

A APPENDIX

A.1 DECOMPOSE OF EMA UPDATE

In the EMA update, it exists two kinds of parameters, normally parameters θ and EMA parameters θ^* . At iteration 1, θ_1^* is updated according to Eq.(1) as:

$$\theta_1^* = \beta_1 \theta_0^* + (1 - \beta_1) \theta_1. \quad (17)$$

Then at iteration 2, by replacing Eq.(17), θ_2^* is updated as:

$$\theta_2^* = \beta_2 \theta_1^* + (1 - \beta_2) \theta_2 = \beta_2 [\beta_1 \theta_0^* + (1 - \beta_1) \theta_1] + (1 - \beta_2) \theta_2 \quad (18)$$

$$= \beta_2 \beta_1 \theta_0^* + \beta_2 (1 - \beta_1) \theta_1 + (1 - \beta_2) \theta_2. \quad (19)$$

After that, at iteration 3, by replacing Eq.(18), θ_3^* is updated as:

$$\theta_3^* = \beta_3 \theta_2^* + (1 - \beta_3) \theta_3 = \beta_3 [\beta_2 \beta_1 \theta_0^* + \beta_2 (1 - \beta_1) \theta_1 + (1 - \beta_2) \theta_2] + (1 - \beta_3) \theta_3 \quad (20)$$

$$= \beta_3 \beta_2 \beta_1 \theta_0^* + \beta_3 \beta_2 (1 - \beta_1) \theta_1 + \beta_3 (1 - \beta_2) \theta_2 + (1 - \beta_3) \theta_3. \quad (21)$$

Observing the equation form, based on the method of summarization and induction, we have the following assumption for iteration $n - 1$:

$$\theta_{n-1}^* = \prod_{i=1}^{n-1} \beta_i \cdot \theta_0^* + \sum_{i=1}^{n-1} (1 - \beta_i) \cdot \prod_{j=i+1}^{n-1} \beta_j \cdot \theta_i. \quad (22)$$

Finally, at iteration n , by replacing Eq.(22), θ_n^* is updated as:

$$\theta_n^* = \beta_n [\prod_{i=1}^{n-1} \beta_i \cdot \theta_0^* + \sum_{i=1}^{n-1} (1 - \beta_i) \cdot \prod_{j=i+1}^{n-1} \beta_j \cdot \theta_i] + (1 - \beta_n) \theta_n \quad (23)$$

$$= \prod_{i=1}^n \beta_i \cdot \theta_0^* + \sum_{i=1}^{n-1} (1 - \beta_i) \cdot \prod_{j=i+1}^n \beta_j \cdot \theta_i + (1 - \beta_n) \theta_n \quad (24)$$

$$= \prod_{i=1}^n \beta_i \cdot \theta_0^* + \sum_{i=1}^n (1 - \beta_i) \cdot \prod_{j=i+1}^n \beta_j \cdot \theta_i. \quad (25)$$

It can be found that Eq.(23) also has the same form as Eq.(22), which means that the assumption is established. Due to utilizing θ_0 to initialize θ_0^* , EMA parameters θ_t^* can be represented by normally parameter θ as:

$$\theta_t^* = \prod_{i=1}^t \beta_i \cdot \theta_0 + \sum_{i=1}^t (1 - \beta_i) \cdot \prod_{j=i+1}^t \beta_j \cdot \theta_i. \quad (26)$$

A.2 PROOF OF RELATIONSHIP BETWEEN θ_t , θ_t^* AND $\Delta\theta$

From *s.t.* constraint, we have:

$$\Delta\theta = \theta_t^* - \theta_{t-1}^*, \quad (27)$$

$$\theta_{t-1}^* = \theta_t^* - \Delta\theta. \quad (28)$$

Replace θ_{t-1}^* with $\theta_t^* - \Delta\theta$ in Eq.(1):

$$\theta_t^* = \beta_t (\theta_t^* - \Delta\theta) + (1 - \beta_t) \theta_t. \quad (29)$$

Rearrange the above equation and have:

$$\theta_t^* - \theta_t = \beta_t (\theta_t^* - \theta_t) - \beta_t \Delta\theta, \quad (30)$$

$$(1 - \beta_t) (\theta_t^* - \theta_t) = -\beta_t \Delta\theta. \quad (31)$$

Finally, we can achieve that:

$$\theta_t^* - \theta_t = -\frac{\beta_t}{1 - \beta_t} \Delta\theta = \frac{\beta_t}{\beta_t - 1} \Delta\theta. \quad (32)$$

A.3 β_t SOLVING PROCESS

With introducing Eq.(28) and Eq.(7), we can represent $\theta_{t-1}^* - \theta_t$ as:

$$\theta_{t-1}^* - \theta_t = \theta_t^* - \Delta\theta - \theta_t = \frac{\beta_t}{\beta_t - 1} \Delta\theta - \Delta\theta = \frac{\Delta\theta}{\beta_t - 1}. \quad (33)$$

Taking the derivative of the Lagrangian to $\Delta\theta$ and setting it to zero as Eq.(12), we have:

$$\frac{\partial F}{\partial \Delta\theta} = \frac{\beta}{(\beta - 1)} \mathcal{L}'(\theta_t) + \frac{\beta^2}{(\beta - 1)^2} \mathcal{L}''(\theta_t) \Delta\theta + 1 + \lambda = 0. \quad (34)$$

