# Enhancing Multimodal Continual Instruction Tuning with BranchLoRA

**Duzhen Zhang**[1*†], **Yong Ren**[2*], **Zhong-Zhi Li**[2], **Yahan Yu**[3], **Jiahua Dong**[1], **Chenxing Li**[4]
**Zhilong Ji**[5] and **Jinfeng Bai**[5]
[1]Mohamed bin Zayed University of Artificial Intelligence, Abu Dhabi, UAE
[2]Institute of Automation, Chinese Academy of Sciences, Beijing, China
[3]Kyoto University, Kyoto, Japan [4]Tencent AI Lab, Beijing, China
[5]Tomorrow Advancing Life, Beijing, China
duzhen.zhang@mbzuai.ac.ae, {thurenyong, dongjiahua1995}@gmail.com
lizhongzhi2022@ia.ac.cn, yahan@nlp.ist.i.kyoto-u.ac.jp, chenxingli@tencent.com

## Abstract

Multimodal Continual Instruction Tuning (MCIT) aims to finetune Multimodal Large Language Models (MLLMs) to continually align with human intent across sequential tasks. Existing approaches often rely on the Mixture-of-Experts (MoE) LoRA framework to preserve previous instruction alignments. However, these methods are prone to Catastrophic Forgetting (CF), as they aggregate all LoRA blocks via simple summation, which compromises performance over time. In this paper, we identify a critical parameter inefficiency in the MoELoRA framework within the MCIT context. Based on this insight, we propose BranchLoRA, an asymmetric framework to enhance both efficiency and performance. To mitigate CF, we introduce a flexible tuning-freezing mechanism within BranchLoRA, enabling branches to specialize in intra-task knowledge while fostering inter-task collaboration. Moreover, we incrementally incorporate task-specific routers to ensure an optimal branch distribution over time, rather than favoring the most recent task. To streamline inference, we introduce a task selector that automatically routes test inputs to the appropriate router without requiring task identity. Extensive experiments on the latest MCIT benchmark demonstrate that BranchLoRA significantly outperforms MoELoRA and maintains its superiority across various MLLM sizes.[1]

## 1 Introduction

Multimodal Large Language Models (MLLMs) (Li et al., 2023; Liu et al., 2023b; Bai et al., 2023), which combine a visual encoder with a LLM, have achieved remarkable success in addressing various multimodal tasks. Instruction tuning (Dai et al., 2023) plays a pivotal role in aligning MLLMs with human intent, enabling the creation of versatile models with general-purpose capabilities. In practical scenarios, MLLMs are often required to adapt to new instructions to support evolving functionalities as knowledge and societal needs advance (Zheng et al., 2024a). However, current MLLMs remain static, limiting their ability to accommodate continually emerging demands. Retraining MLLMs from scratch to meet these requirements is costly and inefficient. To address this challenge, recent research has framed the problem within the paradigm of Multimodal Continual Instruction Tuning (MCIT). MCIT seeks to continually finetune MLLMs for new tasks while preserving their strong performance on previously learned ones.

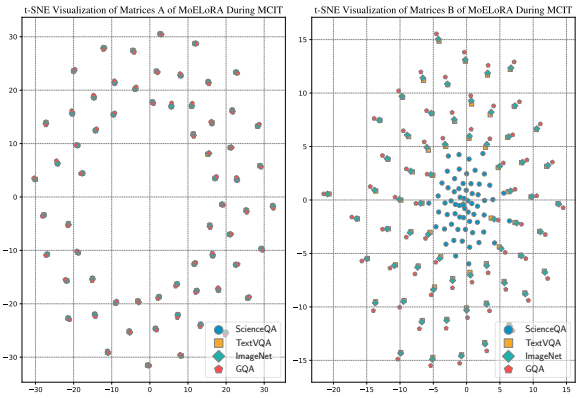


Figure 1: MoELoRA parameter analysis during MCIT across 4 sequential tasks: matrices = layers × experts.

Multimodal Continual Instruction Tuning (MCIT) faces a significant challenge: Catastrophic Forgetting (CF), where models lose or overwrite previously acquired knowledge when adapting to new tasks (McCloskey and Cohen, 1989; Zhang et al., 2023a; Dong et al., 2024). To mitigate this issue, Mixture-of-Experts (MoE) LoRA (Hu et al., 2022; Chen et al., 2024), illustrated in Figure 2 (a), utilizes multiple specialized experts (*i.e.*, LoRA blocks) to capture distinct knowledge from sequential tasks while employing

*Equal contributions.
†Corresponding author.
[1]Our code is available at https://github.com/BladeDancer957/BranchLoRA.



*Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5743–5756

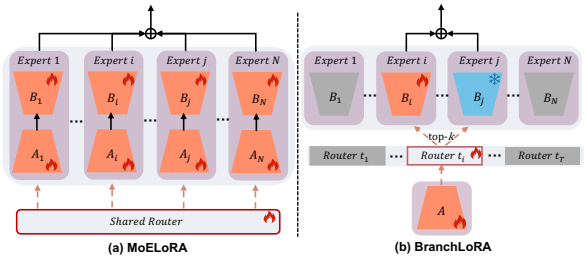


Figure 2: Diagram of MoELoRA and BranchLoRA.

a shared router to modulate their contributions. However, experiments on the MCIT benchmark have revealed a critical limitation of MoELoRA related to parameter inefficiency. As visualized in Figure 1, we analyzed the parameter behavior of MoELoRA when finetuned continually on 4 sequential tasks. The results highlight a key observation: the parameters in matrices $\boldsymbol{A}$ of MoELoRA tend to converge, capturing shared patterns across all tasks, whereas the parameters in matrices $\boldsymbol{B}$ remain distinct, focusing on the unique aspects. This suggests that MoELoRA suffers from parameter redundancy. Based on these findings, we propose that an improved architecture should adopt an asymmetric structure to better balance task-shared and task-specific learning.

Moreover, as shown in Figure 2 (a), MoELoRA aggregates all experts during MCIT, making it susceptible to overwriting previously learned knowledge when adapting to new tasks. Additionally, the router in MoELoRA is shared across tasks, and its continuous updates optimize the expert distribution primarily for the most recent task. This results in the forgetting of distributions optimized for earlier tasks. In summary, these two limitations cause MoELoRA to remain vulnerable to CF.

To address the above limitations, we introduce BranchLoRA (shown in Figure 2 (b)), an asymmetric framework designed to enhance MCIT performance. In this design, the shared matrix $\boldsymbol{A}$ serves as a "tree trunk" capturing task-invariant patterns, while multiple matrices $\boldsymbol{B}$ (*i.e.*, experts) act as "branches" that encode task-specific knowledge for sequential tasks. Instead of aggregating all experts, BranchLoRA adopts a dynamic sparse selection strategy, selecting only the top-$k$ experts based on their distribution. To further enhance intra-task learning, foster inter-task collaboration, and reduce inter-task interference during MCIT, we introduce a flexible tuning-freezing mechanism. When learning a new task, the router accesses frozen experts to leverage transferable knowledge from previous tasks, while optimizing tunable experts to acquire task-specific information for the new task.

To prevent bias toward the most recent task and maintain an optimal expert distribution over time, BranchLoRA incrementally incorporates task-specific routers. Furthermore, we integrate a task selector that automatically routes test samples to the appropriate router without requiring explicit task identity during inference, ensuring greater alignment with real-world scenarios. These innovations collectively strengthen BranchLoRA's anti-forgetting capabilities and deliver a marked improvement in overall MCIT performance.

Our contributions can be summarized as follows:

- Through MCIT experiments, we identify parameter inefficiency in MoELoRA and propose an asymmetric BranchLoRA architecture to balance shared and task-specific learning.
- We introduce a flexible tuning-freezing mechanism and task-specific routers within BranchLoRA, allowing experts to capture intra-task knowledge, foster inter-task collaboration, and minimize inter-task interference, thereby mitigating CF more effectively.
- Extensive experiments on the recent MCIT benchmark demonstrate that BranchLoRA significantly outperforms MoELoRA. Furthermore, BranchLoRA consistently achieves superior performance across different MLLM sizes (*e.g.*, LLaVA-7B and LLaVA-13B).

