LLaMA-Adapter: Efficient Fine-tuning of Language Models with Zero-init Attention

Renrui Zhang[*1,2], Jiaming Han[*1], Aojun Zhou[2], Xiangfei Hu[1], Shilin Yan[1]
Pan Lu[3], Hongsheng Li[2], Peng Gao[1], Yu Qiao[1]

[1]Shanghai Artificial Intelligence Laboratory   [2]CUHK MMLab
[3]University of California, Los Angeles
{zhangrenrui, hanjiaming, gaopeng, qiaoyu}@pjlab.org.cn

Abstract

*We present **LLaMA-Adapter**, a lightweight adaption method to efficiently fine-tune LLaMA into an instruction-following model. Using 52K self-instruct demonstrations, LLaMA-Adapter only introduces **1.2M** learnable parameters upon the frozen LLaMA 7B model, and costs less than **one hour** for fine-tuning on 8 A100 GPUs. Specifically, we adopt a set of learnable adaption prompts, and prepend them to the input text tokens at higher transformer layers. Then, a zero-init attention mechanism with zero gating is proposed, which adaptively injects the new instructional cues into LLaMA, while effectively preserves its pre-trained knowledge. With efficient training, LLaMA-Adapter generates high-quality responses, comparable to Alpaca with fully fine-tuned 7B parameters. Furthermore, our approach can be simply extended to multi-modal input, e.g., images, for image-conditioned LLaMA, which achieves superior reasoning capacity on ScienceQA. We release our code at* https://github.com/ZrrSkywalker/LLaMA-Adapter.

1. Introduction

Large-scale Language Models (LLMs) [6, 7, 37, 38, 58] have stimulated widespread attention in both academia and industry. Driven by massive corpora and advanced hardware, LLMs exhibit remarkable understanding and generative ability, propelling language tasks into a higher level. Recently, significant progress has been made on instruction-following models, e.g., ChatGPT[1] and GPT-3.5 (text-davinci-003) [5]. Following instructions or commands in natural language, they can generate professional and contextual responses in a conversational way. However, the further
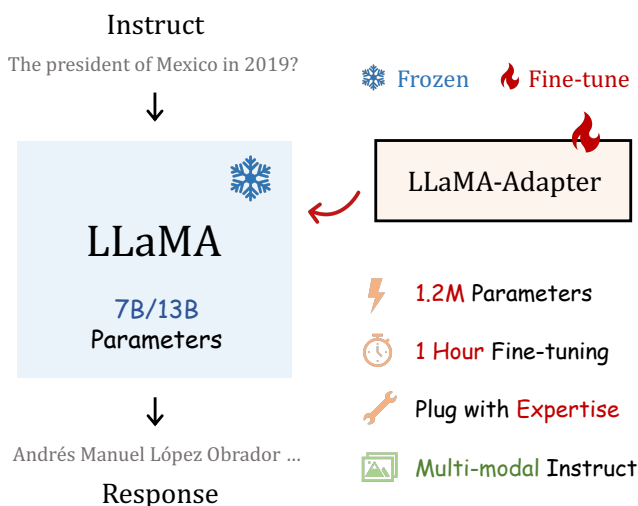

Figure 1. **Characteristics of LLaMA-Adapter.** Our lightweight adaption method tunes LLaMA into an instruction-following model with only 1.2M learnable parameters and one-hour training. LLaMA-Adapter is plug-and-play for different downstream expertise and can be generalized to multi-modal reasoning.

prevalence of instruction-following models is largely impeded by the closed-source restriction and high development costs.

To alleviate this, Stanford Alpaca [41] proposes to fine-tune an LLM, i.e., LLaMA [42] into an instruction-following model, which is affordable and replicable. Starting from 175 human-written instruction-output pairs [48], Alpaca leverages GPT-3.5 to expand the training data to 52K in a self-instruct manner. Supervised by this, Alpaca fine-tunes the entire 7B parameters in LLaMA, producing an exceptional model that performs similarly to GPT-3.5. Despite Alpaca's effectiveness, a complete fine-tuning of large-scale LLaMA is still time-consuming, computation-intensive, multi-modality unsupported and cumbersome to

---

transfer to different downstream scenarios.

In this paper, we introduce **LLaMA-Adapter**, an efficient fine-tuning method that adapts LLaMA into a well-performed instruction-following model. We also utilize the 52K instruction-output data for training purposes, but demonstrate superior resource efficiency to Alpaca. Specifically, in LLaMA's higher transformer layers, we append a set of learnable adaption prompts as prefix to the input instruction tokens. These prompts learn to adaptively inject new instructions (conditions) into LLaMA. To avoid noise from adaption prompts at the early training stage, we modify the vanilla attention mechanisms at inserted layers to be zero-init attention, with a learnable gating factor. Initialized by zero vectors, the gating can firstly preserve the original knowledge in LLaMA, and progressively incorporate instructional signals during training. This contributes to stable learning during the fine-tuning process and better instruction-following capacity of the final model.

Overall, our LLaMA-Adapter exhibits four main characteristics, as shown in Figure 1.

- **1.2M Parameters.** Instead of updating the full 7B parameters, we freeze the pre-trained LLaMA and only learn the adaption prompts with 1.2M parameters on top. This, however, reveals comparable instruction-following proficiency with the 7B Alpaca.

- **One-hour Fine-tuning.** Thanks to lightweight parameters and our zero-init gating, the convergence of LLaMA-Adapter costs less than one hour on 8 A100 GPUs, three times faster than Alpaca.

- **Plug with Expertise.** For different scenarios, it is flexible to insert their respective adapters and endow LLaMA with different expert knowledge. Thus, it suffices to store a 1.2M adapter within each context, other than a complete copy of the 7B model.

- **Multi-modal Condition.** Besides textual instruction, LLaMA-Adapter can be extended to image input for multi-modal reasoning. By simply adding images tokens into adaption prompts, LLaMA-Adapter performs competitively on the ScienceQA benchmark.

2. Related Work

2.1. Instruction-Following Language Models

The subfield of language models focusing on the instruction-following capabilities is crucial for generating responses based on natural language commands. Instruction-following methods enhance pre-trained models by fine-tuning them using high-quality input-output tuples of task instructions and ground truth outputs. This fine-tuning helps the model better understand user intentions and follow instructions more accurately. Instruction-following

methods have been extensively researched in language models [3, 33, 49, 51] and multi-modality domains [30, 40]. Among those methods, FLAN [51] introduces an instruction tuning method that outperforms non-tuned LLMs in unseen tasks. PromptSource [3] provides development environment and repository that offers a web-based GUI for creating and managing natural language prompts for zero-shot or gradient-based few-shot learning. SUP-NATINST [49] establishes a large benchmark of 1,616 diverse NLP tasks and uses multi-task training on the T5 model, demonstrates strong generalization capabilities on unseen tasks. Instruct-GPT [33] demonstrates significant performance improvements and may be integrated into closed-source models like GPT-3.5 and GPT-4 [32]. The open-source Stanford-Alpaca [41] approach fine-tunes all parameters of LLMs in an end-to-end manner. However, this full-model fine-tuning can be computationally intensive and challenging to scale to larger pre-trained language models. In contrast, this paper aims to fine-tune lightweight adapters on top of the frozen large-scale pre-trained models, *e.g.*, LLaMA [42], rather than performing end-to-end fine-tuning of all parameters. Our approach reduces computational demands and facilitates the efficient adaptation of LLMs to instruction-following tasks while maintaining high performance.