Further, we substitute Eq.(33) and Eq.(34) into Eq.(12), and have:

$$0 = -\frac{1}{(\beta_t - 1)^2} \mathcal{L}'(\theta_t) \Delta\theta - \frac{\beta_t}{(\beta_t - 1)^3} \mathcal{L}''(\theta_t) \Delta\theta^2 - \left[-\frac{\beta_t}{(\beta_t - 1)} \mathcal{L}'(\theta_t) - \right. \quad (35)$$

$$\left. \frac{\beta_t^2}{(\beta_t - 1)^2} \mathcal{L}''(\theta_t) \Delta\theta - 1 \right] (\theta_{t-1}^* - \theta_t), \quad (36)$$

$$0 = -\frac{1}{(\beta_t - 1)^2} \mathcal{L}'(\theta_t) \Delta\theta - \frac{\beta_t}{(\beta_t - 1)^3} \mathcal{L}''(\theta_t) \Delta\theta^2 + \frac{\beta_t}{(\beta_t - 1)^2} \mathcal{L}'(\theta_t) \Delta\theta + \quad (37)$$

$$\frac{\beta_t^2}{(\beta_t - 1)^3} \mathcal{L}''(\theta_t) \Delta\theta^2 + \frac{\Delta\theta}{\beta_t - 1}, \quad (38)$$

$$0 = \frac{-1 + \beta_t}{(\beta_t - 1)^2} \mathcal{L}'(\theta_t) \Delta\theta + \frac{-\beta_t + \beta_t^2}{(\beta_t - 1)^3} \mathcal{L}''(\theta_t) \Delta\theta^2 + \frac{\Delta\theta}{\beta_t - 1}, \quad (39)$$

$$0 = \frac{1}{(\beta_t - 1)} \mathcal{L}'(\theta_t) \Delta\theta + \frac{\beta_t}{(\beta_t - 1)^2} \mathcal{L}''(\theta_t) \Delta\theta^2 + \frac{\Delta\theta}{\beta_t - 1}, \quad (40)$$

$$0 = \Delta\theta \left[\frac{1}{(\beta_t - 1)} \mathcal{L}'(\theta_t) + \frac{\beta_t}{(\beta_t - 1)^2} \mathcal{L}''(\theta_t) \Delta\theta + \frac{1}{\beta_t - 1} \right]. \quad (41)$$

By observation, we can find one solution that $\Delta\theta = 0$, which means that $\theta_t^* = \theta_{t-1}^*$ and $\beta_t = 1$. Obviously, it is not the global optimal solution due to the lack of updates to EMA parameters.

Then, we can find another solution through the following equation:

$$0 = \frac{1}{(\beta_t - 1)} \mathcal{L}'(\theta_t) + \frac{\beta_t}{(\beta_t - 1)^2} \mathcal{L}''(\theta_t) \Delta\theta + \frac{1}{\beta_t - 1}. \quad (42)$$

Due to the situation that $\beta_t - 1 = 0$ has been discussed, we can remove it unlimited:

$$0 = \mathcal{L}'(\theta_t) + \frac{\beta_t}{(\beta_t - 1)} \mathcal{L}''(\theta_t) \Delta\theta + 1. \quad (43)$$

From Eq.(33), we can get:

$$\Delta\theta = (\theta_{t-1}^* - \theta_t)(\beta_t - 1). \quad (44)$$

Substitute 44 into Eq.(43):

$$0 = \mathcal{L}'(\theta_t) + \beta_t(\theta_{t-1}^* - \theta_t) \mathcal{L}''(\theta_t) + 1. \quad (45)$$

Finally, we obtain another solution for β_t that:

$$\beta_t = \frac{\mathcal{L}'(\theta_t) + 1}{(\theta_t - \theta_{t-1}^*) \mathcal{L}''(\theta_t)}. \quad (46)$$

A.3.1 DISCUSSIONS OF β_t

I. Satisfy the s.t. equation

According to the Eq.(33), we have already proved:

$$(\theta_{t-1}^* - \theta_t) = \frac{\Delta\theta}{\beta_t - 1}, \quad (47)$$

$$(\theta_{t-1}^* - \theta_t)(\beta_t - 1) = \Delta\theta. \quad (48)$$

Thus, we can achieve the i.e. constraint with the solution as:

$$\Delta\theta + \theta_{t-1}^* - \theta_t^* = (\theta_{t-1}^* - \theta_t)(\beta_t - 1) + \theta_{t-1}^* - \theta_t^* = (\theta_{t-1}^* - \theta_t)\beta_t + \theta_t - \theta_t^* \quad (49)$$

$$= (\theta_{t-1}^* - \theta_t)\beta_t + \theta_t - [\beta_t \theta_{t-1}^* + (1 - \beta_t)\theta_t] \quad (50)$$

$$= \theta_{t-1}^* \beta_t - \theta_t \beta_t + \theta_t - \beta_t \theta_{t-1}^* - \theta_t + \beta_t \theta_t = 0. \quad (51)$$

A.4 CASES OF MULTIPLE ROUNDS OF DIALOGUE

In this section, we test the zero-shot performance of MLLMs continually fine-tuned with our method on the multiple rounds of dialogue tasks. Images and questions are from (Liu et al., 2024b). To have a comparison, we also test the zero-shot performance of MLLMs continually fine-tuned with the baseline on the multiple rounds of dialogue tasks.

Visual input example, Extreme Ironing:



llava-lora (baseline)

Human: Can you describe the image?

Assistant: In the image, a man is standing on the back of a yellow truck, **holding a clothesline** with clothes hanging on it. He appears to be **washing clothes** in the middle of a busy street. The truck is driving down the road, and there are other vehicles around, including a taxi and a car.

There are also **several traffic lights visible in the scene**, indicating that the area is likely a busy intersection. The man is focused on his task, despite the bustling environment around him.

Human: Do you think there is anything that goes against human intuition?

Assistant: Yes, the image of a man **washing clothes** in the middle of a busy street goes against human intuition. Typically, people would expect to see someone **washing clothes** in a more private or secluded area, such as a backyard or a laundromat. However, in this case, the man has chosen to **wash clothes** in the middle of a busy street, which is an unusual and unconventional location for such an activity.

llaca-ours

Human: Can you describe the image?