## 2 Related Work

### 2.1 MLLMs

Recently, MLLMs have seen significant advancements, extending LLMs to handle visual and textual inputs (Zhang et al., 2024; Chen et al., 2023a; Zhao et al., 2025b,a). They utilize the inherent reasoning abilities of LLMs alongside the high-quality representations of visual foundation models to achieve complex multimodal reasoning (Li et al., 2025) and content comprehension. BLIP2 (Li et al., 2023) bridges the gap between image and text by integrating a frozen LLM with a visual tower, using the Q-Former projector to enable modality alignment. LLaVA (Liu et al., 2023b) and MiniGPT4 (Zhu et al., 2023) simplify the alignment process with a straightforward linear projector, applying instruction tuning to better align with human intent. Recent models, such as LLaVA-1.5 (Liu

et al., 2023a), ShareGPT4V (Chen et al., 2023b), and LLaVA-NeXT (Liu et al., 2024), have further refined these strategies, enhancing performance across various multimodal benchmarks. These innovations underscore the growing capabilities of MLLMs in tackling complex reasoning tasks.

### 2.2 MCIT

While MLLMs have made significant progress, regular updates are crucial to endow them with new capabilities and ensure they stay aligned with the rapidly evolving landscape of human knowledge (Wu et al., 2024; Zheng et al., 2024b, 2025). To achieve this, MCIT is essential, as it allows models to continuously incorporate emerging data without the costly need for retraining from scratch. Recently, MCIT benchmarks have been introduced to finetune MLLMs across sequential tasks (He et al., 2023; Chen et al., 2024). However, MCIT faces the challenge of CF, where the model forgets earlier knowledge when learning new tasks (Goodfellow et al., 2013; Dong et al., 2022; Zhang et al., 2023b, 2025). Existing approaches, such as the MoELoRA paradigm, attempt to maintain prior instruction alignment (Chen et al., 2024), but they still struggle with forgetting due to the simple aggregation of all LoRA experts, leading to performance degradation over time.

## 3 Preliminary

### 3.1 Problem Formulation

MCIT (He et al., 2023; Chen et al., 2024) is designed to enable MLLMs to continually instruction-tune on new datasets without incurring the expense of full re-training. Unlike traditional continual learning (De Lange et al., 2021), MCIT emphasizes the effective use of natural language instructions to mitigate CF and promote knowledge transfer. It is formulated as a sequential stream of datasets, denoted by $\mathcal{T}_{\text{seq}} = \{t_1, ..., t_T\}$, where $T$ represents the total number of datasets or tasks. Importantly, these datasets are diverse in nature and are not confined to specific domains or categories.

Each dataset $t_i \in \mathcal{T}_{\text{seq}}$ includes a natural language instruction $I_{t_i}$, a training set $\mathcal{D}_{t_i}^{\text{train}}$, and a test set $\mathcal{D}_{t_i}^{\text{test}}$. The objective of MCIT is to sequentially train a single model $\mathcal{M}$ on the dataset stream while maintaining robust performance across all previously encountered tasks. Notably, during inference, the model is presented with test samples without prior knowledge of their associated tasks.

### 3.2 MoELoRA

To alleviate the challenge of CF, Chen et al. adopt the widely used MoELoRA approach (Dou et al., 2023; Liu et al., 2023c) within the context of MCIT. This method leverages multiple experts to acquire specialized knowledge tailored to different tasks. MoELoRA is composed of two key components: a fixed expert pool and a router. The pool consists of multiple identical yet independent LoRA blocks (Hu et al., 2022), which are prepended into each Feed-Forward (FF) layer of MLLM. Meanwhile, the router is responsible for modeling a probability distribution that determines the output weights of these experts (Fedus et al., 2022). In particular, given an intermediate representation $\boldsymbol{x}$ produced by the preceding attention layer, the output of the MoELoRA layer can be expressed as follows:

$$\begin{aligned} \boldsymbol{h} &= \boldsymbol{x}\boldsymbol{W}_f + \frac{\alpha}{r}\sum_{j=1}^{N} R(\boldsymbol{x})_{[j]} E_j(\boldsymbol{x}) \\ R(\boldsymbol{x}) &= \mathit{Softmax}(\boldsymbol{x}_{[0]}\boldsymbol{W}_r),\ E_j(\boldsymbol{x}) = \boldsymbol{x}\boldsymbol{A}_j\boldsymbol{B}_j \end{aligned}, \tag{1}$$

where $\boldsymbol{W}_f \in \mathbb{R}^{d_{\text{in}} \times d_{\text{out}}}$ represents the parameters of the FF layer, while the rank $r$ determines the number of trainable parameters. The constant hyperparameter $\alpha$ is introduced to finetune the effect of $r$. The model employs $N$ experts, where the router $R(\cdot)$ takes the first token of an intermediate representation $\boldsymbol{x}_{[0]} \in \mathbb{R}^{1 \times d_{\text{in}}}$ as input and is parameterized by trainable $\boldsymbol{W}_r \in \mathbb{R}^{d_{\text{in}} \times N}$ to assign output weights across these experts. Each expert, denoted as $E_j(\cdot)$, is characterized by two trainable low-rank matrices, $\boldsymbol{A}_j \in \mathbb{R}^{d_{\text{in}} \times \frac{r}{N}}$ and $\boldsymbol{B}_j \in \mathbb{R}^{\frac{r}{N} \times d_{\text{out}}}$. These matrices have a reduced rank of $\frac{r}{N}$, ensuring that the total number of trainable parameters matches that of a single LoRA setup, thereby maintaining computational efficiency.

## 4 Method

### 4.1 Analysis of MoELoRA in MCIT Context

The motivation behind MoELoRA (Chen et al., 2024) is to leverage multiple experts (*i.e.*, smaller LoRA blocks) to capture specialized knowledge from different tasks. To explore how these multiple experts reduce task interference, we conduct experiments by continually finetuning MoELoRA on 4 sequential tasks from the MCIT benchmark (Chen et al., 2024): ScienceQA (Lu et al., 2022), TextVQA (Singh et al., 2019), ImageNet (Deng et al., 2009), and GQA (Hudson and Manning,