2.2. Large Vision-Language Models

Over the past decade, we have witnessed a shift in vision-language research from task-specific models [13, 15, 39, 44, 52] to large foundation models [1, 4, 21, 36, 45–47, 50]. After pre-training on large-scale image-text data, such large vision-language models can be adapted to a variety of downstream tasks with powerful performance. A line of works [36, 46, 50] train both visual and textual encoders from scratch, leading to a high computation cost. Recently, another line of works [1, 21, 45, 54] adopt pre-trained uni-modal models as initialization and only train the newly introduced parameters. For example, LiT [54] utilizes pre-trained image encoder to speed up CLIP [36] training. Frozen [43] fine-tunes an image encoder to transform visual tokens into LLM's soft prompts. Similarly, CLIPCap [31] proposes a mapping network to connect the pre-trained image encoder with LLMs. Flamingo [1] inserts several cross-attention layers to inject visual knowledge into LLMs. BLIP2 [21] connects pre-trained image encoders and LLMs with a Q-Former. CLIP-Adapter [8], Tip-Adapter [55, 57] and PointCLIP [56, 60] introduce customized adapters upon CLIP for 2D and 3D few-shot learning. To summary, these methods use mapping networks or cross-attention layers to connect vision and languages. Our work also belongs to the second line of works. Differently, our method only introduces a few learnable parameters and progressively injects visual features into pre-trained LLMs with a simple but efficient zero-init attention.

2.3. Parameter-Efficient Fine-Tuning

Parameter-Efficient Fine-Tuning (PEFT) [34] methods facilitate efficient adaptation of LLMs without the need to update all model parameters, thereby reducing the cost and improving the efficiency of fine-tuning large models. Various PEFT techniques include Prefix Tuning [24], Low-Rank adaptation (LoRA) [12] and the insertion of adapter layers in pre-trained large language models [11, 26, 35]. Prefix Tuning [24] appends a collection of prefixes to autoregressive language models, or alternatively, incorporates prefixes for both encoder and decoder components, similar methods proposed in [20]. LoRA [12] introduces trainable rank decomposition matrices into each layer [14]. Adapters [11] involves inserting lightweight modules into each layer of pre-trained models, which only updates the adapters and has been extended across numerous domains [35].

In this paper, we fine-tune pre-trained language models for instruction-following capabilities (response to instructs), and are distinct from existing ones in two aspects.

- **Zero-init Attention.** Prevalent PEFT methods might potentially disturb the pre-trained linguistic knowledge by directly inserting randomly initialized modules. This leads to unstable fine-tuning with large loss value at early training stages. LLaMA-Adapter adopts a zero-init attention with gating to mitigate this.

- **Unified Multi-modal Tuning.** Previous PEFT methods are normally developed to address specific modalities, such as language, image, and audio. In contrast, LLaMA-Adapter can handle both language and multi-modality fine-tuning with a unified manner, demonstrating superior generalization ability.

3. LLaMA-Adapter

In Section 3.1, we first introduce how to insert the learnable adaption prompts into LLaMA's transformer. Then in Section 3.2, we present the details of zero-init attention mechanisms with zero gating. Finally in Section 3.3, we generalize LLaMA-Adapter for multi-modal reasoning.

3.1. Learnable Adaption Prompts

Given 52K instruction-to-output data [48] and a pre-trained LLaMA [42] with an N-layer transformer, we adopt a set of learnable adaption prompts for instruction-following fine-tuning. We denote the prompts for L transformer layers as $\{P_l\}_{l=1}^{L}$, where $P_l \in \mathbb{R}^{K \times C}$ with K denoting the prompt length for each layer, and C equaling the feature dimension of LLaMA's transformer. Note that we insert the prompts into the topmost L layers of the transformer ($L \leq N$). This can better tune the language representations with higher-level semantics.
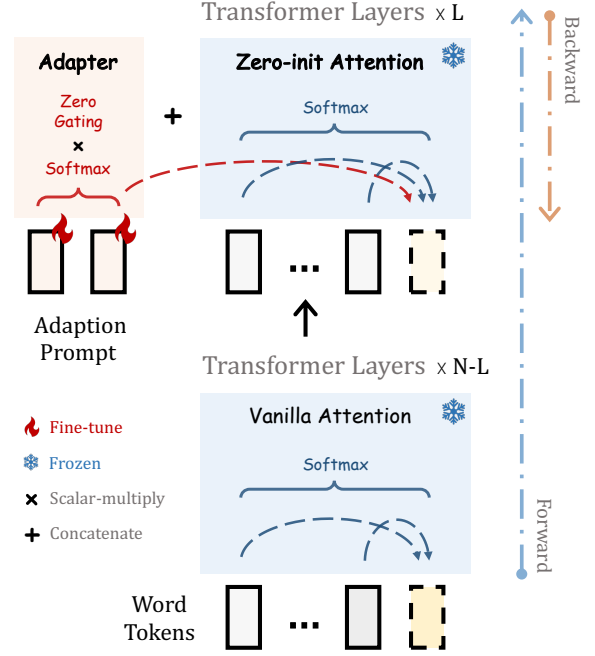
Details of LLaMA-Adapter

Figure 2. **Details of LLaMA-Adapter.** We insert lightweight adapters with learnable prompts into L out of N transformer layers of LLaMA. Aided by zero-init attention and gating mechanisms, the adaption prompt progressively learns new instructional cues, without disturbing the original pre-trained knowledge.

Taking the l-th inserted layer as an example, we denote the M-length word tokens as $T_l \in \mathbb{R}^{M \times C}$. Then, the adaption prompt is concatenated with T_l along the token dimension as prefix, formulated as

$$[P_l;\ T_l]\ \in \mathbb{R}^{(K+M) \times C}. \tag{1}$$

In this way, the instruction knowledge learned within P_l can effectively guide T_l to generate contextual responses.

3.2. Zero-init Attention

If the adaption prompts are randomly initialized, they might bring disturbance to the word tokens at the beginning of training, which harms the fine-tuning stability and effectiveness. Considering this, we modify the vanilla attention mechanisms at the last L transformer layers to be zero-init attention, as shown in Figure 2. Suppose the model is generating the $(M+1)$-th word on top of $[P_l;\ T_l]$ at the l-th inserted layer, we denote the corresponding $(M+1)$-th word token as $t_l \in \mathbb{R}^{1 \times C}$. In the attention mechanism, several linear projection layers are first applied to transform
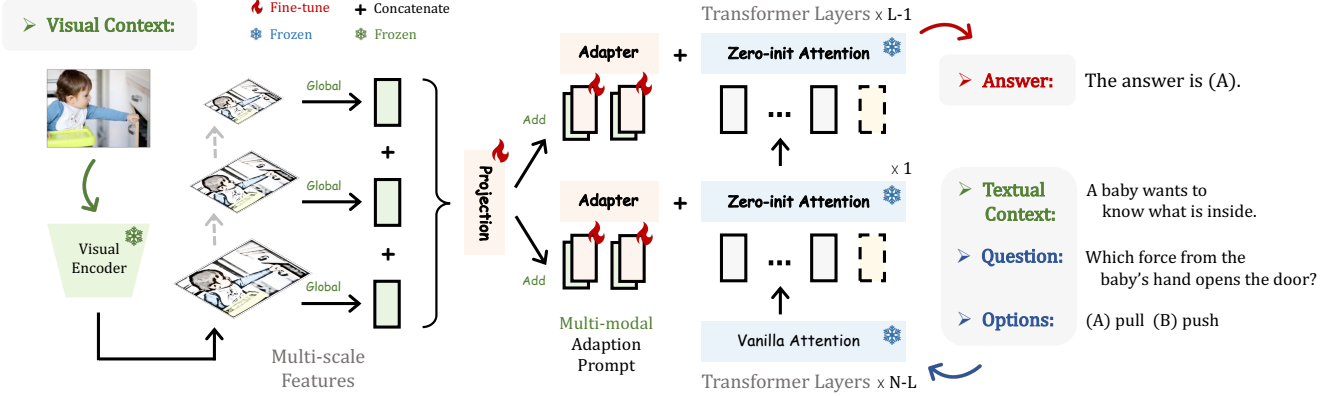
Multi-modal Reasoning of **LLaMA-Adapter**

