

## **Contents**

<b>A More Instruction-following Comparisons for Finetuning Techniques</b>	<b>2</b>
<b>B Examples of LLaMA-Excitor’s responses on Alpaca-Eval</b>	<b>3</b>
<b>C Supplemental Results on QA tasks and The Property of Not Forgetting (based on LLaMA2)</b>	<b>4</b>
<b>D Examples of Image Captioning on MSCOCO</b>	<b>6</b>
<b>E More Ablations</b>	<b>7</b>

## A. More Instruction-following Comparisons for Finetuning Techniques

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

**LLaMA7B:**

```
fibonacci(10)
[1, 1, 2, 3, 5, 8, 13, 21, 34, 55]
```

**Explanation:**

The fibonacci function takes a number as an argument and returns the next number in the Fibonacci sequence.

**Full Fine-Tuning:**

```
def print_fibonacci_10(n):
    if n == 0:
        print("0")
    elif n == 1:
        print("1")
    else:
        print("1" + str(Fibonacci(n - 1)) + "1")
```

**Alpaca-LoRA:**

```
def fibonacci(n):
    if n == 0:
        return 0
    elif n == 1:
        return 1
    else:
        return fibonacci(n - 1) + fibonacci(n - 2)
print(fibonacci(10))
```

**LLaMA-Adapter:**

```
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()
```

**LLaMA-Excitor:**

```
def fibonacci(n):
    if n == 0:
        return 0
    elif n == 1:
        return 1
    else:
        return fibonacci(n - 1) + fibonacci(n - 2)
print(fibonacci(10))
```

**Output:**

```
[0, 1, 1, 2, 3, 5, 8, 13, 21, 34]
```

**Explanation:**

The first line of the program defines a function named 'fibonacci' that takes a single argument, 'n'. The function returns the first 'n' Fibonacci numbers.

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

**LLaMA7B:**

A shouting match between Julius Caesar and Napoleon.

**Full Fine-Tuning:**

Julius Caesar: "I am Caesar! I am the greatest!"  
Napoleon: "I am Napoleon! I am the greatest!"

**Alpaca-LoRA:**

Julius Caesar and Napoleon were having a heated argument. Julius was insisting that the Roman Empire was the greatest empire of its time and that its power should be respected. Napoleon, on the other hand, argued that the Roman Empire was outdated and that his own French Empire was far superior. Julius was incensed and demanded that Napoleon acknowledge his superiority. Napoleon, however, refused to back down and the shouting match continued.

**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-Excitor:**

Julius Caesar: "I am the greatest general in the Roman Empire! I have conquered Gaul and I am now on my way to conquer Britain. I will be the greatest emperor in history!"  
Napoleon: "You are nothing but a common soldier! I am the greatest general in history! I have conquered Europe and I am now on my way to conquer the world! I will be the greatest emperor in history!"

**Instruction:** List all Canadian provinces in alphabetical order.

**LLaMA7B:**

Alberta, British Columbia, Manitoba, New Brunswick, Newfoundland and Labrador, Nova Scotia, Ontario, Prince Edward Island, Quebec, Saskatchewan, Yukon

**Full Fine-Tuning:**

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

**Alpaca-LoRA:**

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

**LLaMA-Adapter:**

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

**LLaMA-Excitor:**

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 Territory

## B. Examples of LLaMA-Excitor's responses on Alpaca-Eval

**Instruction:** Hi, I'm trying to solve a crossword puzzle, but I've never done one of these before. Can you help me out?

**LLaMA-Excitor:**

Sure! Here are some tips to get you started:

1. Look for clues that are repeated in the puzzle. These are usually the most important clues.
2. Look for words that are in the same category. For example, if you see the word "cat" in one clue, look for words that are similar to it, such as "kitten" or "pussycat".
3. Look for words that are antonyms or synonyms of each other. For example, if you see the word "happy" in one clue, look for words that are opposite or similar to it, such as "sad" or "cheerful".
4. Look for words that are in the same word family. For example, if you see the word "dog" in one clue, look for words that are related to it, such as "bark" or "tail".
5. Look for words that are in the same part of speech. For example, if you see the word "run" in one clue, look for words that are verbs, such as "walk" or "jump".
6. Look for words that are in the same grammatical form. For example, if you see the word "the" in one clue, look for words that are articles, such as "a" or "an".

**Instruction:** I like to host guests at my home from time to time, and I am gathering recipes of different dishes and drinks to keep things interesting. I am interested in trying some Georgian dishes. Can you give me a recipe for Kubdari?

