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

Renrui Zhang[*1,2], **Jiaming Han**[*1], **Chris Liu**[*1], **Peng Gao**[*†‡1], **Aojun Zhou**[2]
Xiangfei Hu[1], **Shilin Yan**[1], **Lu Pan**[3], **Hongsheng Li**[†2], **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 word tokens at higher transformer layers. Then, a zero-initialized attention mechanism with zero gating is proposed, which adaptively injects the new instructional cues into LLaMA, while effectively preserves its pre-trained knowledge. With our efficient training, LLaMA-Adapter can generate high-quality responses, comparable to Alpaca with fully fine-tuned 7B parameters. Besides language commands, our approach can be simply extended to multi-modal instructions for learning image-conditioned LLaMA model, which achieves superior reasoning performance on ScienceQA and COCO Caption benchmarks. Furthermore, we also evaluate the zero-initialized attention mechanism for fine-tuning other pre-trained models (ViT, RoBERTa) on traditional vision and language tasks, demonstrating the superior generalization capacity of our approach. Code is released at https://github.com/OpenGVLab/LLaMA-Adapter.

1 Introduction

Large-scale Language Models (LLMs) [13, 52, 73, 53, 15] 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 [2] and GPT-3.5 (text-davinci-003) [4]. Following instructions in natural language, they can generate professional and contextual responses in a conversational way. However, the further prevalence of instruction models is largely impeded by the closed-source restriction and high development costs.

To alleviate this, Stanford Alpaca [60] proposes to fine-tune an LLM, i.e., LLaMA [61] into an instruction-following model, which is affordable and replicable. Starting from 175 human-written instruction-output pairs [62], 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 instruction 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.

[*] Equal contribution [†] Corresponding author [‡] Project leader

Preprint. Under review.

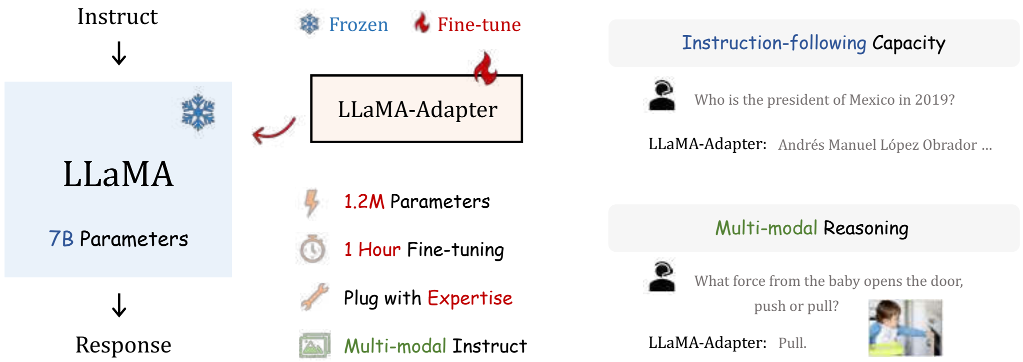

Figure 1: **Characteristics of LLaMA-Adapter.** Our lightweight adaption method efficiently fine-tunes LLaMA [61] 7B model with only 1.2M learnable parameters within one hour. After training, LLaMA-Adapter exhibits superior instruction-following and multi-modal reasoning capacity.

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 freeze the entire LLaMA model with superior resource efficiency. 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 the frozen LLaMA. To avoid noise from adaption prompts at the early training stage, we modify the vanilla attention mechanisms at inserted layers to be zero-initialized 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 our lightweight adaption modules with zero-initialized gating, the training convergence of LLaMA-Adapter costs less than one hour on 8 A100 GPUs, which are 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 Instruction.** Besides textual instruction, our approach can also take images as input for multi-modal reasoning. By adding image tokens into adaption prompts, LLaMA-Adapter performs competitively on ScienceQA [41] and COCO Caption [8] benchmarks.

In addition to instruction-following models, our zero-initialized attention can be generalized to other vision and language models for parameter-efficient fine-tuning. For vision models, we utilize our approach to fine-tune a pre-trained ViT [16] for downstream image classification, obtaining superior performance on VTAB-1k [67] benchmark over various image distributions. For other language models, we evaluate our fine-tuning efficacy on ReBERTa [40] for extractive question answering, which achieves leading results on SQuAD [54] v1.1 and v2.0 benchmarks. By these experiments, we demonstrate the effectiveness of LLaMA-Adapter for traditional vision and language tasks.