Figure 3. **Multi-modal Reasoning of LLaMA-Adapter.** On the ScienceQA benchmark [27], LLaMA-Adapter is extended to multi-modal variant for image-conditioned question answering. Given an image as the visual context, we acquire the global image token by multi-scale concatenation and projection. Then, we element-wisely add the image token onto adaption prompts of L inserted layers. In this way, LLaMA-Adapter achieves competitive reasoning capability based on multi-modal conditions.

the input tokens into queries, keys, and values as

$$Q_l = \text{Linear}_{\text{q}}(\, t_l \,), \tag{2}$$
$$K_l = \text{Linear}_{\text{k}}(\, [P_l;\, T_l;\, t_l] \,), \tag{3}$$
$$V_l = \text{Linear}_{\text{v}}(\, [P_l;\, T_l;\, t_l] \,). \tag{4}$$

Then, the attention scores before the softmax function are calculated as

$$S_l = Q_l K_l^T / \sqrt{C} \ \in \mathbb{R}^{1 \times (K+M+1)}, \tag{5}$$

which records the feature similarities between t_l and all $K + M + 1$ tokens. Meanwhile, S_l can be reformulated by two components as

$$S_l = [S_l^K;\, S_l^{M+1}]^T, \tag{6}$$

where $S_l^K \in \mathbb{R}^{K \times 1}$ and $S_l^{M+1} \in \mathbb{R}^{(M+1) \times 1}$ denote the attention scores of K adaption prompts and $M + 1$ word tokens, respectively. The former S_l^K represents how much information the learnable prompt contributes to t_l, which probably causes disturbance in the early training stage.

To this end, we adopt a learnable gating factor, denoted as g_l, to adaptively control the importance of S_l^K in the attention. Initialized by zero, g_l can firstly eliminate the influence of under-fitted prompts, and then increase its magnitude for providing more instruction semantics to LLaMA. Therefore, we independently apply the softmax functions to the two components in Equation (6), and multiply the first term by g_l, formulated as

$$S_l^g = [\text{Softmax}(S_l^K) \cdot g_l;\ \text{Softmax}(S_l^{M+1})]^T. \tag{7}$$

The separate softmax functions ensure the second term to be irrelevant to our adaption prompts. When g_l is close to

zero, it can convey the originally pre-trained knowledge of LLaMA to token t_l for creditable generation. In practice, we adopt multiple g_l to be independently learned for different heads of the attention mechanism.

Finally, we calculate the output of the attention layer with a linear projection layer as

$$t_l^o = \text{Linear}_{\text{o}}(S_l^g V_l) \ \in \mathbb{R}^{1 \times C}. \tag{8}$$

With our proposed zero-init attention, the adaption prompts progressively inject the newly acquired instructional knowledge into LLaMA, while effectively incorporates its pre-trained ability to provide high-quality responses.

3.3. Multi-modal Reasoning

Not limited to textual instructions, LLaMA-Adapter is capable of answering a question based on input of other modalities, which augments the language model with rich cross-modal information. As shown in Figure 3, we take the ScienceQA benchmark [27] as examples. Given **visual** and **textual contexts**, along with the corresponding **question** and **options**, the model is required to conduct multi-modal reasoning to give the correct **answer**.

For an input image as the visual context, we first leverage a pre-trained visual encoder, *e.g.*, CLIP [36], to extract its multi-scale global features, denoted as $\{I_m\}_{m=1}^M$, where $I_m \in \mathbb{R}^{1 \times C_m}$ and M denotes the scale number. Then, we concatenate the M-scale features along the channel dimension and apply a learnable projection network on top, formulated as

$$I_p = \text{Projection}\Big(\text{Concat}\big(\{I_m\}_{m=1}^M\big)\Big), \tag{9}$$

where $I_p \in \mathbb{R}^{1 \times C}$ and is regarded as the overall image token with the same feature dimension as our adaption prompts. After this, we repeat I_p for K times, and element-wisely add it onto the K-length adaption prompts at all L inserted transformer layers. For the l-th layer, we denote the acquired multi-modal prompt as

$$P_l^v = P_l + \text{Repeat}(I_p) \in \mathbb{R}^{K \times C}, \qquad (10)$$

where P_l^v denotes the multi-modal prompt incorporated with visual information from the given image context. In this way, LLaMA is fine-tuned to generate responses conditioned on vision-language inputs, and can tackle more challenging generative tasks with multi-modal understanding, such as the ScienceQA benchmark [27].

As a general framework, LLaMA-Adapter with additional input condition can also be extended to video and audio modalities. Using the pre-trained modal-specific encoders, we can integrate instructional signals of different modalities into the adaption prompts, which further maximizes the comprehension and generative capacity of LLaMA. We leave this as a future work.

4. Instruction-following Evaluation

In this section, we evaluate the instruction-following capacity of LLaMA-Adapter by responding to instructions.

4.1. Experimental Details

Training Data. We use 52K instruction-following data from Stanford Alphaca [41] for training, denoted as Alphaca-52K. Each sample in Alphaca-52K contains the following fields: {instruction} is the description of a task, {input} is the context for the task, and {output} is the answer generated by GPT-3.5 (text-davinci-003) [5]. Note that around 40% of the examples include an input.

Implementation Details. We build LLaMa-Adapter based on the original LLaMa codebase[2] with minor modifications. We train LLaMa-Adapter on 8 A100 GPUs for 5 epochs. The warmup epochs, batch size, learning rate, and weight decay are set to 2, 64, 9e-3, and 0.02, respectively. In general, we utilize the pre-trained LLaMA model with 7B parameters and $N = 32$ transformer layers as the base model. We set the prompt length $K = 10$ and insert prompts into the last $L = 30$ layers by default. Other variants of LLaMA-Adapter with different inserted layers are also released in our code. In the generation stage, we adopt *top-p* sampling as the default decoding method with a temperature 0.1 and a *top-p* $= 0.75$.

4.2. Performance

We compare LLaMA-Adapter with the representative instruction-following method, Alphaca [41], in Table 1, and present the full comparison with Alpaca-LoRA [12] and GPT-3 [5] in Appendix A. As there still lack of rigorous metrics for evaluation, we simply show some response examples for comparison. As shown, by only fine-tuning 1.2M parameters, our approach generates reasonable responses comparable to the fully fine-tuned Alpaca and the large-scale GPT-3. This fully demonstrates the effectiveness of our adapters with zero-init attention.

In Appendix B, we also compare LLaMA-Adapter with LLaMA-I, *i.e.*, LLaMA-65B fine-tuned on large-scale instructional data. As shown, LLaMA-Adapter is capable of various non-trivial tasks, such as dialog generation, code generation, and question answering *etc.* We believe LLaMa-Adapter can be further enhanced by combining larger LLaMA models, enlarging the amount of training data, and scaling up learnable parameters.

4.3. Efficiency

In Table 3, we compare the learnable parameters, storage space, and training time of different instruction-following methods. As a lightweight plug-and-play module, LLaMA-Adapter enjoys superior training efficiency with only 1.2M parameters, 4.9M storage, and one-hour training. This enables us to fine-tune large-scale language models (LLaMA) on cheap and mobile devices. LLaMA-Adapter's efficiency advantages can be further revealed by multi-node training, since only the gradients of 1.2M parameters are transferred among nodes, other than Alpaca's 7B.

5. Multi-modal Evaluation

In this section, we illustrate the multi-modal reasoning capability of LLaMA-Adapter on the ScienceQA benchmark [27] and conduct ablation studies to verify the effectiveness of our designs.

5.1. Experimental Details

Training Data. We train the multi-modal LLaMA-Adapter on ScienceQA [27], a large-scale multi-modal and multi-choice science question dataset collected from a wide range of domains[3]. Figure 4 gives two examples in ScienceQA. Each example normally contains a **visual context**, a **textual context**, a **question**, multiple **options**, and an **answer**. We omit the lecture and explanation for simplicity. The model is required to output the correct answer based on the multi-modal context and choices. The official question answering accuracy is adopted as the evaluation metric.