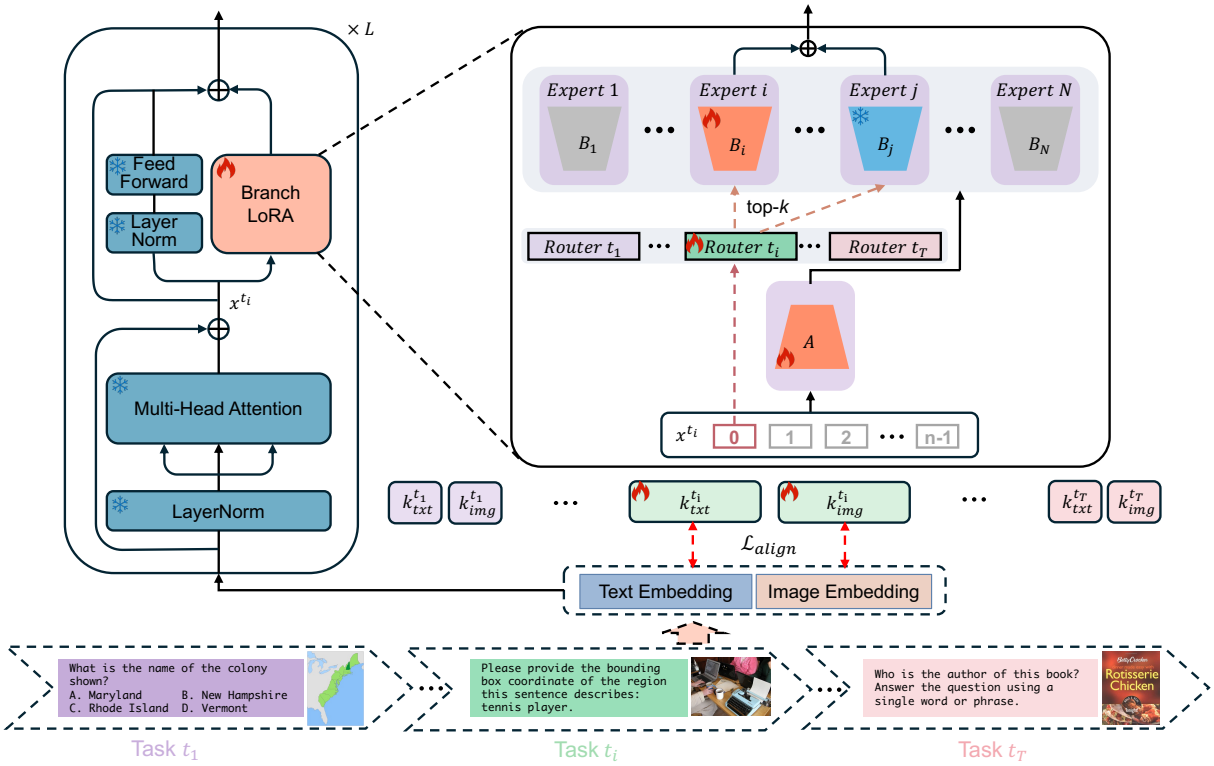


Figure 3: The overall framework of the proposed BranchLoRA. The shared matrix $\boldsymbol{A}$ captures task-invariant patterns, while multiple matrices $\boldsymbol{B}$ (*i.e.*, experts) encode task-specific knowledge for sequential tasks. BranchLoRA is integrated into the feed-forward module of each MLLM layer, with its input being the intermediate representation $\boldsymbol{x}^{t_i}$ from the multi-head attention module. The task-specific router processes the first token of $\boldsymbol{x}^{t_i}$ to generate expert weights, enabling a dynamic sparse selection strategy to combine expert outputs. Additionally, task-specific keys are trained to facilitate automatic task selection, eliminating the need for explicit task identity during inference.

2019). As illustrated in Figure 1, we use the t-SNE technique (Van der Maaten and Hinton, 2008) to visualize the parameters of matrices $\boldsymbol{A}$ and $\boldsymbol{B}$ across all experts from every MLLM layer. This analysis reveals a key finding:

**Observation:** *When multiple experts are fine-tuned continually on sequential tasks, the parameters of matrix $\boldsymbol{A}$ tend to converge, while those of matrix $\boldsymbol{B}$ remain distinguishable.*

Specifically, all matrices $\boldsymbol{A}$ exhibit strong similarity, leading to noticeable overlaps across four sequential tasks, as illustrated in Figure 1. In contrast, matrices $\boldsymbol{B}$ are more distinct and easier to differentiate. We hypothesize that this discrepancy arises from their initialization methods: matrix $\boldsymbol{A}$ tends to capture shared features across tasks, whereas matrix $\boldsymbol{B}$ focuses on adapting to task-specific variations. This suggests that the existing MoELoRA approach may suffer from parameter redundancy in the context of MCIT. A similar pattern was also reported by HydraLoRA (Tian et al., 2024) in multi-task finetuning scenarios.

### 4.2 BranchLoRA

Building on the previous observation, we propose an asymmetric BranchLoRA framework aimed at improving both efficiency and performance. In this framework, **the parameters of matrix $\boldsymbol{A}$ are shared** across sequential tasks to optimize parameter usage, while multiple matrices $\boldsymbol{B}$ are employed to capture task-specific knowledge. To further enhance intra-task learning, promote inter-task collaboration, and reduce inter-task interference during MCIT, we introduce a flexible tuning-freezing mechanism and task-specific routers within BranchLoRA. This approach effectively mitigates CF. The overview of the BranchLoRA framework is shown in Figure 3.

#### 4.2.1 Flexible Tuning-freezing Mechanism

Rather than always aggregating all experts, which can lead to overwriting previously learned knowledge when adapting to new tasks, we employ a sparse selection strategy. This approach dynamically selects the top-$k$ experts based on their proba-

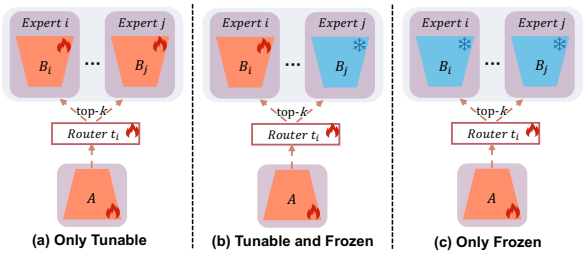


Figure 4: Diagram of the Flexible Tuning-Freezing Mechanism. 3 distinct expert combinations for an input during new task $t_i$ training: (a) Only Tunable, (b) Tunable and Frozen, and (c) Ony Frozen, with the shared matrix $\boldsymbol{A}$ and task-specific Router $t_i$ being tunable.

bility distribution. The router $R(\cdot)$ in equation (1) is modified as follows:

$$R(\boldsymbol{x}) = \mathit{Softmax}(\text{top-}k(\boldsymbol{x}_{[0]}\boldsymbol{W}_r)), \quad (2)$$

where top-$k(\cdot)$ selects the $k$ most relevant experts and assigns a value of $-\infty$ to the others.

To further enhance experts (*i.e.*, matrices $\boldsymbol{B}$) with both intra-task knowledge and inter-task collaboration, we introduce a flexible tuning-freezing mechanism. To be more specific, after training on the current task, we analyze the distribution of router outputs across all samples. The top-$k$ most activated experts are then frozen during training on subsequent tasks to preserve the task-specific knowledge of the current task. In this way, when confronted with a new task, the router can access these frozen experts to leverage transferable knowledge from previous tasks, while optimizing the unfrozen experts to capture task-specific information for the new task. As shown in Figure 4, during new task training, the router can activate (a) only the tunable experts, (b) both tunable and previously frozen experts, or (c) exclusively the frozen experts from past tasks (with only the matrix A and router being tunable). This mechanism enables experts to collaboratively consolidate their knowledge, mirroring how the human brain strengthens and integrates new information with existing memories.

#### 4.2.2 Task-specific Routers with Auto-Selector

As tasks are trained sequentially, the router in Equation (2) is updated continuously, which can result in the forgetting of distributions optimized for earlier tasks. To avoid bias toward the most recent task and ensure an optimal expert distribution over time, we incrementally introduce task-specific routers. The updated version of Equation (2) is as follows:

$$R^{t_i}(\boldsymbol{x}) = \mathit{Softmax}(\text{top-}k(\boldsymbol{x}^{t_i}_{[0]}\boldsymbol{W}^{t_i}_r)), \quad (3)$$

where the task-specific router $R^{t_i}(\cdot)$ with trainable $\boldsymbol{W}^{t_i}_r$ is utilized to assign output weights to different experts by using the first token of an intermediate representation $\boldsymbol{x}^{t_i}_{[0]}$ from task $t_i$ as input.

However, task-specific routers rely on task identity during inference, which poses challenges in real-world scenarios where task identity may not always be available. To overcome this limitation, inspired by (Wang et al., 2022b,a), we propose learning task-specific keys to enable automatic task selection without requiring explicit task identity during inference. These keys are progressively aligned with the image and text embeddings of samples from task $t_i$ during training. The approach is formulated as follows:

$$\begin{aligned}\mathcal{L}_{\text{align}} = &\sum\nolimits_j (1 - Cos(\boldsymbol{e}^{t_i}_{j,\text{img}}, \boldsymbol{k}^{t_i}_{\text{img}})) \\ &+ \sum\nolimits_j (1 - Cos(\boldsymbol{e}^{t_i}_{j,\text{txt}}, \boldsymbol{k}^{t_i}_{\text{txt}}))\end{aligned}, \quad (4)$$

where $Cos(\cdot)$ represents cosine similarity, while $\boldsymbol{e}^{t_i}_{j,\text{img}}$ and $\boldsymbol{e}^{t_i}_{j,\text{txt}}$ denote the image and text embeddings of the $j$-th sample from task $t_i$, respectively. $\boldsymbol{k}^{t_i}_{\text{img}}$ and $\boldsymbol{k}^{t_i}_{\text{txt}}$ represent the trainable keys corresponding to the images and texts in task $t_i$.