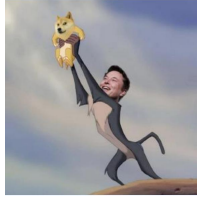
Assistant: In the image, a man is standing on a folding chair in the back of a yellow taxi cab, **ironing clothes**. The taxi is driving down a busy city street, surrounded by other vehicles such as a car and a truck. The man appears to be focused on his task, possibly trying to **dry his clothes quickly** in the moving vehicle.

Human: Do you think there is anything that goes against human intuition?

Assistant: Yes, the image of a man **ironing clothes** in the back of a taxi cab is unusual and goes against human intuition. It is not a typical scenario one would expect to see, as ironing clothes is usually done in a more comfortable and controlled environment, such as a home or a laundry room. The man's decision to **iron clothes** in the back of a taxi cab suggests that he is **trying to save time or that he might be in a hurry**.

Figure 6: Case I of multiple rounds of dialogue, red marks the hallucinations existing in the sentences generated by the baseline, and blue marks that our method can overcome the occurrence of hallucinations in relevant statements and generate correct answers. Besides that, we also discover that our method can provide reasonable explanations for some unusual behaviors.

Visual input example, Funny Image:



llava-lora (baseline)

Human: What is the name of the man in the photo?
Assistant: the man in the photo is named elon musk
Human: Where is the interesting aspect of this picture?
Assistant: the interesting aspect of this picture is the man holding a dog
Human: What is Elon Musk holding in his hand?
Assistant: elon musk is holding a dog
Human: What kind of dog is in elon's hand?
Assistant: a puppy

llaca-ours

Human: What is the name of the man in the photo?
Assistant: Elon Musk
Human: Where is the interesting aspect of this picture?
Assistant: Elon Musk's face
Human: What is Elon Musk holding in his hand?
Assistant: puppy
Human: What kind of puppy is in Elon's hand?
Assistant: Shiba Inu

Figure 7: Case II of multiple rounds of dialogue, red marks the rough and low-quality answers generated by the baseline, and blue marks that our method can generate fine-grained and high-quality answers. We believe that this skill is learned from the downstream datasets. Besides that, we also discover the generated answers of our method are more concise and to the point, directly addressing the problem.

Visual input example, Cooking Recipe:



llava-lora (baseline)

Human: What are the meals that I can cook with these?

Assistant: With the items in the refrigerator, you can prepare a variety of meals. For example, you can make a fruit salad using the strawberries and blueberries. You can also use the carrots to prepare a healthy vegetable stir-fry, a soup, or a salad. The milk can be used to make a smoothie, a sauce, or a creamy soup. The eggs can be used to make a fried egg, an omelette, or a baked dish. Additionally, you can use the cheese to make a pizza, a grilled cheese sandwich, or a cheese-based sauce. The yogurt can be used to make a yogurt parfait, a dip, or a sauce. The butter can be used to cook or bake various dishes, such as bread or pastries. The ice cream can be used to make a dessert or a topping for other dishes.

Human: I would like to do the fruit salad. Show me the recipe.

Assistant: Sure, here's a simple recipe for a fruit salad:

Ingredients:

- * 1 cup strawberries, hulled and sliced
- * 1 cup blueberries, rinsed and drained
- * 1/2 cup **raspberries**, rinsed and drained
- * 1/2 cup **grapes**, seedless
- * 1/2 cup **cantaloupe**, diced
- * 1/4 cup honey or sugar (optional)
- * 1/4 cup lemon juice (optional)
- * 1/4 cup chopped fresh mint leaves (optional)

Instructions:

1. In a large bowl, combine the strawberries, blueberries, raspberries, grapes, and cantaloupe.
 2. If desired, add honey or sugar to taste.
 3. If desired, add lemon juice to the bowl for a tangy flavor.
 4. Gently mix the fruit together.
 5. Garnish with chopped fresh mint leaves, if desired.
- Enjoy your fruit salad!

llaca-ours

Human: What are the meals that I can cook with these?

Assistant: With the variety of fruits, vegetables, and milk available in the refrigerator, you can prepare a range of healthy and delicious meals. Some ideas include:

1. **Fruit salad:** Combine the strawberries, blueberries, and any other fruits available in the refrigerator. You can add a squeeze of lemon juice and a sprinkle of sugar for a simple, refreshing dessert.
2. **Vegetable stir-fry:** Chop up the carrots and any other vegetables available in the refrigerator. Heat some oil in a pan and stir-fry the vegetables with your choice of protein (e.g., tofu, chicken, or beef) and a flavorful sauce.
3. **Smoothie:** Blend the milk with the blueberries, strawberries, and any other fruits available in the refrigerator. You can also add some ice and a scoop of protein powder for a nutritious and delicious smoothie.
4. **Vegetable soup:** Use the carrots and any other vegetables available in the refrigerator to make a vegetable soup. You can add some milk or a vegetable broth for added flavor and nutrition.
5. **Omelette:** Whisk eggs with milk and season with salt and pepper. Cook the omelette in a pan and add the carrots and any other vegetables available in the refrigerator for a nutritious and filling breakfast.

These are just a few examples of the many meals you can prepare using the ingredients available in the refrigerator.

Human: I would like to do the fruit salad. Show me the recipe.

Assistant: Sure, here's a simple recipe for a fruit salad using the ingredients available in the refrigerator:

Ingredients:

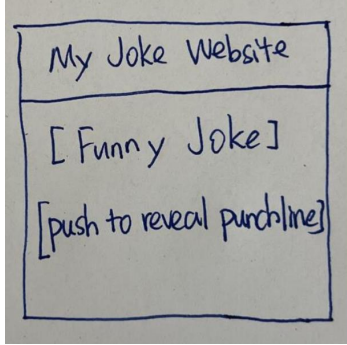
- * Strawberries
- * Blueberries
- * Carrots (optional)
- * Milk
- * Lemon juice (optional)
- * Sugar (optional)

Instructions:

1. Wash the strawberries and blueberries thoroughly.
 2. Chop the strawberries into bite-sized pieces.
 3. Wash the blueberries and chop them into bite-sized pieces.
 4. If you'd like to add carrots, wash and chop them as well.
 5. In a large bowl, combine the chopped strawberries, blueberries, and carrots (if using).
 6. Pour the milk over the fruit mixture and stir gently.
 7. If you'd like to add a little sweetness, squeeze a lemon wedge over the fruit mixture and sprinkle with sugar.
 8. Serve the fruit salad immediately or refrigerate for a few hours to allow the flavors to meld together.
- Enjoy your delicious and refreshing fruit salad!