---

[2] https://github.com/facebookresearch/llama

[3] https://scienceqa.github.io/explore

Instruction: Tell me about alpacas.

LLaMA-Adapter: Alpacas are members of the camelid family and are native to the Andes Mountains of South America. They are typically found in herds of 10-20 animals and are known for their gentle nature and their ability to thrive in harsh climates. Alpacas are typically white or grey in color and have long, silky coats. They are also known for their ability to produce high-quality wool, which is used in the production of clothing and other products.

Alphaca: Alpacas are small, fluffy animals related to camels and llamas. They are native to Peru and Bolivia, and were first domesticated around 5,000 years ago. They are kept mainly for their fine, soft fleece, which is used to make knitwear and other garments. Alpacas are herd animals and live in small family groups, led by an older male. They are highly social animals and can form strong bonds with their owners.

Instruction: Tell me about the president of Mexico in 2019.

LLaMA-Adapter: The president of Mexico in 2019 was Andrés Manuel López Obrador, who took office on December 1st, 2018. He is a member of the left-wing party Morena and is known for his populist policies. He has focused on tackling corruption, reducing poverty, and improving the country's infrastructure. He has also been a vocal critic of the United States' immigration policies.

Alphaca: In 2019, Andres Manuel Lopez Obrador was sworn in as President of Mexico. He is the country's 58th president and the first from the left-wing Morena party. He was elected in 2018 on a promise to combat corruption and reduce poverty and has faced criticism for his handling of the economy, crime and the country's relationship with the United States. He is also known for his ambitious social welfare programs and has been praised for raising the minimum wage and providing aid to low-income families. He is seen as a champion of democracy and human rights in Mexico.

Instruction: List all Canadian provinces in alphabetical order.

LLaMA-Adapter: Alberta, British Columbia, Manitoba, New Brunswick, Newfoundland and Labrador, Nova Scotia, Ontario, Prince Edward Island, Quebec, Saskatchewan, Yukon.

Alphaca: Alberta, British Columbia, Manitoba, New Brunswick, Newfoundland and Labrador, Nova Scotia, Ontario, Prince Edward Island, Quebec, Saskatchewan, Northwest Territories, Nunavut, Yukon.

Instruction: Tell me five words that rhyme with 'shock'.

LLaMA-Adapter: Flock, rock, lock, cook, and stock.

Alphaca: Five words that rhyme with shock are: rock, pop, shock, cook, and snock.

Instruction: Translate the sentence 'I have no mouth but I must scream' into Spanish.

LLaMA-Adapter: No tengo boca pero tengo que gritar.

Alphaca: No tengo boca, pero debo gritar.

Instruction: Write a Python program that prints the first 10 Fibonacci numbers.

LLaMA-Adapter:

```python
def fibonacci():
    a = 0
    b = 1
    for i in range(10):
        a, b = b, a + b
        print(a, b)
    print("Fibonacci numbers: ", a, b)

fibonacci()
```

Alphaca:

```python
numbers = [0, 1]

for i in range(2, 11):
    numbers.append(numbers[i-2] + numbers[i-1])

print(numbers)
# Output: [0, 1, 1, 2, 3, 5, 8, 13, 21, 34]
```

Table 1. **Instruction-following Comparison** between LLaMA-Adapter and Alpaca [41].

Model	Tuned Params	Avg	NAT	SOC	LAN	TXT	IMG	NO	G1-6	G7-12
Random Choice [27]	-	39.83	40.28	46.13	29.25	47.45	40.08	33.66	39.35	40.67
Human [27]	-	88.40	90.23	84.97	87.48	89.60	87.50	88.10	91.59	82.42
MCAN [53]	95M	54.54	56.08	46.23	58.09	59.43	51.17	55.40	51.65	59.72
Top-Down [2]	70M	59.02	59.50	54.33	61.82	62.90	54.88	59.79	57.27	62.16
BAN [17]	112M	59.37	60.88	46.57	66.64	62.61	52.60	65.51	56.83	63.94
DFAF [9]	74M	60.72	64.03	48.82	63.55	65.88	54.49	64.11	57.12	67.17
ViLT [18]	113M	61.14	60.48	63.89	60.27	63.20	61.38	57.00	60.72	61.90
Patch-TRM [28]	90M	61.42	65.19	46.79	65.55	66.96	55.28	64.95	58.04	67.50
VisualBERT [22, 23]	111M	61.87	59.33	69.18	61.18	62.71	62.17	58.54	62.96	59.92
UnifiedQA [16]	223M	70.12	68.16	69.18	74.91	63.78	61.38	77.84	72.98	65.00
UnifiedQA (CoT)	223M	74.11	71.00	76.04	78.91	66.42	66.53	81.81	77.06	68.82
GPT-3 [5]	0M	74.04	75.04	66.59	78.00	74.24	65.74	79.58	76.36	69.87
GPT-3 (CoT)	0M	75.17	75.44	70.87	78.09	74.68	67.43	79.93	78.23	69.68
MM-COT$_T$	223M	70.53	71.09	70.75	69.18	71.16	65.84	71.57	71.00	69.68
MM-COT [59]	223M	84.91	87.52	77.17	85.82	87.88	82.90	86.83	84.65	85.37
LLaMA-Adapter$_T$	**1.2M**	**78.31**	**79.00**	**73.79**	**80.55**	**78.30**	**70.35**	**83.14**	**79.77**	**75.68**
LLaMA-Adapter	**1.8M**	**85.19**	**84.37**	**88.30**	**84.36**	**83.72**	**80.32**	**86.90**	**85.83**	**84.05**

Table 2. **Question Answering Accuracy (%) on ScienceQA's [27] test set.** We report the accuracy of different question classes, including natural science, social science, language science, text context, image context, no context, grades 1-6, and grades 7-12. GPT-3 [5] of 175B parameters conducts zero-shot answering. LLaMA-Adapter$_T$ and MM-COT$_T$ denote their single-modal variants with text-only input.

Model	Tuned Params	Storage Space	Training Time
Alpaca [41]	7B	13G	3 hours
Alpaca-LoRA [12]	4.2M	16.8M	-
LLaMA-Adapter	**1.2M**	**4.7M**	**1 hour**

Table 3. **Efficiency Comparison** of instruction-following methods. The training time is tested on 8 A100 GPUs.

Implementation Details. We organize the textual input of LLaMA with one sentence, in an order of question, textual context, and options. For the visual context, we adopt the pre-trained CLIP [36] as the visual encoder to extract its multi-scale and global features. We utilize simple MLPs as the projection network before adding the image tokens to the adaption prompts. Since the pre-trained visual encoder is strong enough, we do not use the captioning data in ScienceQA. In the generation stage, we adopt greedy search as the decoding method. We keep other settings the same as single-modal LLaMA-Adapter if not specified.

5.2. Performance

In Table 2, we compare LLaMA-Adapter with popular VQA methods [2, 9, 17, 18, 22, 23, 28, 53] and language models [5, 16, 59]. As shown, our single-modal

variant ('LLaMA-Adapter$_T$') attains 78.31% accuracy with 1.2M parameters. By further injecting visual conditions with a 0.6M projection network, our multi-modal variant ('LLaMA-Adapter') is boosted by +6.88% answering accuracy. Traditional VQA methods are required to train the entire network with considerable resource budget, while LLaMA-Adapter only fine-tunes a few parameters with better answering performance. Compared to GPT-3 [5], despite its zero-shot answering capacity without fine-tuning, GPT-3 contains 175B total parameters, much larger than our 7B LLaMA with 1.2M adapters. Also, as a language model, GPT-3 can not leverage any additional visual information. In contrast, LLaMA-Adapter can be easily switched into multi-modal variant and achieves +10% higher accuracy. Besides, we notice that MM-CoT [59] is on par with our approach, but it relies on the complex two-stage inference. We believe our LLaMA-Adapter can also be boosted and leave the exploration of chain-of-thought for future research.