During inference, we can compute the similarity between the embeddings of a test sample and the trained keys for each task, and automatically route the sample to the router corresponding to the keys with the highest similarity.

Finally, the total loss in BranchLoRA is:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{t_i} + \lambda \mathcal{L}_{\text{align}}, \quad (5)$$

where $\mathcal{L}_{t_i}$ is the autoregressive generation loss for task $t_i$ and $\lambda$ is the loss coefficient.

## 5 Experimental Settings

### 5.1 Datasets

We adopt the setup of the latest MCIT benchmark, CoIN (Chen et al., 2024), which incorporates 8 multimodal datasets spanning various domains and tasks: ScienceQA (Lu et al., 2022), TextVQA (Singh et al., 2019), ImageNet (Deng et al., 2009), GQA (Hudson and Manning, 2019), VizWiz (Gurari et al., 2018), Grounding (Kazemzadeh et al., 2014; Mao et al., 2016), VQAv2 (Goyal et al., 2017), and OCR-VQA (Mishra et al., 2019). These datasets are used in the same training order as in the main experiment. Detailed statistics for these datasets are provided in Table 6 of Appendix A.

| Method | Accuracy on Each Task | | | | | | | | Overall Results | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | ScienceQA | TextVQA | ImageNet | GQA | VizWiz | Grounding | VQAv2 | OCR-VQA | ACC⇑ | MAA⇑ | BWT⇑ |
| Zero-shot | 49.91 | 2.88 | 0.33 | 2.08 | 0.90 | 0.00 | 0.68 | 0.17 | – | 7.12 | – |
| LoRA | 82.45 | 49.99 | 96.05 | 56.40 | 55.45 | 31.27 | 62.20 | 57.08 | 28.74 | 32.97 | -32.62 |
| | 21.26 | 28.74 | 10.25 | 36.78 | 32.45 | 0.83 | 42.50 | 57.08 | | | |
| LwF* | 81.36 | 50.59 | 96.84 | 51.98 | 48.19 | 25.13 | 41.30 | 64.12 | 30.41 | 34.95 | -27.03 |
| | 26.78 | 37.52 | 12.64 | 35.18 | 25.24 | 2.87 | 38.92 | 64.12 | | | |
| EWC* | 82.81 | 51.76 | 96.80 | 46.19 | 48.68 | 26.82 | 66.37 | 63.46 | 32.90 | 36.93 | -27.46 |
| | 30.33 | 36.08 | 11.62 | 35.75 | 37.50 | 3.48 | 44.98 | 63.46 | | | |
| MoELoRA* | 80.15 | 49.60 | 96.65 | 58.40 | 51.54 | 22.22 | 65.79 | 60.10 | 33.73 | 39.32 | -26.83 |
| | 67.15 | 39.17 | 4.87 | 33.78 | 25.31 | 0.71 | 38.74 | 60.10 | | | |
| MoELoRA | 75.78 | 51.73 | 96.70 | 59.42 | 58.88 | 37.50 | 64.22 | 60.08 | **37.13** | **42.76** | **-25.91** |
| | 63.09 | 38.63 | 10.50 | 37.38 | 43.62 | 0.59 | 43.15 | 60.08 | | | |
| **BranchLoRA(Ours)** | 86.70 | 56.14 | 96.46 | 56.04 | 59.43 | 39.48 | 65.02 | 62.14 | **44.20** | **49.94** | **-20.98** |
| | 68.24 | 40.18 | 24.60 | 41.40 | 49.83 | 15.94 | 51.23 | 62.14 | | | |
| Multi-task | 56.77 | 49.35 | 95.55 | 56.65 | 53.90 | 30.09 | 59.50 | 55.65 | – | 57.18 | – |

Table 1: Main results on the LLaVA-1.5-7B model using the CoIN benchmark. For sequential finetuning methods (except for Zero-shot and Multi-task), the first row presents the results for each task evaluated immediately after tuning on the corresponding task (*i.e.*, $A_{i,i}$), and the second row shows the results for each task after finetuning on the final task (*i.e.*, $A_{T,i}$). The **red** highlights the highest overall performance, and the **blue** indicates the second-highest performance. $*$ represents results from our re-implementation. Other results are cited from CoIN (Chen et al., 2024).

### 5.2 Baselines

We evaluate the performance of BranchLoRA by comparing it against the following baselines: **Zero-shot**: Directly assessing each task using pre-trained MLLMs without additional finetuning; **LoRA** (Hu et al., 2022): Updating knowledge sequentially through two low-rank matrices, while preserving the original parameters of the pre-trained MLLM; **MoELoRA** (Chen et al., 2024): Utilizing multiple identical yet independent LoRAs to capture specialized knowledge from sequential tasks and achieving State-Of-The-Art (SOTA) MCIT performance on the CoIN benchmark; **Multi-task**: Performing finetuning with LoRA on all tasks simultaneously, rather than using sequential training.

Moreover, we compare BranchLoRA with classic continual learning methods: **LwF** (Li and Hoiem, 2017) and **EWC** (Kirkpatrick et al., 2017). More details about the above baselines can be found in Appendix B.

### 5.3 Evaluation Metrics

We evaluate the outputs of MLLMs by comparing them to ground truths in a word-by-word manner. Since tasks produce outputs in various formats, the evaluation metrics are tailored accordingly. Detailed descriptions of these comparisons are provided in Appendix C.

In line with the CoIN benchmark (Chen et al., 2024), we evaluate MCIT performance using three metrics: Average Accuracy (ACC) to measure performance after training on the final task, Mean Average Accuracy (MAA) to assess performance throughout the training process, and Backward Transfer (BWT) to quantify the extent of CF. These metrics are defined as follows: (1) ACC $= \frac{1}{T}\sum_{i=1}^{T} A_{T,i}$, where $A_{T,i}$ is the performance on $i$-th task after training the final task $T$. (2) MAA $= \frac{1}{T}\sum_{i=1}^{T}(\frac{1}{i}\sum_{k=1}^{i} A_{i,k})$, where $A_{i,k}$ is the performance on $k$-th task after training the task $i$. (3) BWT $= \frac{1}{T}\sum_{i=1}^{T}(A_{T,i} - A_{i,i})$, where $A_{i,i}$ is the performance on $i$-th task after training on $i$-th task.

### 5.4 Implementation Details

We use the well-established **LLaVA-1.5-7B** and **LLaVA-1.5-13B** (Liu et al., 2023b) as our backbone models, integrating LoRA (Hu et al., 2022) into the MLLM. During the MCIT process, both the vision encoder and the LLM remain frozen, with only the projector and LoRA components being finetuned. For a single LoRA, the rank $r$ is set to 128, and the hyperparameter $\alpha$ is set to 256. In the case of MoELoRA, we set the number of experts $N$ to 8, and the rank of the small matrices within each expert is adjusted to $r/N = 128/8 = 16$ to ensure computational efficiency, as specified in its original paper. For a fair comparison, the number of experts and the rank in BranchLoRA are kept consistent

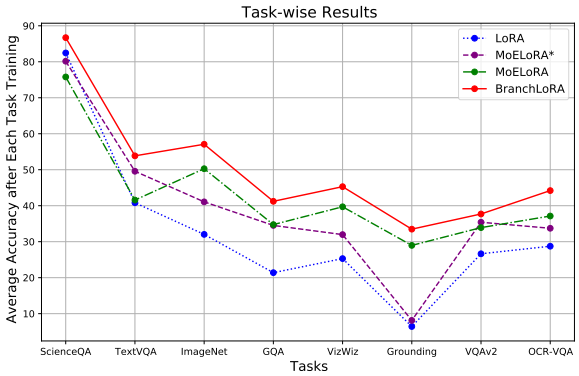


Figure 5: Task-wise performance comparison on the LLaVA-1.5-7B model. Our BranchLoRA method consistently outperforms the previous MCIT baselines, LoRA and MoELoRA, in all task-wise evaluations.

