoning.

Our contributions can be summarized as follows: (1) We demonstrate that applying PEFT to the Q-Former reduces trainable parameters to less than 2% while maintaining comparable performances. (2) Our experiments reveal that applying PEFT on the LLM, rather than freezing, consistently results in better performance regardless of how the Q-Former is fine-tuned. (3) We show that, in contrast to full fine-tuning the Q-Former and freezing the LLM, applying PEFT on both components can achieve superior results and bring down the total trainable parameters to less than 10%.

2 Method

In this work, we apply the PEFT method LoRA [11] to two different components in InstructBLIP, the Q-Former and the LLM, and evaluate the performance on two visual reasoning benchmarks, ScienceQA and IconQA.

LoRA greatly reduces trainable parameters by decomposing weight update matrix into the product of two low rank matrices \( \Delta W = BA \). After the fine-tuning, weight matrix can be reparametrized by adding the weight update to the original pre-trained model weights: \( W + \Delta W = W + BA \), where \( W \in \mathbb{R}^{d \times k} \), \( B \in \mathbb{R}^{d \times r} \), \( A \in \mathbb{R}^{r \times k} \), \( r \ll \min(d, k) \). This process can prevent additional latency during inference. Unlike the original LoRA implementation, which confines its application to only the self-attention modules [11], our approach extends the use of LoRA to multiple transformer sublayers in both the Q-Former and the LLM. For the Q-Former, we apply LoRA to the query and value projection layers in the self-attention layers and to the query, key, value, and output projection layers with the cross-attention layers. We also apply LoRA to the outer layer of the feed-forward...
<table>
<thead>
<tr>
<th>Method</th>
<th>LLM</th>
<th>Q-Former</th>
<th>Sublayer</th>
<th>Base Model</th>
<th>q=1</th>
<th>q=2</th>
<th>q=4</th>
<th>q=8</th>
<th>q=1</th>
<th>q=2</th>
<th>q=4</th>
<th>q=8</th>
</tr>
</thead>
<tbody>
<tr>
<td>LoRA Full ffn</td>
<td>Flan-T5-XL</td>
<td>86.42</td>
<td>86.37</td>
<td>85.32</td>
<td>86.27</td>
<td>73.40</td>
<td>74.76</td>
<td>74.00</td>
<td>71.52</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>LoRA Full attn</td>
<td>Flan-T5-XL</td>
<td>87.36</td>
<td>86.17</td>
<td>86.91</td>
<td>86.42</td>
<td><strong>73.81</strong></td>
<td>72.88</td>
<td>73.45</td>
<td>72.86</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>LoRA Full all</td>
<td>Flan-T5-XL</td>
<td>87.41</td>
<td>87.36</td>
<td><strong>88.20</strong></td>
<td>87.90</td>
<td>73.43</td>
<td><strong>75.06</strong></td>
<td>73.23</td>
<td>73.77</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Freeze LoRA ffn</td>
<td>Flan-T5-XL</td>
<td>85.08</td>
<td>84.73</td>
<td><strong>85.42</strong></td>
<td>83.49</td>
<td><strong>72.23</strong></td>
<td>70.87</td>
<td>72.12</td>
<td>71.22</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Freeze LoRA self-attn</td>
<td>Flan-T5-XL</td>
<td><strong>86.02</strong></td>
<td>83.74</td>
<td>84.38</td>
<td>86.02</td>
<td>71.82</td>
<td><strong>72.55</strong></td>
<td>72.06</td>
<td>71.64</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Freeze LoRA cross-attn</td>
<td>Flan-T5-XL</td>
<td>84.13</td>
<td>71.29*</td>
<td>84.88</td>
<td><strong>85.18</strong></td>
<td>72.32</td>
<td>72.42</td>
<td>72.32</td>
<td><strong>73.92</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Freeze LoRA all</td>
<td>Flan-T5-XL</td>
<td>84.04</td>
<td>83.79</td>
<td>84.88</td>
<td><strong>84.93</strong></td>
<td>42.40*</td>
<td>71.3</td>
<td><strong>72.02</strong></td>
<td><strong>72.02</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>LoRA LoRA ffn</td>
<td>Vicuna-7B</td>
<td>86.32</td>
<td>86.42</td>
<td>85.87</td>
<td>85.97</td>
<td><strong>73.72</strong></td>
<td>72.97</td>
<td>73.02</td>
<td>73.24</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>LoRA LoRA attn</td>
<td>Vicuna-7B</td>
<td>86.42</td>
<td>86.32</td>
<td>85.08</td>
<td>85.23</td>
<td>72.36</td>
<td>73.16</td>
<td><em>72.29</em></td>
<td><strong>73.72</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>LoRA LoRA all</td>
<td>Vicuna-7B</td>
<td>85.03</td>
<td>86.32</td>
<td>85.57</td>
<td>85.72</td>
<td><strong>73.77</strong></td>
<td>72.58</td>
<td>72.93</td>
<td>73.15</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Freeze LoRA ffn</td>
<td>Vicuna-7B</td>
<td>82.50</td>
<td>80.76</td>
<td><strong>82.80</strong></td>
<td>81.01</td>
<td>72.53</td>
<td><strong>73.01</strong></td>
<td>72.25</td>
<td>71.85</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Freeze LoRA self-attn</td>
<td>Vicuna-7B</td>
<td><strong>83.19</strong></td>
<td>81.51</td>
<td>82.25</td>
<td>83.14</td>
<td>71.23</td>
<td>71.45</td>
<td>71.42</td>
<td><strong>71.74</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Freeze LoRA cross-attn</td>
<td>Vicuna-7B</td>
<td>83.29</td>
<td>83.24</td>
<td>83.14</td>
<td>82.75</td>
<td>71.11</td>
<td>72.40</td>
<td>71.99</td>
<td><strong>73.39</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Freeze LoRA all</td>
<td>Vicuna-7B</td>
<td>84.04</td>
<td>58.35*</td>
<td><strong>84.09</strong></td>
<td>83.34</td>
<td>72.40</td>
<td>71.64</td>
<td>71.61</td>
<td><strong>73.51</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>LoRA LoRA all</td>
<td>Vicuna-7B</td>
<td>84.63</td>
<td>85.18</td>
<td><strong>88.10</strong></td>
<td>86.61</td>
<td><strong>73.46</strong></td>
<td>63.55*</td>
<td>72.07</td>
<td>72.67</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 1: Overall performance results. "Full" indicates full fine-tuning, and the best results among 4 r values are bolded. The best results for each PEFT category, benchmark, and base language models are underlined. The underlined performances are used to compare the best performances between PEFT methods in Figure 2. (Few results annotated as * are expected to improve after additional hyperparameter tuning.)