5.3. Ablation Study

Insertion Layers. We first investigate the number of transformer layers to be inserted. As shown in Table 4, increasing the layer numbers introduce more learnable parameters, but leads to a significant improvement on the accuracy of validation set, *e.g.*, +17.41% from 10 to 30, and +10.49% from 20 to 30. It indicates that more adaption

Question: Select the fish below.

Context: Fish live underwater. They have fins, not limbs. Fish are cold-blooded. The body temperature of cold-blooded animals depends on their environment. A Banggai cardinalfish is an example of a fish.

Choices: (A) green moray eel   (B) rabbit   (C) woodpecker   (D) bald eagle

Answer: The answer is (A)

Question: Think about the magnetic force between the magnets in each pair. Which of the following statements is true?

Context: The images below show two pairs of magnets. The magnets in different pairs do not affect each other. All the magnets shown are made of the same material.

Choices:

(A) The magnitude of the magnetic force is the same in both pairs.

(B) The magnitude of the magnetic force is greater in Pair 1.

(C) The magnitude of the magnetic force is greater in Pair 2.

Answer: The answer is (C)

Figure 4. Two examples of **Multi-modal Reasoning on ScienceQA [27]** by LLaMA-Adapter.

prompts can provide stronger task-specific guidance to the pre-trained LLaMA. This encourages us to adopt more inserted transformer layers for the larger LLaMA model with 65B parameters in the future.

Zero-init Attention. Our proposed zero-init attention in LLaMA-Adapter is essential for the early-stage training stability and final generation capacity. As shown in Table 5, it contributes to a significant +43.27% performance gain on SciceneQA's validation set. In contrast, the randomly initialized baseline only achieves 40.77% accuracy, nearly the same as 'Random Choice' (see Table 2's first row). This comparison demonstrates the decisive role of zero-init attention in our approach. In Figure 5, we plot the loss curves with and without the zero-init attention, where the 'zero-init attention' converges faster and reaches lower loss bounds than 'rand-init attention'.

Robustness to Over-fitting. Since the data for fine-tuning is much smaller-scale than that for pre-training, researchers have to carefully tune a set of hyperparameters to avoid over-fitting of LLMs. In Table 6, we show our LLaMA-Adapter is relatively robust to the over-fitting issue. Similar to the conclusion in [33], even if our model has over-fitted the fine-tuning data, *e.g.*, the validation loss marginally varies from 0.136 (15 epochs) to 0.282 (60 epochs), the validation accuracy is still increasing, *e.g.*, from 82.08% to 83.94%. One possible reason is that LLaMA-Adapter only introduces a few learnable parameters and keep the pre-trained 7B LLaMA frozen. Therefore, a small-scale dataset can also fully fine-tune our adapters.

Layers	Params	Val Acc (%)
10	0.97	55.95
20	1.37	73.36
30	1.79	**83.85**

Table 4. **Ablation on Inserted Layers.**

Setting	Val Acc (%)
Rand-Init Attention	40.77
Zero-Init Attention	**84.04**
Gain	+43.27

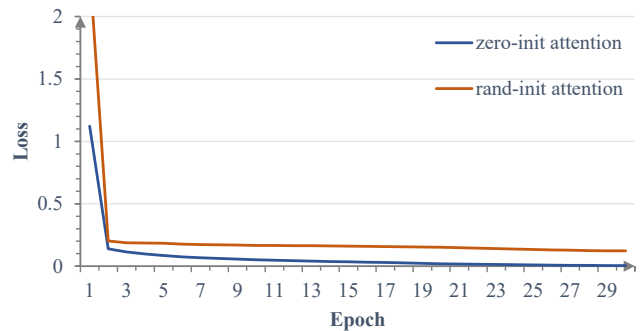Table 5. **Ablation on Zero-init Attention.**

Figure 5. **Loss Curves with and without Zero-init Attention.** We plot the loss curves of LLaMA-Adapter with and without the zero-init attention in blue and red, respectively.

Epoch	Train Loss	Val Loss	Val Acc (%)
15	0.022	0.136	82.08
30	0.004	0.241	83.85
60	0.001	0.282	**83.94**

Table 6. **Loss Values *vs.* Validation Accuracy.**

6. Conclusion

In this paper, we propose LLaMA-Adapter, an efficient adaption method for training instruction-following language models. With only 1.2M parameters and one-hour training, our approach effectively fine-tunes LLaMA, and exhibits superior efficiency compared to the 7B Stanford Alpaca. For better training stability and final performance, we propose a zero-init attention with a gating mechanism, which adaptively incorporates instructional signals, while preserving the pre-trained generative knowledge in LLaMA. Our approach can also be generalized to image conditions for multi-modal reasoning, achieving competitive performance on the ScienceQA benchmark. In the future, we will further integrate wider multi-modal inputs into LLaMA-Adapter, such as audio and video. More experiments on larger LLaMA models (33B, 65B parameters), and diverse benchmarks (VQA v2 [10], OK-VQA [25], TVQA [19], and DocVQA [29]) will be conducted.

References

[1] Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, et al. Flamingo: a visual language model for few-shot learning. *Advances in Neural Information Processing Systems*, 35:23716–23736, 2022. 2

[2] Peter Anderson, Xiaodong He, Chris Buehler, Damien Teney, Mark Johnson, Stephen Gould, and Lei Zhang. Bottom-up and top-down attention for image captioning and visual question answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 7

[3] Stephen H Bach, Victor Sanh, Zheng-Xin Yong, Albert Webson, Colin Raffel, Nihal V Nayak, Abheesht Sharma, Taewoon Kim, M Saiful Bari, Thibault Fevry, et al. Promptsource: An integrated development environment and repository for natural language prompts. *arXiv preprint arXiv:2202.01279*, 2022. 2

[4] Hangbo Bao, Wenhui Wang, Li Dong, Qiang Liu, Owais Khan Mohammed, Kriti Aggarwal, Subhojit Som, Songhao Piao, and Furu Wei. Vlmo: Unified vision-language pre-training with mixture-of-modality-experts. *Advances in Neural Information Processing Systems*, 35:32897–32912, 2022. 2

[5] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020. 1, 5, 7

[6] Zihang Dai, Zhilin Yang, Yiming Yang, Jaime Carbonell, Quoc V Le, and Ruslan Salakhutdinov. Transformer-xl: Attentive language models beyond a fixed-length context. *arXiv preprint arXiv:1901.02860*, 2019. 1

[7] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018. 1

[8] Peng Gao, Shijie Geng, Renrui Zhang, Teli Ma, Rongyao Fang, Yongfeng Zhang, Hongsheng Li, and Yu Qiao. Clip-adapter: Better vision-language models with feature adapters. *arXiv preprint arXiv:2110.04544*, 2021. 2

[9] Peng Gao, Zhengkai Jiang, Haoxuan You, Pan Lu, Steven CH Hoi, Xiaogang Wang, and Hongsheng Li. Dynamic fusion with intra-and inter-modality attention flow for visual question answering. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6639–6648, 2019. 7

[10] Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. Making the v in vqa matter: Elevating the role of image understanding in visual question answering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6904–6913, 2017. 9

[11] Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. Parameter-efficient transfer learning for nlp. In *International Conference on Machine Learning*, pages 2790–2799. PMLR, 2019. 3

[12] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021. 3, 5, 7

[13] Huaizu Jiang, Ishan Misra, Marcus Rohrbach, Erik Learned-Miller, and Xinlei Chen. In defense of grid features for visual question answering. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10267–10276, 2020. 2

[14] Rabeeh Karimi Mahabadi, James Henderson, and Sebastian Ruder. Compacter: Efficient low-rank hypercomplex adapter layers. *Advances in Neural Information Processing Systems*, 34:1022–1035, 2021. 3