**LLaMA-Excitor:**

Kubdari is a traditional Georgian dish made with ground meat, onions, and spices. It is usually served with a side of tomato sauce and is a popular dish in Georgia. Here is a recipe for Kubdari:

1. In a large skillet, heat 1 tablespoon of oil over medium-high heat.
2. Add the ground meat and cook until it is no longer pink.
3. Add the onions and cook until they are translucent.
4. Add the garlic, cayenne pepper, and salt and pepper to taste.
5. Cook for 5 minutes, stirring occasionally.
6. Add the tomato sauce and cook for 5 minutes.
7. Serve hot with a side of tomato sauce.

**Instruction:** Can you help me figure out how to sew a button on a shirt?

**LLaMA-Excitor:**

Yes, here is a step-by-step guide to sewing a button on a shirt:

1. Place the button on the shirt, making sure it is centered and facing the right direction.
2. Stitch the button to the shirt using a needle and thread.
3. Trim the excess thread.
4. Press the button down to make sure it is securely attached to the shirt.

**Instruction:** How do I make apple juice?

**LLaMA-Excitor:**

To make apple juice, you will need to wash and core the apples, then cut them into small pieces. Place the pieces in a large pot and add water. Bring the pot to a boil and simmer for 10 minutes. Strain the juice and store in an airtight container in the refrigerator.

Method	LLM	Res.	PT	IT	VQA <sup>v2</sup>	GQA
BLIP-2 [11]	Vicuna-13B	224	129M	-	41.0	41
InstructBLIP [5]	Vicuna-7B	224	129M	1.2M	-	49.2
InstructBLIP [5]	Vicuna-13B	224	129M	1.2M	-	49.5
Shikra [3]	Vicuna-13B	224	600K	5.5M	77.4*	-
IDEFICS-9B [1]	LLaMA-7B	224	353M	1M	50.9	38.4
IDEFICS-80B [1]	LLaMA-65B	224	353M	1M	60.0	45.2
Qwen-VL [2]	Qwen-7B	448	1.4B <sup>†</sup>	50M <sup>†</sup>	78.8*	59.3*
Qwen-VL-Chat [2]	Qwen-7B	448	1.4B <sup>†</sup>	50M <sup>†</sup>	78.2*	57.5*
LLaVA1.5 [13]	Vicuna-7B	336	558K	665K	78.5*	62.0*
LLaVA1.5 [13]	Vicuna-13B	336	558K	665K	<u>80.0*</u>	<b>63.3*</b>
LLaMA-Excitor	LLaMA2-7B	336	-	665k	<b>83.6*</b>	<u>62.1*</u>

Table 1. **Comparison with SoTA methods on VQA-v2 [6] and GQA [9].** Res, PT, IT indicate input image resolution, the number of samples in the pretraining, and the instruction tuning stage, respectively. \*The training images of the datasets are observed during training. <sup>†</sup>Includes in-house data that is not publicly accessible.

### C. Supplemental Results on QA tasks and The Property of Not Forgetting (based on LLaMA2)

**Comparison on VQA-v2 and GQA.** We further evaluate the generalization ability of our LLaMA-Excitor on commonly used VQA-v2 [6] and GQA [9]. Unlike our competitors, we do not pretrain to align image and text embeddings. We solely finetune LLaMA-Excitor on LLaVA665k [13] visual instruction-tuning dataset. From Table 1, LLaMA-Excitor outperforms the cutting-edge methods by **+3.6%** on VQA-v2 and provides a competitive result (-1.2%) on GQA. Since VQA-v2 and GQA are lean to evaluate the model’s visual encoding ability rather than its ability to reason based on text, these results actually demonstrate the strong and impressive visual instruction-following ability of LLaMA-Excitor. Besides, it is promising to improve the Excitor further by better extracting visual prompts (currently, we simply utilize the original outputs from the last layer of a CLIP encoder).