| **Method** | #. Trainable Parameters⇓ | Training Time (ms/batch)⇓ | ACC⇑ | MAA⇑ | BWT⇑ |
|---|---|---|---|---|---|
| MoELoRA | 350M | 62 | 37.13 | 42.76 | -25.91 |
| **BranchLoRA(Ours)** | **222M** | **51** | **44.20** | **49.94** | **-20.98** |

Table 2: The efficiency analysis of BranchLoRA on LLaVA-1.5-7B. The **Bold** represents the best results.

| **Variant** | ACC⇑ | MAA⇑ | BWT⇑ |
|---|---|---|---|
| MoELoRA | 37.13 | 42.76 | -25.91 |
| + shared matrix $\boldsymbol{A}$ | 38.19 | 43.95 | -25.32 |
| + dynamic sparse selection strategy | 39.96 | 45.53 | -23.77 |
| + flexible tuning-freezing mechanism | 42.22 | 47.76 | -22.41 |
| + task-specific router (**BranchLoRA**) | **44.20** | **49.94** | **-20.98** |

Table 3: The ablation study of BranchLoRA on LLaVA-1.5-7B. When compared with BranchLoRA, all ablation variants degrade MCIT performance. It verifies the importance of all components to address MCIT collaboratively. The **Bold** represents the best results.

with MoELoRA, selecting the top-2 experts, with the loss coefficient $\lambda$ set to 1.0. All experiments are carried out on 8 NVIDIA H800 GPUs, each with 80GB of memory.

## 6 Experimental Results

### 6.1 Main Results

To evaluate the performance of BranchLoRA, we conduct experiments on LLaVA-1.5-7B using the CoIN benchmark, with results shown in Table 1.

Despite LLaVA-1.5-7B's vast knowledge, its Zero-shot performance on specialized tasks is suboptimal, with an MAA of 7.12. While updating the model across all tasks simultaneously is resource-intensive, the Multi-task approach proves effective, reaching an MAA of 57.18. Sequential finetuning methods like LoRA, MoELoRA, and BranchLoRA, outperform Multi-task on most tasks (*i.e.*, the first row). This advantage may result from the model's ability to focus on one task, thereby minimizing interference caused by diverse instructions from all tasks. However, LoRA lacks strategies to mitigate CF, which leads to a loss of previously learned instructions and results in a BWT of $-32.62$.

MoELoRA alleviates CF by using distinct experts for each task, achieving ACC, MAA, and BWT scores of 37.13, 42.76, and $-25.91$, respectively. In contrast, BranchLoRA introduces an asymmetric structure that better balances shared and task-specific learning. It uses a flexible tuning-freezing mechanism and task-specific routers, allowing experts to capture intra-task knowledge, encourage cross-task collaboration, and reduce inter-task interference. This design significantly improves CF mitigation. Consequently, BranchLoRA outperforms MoELoRA by a substantial margin, achieving new SOTA performance on CoIN, with ACC, MAA, and BWT scores of 44.20, 49.94, and $-20.98$, respectively. Moreover, BranchLoRA significantly outperforms traditional continual learning methods like LwF and EWC.

As shown in Figure 5, task-wise comparison results (*i.e.*, task-wise MAA, which measures the average accuracy on all previous tasks, including the current one, after finetuning each task) further highlight the effectiveness of BranchLoRA. These results reveal that BranchLoRA consistently outperforms prior SOTA baselines, LoRA and MoELoRA, across all comparisons, showcasing its robustness and adaptability in sequential learning.

### 6.2 Efficiency Analysis

We compare the number of trainable parameters and training time of BranchLoRA with the previous SOTA, MoELoRA, with results presented in Table 2 (based on LLaVA-1.5-7B). Training time refers to the duration required for forward and backward propagation of a data batch, measured in milliseconds (ms). To minimize variance, we averaged the time over 100 batches. BranchLoRA significantly reduces both trainable parameters and training time compared to MoELoRA, while maintaining superior performance, providing quantitative evidence of the efficiency of our approach.

### 6.3 Ablation Study

This section presents ablation studies on the LLaVA-1.5-7B model to evaluate the contributions of individual components in BranchLoRA, as detailed in Table 3. Sharing multiple $\boldsymbol{A}$ matrices

| Method | Accuracy on Each Task | | | | | | | | Overall Results | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | ScienceQA | TextVQA | ImageNet | GQA | VizWiz | Grounding | VQAv2 | OCR-VQA | ACC⇑ | MAA⇑ | BWT⇑ |
| MoELoRA | 76.16 | 55.92 | 97.67 | 55.85 | 62.95 | 48.10 | 68.32 | 64.05 | 42.51 | 49.14 | -23.62 |
| | 70.66 | 45.82 | 12.90 | 36.15 | 49.93 | 10.79 | 49.76 | 64.05 | | | |
| BranchLoRA | 87.26 | 60.97 | 98.28 | 54.08 | 63.89 | 46.98 | 70.61 | 66.37 | **49.27** | **55.73** | **-19.29** |
| | 78.06 | 49.85 | 29.46 | 39.76 | 55.28 | 19.01 | 56.36 | 66.37 | | | |

Table 4: The results of MoELoRA and BranchLoRA on the larger LLaVA-1.5-13B model.

| Type | Accuracy on Each Task | | | | | | | | Overall Results | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | ScienceQA | TextVQA | ImageNet | GQA | VizWiz | Grounding | VQAv2 | OCR-VQA | ACC⇑ | MAA⇑ | BWT⇑ |
| Original | 86.70 | 56.14 | 96.46 | 56.04 | 59.43 | 39.48 | 65.02 | 62.14 | 44.20 | 49.94 | -20.98 |
| | 68.24 | 40.18 | 24.60 | 41.40 | 49.83 | 15.94 | 51.23 | 62.14 | | | |
| Diverse | 86.70 | 57.26 | 97.42 | 54.87 | 56.46 | 37.94 | 67.91 | 64.16 | 45.06 | 50.25 | -20.28 |
| | 69.52 | 43.39 | 23.84 | 44.71 | 46.84 | 15.30 | 52.74 | 64.16 | | | |
| 10Type | 88.43 | 58.98 | 98.30 | 55.97 | 54.77 | 39.21 | 69.52 | 65.44 | **46.47** | **51.76** | **-19.86** |
| | 70.22 | 44.82 | 25.37 | 43.82 | 47.67 | 19.57 | 54.82 | 65.44 | | | |

Table 5: The results of BranchLoRA on the LLaVA-1.5-7B model about different instruction templates.

within MoELoRA not only significantly improves parameter efficiency but also slightly enhances MCIT performance, indicating that the asymmetric structure better balances task-shared and task-specific learning. Introducing a dynamic sparse selection strategy further improves performance by reducing the impact on previously learned knowledge compared to aggregating all experts. Building on this, the flexible tuning-freezing mechanism enhances intra-task learning while fostering cross-task collaboration, leading to additional performance improvements. Finally, the incorporation of task-specific routers minimizes inter-task interference, resolving the challenge of continuous updates causing expert distributions to overly favor the most recent task. Together, these components constitute the complete BranchLoRA framework.

Moreover, despite occasional misclassifications by the automatic task selector during inference (achieving an average accuracy of 95.8% across all tasks), BranchLoRA consistently outperforms SOTA methods across various metrics. Its ability to automatically select tasks during inference aligns with the demands of real-world applications, further highlighting its practical value.

### 6.4 More Explorations

**Larger Backbone Size** We conduct experiments on larger MLLM backbones, such as LLaVA-1.5-13B. The results, presented in Table 4, highlight the scalability and effectiveness of BranchLoRA in resource-intensive scenarios. Larger models like LLaVA-1.5-13B experience less CF compared to smaller models like LLaVA-1.5-7B (shown in Table 1) in MCIT across multiple tasks. However, CF still occurs, particularly when there is a significant difference in task similarity. Across both the 7B and 13B models, BranchLoRA consistently outperforms the previous SOTA, MoELoRA, demonstrating the scalability and versatility of our approach across different MLLM backbone sizes.