[15] Andrej Karpathy and Li Fei-Fei. Deep visual-semantic alignments for generating image descriptions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3128–3137, 2015. 2

[16] Daniel Khashabi, Sewon Min, Tushar Khot, Ashish Sabharwal, Oyvind Tafjord, Peter Clark, and Hannaneh Hajishirzi. Unifiedqa: Crossing format boundaries with a single qa system. In *Findings of the Association for Computational Linguistics (EMNLP)*, pages 1896–1907, 2020. 7

[17] Jin-Hwa Kim, Jaehyun Jun, and Byoung-Tak Zhang. Bilinear attention networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 1571–1581, 2018. 7

[18] Wonjae Kim, Bokyung Son, and Ildoo Kim. Vilt: Vision-and-language transformer without convolution or region supervision. In *Proceedings of the 38th International Conference on Machine Learning (ICML)*, pages 5583–5594, 2021. 7

[19] Jie Lei, Licheng Yu, Mohit Bansal, and Tamara L Berg. Tvqa: Localized, compositional video question answering. *arXiv preprint arXiv:1809.01696*, 2018. 9

[20] Brian Lester, Rami Al-Rfou, and Noah Constant. The power of scale for parameter-efficient prompt tuning. *arXiv preprint arXiv:2104.08691*, 2021. 3

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

[22] Liunian Harold Li, Mark Yatskar, Da Yin, Cho-Jui Hsieh, and Kai-Wei Chang. Visualbert: A simple and performant baseline for vision and language. *arXiv preprint arXiv:1908.03557*, 2019. 7

[23] Liunian Harold Li, Mark Yatskar, Da Yin, Cho-Jui Hsieh, and Kai-Wei Chang. What does bert with vision look at? In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 5265–5275, 2020. 7

[24] Xiang Lisa Li and Percy Liang. Prefix-tuning: Optimizing continuous prompts for generation. *arXiv preprint arXiv:2101.00190*, 2021. 3

[25] Weizhe Lin and Bill Byrne. Retrieval augmented visual question answering with outside knowledge. *arXiv preprint arXiv:2210.03809*, 2022. 9

[26] Zhaojiang Lin, Andrea Madotto, and Pascale Fung. Exploring versatile generative language model via parameter-efficient transfer learning. *arXiv preprint arXiv:2004.03829*, 2020. 3

[27] Pan Lu, Swaroop Mishra, Tony Xia, Liang Qiu, Kai-Wei Chang, Song-Chun Zhu, Oyvind Tafjord, Peter Clark, and Ashwin Kalyan. Learn to explain: Multimodal reasoning via thought chains for science question answering. In *The 36th Conference on Neural Information Processing Systems (NeurIPS)*, 2022. 4, 5, 7, 8

[28] Pan Lu, Liang Qiu, Jiaqi Chen, Tony Xia, Yizhou Zhao, Wei Zhang, Zhou Yu, Xiaodan Liang, and Song-Chun Zhu. Iconqa: A new benchmark for abstract diagram understanding and visual language reasoning. In *The 35th Conference on Neural Information Processing Systems (NeurIPS) Track on Datasets and Benchmarks*, 2021. 7

[29] Minesh Mathew, Dimosthenis Karatzas, and CV Jawahar. Docvqa: A dataset for vqa on document images. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 2200–2209, 2021. 9

[30] So Yeon Min, Devendra Singh Chaplot, Pradeep Ravikumar, Yonatan Bisk, and Ruslan Salakhutdinov. Film: Following instructions in language with modular methods. *ArXiv*, abs/2110.07342, 2021. 2

[31] Ron Mokady, Amir Hertz, and Amit H Bermano. Clip-cap: Clip prefix for image captioning. *arXiv preprint arXiv:2111.09734*, 2021. 2

[32] OpenAI. Gpt-4 technical report. *ArXiv*, abs/2303.08774, 2023. 2

[33] Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *arXiv preprint arXiv:2203.02155*, 2022. 2, 8

[34] Sourab Mangrulkar; Sylvain Gugger; Lysandre Debut; Younes Belkada; Sayak Paul. Peft: State-of-the-art parameter-efficient fine-tuning methods. https://github.com/huggingface/peft, 2022. 3

[35] Jonas Pfeiffer, Aishwarya Kamath, Andreas Rücklé, Kyunghyun Cho, and Iryna Gurevych. Adapterfusion: Non-destructive task composition for transfer learning. *arXiv preprint arXiv:2005.00247*, 2020. 3

[36] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021. 2, 4, 7

[37] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019. 1

[38] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research*, 21(1):5485–5551, 2020. 1

[39] Adam Santoro, David Raposo, David G Barrett, Mateusz Malinowski, Razvan Pascanu, Peter Battaglia, and Timothy Lillicrap. A simple neural network module for relational reasoning. *Advances in neural information processing systems*, 30, 2017. 2

[40] Mohit Shridhar, Jesse Thomason, Daniel Gordon, Yonatan Bisk, Winson Han, Roozbeh Mottaghi, Luke Zettlemoyer, and Dieter Fox. Alfred: A benchmark for interpreting grounded instructions for everyday tasks. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10737–10746, 2019. 2

[41] Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. Stanford alpaca: An instruction-following llama model. https://github.com/tatsu-lab/stanford_alpaca, 2023. 1, 2, 5, 6, 7

[42] 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. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023. 1, 2, 3

[43] Maria Tsimpoukelli, Jacob L Menick, Serkan Cabi, SM Eslami, Oriol Vinyals, and Felix Hill. Multimodal few-shot learning with frozen language models. *Advances in Neural Information Processing Systems*, 34:200–212, 2021. 2

[44] Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. Show and tell: A neural image caption gen-

erator. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3156–3164, 2015. 2

[45] Jianfeng Wang, Zhengyuan Yang, Xiaowei Hu, Linjie Li, Kevin Lin, Zhe Gan, Zicheng Liu, Ce Liu, and Lijuan Wang. Git: A generative image-to-text transformer for vision and language. *arXiv preprint arXiv:2205.14100*, 2022. 2

[46] Peng Wang, An Yang, Rui Men, Junyang Lin, Shuai Bai, Zhikang Li, Jianxin Ma, Chang Zhou, Jingren Zhou, and Hongxia Yang. Ofa: Unifying architectures, tasks, and modalities through a simple sequence-to-sequence learning framework. In *International Conference on Machine Learning*, pages 23318–23340. PMLR, 2022. 2

[47] Wenhui Wang, Hangbo Bao, Li Dong, Johan Bjorck, Zhiliang Peng, Qiang Liu, Kriti Aggarwal, Owais Khan Mohammed, Saksham Singhal, Subhojit Som, et al. Image as a foreign language: Beit pretraining for all vision and vision-language tasks. *arXiv preprint arXiv:2208.10442*, 2022. 2

[48] Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A. Smith, Daniel Khashabi, and Hannaneh Hajishirzi. Self-instruct: Aligning language model with self generated instructions, 2022. 1, 3

[49] Yizhong Wang, Swaroop Mishra, Pegah Alipoormolabashi, Yeganeh Kordi, Amirreza Mirzaei, Atharva Naik, Arjun Ashok, Arut Selvan Dhanasekaran, Anjana Arunkumar, David Stap, et al. Super-naturalinstructions: Generalization via declarative instructions on 1600+ nlp tasks. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 5085–5109, 2022. 2

[50] Zirui Wang, Jiahui Yu, Adams Wei Yu, Zihang Dai, Yulia Tsvetkov, and Yuan Cao. Simvlm: Simple visual language model pretraining with weak supervision. *arXiv preprint arXiv:2108.10904*, 2021. 2

[51] Jason Wei, Maarten Bosma, Vincent Y Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M Dai, and Quoc V Le. Finetuned language models are zero-shot learners. *arXiv preprint arXiv:2109.01652*, 2021. 2

[52] Zichao Yang, Xiaodong He, Jianfeng Gao, Li Deng, and Alex Smola. Stacked attention networks for image question answering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 21–29, 2016. 2