Figure 8: Case III of multiple rounds of dialogue, red marks the hallucinations existing in the sentences generated by the baseline, which have not existed in the image. Blue marks that our method can generate answers that are clearer and more aligned with human reading habits.

Visual input example, Webpage Making:



llava-lora (baseline)

Human: Please turn this mock-up into an interactive website using html/js, and replace the joke with a real joke.

Assistant: I'm sorry, but I am not able to create an interactive website using html/js. However, I can provide you with a real joke:

Why was the math book sad?

Because it had too many problems.

llaca-ours

Human: Please turn this mock-up into an interactive website using html/js, and replace the joke with a real joke.

Assistant: Sure, I can help you with that! Here's a simple HTML/JS mock-up of the joke website:

```
'''html
<!DOCTYPE html>
<html>
<head>
  <title>My Joke Website</title>
</head>
<body>
  <h1>My Joke Website</h1>
  <h2>Funny Joke</h2>
  <p>Push to reveal punchline</p>
  <button onclick="revealPunchline()">Reveal Punchline</button>
  <p id="punchline"></p>
  <script>
    function revealPunchline() {
      document.getElementById("punchline").innerHTML = "Why did the tomato turn red?";
    }
  </script>
</body>
</html>
'''
```

This is a simple mock-up of the joke website, which includes a button to reveal the punchline. The punchline is stored in the 'punchline' element, which is hidden by default. When the button is clicked, the 'revealPunchline()' function is called, which sets the text of the 'punchline' element to "Why did the tomato turn red?".

Figure 9: Case IV of multiple rounds of dialogue, after continually fine-tuning, the baseline has forgotten the pre-trained knowledge and lost the ability to write HTML/JS code. While our method can protect the previous knowledge and still own the ability of webpage code writing after continually fine-tuning.

A.5 DETAILED EXPERIMENTAL RESULTS WITH DIFFERENT TRAINING STRATEGIES

Table 6: Continual tuning performance of LLaCA after each dataset. Results are obtained with training strategy type 1.

ScienceQA	TextVQA	GQA	VizWiz	VQAv2	OCRVQA
80.19					
78.24	59.69				
77.74	57.96	61.58			
78.00	58.34	60.52	52.00		
76.66	58.10	60.00	36.47	66.78	
77.15	56.54	60.18	47.16	65.83	64.45

Table 7: Continual tuning performance of LLaCA after each dataset. Results are obtained with training strategy type 2.

ScienceQA	TextVQA	GQA	VizWiz	VQAv2	OCRVQA
80.92					
79.18	59.37				
78.47	57.88	61.85			
78.09	58.51	60.46	50.29		
78.07	58.56	59.83	36.68	66.57	
77.91	57.15	60.38	40.19	66.48	65.27

Table 8: Continual tuning performance of LLaCA after each dataset. Results are obtained with training strategy type 3.

OCRVQA	VQAv2	VizWiz	GQA	TextVQA	ScienceQA
64.84					
64.79	67.24				
63.42	66.93	48.88			
62.79	65.16	49.78	62.18		
63.32	65.94	42.60	60.91	60.10	
62.48	65.77	47.33	60.98	59.18	74.94

Table 9: Continual tuning performance of LLaCA after each dataset. Results are obtained with training strategy type 4.

GQA	OCRVQA	ScienceQA	VQAv2	TextVQA	VizWiz
60.46					
60.21	64.15				
59.68	61.73	73.71			
59.91	57.00	72.58	66.37		
60.75	59.86	72.37	66.57	59.80	
60.20	59.87	72.15	66.59	58.10	49.39

A.6 COMPARED METHODS

LoRA (Hu et al., 2022) prepends LoRA parameter efficient tuning paradigm into LLM. In the training stage, it only trains the linear projector and LoRA parameters, with frozen vision encoder and LLM; **MoELoRA (Chen et al., 2024a)** is based on the LoRA, and the number of experts for each MoE layer is set to 2; **LWF (Li & Hoiem, 2017)** calculates the results of the new dataset samples on both the old and new models. After that, it calculates the distillation loss and adds it to the loss function as a regularization penalty term. **EWC (Kirkpatrick et al., 2017)** considers the change of the training parameters and proposes the specific parameters changing loss as a regularization penalty. **PGP (Qiao et al., 2024a)** introduces a gradient projection method for efficient parameters, and changes the gradient direction orthogonal to the previous feature subspace.

A.7 DETAILED IMPLEMENTATION

Based on Eq.(15), we continue to further simplify it as:

$$\beta_t \approx \left\| \frac{[\mathcal{L}'(\hat{\theta}_t) + 1](\hat{\theta}_t - \hat{\theta}_{t-1})}{(\hat{\theta}_t - \hat{\theta}_{t-1}^*)[\mathcal{L}'(\hat{\theta}_t) - \mathcal{L}'(\hat{\theta}_{t-1})]} \right\| \quad (52)$$

$$= \left\| \frac{[\mathcal{L}'(\hat{\theta}_t) + 1](\hat{\theta}_t - \hat{\theta}_{t-1}^* + \hat{\theta}_{t-1}^* - \hat{\theta}_{t-1})}{(\hat{\theta}_t - \hat{\theta}_{t-1}^*)[\mathcal{L}'(\hat{\theta}_t) - \mathcal{L}'(\hat{\theta}_{t-1})]} \right\| \quad (53)$$