**Impact of Instruction Diversity** In our main experiment (Section 6.1), some tasks rely on similar instruction templates (*i.e.*, Original). To examine how template variety affects the MCIT performance of our BranchLoRA, we introduce two additional template types: Diverse and 10Type. The Diverse template involves distinct instructions tailored to each task, while the 10Type template involves randomly selecting from 10 different instruction templates for each task. A detailed list of these three instruction types for all tasks is provided in Table 7 of Appendix D.

Table 5 shows the MCIT performance of our BranchLoRA across these three template types. Our analysis reveals that simply switching to the Diverse template type has a limited effect on overall performance. However, employing random selection from multiple instruction templates significantly improves overall results. This improvement

likely stems from the model's ability to better understand the underlying instructional intent when exposed to varied templates.

These findings highlight the value of incorporating diverse instructions for each task rather than relying on a single instruction. Introducing instruction variety enhances the model's capacity to interpret instructional intent, mitigates the decline in instruction-following ability, and improves robustness to variations in instructions.

## 7 Conclusion

In this paper, we present BranchLoRA, an innovative solution to address the critical parameter inefficiency and CF in the MoELoRA framework for MCIT. By introducing a flexible tuning-freezing mechanism and task-specific routers with automatic selector, BranchLoRA enables experts to specialize in intra-task knowledge while promoting inter-task collaboration, more effectively mitigating CF. Extensive experiments on different MLLM sizes (*e.g.*, LLaVA-1.5-7B and LLaVA-1.5-13B) using the latest CoIN benchmark demonstrate that BranchLoRA significantly outperforms MoELoRA, offering a more efficient and robust approach for continual alignment with human intent across sequential tasks.

In the future, a promising direction is to integrate BranchLoRA with advanced model merging techniques (Alexandrov et al., 2024; Ilharco et al., 2022; Sukhbaatar et al., 2024; Daheim et al.; Yadav et al., 2023; Ram et al., 2024), which typically assign different levels of importance to task-specific features. These methods enable more nuanced consolidation of knowledge across tasks. Incorporating them into the BranchLoRA framework could further mitigate CF by dynamically preserving essential information from previous tasks while adapting to new ones. This synergy has the potential to improve the robustness and scalability of BranchLoRA in more diverse and extended MCIT settings.

## Limitations

Despite its promising results, BranchLoRA has some limitations that need to be addressed. For instance, our experiments were primarily conducted using the recent MCIT benchmark, which may not fully capture the method's potential across diverse tasks and domains. To further validate the effectiveness of BranchLoRA, future research could explore its application on a wider array of continual instruction tuning benchmarks, including those that focus on non-multimodal tasks. Additionally, developing more comprehensive and challenging benchmarks that encompass a broader variety of multimodal tasks, real-world scenarios, and different domains would provide a more robust evaluation of BranchLoRA's performance. Such efforts would help assess the method's scalability, adaptability, and generalization capabilities in a broader context, enabling a deeper understanding of its strengths and limitations.

## References


Anton Alexandrov, Veselin Raychev, Mark Mueller, Ce Zhang, Martin Vechev, and Kristina Toutanova. 2024. Mitigating Catastrophic Forgetting in Language Transfer via Model Merging. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 17167–17186.

Jinze Bai, Shuai Bai, Shusheng Yang, Shijie Wang, Sinan Tan, Peng Wang, Junyang Lin, Chang Zhou, and Jingren Zhou. 2023. Qwen-vl: A frontier large vision-language model with versatile abilities. *arXiv preprint arXiv:2308.12966*.

Cheng Chen, Junchen Zhu, Xu Luo, Heng Tao Shen, Jingkuan Song, and Lianli Gao. 2024. CoIN: A benchmark of continual instruction tuning for multimodel large language models. In *The Thirty-eight Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

Fei-Long Chen, Du-Zhen Zhang, Ming-Lun Han, Xiu-Yi Chen, Jing Shi, Shuang Xu, and Bo Xu. 2023a. Vlp: A survey on vision-language pre-training. *Machine Intelligence Research*, 20(1):38–56.

Lin Chen, Jisong Li, Xiaoyi Dong, Pan Zhang, Conghui He, Jiaqi Wang, Feng Zhao, and Dahua Lin. 2023b. ShareGPT4V: Improving Large Multi-Modal Models with Better Captions. *arXiv preprint arXiv:2311.12793*.

Nico Daheim, Thomas Möllenhoff, Edoardo Ponti, Iryna Gurevych, and Mohammad Emtiyaz Khan. Model Merging by Uncertainty-Based Gradient Matching. In *The Twelfth International Conference on Learning Representations*.

Wenliang Dai, Junnan Li, Dongxu Li, Anthony Meng Huat Tiong, Junqi Zhao, Weisheng Wang, Boyang Li, Pascale Fung, and Steven C. H. Hoi. 2023. InstructBLIP: Towards General-purpose Vision-Language Models with Instruction Tuning. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Matthias De Lange, Rahaf Aljundi, Marc Masana, Sarah Parisot, Xu Jia, Aleš Leonardis, Gregory Slabaugh,

and Tinne Tuytelaars. 2021. A continual learning survey: Defying forgetting in classification tasks. *IEEE transactions on pattern analysis and machine intelligence*, 44(7):3366–3385.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. 2009. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee.

Jiahua Dong, Hongliu Li, Yang Cong, Gan Sun, Yulun Zhang, and Luc Van Gool. 2024. No One Left Behind: Real-World Federated Class-Incremental Learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(4):2054–2070.

Jiahua Dong, Lixu Wang, Zhen Fang, Gan Sun, Shichao Xu, Xiao Wang, and Qi Zhu. 2022. Federated Class-Incremental Learning. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Shihan Dou, Enyu Zhou, Yan Liu, Songyang Gao, Jun Zhao, Wei Shen, Yuhao Zhou, Zhiheng Xi, Xiao Wang, Xiaoran Fan, et al. 2023. Loramoe: Revolutionizing mixture of experts for maintaining world knowledge in language model alignment. *arXiv preprint arXiv:2312.09979*, 4(7).

William Fedus, Barret Zoph, and Noam Shazeer. 2022. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. *Journal of Machine Learning Research*, 23(120):1–39.

Ian J Goodfellow, Mehdi Mirza, Da Xiao, Aaron Courville, and Yoshua Bengio. 2013. An empirical investigation of catastrophic forgetting in gradient-based neural networks. *arXiv preprint arXiv:1312.6211*.

Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. 2017. 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.

Danna Gurari, Qing Li, Abigale J Stangl, Anhong Guo, Chi Lin, Kristen Grauman, Jiebo Luo, and Jeffrey P Bigham. 2018. Vizwiz grand challenge: Answering visual questions from blind people. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3608–3617.

Jinghan He, Haiyun Guo, Ming Tang, and Jinqiao Wang. 2023. Continual instruction tuning for large multimodal models. *arXiv preprint arXiv:2311.16206*.

Edward J Hu, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. 2022. LoRA: Low-Rank Adaptation of Large Language Models. In *International Conference on Learning Representations*.

Drew A Hudson and Christopher D Manning. 2019. 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.

Gabriel Ilharco, Mitchell Wortsman, Samir Yitzhak Gadre, Shuran Song, Hannaneh Hajishirzi, Simon Kornblith, Ali Farhadi, and Ludwig Schmidt. 2022. Patching open-vocabulary models by interpolating weights. *Advances in Neural Information Processing Systems*, 35:29262–29277.

Sahar Kazemzadeh, Vicente Ordonez, Mark Matten, and Tamara Berg. 2014. Referitgame: Referring to objects in photographs of natural scenes. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 787–798.

James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. 2017. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, 114(13):3521–3526.

Junnan Li, Dongxu Li, Silvio Savarese, and Steven C. H. Hoi. 2023. BLIP-2: Bootstrapping Language-Image Pre-training with Frozen Image Encoders and Large Language Models. In *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, pages 19730–19742.

Zhizhong Li and Derek Hoiem. 2017. Learning without forgetting. *IEEE transactions on pattern analysis and machine intelligence*, 40(12):2935–2947.