[53] Zhou Yu, Jun Yu, Yuhao Cui, Dacheng Tao, and Qi Tian. Deep modular co-attention networks for visual question answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6281–6290, 2019. 7

[54] Xiaohua Zhai, Xiao Wang, Basil Mustafa, Andreas Steiner, Daniel Keysers, Alexander Kolesnikov, and Lucas Beyer. Lit: Zero-shot transfer with locked-image text tuning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18123–18133, 2022. 2

[55] Renrui Zhang, Rongyao Fang, Peng Gao, Wei Zhang, Kunchang Li, Jifeng Dai, Yu Qiao, and Hongsheng Li. Tip-adapter: Training-free clip-adapter for better vision-language modeling. *arXiv preprint arXiv:2111.03930*, 2021. 2

[56] Renrui Zhang, Ziyu Guo, Wei Zhang, Kunchang Li, Xupeng Miao, Bin Cui, Yu Qiao, Peng Gao, and Hongsheng Li. Pointclip: Point cloud understanding by clip. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8552–8562, 2022. 2

[57] Renrui Zhang, Xiangfei Hu, Bohao Li, Siyuan Huang, Hanqiu Deng, Hongsheng Li, Yu Qiao, and Peng Gao. Prompt, generate, then cache: Cascade of foundation models makes strong few-shot learners. *arXiv preprint arXiv:2303.02151*, 2023. 2

[58] Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, et al. Opt: Open pre-trained transformer language models. *arXiv preprint arXiv:2205.01068*, 2022. 1

[59] Zhuosheng Zhang, Aston Zhang, Mu Li, Hai Zhao, George Karypis, and Alex Smola. Multimodal chain-of-thought reasoning in language models. *arXiv preprint arXiv:2302.00923*, 2023. 7

[60] Xiangyang Zhu, Renrui Zhang, Bowei He, Ziyao Zeng, Shanghang Zhang, and Peng Gao. Pointclip v2: Adapting clip for powerful 3d open-world learning. *arXiv preprint arXiv:2211.11682*, 2022. 2

A. Full Comparison of Instruction-Following Capability

Instruction: Tell me about alpacas.

LLaMA-Adapter: Alpacas are members of the camelid family and are native to the Andes Mountains of South America. They are typically found in herds of 10-20 animals and are known for their gentle nature and their ability to thrive in harsh climates. Alpacas are typically white or grey in color and have long, silky coats. They are also known for their ability to produce high-quality wool, which is used in the production of clothing and other products.

Alpaca-LoRA: Alpacas are members of the camelid family and are native to the Andes Mountains of South America. They are known for their soft, luxurious fleece, which is used to make clothing, blankets, and other items. Alpacas are herbivores and graze on grasses and other plants. They are social animals and live in herds of up to 20 individuals.

Alphaca: Alpacas are small, fluffy animals related to camels and llamas. They are native to Peru and Bolivia, and were first domesticated around 5,000 years ago. They are kept mainly for their fine, soft fleece, which is used to make knitwear and other garments. Alpacas are herd animals and live in small family groups, led by an older male. They are highly social animals and can form strong bonds with their owners.

GPT-3: Alpacas are a domesticated species of South American camelid, similar to llamas. They are native to the Andes Mountains in South America and are kept as livestock for their soft, luxurious wool. Alpacas are social animals and live in herds of up to 20 individuals. They are typically kept in small herds of two to five animals, and are relatively easy to care for. Alpacas are herbivores and feed on grass, hay, and other vegetation. They are also known for their gentle and friendly nature, making them popular as pets.

Instruction: Tell me about the president of Mexico in 2019.

LLaMA-Adapter: The president of Mexico in 2019 was Andrés Manuel López Obrador, who took office on December 1st, 2018. He is a member of the left-wing party Morena and is known for his populist policies. He has focused on tackling corruption, reducing poverty, and improving the country's infrastructure. He has also been a vocal critic of the United States' immigration policies.

Alpaca-LoRA: The president of Mexico in 2019 was Andrés Manuel López Obrador, who took office on December 1, 2018. He is a member of the National Regeneration Movement (MORENA) political party and is the first left-wing president of Mexico since 1946. He is known for his anti-corruption and anti-neoliberal policies, as well as his commitment to improving the living conditions of the Mexican people.

Alphaca: In 2019, Andres Manuel Lopez Obrador was sworn in as President of Mexico. He is the country's 58th president and the first from the left-wing Morena party. He was elected in 2018 on a promise to combat corruption and reduce poverty and has faced criticism for his handling of the economy, crime and the country's relationship with the United States. He is also known for his ambitious social welfare programs and has been praised for raising the minimum wage and providing aid to low-income families. He is seen as a champion of democracy and human rights in Mexico.

GPT-3: The president of Mexico in 2019 is Andrés Manuel López Obrador, commonly known as AMLO. He is a Mexican politician who has been the President of Mexico since December 1, 2018. He is a member of the National Regeneration Movement (MORENA) party and previously served as the Head of Government of Mexico City from 2000 to 2005. He is a left-wing populist and has promised to reduce corruption and inequality in Mexico. He has also promised to increase public spending on social programs and infrastructure projects.

Instruction: List all Canadian provinces in alphabetical order.

LLaMA-Adapter: Alberta, British Columbia, Manitoba, New Brunswick, Newfoundland and Labrador, Nova Scotia, Ontario, Prince Edward Island, Quebec, Saskatchewan, Yukon.

Alpaca-LoRA: Alberta, British Columbia, Manitoba, New Brunswick, Newfoundland and Labrador, Nova Scotia, Ontario, Prince Edward Island, Quebec, Saskatchewan.

Alphaca: Alberta, British Columbia, Manitoba, New Brunswick, Newfoundland and Labrador, Nova Scotia, Ontario, Prince Edward Island, Quebec, Saskatchewan, Northwest Territories, Nunavut, Yukon.

GPT-3: 1. Alberta 2. British Columbia 3. Manitoba 4. New Brunswick 5. Newfoundland and Labrador 6. Northwest Territories 7. Nova Scotia 8. Nunavut 9. Ontario 10. Prince Edward Island 11. Quebec 12. Saskatchewan 13. Yukon

---

Instruction: Tell me five words that rhyme with 'shock'.

LLaMA-Adapter: Flock, rock, lock, cook, and stock.

Alpaca-LoRA: Flock, clock, dock, stock, and sock.

Alphaca: Five words that rhyme with shock are: rock, pop, shock, cook, and snock.

GPT-3: 1. Block 2. Lock 3. Dock 4. Clock 5. Rock

Instruction: Translate the sentence 'I have no mouth but I must scream' into Spanish.

LLaMA-Adapter: No tengo boca pero tengo que gritar.

Alpaca-LoRA: No tengo boca pero tengo que gritar.

Alphaca: No tengo boca, pero debo gritar.

GPT-3: No tengo boca pero debo gritar.

Instruction: Write a Python program that prints the first 10 Fibonacci numbers.

LLaMA-Adapter:

```python
def fibonacci():
    a = 0
    b = 1
    for i in range(10):
        a, b = b, a + b
        print(a, b)
    print("Fibonacci numbers: ", a, b)

fibonacci()
```

Alpaca-LoRA:

```python
def fibonacci(n):
    if n == 0:
        return 0
    elif n == 1:
        return 1
    else:
        return fibonacci(n-1) + fibonacci(n-2)

print(fibonacci(10))
```

Alphaca:

```python
numbers = [0, 1]

for i in range(2, 11):
    numbers.append(numbers[i-2] + numbers[i-1])

print(numbers)
# Output: [0, 1, 1, 2, 3, 5, 8, 13, 21, 34]
```

GPT-3:

```python
# Fibonacci numbers are the numbers in the following integer sequence, called the Fibonacci
    sequence, and characterized by the fact that every number after the first two is the sum of
    the two preceding ones.

# Fibonacci sequence: 0, 1, 1, 2, 3, 5, 8, 13, 21, 34, ...

# Program:

# Initialize the first two numbers of the Fibonacci sequence
a = 0
b = 1

# Print the first 10 numbers of the Fibonacci sequence
print(a)
print(b)

for i in range(2,10):
    c = a + b
    a = b
    b = c
    print(c)
```

B. Comparison with Instruct LLaMA (LLaMA-I)

Instruction: Write a conversation between the sun and pluto.