2 Related Work

Instruction-Following Language Models. The subfield of language models learning instruction-following capabilities aims to generate responses based on natural language commands, which have been extensively researched in language [64, 63, 3, 46], and multi-modality [59, 42] domains. These methods normally enhance the pre-trained LLMs by fine-tuning them using high-quality instruction-output data pairs. Such fine-tuning process boosts the model to better comprehend user intentions and follow instructions more accurately. Therein, FLAN [64] introduces an instruction tuning method that outperforms non-tuned LLMs in unseen tasks. PromptSource [3] provides a development environment with a web-based GUI, which creates and manages natural language prompts for zero-shot and gradient-based few-shot learning. SUP-NATINST [63] establishes a large benchmark of 1,616 diverse language tasks, and adopts a multi-task training on the T5 model. InstructGPT [46] demonstrates significant improvement of the instruction-following power, and is probably integrated into the closed-source GPT-3.5 [4] and GPT-4 [45]. Stanford Alpaca [60] fine-tunes all the 7B parameters of an LLM, i.e., LLaMA [61] in an end-to-end manner, which is open-source and replicable. However, this full-model fine-tuning can be inefficient in both time and memory, limiting its transferability to downstream applications. In contrast, our LLaMA-Adapter aims to fine-tune only lightweight adapters on top of the frozen LLaMA, other than updating parameters of the entire model. Compared to a concurrent work Alpaca-LoRA [1], our approach further reduces the computational demands, and can be generalized to follow visual instructions for multi-modal reasoning.

Parameter-Efficient Fine-Tuning. The pre-training and fine-tuning paradigms have been proven to be highly effective in different language and vision tasks. Compared to full fine-tuning, Parameter-Efficient Fine-Tuning (PEFT) [47] methods freeze most parameters of pre-trained models, and can still exhibit comparable capabilities on downstream tasks. Various PEFT techniques have been explored, including prompt tuning [35, 30, 39, 38, 50, 72], Low-Rank Adaptation (LoRA) [23, 69, 20], and adapters [22, 48, 37, 9, 55]. Prompt tuning appends a collection of trainable prompt tokens to pre-trained large models, which are inserted either to the input embeddings only [30, 39], or to all of the intermediate layers [35, 38]. LoRA [23] introduces trainable rank decomposition matrices into each network weights [25], which have indicated promising fine-tuning ability on large generative models [12, 61]. Adapters [22] insert lightweight adaption modules into each layer of the pre-trained transformer and have been extended across numerous domains [19, 18, 70, 71]. In this paper, we propose a new PEFT method, LLaMA-Adapter, specially designed for LLaMA [61] and instruction-following fine-tuning. Existing 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 values at early training stages. To this end, LLaMA-Adapter adopts a zero-initialized attention with gating factors to well mitigate such a issue, which progressively incorporates the instructional cues with the frozen LLaMA. Moreover, we verify the effectiveness of our approach to fine-tune large models in other domains. Aided by the adaption prompts with zero gating, our efficient fine-tuning of ViT [16] and RoBERTa [40] exhibit competitive downstream performance respectively on vision and language tasks, demonstrating superior generalization capacity.

3 LLaMA-Adapter

In Section 3.1, we first introduce how to insert the learnable adaption prompts into LLaMA's [61] transformer. Then, we present the details of zero-initialized attention mechanisms with zero gating in Section 3.2, and generalize LLaMA-Adapter for multi-modal reasoning in Section 3.3. Finally, we extend our approach for efficient fine-tuning of vision and vision-language models in Section 3.4.

3.1 Learnable Adaption Prompts

"C" is Embedding
Dimension
For BERT, say 768

Given 52K instruction-output data [62] and a pre-trained LLaMA [61] 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.

Taking the l-th inserted layer as an example ($l \leq L$), we denote the M-length word tokens as $T_l \in \mathbb{R}^{M\times C}$, which represent the input instruction and the already generated response. The learnable 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 the subsequent contextual response via attention layers in the transformer block.