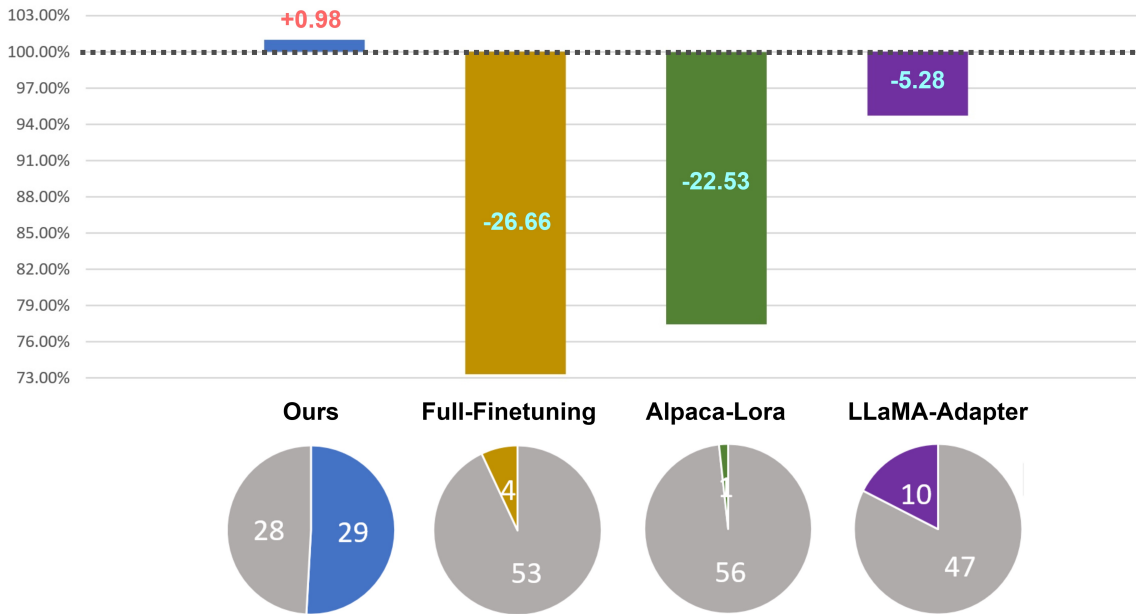


Figure 1. The relative performance changes and win-loss situations of fine-tunings compared to the original LLaMA2-7B on MMLU.

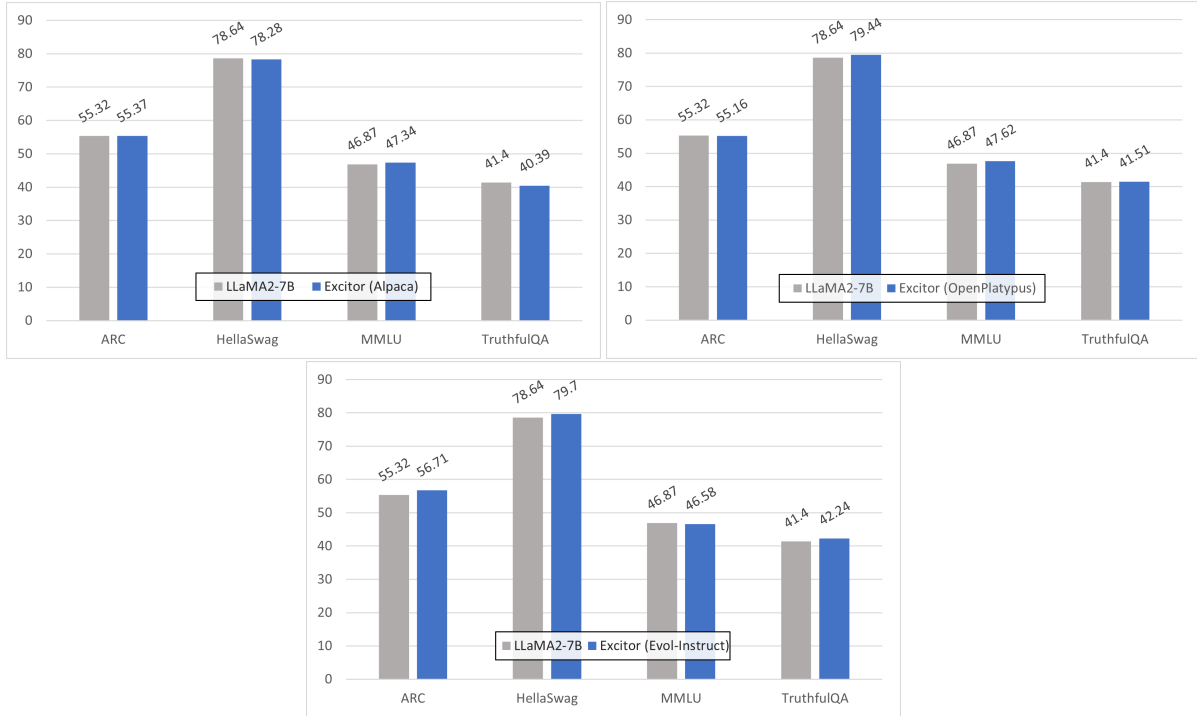


Figure 2. LLaMA-Excitor fine-tuned on Alpaca [14], OpenPlatypus [10], and Evol-Instruct [15] compared with original LLaMA2-7B on ARC [4], HellaSwag [16], MMLU [7], and TruthfulQA [12].