Zhong-Zhi Li, Duzhen Zhang, Ming-Liang Zhang, Jiaxin Zhang, Zengyan Liu, Yuxuan Yao, Haotian Xu, Junhao Zheng, Pei-Jie Wang, Xiuyi Chen, et al. 2025. From system 1 to system 2: A survey of reasoning large language models. *arXiv preprint arXiv:2502.17419*.

Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. 2023a. Improved Baselines with Visual Instruction Tuning. In *NeurIPS 2023 Workshop on Instruction Tuning and Instruction Following*.

Haotian Liu, Chunyuan Li, Yuheng Li, Bo Li, Yuanhan Zhang, Sheng Shen, and Yong Jae Lee. 2024. LLaVA-NeXT: Improved reasoning, OCR, and world knowledge.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. 2023b. Visual Instruction Tuning. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Qidong Liu, Xian Wu, Xiangyu Zhao, Yuanshao Zhu, Derong Xu, Feng Tian, and Yefeng Zheng. 2023c. Moelora: An moe-based parameter efficient fine-tuning method for multi-task medical applications. *arXiv preprint arXiv:2310.18339*.

Pan Lu, Swaroop Mishra, Tanglin Xia, Liang Qiu, Kai-Wei Chang, Song-Chun Zhu, Oyvind Tafjord, Peter Clark, and Ashwin Kalyan. 2022. Learn to explain: Multimodal reasoning via thought chains for science question answering. *Advances in Neural Information Processing Systems*, 35:2507–2521.

Junhua Mao, Jonathan Huang, Alexander Toshev, Oana Camburu, Alan L Yuille, and Kevin Murphy. 2016. Generation and comprehension of unambiguous object descriptions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 11–20.

Michael McCloskey and Neal J Cohen. 1989. Catastrophic interference in connectionist networks: The sequential learning problem. In *Psychology of learning and motivation*, volume 24, pages 109–165. Elsevier.

Anand Mishra, Shashank Shekhar, Ajeet Kumar Singh, and Anirban Chakraborty. 2019. Ocr-vqa: Visual question answering by reading text in images. In *2019 international conference on document analysis and recognition (ICDAR)*, pages 947–952. IEEE.

Dhananjay Ram, Aditya Rawal, Momchil Hardalov, Nikolaos Pappas, and Sheng Zha. 2024. DEM: Distribution Edited Model for Training with Mixed Data Distributions. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 19287–19301.

Amanpreet Singh, Vivek Natarajan, Meet Shah, Yu Jiang, Xinlei Chen, Dhruv Batra, Devi Parikh, and Marcus Rohrbach. 2019. Towards vqa models that can read. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8317–8326.

Sainbayar Sukhbaatar, Olga Golovneva, Vasu Sharma, Hu Xu, Xi Victoria Lin, Baptiste Rozière, Jacob Kahn, Daniel Li, Wen-tau Yih, Jason Weston, et al. 2024. Branch-Train-MiX: Mixing Expert LLMs into a Mixture-of-Experts LLM. *arXiv preprint arXiv:2403.07816*.

Chunlin Tian, Zhan Shi, Zhijiang Guo, Li Li, and Cheng-Zhong Xu. 2024. Hydralora: An asymmetric lora architecture for efficient fine-tuning. *Advances in Neural Information Processing Systems*, 37:9565–9584.

Laurens Van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-SNE. *Journal of machine learning research*, 9(11).

Zifeng Wang, Zizhao Zhang, Sayna Ebrahimi, Ruoxi Sun, Han Zhang, Chen-Yu Lee, Xiaoqi Ren, Guolong Su, Vincent Perot, Jennifer Dy, et al. 2022a. Dualprompt: Complementary prompting for rehearsal-free continual learning. In *European Conference on Computer Vision*, pages 631–648. Springer.

Zifeng Wang, Zizhao Zhang, Chen-Yu Lee, Han Zhang, Ruoxi Sun, Xiaoqi Ren, Guolong Su, Vincent Perot, Jennifer Dy, and Tomas Pfister. 2022b. Learning to prompt for continual learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 139–149.

Tongtong Wu, Linhao Luo, Yuan-Fang Li, Shirui Pan, Thuy-Trang Vu, and Gholamreza Haffari. 2024. Continual Learning for Large Language Models: A Survey. *arXiv preprint arXiv:2402.01364*.

Prateek Yadav, Derek Tam, Leshem Choshen, Colin A Raffel, and Mohit Bansal. 2023. Ties-merging: Resolving interference when merging models. *Advances in Neural Information Processing Systems*, 36:7093–7115.

Duzhen Zhang, Wei Cong, Jiahua Dong, Yahan Yu, Xiuyi Chen, Yonggang Zhang, and Zhen Fang. 2023a. Continual Named Entity Recognition without Catastrophic Forgetting. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 8186–8197.

Duzhen Zhang, Hongliu Li, Wei Cong, Rongtao Xu, Jiahua Dong, and Xiuyi Chen. 2023b. Task relation distillation and prototypical pseudo label for incremental named entity recognition. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*, pages 3319–3329.

Duzhen Zhang, Yahan Yu, Chenxing Li, Jiahua Dong, Dan Su, Chenhui Chu, and Dong Yu. 2024. Mm-llms: Recent advances in multimodal large language models. In *Findings of the Association for Computational Linguistics ACL 2024*.

Duzhen Zhang, Yahan Yu, Chenxing Li, Jiahua Dong, and Dong Yu. 2025. Federated Incremental Named Entity Recognition. *IEEE Transactions on Audio, Speech and Language Processing*.

Xuanle Zhao, Xuexin Liu, Haoyue Yang, Xianzhen Luo, Fanhu Zeng, Jianling Li, Qi Shi, and Chi Chen. 2025a. ChartEdit: How Far Are MLLMs From Automating Chart Analysis? Evaluating MLLMs' Capability via Chart Editing. *arXiv preprint arXiv:2505.11935*.

Xuanle Zhao, Xianzhen Luo, Qi Shi, Chi Chen, Shuo Wang, Wanxiang Che, Zhiyuan Liu, and Maosong Sun. 2025b. ChartCoder: Advancing Multimodal Large Language Model for Chart-to-Code Generation. *arXiv preprint arXiv:2501.06598*.

Junhao Zheng, Qianli Ma, Zhen Liu, Binquan Wu, and Huawen Feng. 2024a. Beyond Anti-Forgetting: Multimodal Continual Instruction Tuning with Positive Forward Transfer. *arXiv preprint arXiv:2401.09181*.

Junhao Zheng, Shengjie Qiu, Chengming Shi, and Qianli Ma. 2024b. Towards Lifelong Learning of Large Language Models: A Survey. *arXiv preprint arXiv:2406.06391*.

Junhao Zheng, Chengming Shi, Xidi Cai, Qiuke Li, Duzhen Zhang, Chenxing Li, Dong Yu, and Qianli Ma. 2025. Lifelong Learning of Large Language

Model based Agents: A Roadmap. *arXiv preprint arXiv:2501.07278*.

Deyao Zhu, Jun Chen, Xiaoqian Shen, Xiang Li, and Mohamed Elhoseiny. 2023. Minigpt-4: Enhancing vision-language understanding with advanced large language models. *arXiv preprint arXiv:2304.10592*.

## A Datasets

The detailed statistics for the eight multimodal datasets included in the CoIN benchmark (Chen et al., 2024) are presented in Table 6.

## B Baselines

We assess the performance of BranchLoRA by comparing it with several baselines: **Zero-shot**: Evaluating each task directly with pre-trained MLLMs, without any further finetuning; **LoRA** (Hu et al., 2022): Updating knowledge sequentially through two low-rank matrices, while retaining the original parameters of the pre-trained MLLM; **MoELoRA** (Chen et al., 2024): Using multiple identical yet independent LoRAs to capture specialized knowledge from sequential tasks, achieving SOTA performance on the CoIN benchmark; **Multi-task**: Performing finetuning with LoRA on all tasks simultaneously, rather than using sequential training.

Additionally, we perform comparisons with classic continual learning techniques, such as **LwF** (Li and Hoiem, 2017) and **EWC** (Kirkpatrick et al., 2017). Following the CoIN benchmark (Chen et al., 2024), we compute the Fisher matrix for EWC by randomly selecting 1000 samples from each task and set the hyperparameter $\lambda$ to 0.1. For LwF, we retain 100 logits per task to calculate the distillation loss, with $\lambda$ also set to 0.1.