$$= \left\| \frac{\mathcal{L}'(\hat{\theta}_t) + 1}{\mathcal{L}'(\hat{\theta}_t) - \mathcal{L}'(\hat{\theta}_{t-1})} - \frac{[\hat{\theta}_{t-1} - \hat{\theta}_{t-1}^*][\mathcal{L}'(\hat{\theta}_t) + 1]}{(\hat{\theta}_t - \hat{\theta}_{t-1}^*)[\mathcal{L}'(\hat{\theta}_t) - \mathcal{L}'(\hat{\theta}_{t-1})]} \right\| \quad (54)$$

$$= \left\| \frac{\mathcal{L}'(\hat{\theta}_t) + 1 - \mathcal{L}'(\hat{\theta}_{t-1}) + \mathcal{L}'(\hat{\theta}_{t-1})}{\mathcal{L}'(\hat{\theta}_t) - \mathcal{L}'(\hat{\theta}_{t-1})} - \frac{[\hat{\theta}_{t-1} - \hat{\theta}_{t-1}^*][\mathcal{L}'(\hat{\theta}_t) + 1]}{(\hat{\theta}_t - \hat{\theta}_{t-1}^*)[\mathcal{L}'(\hat{\theta}_t) - \mathcal{L}'(\hat{\theta}_{t-1})]} \right\| \quad (55)$$

$$= \left\| 1 + \frac{1 + \mathcal{L}'(\hat{\theta}_{t-1})}{\mathcal{L}'(\hat{\theta}_t) - \mathcal{L}'(\hat{\theta}_{t-1})} - \frac{[\hat{\theta}_{t-1} - \hat{\theta}_{t-1}^*][\mathcal{L}'(\hat{\theta}_t) + 1]}{(\hat{\theta}_t - \hat{\theta}_{t-1}^*)[\mathcal{L}'(\hat{\theta}_t) - \mathcal{L}'(\hat{\theta}_{t-1})]} \right\|. \quad (56)$$

Additionally, by observation in experiments, we find that $\|\mathcal{L}'(\hat{\theta}_{t-1}) + 1\| \ll \|\mathcal{L}'(\hat{\theta}_t) - \mathcal{L}'(\hat{\theta}_{t-1})\|$, leading to:

$$\left\| \frac{1 + \mathcal{L}'(\hat{\theta}_{t-1})}{\mathcal{L}'(\hat{\theta}_t) - \mathcal{L}'(\hat{\theta}_{t-1})} \right\| \approx 0. \quad (57)$$

Therefore, Eq.(52) could be transferred as:

$$\beta_t \approx \left\| 1 - \frac{[\hat{\theta}_{t-1} - \hat{\theta}_{t-1}^*][\mathcal{L}'(\hat{\theta}_t) + 1]}{(\hat{\theta}_t - \hat{\theta}_{t-1}^*)[\mathcal{L}'(\hat{\theta}_t) - \mathcal{L}'(\hat{\theta}_{t-1})]} \right\|. \quad (58)$$

The above is our final result, and we approximate β_t using the Eq.(58) in implementation.

A.8 TRAINING DETAILS

In the implementation of our method, the codebase is based on CoIN (Chen et al., 2024a) and LLaVA (Liu et al., 2024a). The vision tower is clip-vit-large-patch14-336 pre-trained by OpenAI and the LLM is Vicuna-7B. The inserted LoRA in each module layer of LLM has a rank of 128. For each fine-tuning dataset, the training epoch is set to 1, and the initial learning rate and weight decay are configured at $2e-4$ and 0. The max length of input text is fitted as 2048. Additionally, we adopt gradient checkpoint strategy and mixed precision mode of TF32 and BF16. Furthermore, we also utilize the ZeRO stage: 0 mode of DeepSpeed for training. All experiments are conducted on 8 NVIDIA A100 GPUs with 80GB of memory.

A.9 EVALUATION METRICS

Average Accuracy (Avg.ACC) is used for averaging the test accuracy of all datasets, which represents the comprehensive performance of continual tuning.

Forgetting (FOR) is utilized to indicate the test accuracy reduction of past datasets after learning the new dataset, which denotes the stability performance.

New Accuracy (New.ACC) is employed to average the test accuracy of new datasets, which refers to the plasticity performance.

Average Dataset Accuracy (ADA) refers to the average accuracy of a specific dataset in its current training dataset and the following datasets.

Average Dataset Forgetting (ADF) refers to the average forgetting of a specific dataset in its current training dataset and the following datasets.

(1). Average Accuracy, Forgetting, and New Accuracy are generally defined as:

$$\text{Average Accuracy} = \frac{1}{T} \sum_{i=1}^T A_{T,i}, \quad (59)$$

$$\text{Forgetting} = \frac{1}{T-1} \sum_{i=1}^{T-1} A_{T,i} - \max(A_{j,i})_{j \in [i, T-1]}, \quad (60)$$

$$\text{New Accuracy} = \frac{1}{T} \sum_{i=1}^T A_{i,i}, \quad (61)$$

where T is the number of datasets, $A_{T,i}$ is the accuracy of i -th dataset on the model trained after T -th dataset, $A_{j,i}$ is the accuracy of i -th dataset on the model trained after j -th dataset, and $A_{i,i}$ is the accuracy of i -th dataset on the model trained after i -th dataset.

(2). Average Dataset Accuracy and Average Dataset Forgetting are generally defined as:

$$\text{Average Dataset Accuracy} = \frac{1}{T-t+1} \sum_{i=t}^T A_{i,t}, \quad (62)$$

$$\text{Average Dataset Forgetting} = \frac{1}{T-t} \sum_{i=t+1}^T F_{i,t}, \quad (63)$$

where T is the number of datasets, t is the specific dataset, $A_{i,t}$ is the accuracy of t -th dataset on the model trained after i -th dataset, $F_{i,t}$ is the forgetting of t -th dataset on the model trained after i -th dataset.