networks. Conversely, in the LLM, we apply LoRA to both the query and value layers in the attention module and to the feed-forward network.

**Base Models and benchmarks.** We selected InstructBLIP as the base model given its reported state-of-the-art performance for fine-tuning on several downstream tasks [7], including ScienceQA (IMG) [21], OCR-VQA [25], and A-OKVQA [26]. We use the InstructBLIP implementation of LAVIS [27] and use pre-trained Vicuna-7B\(^1\) and Flan-T5-XL\(^2\) HuggingFace checkpoints in our experiments.

We use two benchmarks from InstructBLIP covering tasks of Knowledge Grounded Visual Reasoning (ScienceQA) [21] and Abstract Visual Reasoning (IconQA) [22]. These benchmarks were held-out datasets of InstructBLIP, and were not involved in training the baseline InstructBLIP model.

**Knowledge Grounded Visual Reasoning** is a task of answering questions with a provided image related to the knowledge in diverse academic areas including physics, biology, and math. We use the ScienceQA dataset which covers a variety of science topics with corresponding extensive explanations. We only use the questions with image context (IMG). ScienceQA (IMG) has 6.2k training samples and 2.1k, 2.0k samples for validation and testing.

**Abstract Visual Reasoning** is a task of answering questions after comprehending the abstract meanings from an image. We use IconQA which contains question-answer pairs for natural images that require comprehensive reasoning abilities to understand abstract diagrams.

**Experimental setup.** In the original InstructBLIP [7], full fine-tuning was applied to the Q-Former, while the LLM was frozen. In this work, we empirically analyze the effectiveness of training LoRA on the Q-Former and the LLM. (1) First, we apply LoRA to the LLM while still full fine-tuning the Q-Former, so the LLM is further trained to adapt to visual reasoning tasks. (2) Second, we apply LoRA to the Q-Former while freezing the LLM, resulting in efficient fine-tuning of the Q-Former. (3) Finally, we apply LoRA to both the Q-Former and the LLM. The evaluation entail testing with different ranks (1, 2, 4, 8), and for the base models of Vicuna-7B and Flan-T5-XL. The main results of the overall experiments are in Table 1 and the performance comparison of (1), (2), (3) and the original InstructBLIP is in Figure 2. The implementation and training details can be found in Appendix A and instruction templates used for instruction tuning can be found in Appendix B.