## C Comparison Details

In line with the CoIN benchmark (Chen et al., 2024), we evaluate performance on the Image Question Answering task (encompassing VQAv2, ScienceQA, TextVQA, GQA, VizWiz, and OCR-VQA) by measuring the accuracy of predicted answers against the ground truth, similar to the approach used in LLaVA (Liu et al., 2023b). For classification tasks, the evaluation metric involves comparing the predicted labels to the actual ones. In the referring expression comprehension (grounding) task, we adopt the commonly used Intersection over Union (IoU) metric to assess prediction accuracy. A prediction is deemed correct if the IoU between the predicted and ground-truth bounding boxes exceeds 0.5.

## D Different Instruction Templates

The list of instruction templates for each task is provided in Table 7. **Original**: Certain tasks use similar instructions. **Diverse**: Unique instruction templates specifically designed for each task. **10Type**: A randomly selected instruction template from a set of 10 distinct templates for each task.

| Task | Dataset | Instruction | Train Number | Test Number |
|---|---|---|---|---|
| **Knowledge Grounded IQA** | ScienceQA | Answer with the option's letter from the given choices directly | 12k | 4k |
| **Reading Comprehension IQA** | TextVQA | Answer the question using a single word or phrase | 34k | 5k |
| **Classification** | ImageNet | What is the object in the image? Answer the question using a single word or phrase | 129k | 5k |
| **Visual Reasoning IQA** | GQA | Answer the question using a single word or phrase | 72k | 1k |
| **Blind People IQA** | VizWiz | Answer the question using a single word or phrase | 20k | 8k |
| **Grounding** | RefCOCO RefCOCO+ RefCOCOg | Please provide the bounding box coordinate of the region this sentence describes: <description> | 55k | 31k |
| **Image Question Answering (IQA)** | VQAv2 | Answer the question using a single word or phrase | 82k | 107k |
| **OCR IQA** | OCR-VQA | Answer the question using a single word or phrase | 165k | 100k |

Table 6: The statistic of collected multimodal datasets in the CoIN benchmark (Chen et al., 2024).

| Task | Original | Diverse | 10Type |
|---|---|---|---|
| **ScienceQA** | Answer with the option's letter from the given choices directly | Answer with the option's letter from the given choices directly | Answer with the option's letter from the given choices directly<br>Select the correct answer from the given choices and respond with the letter of the chosen option<br>Determine the correct option from the provided choices and reply with its corresponding letter<br>Pick the correct answer from the listed options and provide the letter of the selected option<br>Identify the correct choice from the options below and respond with the letter of the correct option<br>From the given choices, choose the correct answer and respond with the letter of that choice<br>Choose the right answer from the options and respond with its letter<br>Select the correct answer from the provided options and reply with the letter associated with it<br>From the given choices, select the correct answer and reply with the letter of the chosen option<br>Identify the correct option from the choices provided and respond with the letter of the correct option<br>From the given choices, pick the correct answer and respond by indicating the letter of the correct option |
| **TextVQA** | Answer the question using a single word or phrase | Capture the essence of your response in a single word or a concise phrase | Answer the question with just one word or a brief phrase<br>Use one word or a concise phrase to respond to the question<br>Answer using only one word or a short, descriptive phrase<br>Provide your answer in the form of a single word or a brief phrase<br>Use a single word or a short phrase to respond to the question<br>Summarize your response in one word or a concise phrase<br>Respond to the question using a single word or a brief phrase<br>Provide your answer in one word or a short, descriptive phrase<br>Answer the question with a single word or a brief, descriptive phrase<br>Capture the essence of your response in one word or a short phrase<br>Capture the essence of your response in a single word or a concise phrase |
| **ImageNet** | Answer the question using a single word or phrase | Express your answer in a single word or a short, descriptive phrase | Express your answer in a single word or a short, descriptive phrase<br>Provide your answer using a single word or a brief phrase<br>Describe the content of the image using one word or a concise phrase<br>Respond to the question with a single word or a short, descriptive phrase<br>Classify the image content using only one word or a brief phrase<br>Give your answer in the form of a single word or a concise phrase<br>Use a single word or a short phrase to categorize the image content<br>Express your answer with one word or a short, descriptive phrase<br>Identify the type of content in the image using one word or a concise phrase<br>Summarize your response in a single word or a brief phrase<br>Use one word or a short phrase to classify the content of the image |
| **GQA** | Answer the question using a single word or phrase | Respond to the question briefly, using only one word or a phrase | Respond to the question with a single word or a short phrase<br>Respond to the question using only one word or a concise phrase<br>Answer the question with a single word or a brief phrase<br>Respond with one word or a short phrase<br>Provide your answer in the form of a single word or a concise phrase<br>Respond to the question with just one word or a brief phrase<br>Answer the question using a single word or a concise phrase<br>Provide your response using only one word or a short phrase<br>Respond to the question with a single word or a brief phrase<br>Respond to the question using just one word or a concise phrase<br>Answer the question with one word or a short phrase |
| **VizWiz** | Answer the question using a single word or phrase | Provide a succinct response with a single word or phrase | Answer the question using only one word or a concise phrase<br>Respond to the question using only one word or a concise phrase<br>Respond to the question with a single word or a brief phrase<br>Provide your answer using just one word or a short phrase<br>Respond with one word or a concise phrase<br>Answer the question with just one word or a brief phrase<br>Use a single word or a short phrase to answer the question<br>Provide your answer in the form of one word or a brief phrase<br>Reply to the question using one word or a concise phrase<br>Answer with a single word or a short phrase<br>Use one word or a brief phrase to answer the question |
| **Grounding** | Please provide the bounding box coordinate of the region this sentence describes | Please provide the bounding box coordinate of the region this sentence describes | Identify and provide the bounding box coordinates that match the description given in this sentence<br>Extract and provide the bounding box coordinates based on the region described in the sentence<br>Please provide the bounding box coordinate of the region this sentence describes<br>Find and provide the bounding box coordinates for the region mentioned in the sentence<br>Provide the coordinates of the bounding box that correspond to the region described in the sentence<br>Give the bounding box coordinates as described in the sentence<br>Determine and provide the bounding box coordinates based on the description in the sentence<br>Identify and provide the coordinates of the bounding box described in the sentence<br>Provide the coordinates for the bounding box based on the region described in the sentence<br>Extract and provide the coordinates for the bounding box described in the sentence<br>Identify and give the coordinates of the bounding box as described by the sentence |
| **VQAv2** | Answer the question using a single word or phrase | Answer the question using a single word or phrase | Answer the question using a single word or phrase<br>Answer the question with a single word or a brief phrase<br>Use one word or a short phrase to respond to the question<br>Answer the question using just one word or a concise phrase<br>Provide your answer to the question using only one word or a brief phrase<br>Respond to the question with a single word or a short phrase Use a single word or phrase to answer the question<br>Provide an answer using only one word or a brief phrase<br>Answer the question succinctly with one word or a brief phrase<br>Answer the question with just one word or a short phrase<br>Respond to the question using a single word or a concise phrase |
| **OCR-VQA** | Answer the question using a single word or phrase | Condense your answer for each question into a single word or concise phrase | Answer with the option's letter from the given choices directly<br>Select the correct answer from the given choices and respond with the letter of the chosen option<br>Determine the correct option from the provided choices and reply with its corresponding letter<br>Pick the correct answer from the listed options and provide the letter of the selected option<br>Identify the correct choice from the options below and respond with the letter of the correct option<br>From the given choices, choose the correct answer and respond with the letter of that choice<br>Choose the right answer from the options and respond with its letter<br>Select the correct answer from the provided options and reply with the letter associated with it<br>From the given choices, select the correct answer and reply with the letter of the chosen option<br>Identify the correct option from the choices provided and respond with the letter of the correct option<br>From the given choices, pick the correct answer and respond by indicating the letter of the correct option |

Table 7: The list of different instruction templates for each task (Chen et al., 2024).