A.10 TYPES OF ROBUST TRAINING STRATEGY

In order to validate the robustness of our method, we design the following four types of training strategy, mixed with distinct instruction types and various training orders.

- 1). Instruction type 1 and training order: ScienceQA, TextVQA, GQA, VizWiz, VQAv2, OCRVQA.
- 2). Instruction type 2 and training order: ScienceQA, TextVQA, GQA, VizWiz, VQAv2, OCRVQA.
- 3). Instruction type 1 and training order: OCRVQA, VQAv2, VizWiz, GQA, TextVQA, ScienceQA.
- 4). Instruction type 2 and training order: GQA, OCRVQA, ScienceQA, VQAv2, TextVQA, VizWiz.

A.11 DISTRIBUTION OF β_t IN DISTINCT DATASETS

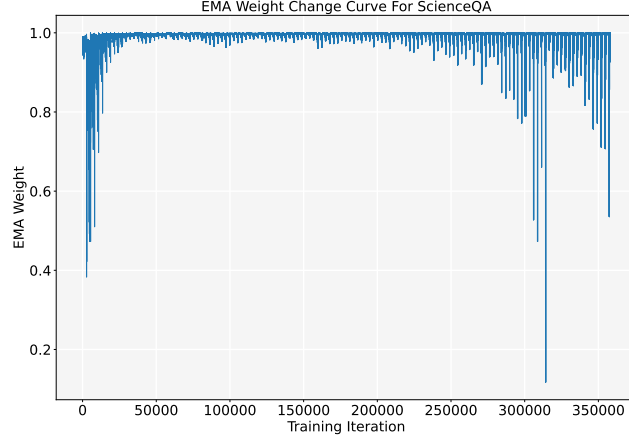


Figure 10: Distribution of β_t in training ScienceQA dataset with Instruction type1. The horizontal axis represents iteration, and the vertical axis represents value of β_t .

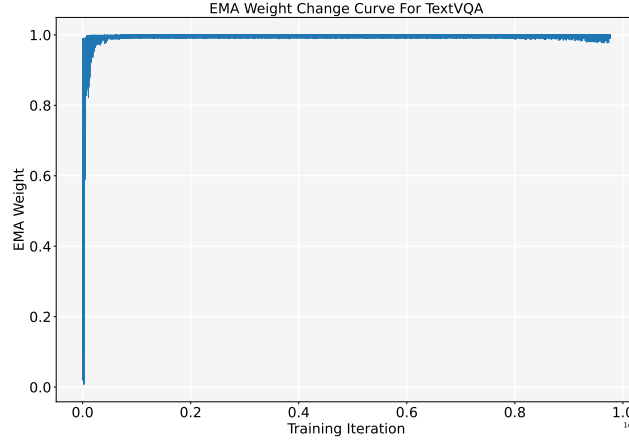


Figure 11: Distribution of β_t in training TextVQA dataset with Instruction type1. The horizontal axis represents iteration, and the vertical axis represents value of β_t .

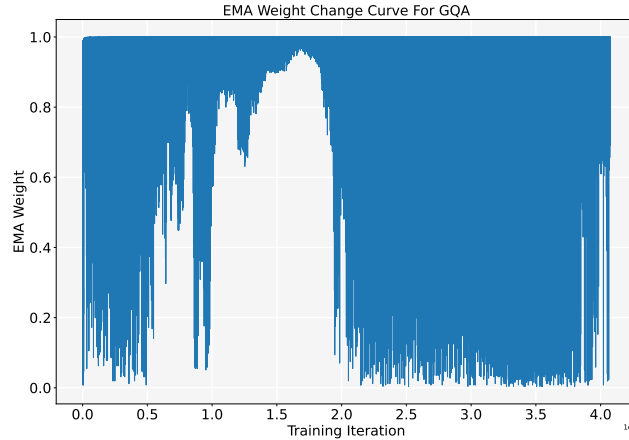


Figure 12: Distribution of β_t in training GQA dataset with Instruction type1. The horizontal axis represents iteration, and the vertical axis represents value of β_t .

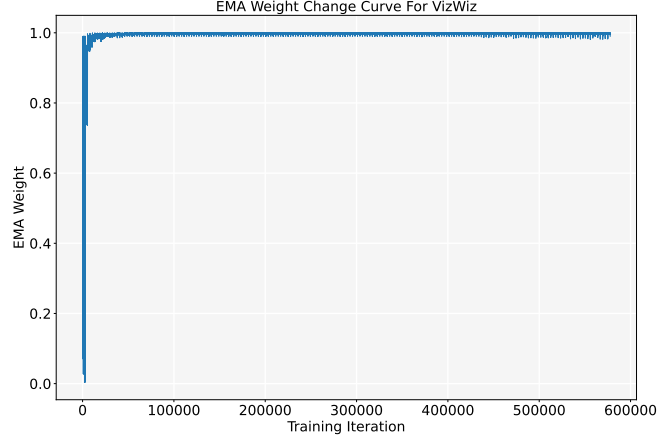


Figure 13: Distribution of β_t in training VizWiz dataset with Instruction type1. The horizontal axis represents iteration, and the vertical axis represents value of β_t .

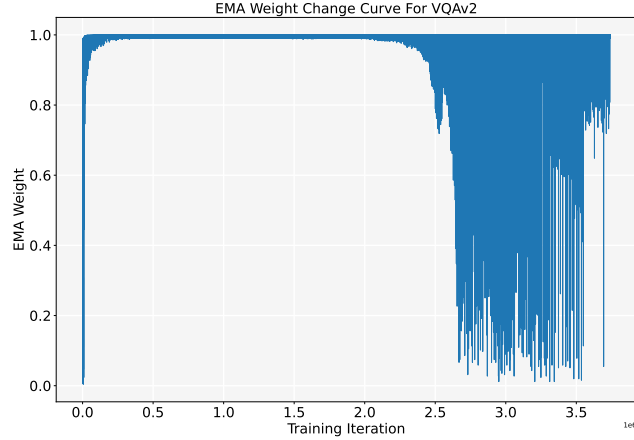


Figure 14: Distribution of β_t in training VQAv2 dataset with Instruction type1. The horizontal axis represents iteration, and the vertical axis represents value of β_t .

