From Experts to Bases: Orthogonal Subspace Mixture for Continual Multimodal Instruction Tuning

Pei Chen[1], Xilai Wang[1], Qixu Shi[1], Zejian Li[1], Lingyun Sun[1]*
[1]Zhejiang University, Hangzhou, China
{chenpei, xilaiwang, shiqixu, zejianlee, sunly}@zju.edu.cn

Abstract

Multimodal Continual Instruction Tuning (MCIT) is essential for adapting Multimodal Large Language Models (MLLMs) to dynamic data streams, yet preventing catastrophic forgetting remains a major challenge. Existing parameter-efficient approaches often face a dilemma: fixed architectures suffer from knowledge interference, while dynamic strategies incur inefficient capacity expansion, limiting scalability. We propose **MoBLoRA** (Mixture-of-Bases LoRA), a novel framework for MCIT. Motivated by our geometric analysis revealing subspace redundancy across sequential tasks, MoBLoRA shifts the paradigm from expert selection to subspace mixing: it decomposes adaptation weights into a globally shared pool of orthonormal bases to capture task-invariant knowledge, and lightweight mixing matrices to encode task-specific variations. This design effectively decouples knowledge accumulation from task reconstruction. Experiments on standard benchmarks show MoBLoRA significantly outperforms state-of-the-art methods while maintaining superior parameter efficiency.[1]

1 Introduction

Recent advancements in Multimodal Large Language Models (MLLMs) have enabled impressive vision-language reasoning on diverse tasks (Yin et al., 2024; Liu et al., 2023). In practice, MLLMs must adapt to evolving instructions and data streams to keep pace with new knowledge. Since retraining from scratch is costly, recent studies formulate this challenge as Multimodal Continual Instruction Tuning (MCIT) (Chen et al., 2024; He et al., 2023), which requires incrementally tuning MLLMs on a sequence of tasks while maintaining performance on previously learned ones.

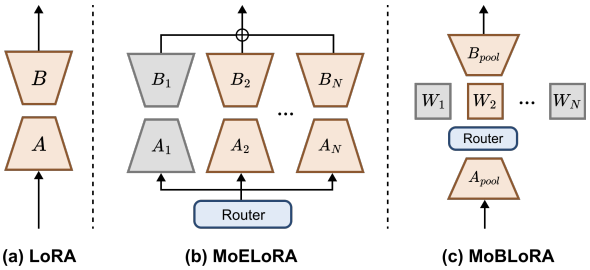


Figure 1: A comparison among LoRA, MoELoRA, and our proposed MoBLoRA.

However, this paradigm is hindered by "Catastrophic Forgetting" (Zhai et al., 2023), where learning new capabilities often leads to forgetting prior knowledge. Consequently, balancing memory stability and learning plasticity remains a challenge in MCIT (Wang et al., 2024).

Low-Rank Adaptation (LoRA) (Hu et al., 2022), as shown in Fig. 1 (a), has become a common solution for adapting MLLMs. As a Parameter-Efficient Fine-Tuning (PEFT) (Xu et al., 2023) method, LoRA freezes the backbone and learns only lightweight low-rank updates, substantially reducing the cost of full-parameter fine-tuning while maintaining strong generalization. In MCIT, this shifts the focus to how LoRA modules should be organized over a task sequence to acquire new capabilities without degrading previously learned ones.

Most existing LoRA-based MCIT methods follow either sharing or isolation. Fixed-architecture methods, typically employing Mixture-of-Experts (MoE) frameworks (Shazeer et al., 2017) as shown in Fig. 1 (b), reuse a shared pool of LoRA experts to enhance parameter efficiency. However, updating these shared experts often leads to interference and catastrophic forgetting (Chen et al., 2024). In contrast, dynamic-architecture strategies such as ProgLoRA (Yu et al., 2025) allocate new LoRA modules to each incoming task to isolate knowledge, which better preserves past

*Corresponding author.
[1]Our code is available at https://github.com/ChouChouisme/MoBLoRA

Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 10545–10561

performance but incurs linear parameter growth and limits positive transfer due to strict separation. Consequently, existing approaches face a dilemma: one must either compromise stability to gain plasticity, or sacrifice efficiency and transferability to preserve memory.

We argue that the above dilemma arises from overlooking latent redundancy in the LoRA parameter space. Prior work demonstrates that within a single fine-tuning task, LoRA updates exhibit a fine-grained subspace structure (Wu et al., 2024) and contain substantial internal redundancy (Zhu et al., 2025). We extend this insight to the cross-task setting, finding that LoRA adapters learned independently on diverse domains spontaneously converge towards shared subspace components (see Section 3.2). This cross-task redundancy suggests a new design point beyond sharing entire experts or isolating entire modules.

Building on this insight, we introduce **M**ixture-**o**f-**B**ases **LoRA** (**MoBLoRA**) to better reconcile the stability-plasticity dilemma, as illustrated in Fig. 1 (c). In contrast to existing MoELoRA approaches (Chen et al., 2024) that route inputs to coarse-grained experts, MoBLoRA shifts the paradigm to fine-grained orthogonal subspace mixing. Specifically, we decompose the parameter space of LoRA into globally shared orthonormal basis pools, which serve as "skill primitives" to capture task-invariant patterns, and task-specific mixing matrices that encode the unique linear combinations of these bases. This design moves beyond the discrete expert selection of MoE, enabling the model to accumulate general knowledge within the shared orthonormal bases while maintaining task-specific distinctiveness through independent mixing matrices. Consequently, MoBLoRA achieves knowledge sharing at the subspace level while ensuring interference-free isolation at the combination level. Our contributions are threefold.