3.2 Zero-initialized 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-initialized 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 the input tokens into queries, keys, and values as

"C": Embedding Dimension

"l": One of the Inserted Adapter Layer

P_l: Prompt matrix "l"

[P_L; T_l]:

Matrix of size ["K" * Embedding Dimension]

-CONCAT WITH-

Matrix of of size [Input Sequence Legth (M) * Embedding Dimension (C)]

$$Q_l = \text{Linear}_\text{q}(\ t_l\ ); \tag{2}$$

$$K_l = \text{Linear}_\text{k}(\ [P_l;\ T_l;\ t_l]\ ); \tag{3}$$

$$V_l = \text{Linear}_\text{v}(\ [P_l;\ T_l;\ t_l]\ ). \tag{4}$$

Then, the attention scores of Q_l and K_l 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 the new word t_l and all $K+M+1$ tokens. Meanwhile, S_l can be reformulated by two components as

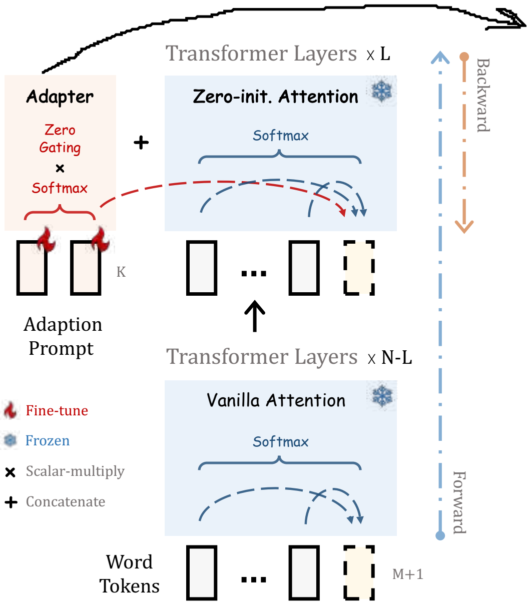

Figure 2: **Details of LLaMA-Adapter.** We insert lightweight adapters with learnable prompts into L out of N transformer layers of LLaMA. To progressively learn the instructional knowledge, we adopt zero-initialized attention with gating mechanisms for stable training in early stages.

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

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 generating 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

It is saying that when the Adapter is immature, keep it's incluence 0 and Original Attention is used

When it learns with time, gating will increase it's influence and final Attention will be a % of Original & learned Attention

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

The separate softmax functions ensure the second term to be irrelevant to the adaption prompts. When g_l is close to zero, it can mostly convey the originally pre-trained knowledge of LLaMA to token t_l for a creditable generation. In practice, we adopt multiple g_l to be independently learned for different heads within the attention, benefiting the learning diversity of multi-head mechanisms.

Finally, we calculate the output of the l-th 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-initialized attention, the adaption prompts can progressively inject the newly acquired instructional signals into the transformer, while simultaneously incorporating the pre-trained knowledge of LLaMA to provide high-quality responses.

3.3 Multi-modal Reasoning

Apart from 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 [41] as examples, which is analogous to the COCO

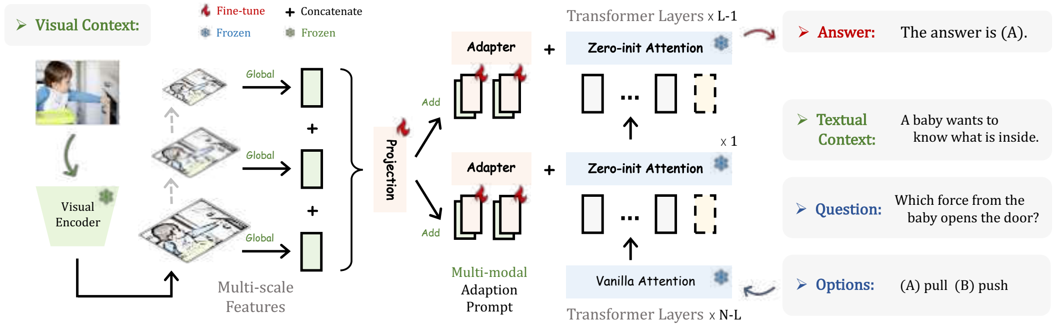

Figure 3: **Multi-modal Reasoning of LLaMA-Adapter.** On ScienceQA benchmark [41], LLaMA-Adapter is extended to a multi-modal variant for image-conditioned question answering. Given an image as the visual context, we acquire the global image token by multi-scale aggregation, and element-wisely add it onto the adaption prompts for visual instruction following.