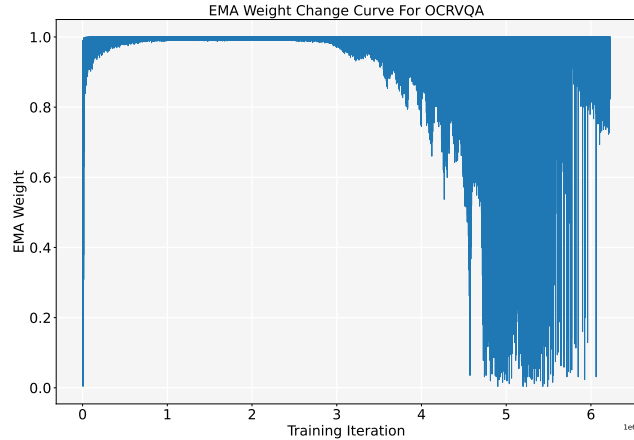


Figure 15: Distribution of β_t in training OCRVQA dataset with Instruction type1. The horizontal axis represents iteration, and the vertical axis represents value of β_t .

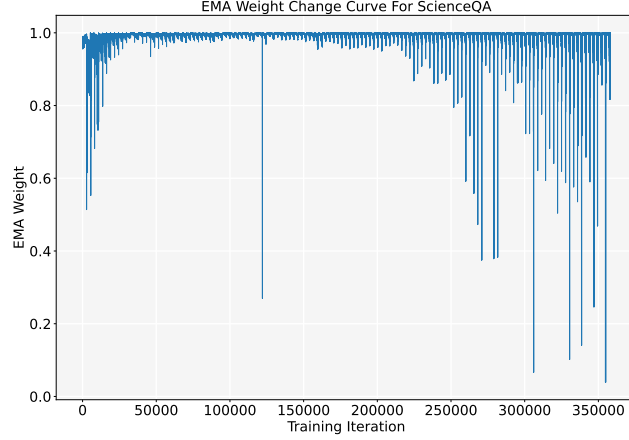


Figure 16: Distribution of β_t in training ScienceQA dataset with Instruction type2. The horizontal axis represents iteration, and the vertical axis represents value of β_t .

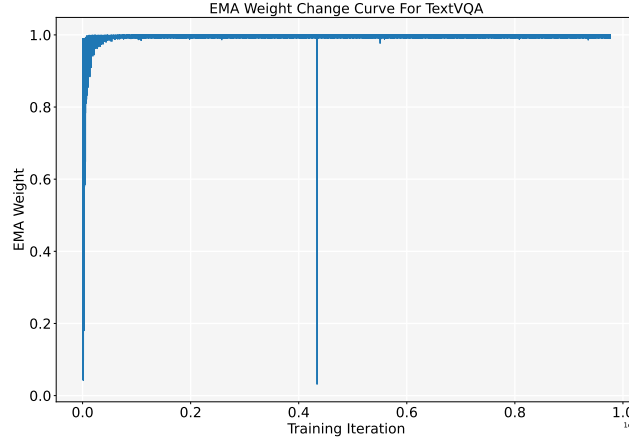


Figure 17: Distribution of β_t in training TextVQA dataset with Instruction type2. The horizontal axis represents iteration, and the vertical axis represents value of β_t .

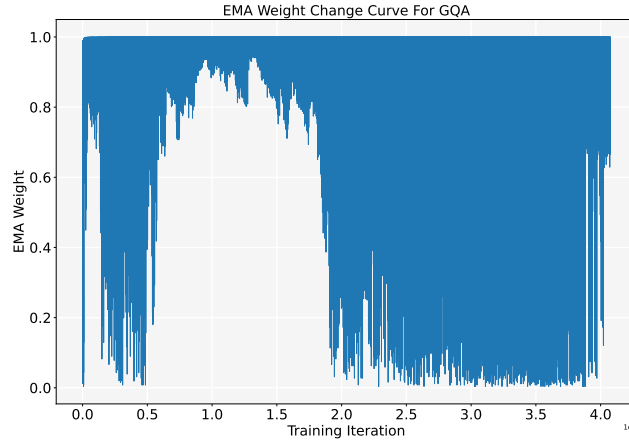


Figure 18: Distribution of β_t in training GQA dataset with Instruction type2. The horizontal axis represents iteration, and the vertical axis represents value of β_t .

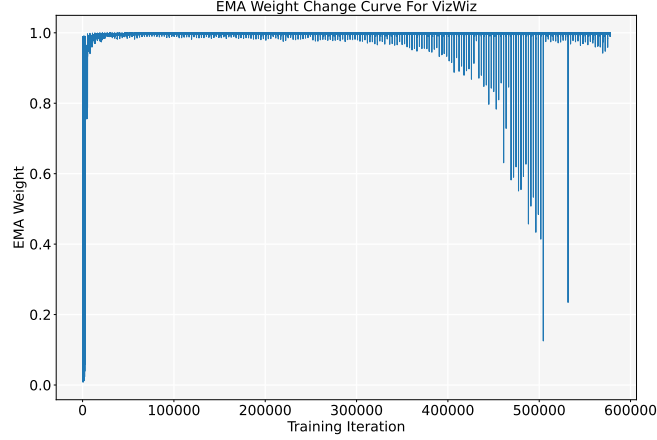


Figure 19: Distribution of β_t in training VizWiz dataset with Instruction type2. The horizontal axis represents iteration, and the vertical axis represents value of β_t .