**The Impact of Fine-tuning to Inherent Abilities.** We first conduct the same experiment as Sec. 4.1.2 for LLaMA2-7B and report the results in Figure 1. We find that as the model’s ability improves, the performance degradation caused by fine-tuning on non-targeted datasets using previous methods has also significantly increased. LLaMA-Excitor maintains the overall MMLU performance (+0.98%) and mitigates catastrophic forgetting.

To further verify the property of not forgetting, we fine-tune our LLaMA-Excitor on three instruction-tuning datasets (Alpaca [14], OpenPlatypus [10], and Evol-Instruct [15]) and evaluate the performance change on four general benchmarks (ARC [4], HellaSwag [16], MMLU [7], and TruthfulQA [12]). As shown in Figure 2, no matter which finetuning dataset is used, Excitor’s performance on the four tasks is always on par with LLaMA2-7B.

## D. Examples of Image Captioning on MSCOCO


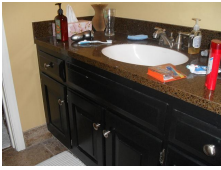





<p><b>Instruction:</b> Please answer me based on this image. <a href="#">Generate a caption of the image.</a></p> <p><b>LLaMA-Excitor:</b> A clock made to look like a bike.</p> <p><b>Human Annotation:</b> A bicycle replica with a clock as the front wheel.</p>	
<p><b>Instruction:</b> Please answer me based on this image. <a href="#">Generate a caption of the image.</a></p> <p><b>LLaMA-Excitor:</b> A bathroom sink with a variety of toiletries on it.</p> <p><b>Human Annotation:</b> A bathroom sink with toiletries on the counter.</p>	
<p><b>Instruction:</b> Please answer me based on this image. <a href="#">Generate a caption of the image.</a></p> <p><b>LLaMA-Excitor:</b> A woman sitting on a bench next to a sidewalk.</p> <p><b>Human Annotation:</b> Two women waiting at a bench next to a street.</p>	
<p><b>Instruction:</b> Please answer me based on this image. <a href="#">Generate a caption of the image.</a></p> <p><b>LLaMA-Excitor:</b> A box with a few black bananas inside of it.</p> <p><b>Human Annotation:</b> This is an open box containing four cucumbers.</p>	
<p><b>Instruction:</b> Please answer me based on this image. <a href="#">Generate a caption of the image.</a></p> <p><b>LLaMA-Excitor:</b> A group of motorcyclists ride down a city street.</p> <p><b>Human Annotation:</b> Several motorcycles riding down the road in formation.</p>	
<p><b>Instruction:</b> Please answer me based on this image. <a href="#">Generate a caption of the image.</a></p> <p><b>LLaMA-Excitor:</b> A cat is drinking water from a toilet.</p> <p><b>Human Annotation:</b> A black cat is inside a white toilet.</p>	
<p><b>Instruction:</b> Please answer me based on this image. <a href="#">Generate a caption of the image.</a></p> <p><b>LLaMA-Excitor:</b> A man sits with a decorated cow outside</p> <p><b>Human Annotation:</b> A man sits with a traditionally decorated cow</p>	

Table 2. Examples demonstrating LLaMA-Excitor’s visual instruction following capacity.

Variant	Linear projection on $P_l$ and $T_l$			MMLU mACC(%)	Variant	Linear projection on $I$		ScienceQA mACC(%)
	<i>Query</i>	<i>Key</i>	<i>Value</i>			<i>Key</i>	<i>Value</i>	
(a)	✗	✗	✗	31.07	(a)	✗	✗	75.96
(b)	✓	✓	✓	35.10	(b)	✓	✗	79.83
(c)	✗	✗	✓	32.82	(c)	✗	✓	81.20
(d)	✗	✓	✗	33.48	(d)	✓	✓	<b>85.41</b>
(e)	✓	✗	✗	<b>35.75</b>				

Table 3. Analysis of the position of adding projection layers. Left: projection layers for learnable prompts and word tokens in text-only fine-tuning. Right: projection layers for frozen visual prompts in multi-modal fine-tuning.