\(^1\) [https://huggingface.co/lmsys/vicuna-7b-v1.3](https://huggingface.co/lmsys/vicuna-7b-v1.3)

\(^2\) [https://huggingface.co/google/flan-t5-xl](https://huggingface.co/google/flan-t5-xl)
Figure 2: Performance and trainable parameter comparison among three PEFT methodologies using base models Vicuna-7B and Flan-T5-XL on ScienceQA and IconQA benchmarks. This compares the best performing configurations (rank value and LoRA-applied sublayers) between InstructBLIP (Q-Former full fine-tuning with frozen LLM), LLM PEFT with Q-Former full fine-tuning, Q-Former PEFT with frozen LLM, and Q-Former PEFT with LLM PEFT. “QF” denotes Q-Former. “FFT” denotes full fine-tuning.

3 Experiments

PEFT on LLM. We assess the efficacy of fine-tuning the LLM in InstructBLIP using LoRA. We consider 3 configurations: applying LoRA to the attention modules, the feed-forward network (FFN), and both the attention modules and the FFN. Across all tasks, with both Vicuna-7B and Flan-T5-XL as base models, fine-tuning the LLM with LoRA consistently outperforms InstructBLIP, as shown in Figure 2. These results suggest that introducing additional trainable parameters in the LLM enhances its language reasoning abilities for visual reasoning tasks. We find no clear performance differences among the LoRA ranks (1, 2, 4, 8). Also, unlike previous studies on language models [14, 28], no particular sublayer stood out in performance with LoRA.

PEFT on Q-Former. We examine the effectiveness of applying LoRA to different sublayers in the Q-Former while keeping the LLM frozen. This involves training LoRA on the self-attention, cross-attention, and FFN layers individually, and collectively on all three layers. We initially hypothesized that the cross-attention layer, given its direct role in image feature extraction, would be the most effective. However we observe no notable performance differences among the LoRA sublayer configurations. All configurations either outperform or match the results of full Q-Former fine-tuning while utilizing at most 2% of the Q-Former parameters (Figure 2). This suggests that training LoRA on the Q-Former offers significantly more efficient training while maintaining competitive performance. Furthermore, higher LoRA ranks do not result in better performance, indicating that the Q-Former’s low-rank weight updates for learning visual reasoning only require small intrinsic ranks [11].

PEFT on both LLM and Q-Former. Finally we apply LoRA to both the Q-Former and the LLM, using the same rank for all possible sublayers in both components. Our results show that this approach outperforms InstructBLIP for both base LLMs across both benchmarks, using fewer than 10% of trainable parameters (as depicted in Figure 2). A notable observation is that the performance gap is higher in ScienceQA than in IconQA. This discrepancy can be attributed to ScienceQA’s richer
language context. Given that ScienceQA entails more language information than IconQA, training the language model appears to yield a greater boost in performance.

4 Conclusion

In this study, we systematically evaluate the benefits of applying LoRA to the Q-Former and LLM of InstructBLIP for visual reasoning tasks. Our results show that applying PEFT to the LLM leads to improved performance. Additionally, by employing PEFT on the Q-Former, we achieve outcomes comparable to full fine-tuning while only utilizing less than 2% of its parameters. Finally, we find that training both the LLM and Q-Former with PEFT yields superior results while training on less than 10% of the parameters compared to InstructBLIP. These findings hold practical importance, our results recommend jointly training both the Q-Former and LLM using PEFT, especially when computational resources are limited. Given our findings that demonstrate the efficiency and effectiveness of PEFT methods on InstructBLIP, we believe this work lays the foundation and motivate further research into efficient visual instruction tuning methods.

5 Acknowledgements

This work was supported by Seoul National University’s Engineering Department and the NLP Research Group AttentionX. We are grateful for the support of A100 GPUs.

References


A Model Training Details

We conduct each experiment in Table 1 and Figure 2 using a single A100 GPU. For the initial learning rate, we use 2e-5 for experiments which involves full fine-tuning of the Q-Former, and otherwise 5e-4. We use AdamW optimizer with linear decay as a learning rate scheduler. Additionally, we set the weight decay to 0.05 and maximum epochs to 15. We apply early stopping with 3 patience steps, except in some cases when the early stopping stops the training before reaching 8th epoch (the halfway point). We remove early stopping in this case for effective training. For batch size, we use 16 as an effective batch size across all experiments. Only difference is that (batch size, gradient accumulation iterations) were set to (8, 2) for Vicuna-7B and (16, 1) for Flan-T5-XL.

B Instruction Templates

We provide instructions used in ScienceQA and IconQA. We use the same format from the Instruct- BLIP paper. We add alphabet labels for each choices and the answer. For ScienceQA, we construct the "context" section of the instruction by incorporating information from both the 'hint' and 'lecture' fields, if they are available in the dataset.

**ScienceQA**

<Image> Context: { {hint} {lecture} } Question: { {question} } Options: { {choices} }.

Answer:

![Figure 3: Example ScienceQA instruction template](image)

**IconQA**

<Image> Question: { {question} } Options: { {choices} }.

Short answer:

![Figure 4: Example IconQA instruction template](image)