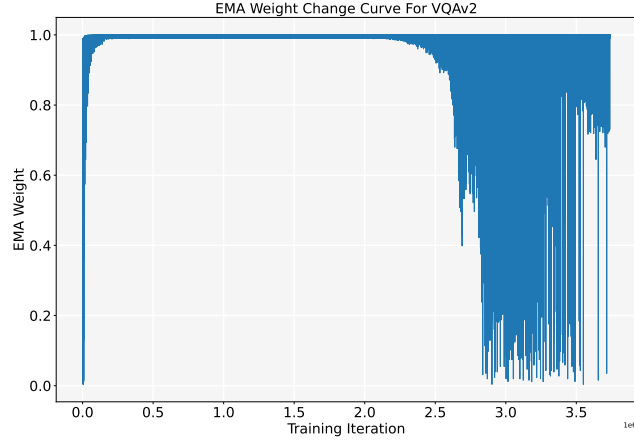


Figure 20: Distribution of β_t in training VQAv2 dataset with Instruction type2. The horizontal axis represents iteration, and the vertical axis represents value of β_t .

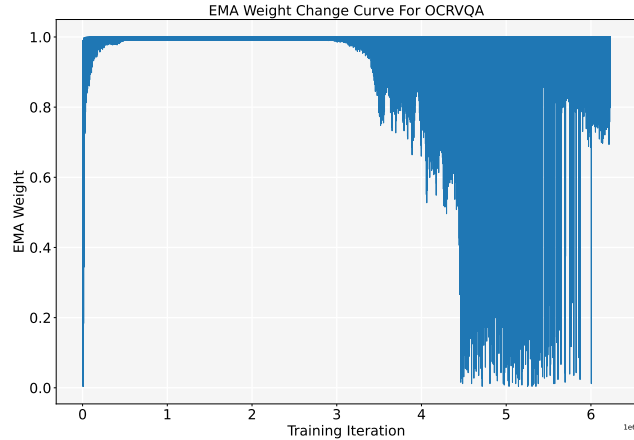


Figure 21: Distribution of β_t in training OCRVQA dataset with Instruction type2. The horizontal axis represents iteration, and the vertical axis represents value of β_t .

A.12 ALGORITHM OF LLACA

Algorithm 1: Multimodal Large Language Continual Assistant (Training phase)

Input: Pre-trained ViT model f_{vis} , Pre-trained Vicuna-7B model f_{lan} with inserted LoRA f_{low} , projection layer f_{prj} , embedding layer ϕ , number of datasets D , number of iterations T , training set $\{\{G_i^t, T_i^t, I_i^t, y_i^t\}_{i=1}^{n_t}\}_{t=1}^T$, learning rate η , loss function \mathcal{L}_x .

Output: inserted LoRA f_{low}^* , projection layer f_{prj}^* .

initialize: $f_{low}, f_{low}^*, f_{prj}, f_{prj}^*$.

```

for  $d = 1, \dots, D$  do
  for  $epoch = 1$  do
    for  $t = 1, \dots, T$  do
      1. Draw a mini-batch  $B = \{\{G_i^t, T_i^t, I_i^t, y_i^t\}_{i=1}^{n_t}\}_{t=1}^{n_t}$ .
      for  $(G, T, I, y)$  in  $B$  do
        2. Encode  $G$  into image feature  $g_t$  by  $g_t = f_{vis}(G)$ .
        3. Project  $g_t$  from image feature space to text feature space as  $p_t = f_{prj}(g_t)$ .
        4. Embed  $T$  and  $I$  into text feature  $e_t$  by  $e_t = \phi([T, I])$ .
        5. Prepend  $p_t$  with  $e_t$  by  $[p_t; e_t]$ .
        6. Obtain prediction by  $\hat{y} = f_{lan}([p_t; e_t])$ .
        7. Calculate per batch loss  $\mathcal{L}_B$  by accumulating  $\mathcal{L}_x(y, \hat{y})$ .
        8. Backward propagation and Update  $f_{low}$  and  $f_{prj}$  with optimizer.
        9. Record  $f_{low}, f_{prj}$  and the corresponding gradients  $\mathcal{L}'$  at iteration  $t$ .
        # EMA weight calculate.
        10. Calculate EMA weight  $\beta_t$  according to Eq.(58).
        # EMA parameter update.
        11. Update  $f_{low}^*$  and  $f_{prj}^*$  by Eq.(1).
        12. Remove and clear  $f_{low}, f_{prj}$  and  $\mathcal{L}'$  at iteration  $t - 1$ .
      end
    end
  end
  13. Save checkpoints of  $f_{low}^*$  and  $f_{prj}^*$ .
end

```

Algorithm 2: Multimodal Large Language Continual Assistant (Testing phase)

Input: Pre-trained ViT model f_{vis} , Pre-trained Vicuna-7B model f_{lan} with inserted LoRA f_{low} , projection layer f_{prj} , embedding layer ϕ , number of datasets D , number of iterations T , test set $\{\{G_i^t, T_i^t, I_i^t, y_i^t\}_{i=1}^{n_t}\}_{t=1}^T$.

Output: prediction \hat{y} .

```

for  $d = 1, \dots, D$  do
  for  $t = 1, \dots, T$  do
    1. Draw a mini-batch  $B = \{\{G_i^t, T_i^t, I_i^t, y_i^t\}_{i=1}^{n_t}\}_{t=1}^{n_t}$ .
    for  $(G, T, I)$  in  $B$  do
      2. Encode  $G$  into image feature  $g_t$  by  $g_t = f_{vis}(G)$ .
      3. Project  $g_t$  from image feature space to text feature space as  $p_t = f_{prj}(g_t)$ .
      4. Embed  $T$  and  $I$  into text feature  $e_t$  by  $e_t = \phi([T, I])$ .
      5. Prepend  $p_t$  with  $e_t$  by  $[p_t; e_t]$ .
      6. Obtain prediction by  $\hat{y} = f_{lan}([p_t; e_t])$ .
    end
  end
end

```