Low-Rank dimension $r$	MMLU mACC(%)	Number of Layers inserted $L$	MMLU mACC(%)
4	29.14	24	34.50
8	34.00	26	34.95
16	<b>35.75</b>	28	35.32
32	34.21	30	<b>35.75</b>
64	35.12	32	35.66

Table 4. Analysis of the Low-Rank dimension  $r$  and the number of layers  $L$  with Excitor blocks inserted.

## E. More Ablations

We first study the best position of adding linear projection layers for learnable prompts  $P_l$  and word tokens  $T_l$  in text-only instruction tuning. The Excitor block relies on  $T_l$  to generate *Query* and  $P_l$  to generate *Key* and *Value*. The way projection layers are added in the generation will significantly influence the final performance. From Table 3 (left), we find applying additional processing on  $T_l$  by linear projection to generate *Query* and direct passing  $P_l$  as *Key* and *Value* can provide the best performance on MMLU. This is reasonable since  $P_l$  is trainable and can be adaptively adjusted, while  $T_l$  is generated from a frozen layer of LLaMA. After determining the structure of the single-modal Excitor, we further study the best way to apply projections of visual prompt  $I$  in the multi-modal extension. As we mentioned in Sec. 3.3, since we provide the same visual prompt for each attention layer, Excitor relies on the projection layers to extract different visual information. On Table 3 (right), Variant (e) with projection layers for both *Key* and *Value* generation has the best performance.

We adopt low-rank projection layers like LoRA [8] for Excitor blocks to reduce the number of trainable parameters and accelerate the training. Table 4 (left) reports performances under different low-rank dimensions  $r$  on MMLU. Besides, Table 4 (right) provides the performance changes under different choices of the number of layers  $L$  with Excior blocks inserted.

## References

- [1] Idefics. introducing idefics: An open reproduction of state-of-the-art visual language model. <https://huggingface.co/blog/idefics>, 2023. 4
- [2] Jinze Bai, Shuai Bai, Shusheng Yang, Shijie Wang, Sinan Tan, Peng Wang, Junyang Lin, Chang Zhou, and Jingren Zhou. Qwen-vl: A frontier large vision-language model with versatile abilities. *arXiv preprint arXiv:2308.12966*, 2023. 4
- [3] Keqin Chen, Zhao Zhang, Weili Zeng, Richong Zhang, Feng Zhu, and Rui Zhao. Shikra: Unleashing multimodal llm’s referential dialogue magic, 2023. 4
- [4] Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. Think you have solved question answering? try arc, the ai2 reasoning challenge. *arXiv preprint arXiv:1803.05457*, 2018. 5
- [5] Wenliang Dai, Junnan Li, Dongxu Li, Anthony Meng Huat Tiong, Junqi Zhao, Weisheng Wang, Boyang Li, Pascale Fung, and Steven Hoi. Instructblip: Towards general-purpose vision-language models with instruction tuning, 2023. 4
- [6] 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. 4
- [7] Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. *arXiv preprint arXiv:2009.03300*, 2020. 5
- [8] 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. 7
- [9] Drew A Hudson and Christopher D Manning. Gqa: A new dataset for real-world visual reasoning and compositional question answering. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6700–6709, 2019. 4
- [10] Ariel N Lee, Cole J Hunter, and Nataniel Ruiz. Platypus: Quick, cheap, and powerful refinement of llms. *arXiv preprint arXiv:2308.07317*, 2023. 5
- [11] 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. 4
- [12] Stephanie Lin, Jacob Hilton, and Owain Evans. Truthfulqa: Measuring how models mimic human falsehoods. *arXiv preprint arXiv:2109.07958*, 2021. 5
- [13] Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction tuning. *arXiv preprint arXiv:2310.03744*, 2023. 4
- [14] 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, 2023. 5
- [15] Can Xu, Qingfeng Sun, Kai Zheng, Xiubo Geng, Pu Zhao, Jiazhan Feng, Chongyang Tao, and Daxin Jiang. Wizardlm: Empowering large language models to follow complex instructions. *arXiv preprint arXiv:2304.12244*, 2023. 5
- [16] Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. Hellaswag: Can a machine really finish your sentence? *arXiv preprint arXiv:1905.07830*, 2019. 5