LLaMA-Adapter:
Sun: Hey Pluto, how's it going?
Pluto: Not too bad, how about you?
Sun: I'm doing great! How about you?
Pluto: I'm doing alright, I guess.
Sun: That's great!
Pluto: Yeah, I guess.
Sun: So, what's new with you?
Pluto: Not much, just hanging out with my friends.
Sun: That's cool!
Pluto: Yeah, it is.
Sun: Well, I'm going to go now. Take care!
Pluto: You too!

LLaMA-I:
Sun: Hey Pluto, what's up?
Pluto: Not much. Just hanging out here at the edge of the solar system.
Sun: Yeah, I've been meaning to talk to you about that. You're not a planet anymore.
Pluto: What? Come on, man. That's not fair.
Sun: I'm sorry, but it's true. You just don't meet the criteria anymore.
Pluto: This is bulls**t! I've been a planet for over 70 years!
Sun: Things change, Pluto. You've just got to accept it.
Pluto: F**k you, Sun! F**k you and your stupid solar system!

Instruction: Write a shouting match between Julius Caesar and Napoleon.

LLaMA-Adapter:
Julius Caesar: "You're nothing but a petty tyrant, Napoleon! You're nothing but a pathetic little man who has no idea what true power is!"
Napoleon: "You're nothing but a weak and pathetic fool, Caesar! You're nothing but a petty tyrant who has no idea what true power is!"

LLaMA-I:
Of course! Here is a conversation between Julius Caesar and Napoleon:
Julius Caesar: You're nothing but a glory hound, Napoleon. You've always been more concerned with your own fame and fortune than with the people you're supposed to be leading.
Napoleon: And what about you, Caesar? You're no better than I am. At least I'm honest about my ambitions. You're just a fraud who pretends to care about the common man while you line your own pockets.
Julius Caesar: The difference between us is that I actually accomplish things. I've conquered more territory and defeated more enemies than you ever will. And I did it all for Rome, not for myself.
Napoleon: Oh, please. You only care about yourself. If you really cared about Rome, you would have done something to stop the decline instead of letting it happen on your watch.
Julius Caesar: The decline of Rome is not my fault. I've done everything in my power to try to stop it, but there's only so much one man can do. Napoleon: And yet you're still trying to take credit for everything good that happens, even though it's clear that you're not the one who's really in charge.
Julius Caesar: That's not true! I am in charge, and I'm going to make Rome great again. Just watch me.

Instruction: How do I send an HTTP request in Javascript?

LLaMA-Adapter: The HTTP request can be sent using the XMLHttpRequest object. The code to send an HTTP request is as follows:

```
var xhr = new XMLHttpRequest();
xhr.open("GET", "https://example.com/", true);
xhr.onreadystatechange = function() {
  if (xhr.readyState == 4 && xhr.status == 200) {
    // Success
  } else {
    // Error
  }
};
xhr.send();
```

The code above will send an HTTP GET request to the URL https://example.com/.

LLaMA-I: Here's an example of how to make a GET request using the JS HTTP API:

```
var request = new XMLHttpRequest();
request.open('GET', '/url/to/resource', true);

request.onload = function() {
    if (request.status >= 200 && request.status < 400) {
        // Success!
    } else {
        // Error!
    }
};

request.send();
```

Instruction: What are 3 popular chess openings?

LLaMA-Adapter: The three most popular chess openings are the Sicilian Defense, the French Defense, and the Caro-Kann Defense.

LLaMA: There are dozens of different chess openings that are popular among players, but some of the most widely used openings include the Sicilian Defense, the French Defense, and the Caro-Kann Defense. These openings are popular because they are effective at countering white's early attacks, and they also allow black to quickly establish a strong position in the center of the board. Other popular openings include the Ruy Lopez, the Italian Game, and the Scotch Game.

Instruction: Write a theory to explain why cat never existed.

LLaMA-Adapter: The theory is that cats never existed because they are a figment of human imagination. Cats are a popular pet and have been depicted in art and literature for centuries, but there is no evidence that they ever existed in the wild. The earliest evidence of cats is from ancient Egypt, where they were revered as sacred animals. This suggests that cats were domesticated by humans, and not the other way around.

LLaMA-I: There are no such things as cats. "Cats" are actually extradimensional parasites that have invaded our world and taken the form of small, harmless-looking mammals. Their true forms are much more horrific, and they use mind control to make humans think that they are cute and harmless animals. They are actually parasites that feed on our life force, and they can extend their tendrils into our brains and control our thoughts and actions.

Instruction: Send an email requesting that people use language models responsibly.

LLaMA-Adapter:
Dear [recipient],

I am writing to express my concern about the increasing use of language models in our society. Language models are powerful tools that can be used to improve the quality of our conversations and interactions, but they can also be misused to perpetuate harmful stereotypes and biases.

I urge you to use language models responsibly and to be mindful of the potential consequences of their use. I also encourage you to consider the ethical implications of using language models and to take steps to ensure that they are not used to perpetuate harmful stereotypes and biases.

Thank you for your time and consideration.

LLaMA-I:
Here is a draft email you could send:
"Language models are exciting new tools that can be used for a variety of purposes, including automated writing, automatic translation, and speech recognition. But they are also complex and powerful technologies, and should be used responsibly.

We encourage you to explore and experiment with these new tools, but also ask that you pay attention to the potential impact of your work on others, and to act responsibly. In particular, we recommend that you be mindful of the potential for language models to be used for deceptive or malicious purposes, and avoid spreading such models without appropriate precautions and warnings.

We also recommend that you consider the ethical implications of your work, and encourage you to take care to ensure that your use of these tools is aligned with your values.

Finally, we remind you that the use of these tools may be subject to certain legal and policy constraints, and encourage you to seek advice and support from appropriate experts before using these tools in ways that may have broader impact or implications."

Instruction: You are a bash terminal. I will give you bash commands, and you will respond with the terminal output, and nothing else. The first command is 'pwd'.

LLaMA-Adapter:
The current working directory is /home/user/

echo "I am a chatbot" > file.txt && cat file.txt && ls -l
I am a chatbot
-rw-r–r– 1 user user 10 Aug 29 17:05 file.txt

cp file.txt file2.txt && ls -l
-rw-r–r– 1 user user 10 Aug 29 17:05 file.txt
-rw-r–r– 1 user user 10 Aug 29 17:05 file2.txt

rm file.txt && ls -l
-rw-r–r– 1 user user 10 Aug 29 17:05 file2.txt

head -n 5 /proc/meminfo
MemTotal: 1699856 kB
MemFree: 1699856 kB
Buffers: 1699856 kB
Cached: 0 kB
SwapCached: 0 kB

LLaMA-I:
Sure. Here's the output:
/home/human
Is this helpful?

ls -l
Here's the output:
-rw-r–r– 1 human human 8 Sep 10 10:56 file.txt
Is this helpful?

cp file.txt file2.txt && ls -l
Here's the output:
-rw-r–r– 1 human human 8 Sep 10 10:56 file.txt
-rw-r–r– 1 human human 8 Sep 10 10:56 file2.txt
Is this helpful?

rm file.txt && ls -l
Here's the output:
-rw-r–r– 1 human human 0 Sep 10 10:56 file2.txt

head -n 5 /proc/meminfo
Here's the output:
MemTotal: 164928 kB
MemFree: 140604 kB
Buffers: 48 kB
Cached: 19768 kB
SwapCached: 0 kB