Caption dataset [8]. Given **visual** and **textual contexts**, along with the corresponding **question** and **options**, the model is required to conduct multi-modal understanding to give the correct **answer**.

For an input image as the visual context, we first leverage a pre-trained visual encoder, e.g., CLIP [51], 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

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

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

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

where P_l^v denotes the adaption prompt incorporating 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.

3.4 Zero-initialized Attention for other Large Models

Our approach, i.e., adaption prompts with zero-initialized attention, is not limited to the domain of instruction models, and can be further utilized to fine-tune large models in traditional vision and language tasks, exerting superior generalization capacity.

Vision Models. We select a pre-trained ViT [16] as the foundation vision model for downstream image classification tasks. Similar to LLaMA, we insert the adaption prompts as prefix into the topmost L transformer layers in ViT, and modify the attention operations to be zero-initialized at all inserted layers. By increasingly injecting the downstream visual semantics, we only introduce a few parameters on top of the frozen ViT, and attain comparable classification accuracy to full fine-tuning on VTAB-1k [67] benchmark, which indicates our attention operator's efficacy in vision domains.

Language Models. We utilize RoBERTa [40] pre-trained on large-scale unlabeled text corpus, and evaluate our proposed zero-initialized attention on SQuAD [54] benchmark for extractive question answering. We implement the zero-initialized attention on top of P-tuning v2 [38], a prompt tuning method for efficiently adapting large language models. Likewise, we only enable the prompt tokens in P-tuning v2 and our zero gating factors to be learnable during fine-tuning. The leading results demonstrate our superiority for traditional language tasks. Please refer to Supplementary Material for applying zero-initialized attention mechanisms to more large models and tasks.

Instruction: Tell me about alpacas.

Alpaca: 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. They are highly social animals and can form strong bonds with their owners.

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.

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

Alpaca: No tengo boca, pero debo gritar.

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

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

Alpaca:

```python
numbers = [0, 1]
for i in range(2, 11):
    numbers.append(numbers[i-2] + numbers[i-1])
print(numbers)
```

LLaMA-Adapter:

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

Figure 4: **Instruction-following Comparison** between Alpaca [60] and LLaMA-Adapter.

4 Experiment

In Section 4.1, we first evaluate the instruction-following capacity of LLaMA-Adapter. Then, we present our multi-modal performance on ScienceQA [41] benchmark in Section 4.2, and conduct ablation study on ScienceQA's validation set in Section 4.3. Finally, we report the fine-tuning results of our approach on other vision and language models in Section 4.4.

4.1 Instruction-following Evaluation

Settings. Following Stanford Alpaca [60], we utilize 52K instruction-following data for training, which is extended from 175 instruction-output pairs [62]. We fine-tune LLaMA-Adapter on 8 A100 GPUs for 5 epochs. The warmup epochs, batch size, learning rate, and weight decay are set to 2, 64, 0.009, and 0.02, respectively. By default, we utilize the pre-trained LLaMA model with 7B parameters and $N = 32$ transformer layers. We adopt a prompt length $K = 10$ and insert the adaption prompts into the last $L = 30$ layers. In the generation stage, we adopt *top-p* sampling [21] as the default decoding method with a temperature 0.1 and a *top-p* $= 0.75$. For quantitative evaluation [10], we ask GPT-4 [45] to assess the response quality of instruction-following models on 80 questions. Since we observed that GPT-4 has a preference to give higher scores to the first response in comparison, we also switch the position of two responses, resulting in a total of 160 evaluation items.

Performance. We compare the generated responses of LLaMA-Adapter and Alpaca [60] in Figure 4, and report the quantitative results in Figure 6. Please refer to Supplementary Material for a full comparison with Alpaca-LoRA [1], GPT-3 [4], and LLaMA-I [61]. For different kinds of instructions