- We propose MoBLoRA, a framework that employs orthogonal subspace mixing to decouple shared basis accumulation from task-specific reconstruction, effectively resolving the stability-plasticity dilemma.
- We provide a geometric analysis of sequential LoRA modules, uncovering inherent subspace redundancy that motivates our transition from discrete experts to shared bases.
- Extensive experiments on MCIT benchmarks demonstrate that MoBLoRA achieves state-of-the-art performance, surpassing dynamic baselines with superior parameter efficiency.

2 Related Work

2.1 Multimodal Large Language Models

MLLMs extend LLMs to images and other modalities, enabling multimodal perception and reasoning (Yin et al., 2024). Typical MLLM architectures integrate a pre-trained vision encoder with an LLM through alignment modules (Yin et al., 2024; Alayrac et al., 2022; Li et al., 2023). For example, Flamingo (Alayrac et al., 2022) fuses visual features via interleaved cross-attention layers, whereas LLaVA (Liu et al., 2023) uses a lightweight projector to embed visual tokens. Subsequent research has optimized these foundations by refining alignment mechanisms (Dai et al., 2023) and scaling instruction-tuning data (Bai et al., 2023). Specifically, LLaVA-1.5 (Liu et al., 2024) improves zero-shot performance via an MLP connector and academic data. State-of-the-art closed source models such as GPT-4o (Hurst et al., 2024) and Gemini (Team, 2025) have pushed the boundaries of world modeling, demonstrating exceptional capabilities in complex visual reasoning.

2.2 Multimodal Continual Instruction Tuning

Instruction tuning (Ouyang et al., 2022) aligns MLLMs with human intent. However, static tuning fails in real-world scenarios where data distributions evolve rapidly. Thus, MCIT formulates this problem as balancing learning plasticity and memory stability (Chen et al., 2024; He et al., 2023). Current MCIT methods predominantly rely on LoRA and generally fall into two paradigms. Regularization methods (Qiao et al., 2024a; Chen et al., 2025) mitigate forgetting by constraining weight updates or gradients to preserve historical knowledge. Parameter isolation strategies include fixed-network approaches (Zhang et al., 2023), which optimize parameter reuse within a static capacity, and dynamic-architecture methods (He et al., 2023; Zhang et al., 2025; Yu et al., 2025), which explicitly expand model capacity to accommodate new task knowledge. Regularization strategies like Model Tailor (Zhu et al., 2024) and CIA (Qiao et al., 2025) preserve knowledge via sparse masking and dynamic exponential moving averages, respectively. In parameter isolation, fixed-network methods such as MoELoRA (Chen et al., 2024) utilize static MoE

adapters, while dynamic-architecture methods like Eproj (He et al., 2023), BranchLoRA (Zhang et al., 2025), and ProgLoRA (Yu et al., 2025) reduce conflicts by expanding capacity through task grouping, asymmetric matrices, and progressive module instantiation. We propose MoBLoRA to mitigate the stability-efficiency dilemma by shifting the paradigm from discrete expert selection to orthogonal subspace mixing.

3 Preliminary

3.1 Problem Formulation

Following standard MCIT settings (Chen et al., 2024), we adapt a pre-trained MLLM θ_0 to a sequence of K tasks $\mathcal{T} = \{T_1, \ldots, T_K\}$. Each task T_k ($k \in \{1, \ldots, K\}$) comprises a dataset $\mathcal{D}_k = \{(V_i, Q_i, A_i)\}_{i=1}^{N_k}$, where N_k is the sample size of task T_k, and (V_i, Q_i, A_i) denote the image, instruction, and response. When training on task T_k, the model accesses only $\mathcal{D}_k$ to optimize parameters θ_k. The objective is to minimize prediction loss on the current task while preserving performance on all prior tasks $T_{1:k-1}$, thereby balancing plasticity and stability.

3.2 Cross-Task Subspace Redundancy

Standard LoRA (Hu et al., 2022) modulates a frozen pre-trained weight matrix $W_0 \in \mathbb{R}^{d_{out} \times d_{in}}$ by injecting a trainable update $\Delta W = BA$, where $B \in \mathbb{R}^{d_{out} \times r}$ and $A \in \mathbb{R}^{r \times d_{in}}$ are low-rank task-specific matrices ($r \ll d_{in}, d_{out}$). Motivated by the significant internal redundancy observed within LoRA modules (Zhu et al., 2025), we investigate whether similar subspace redundancy persists across diverse tasks.

Specifically, we independently fine-tuned LLaVA-1.5-7B (Liu et al., 2023) on four instruction tuning datasets: ScienceQA (Lu et al., 2022), TextVQA (Singh et al., 2019), ImageNet (Deng et al., 2009), and GQA (Hudson and Manning, 2019). These datasets cover diverse multimodal tasks, including knowledge-grounded QA, reading comprehension, image classification, and visual reasoning. To investigate the intrinsic redundancy, we decompose the LoRA matrices into rank-1 vectors denoted as $A = [a_1, \ldots, a_r]^\top$ and $B = [b_1, \ldots, b_r]$, where a_i ($i \in \{1, \ldots, r\}$) represents the direction of feature projection, while b_i signifies the direction of feature reconstruction (Zhu et al., 2025). Subsequently, we quantify the cross-task structural overlap between tasks m and n ($m, n \in \{1, \ldots, K\}$, $m \neq n$) via the pairwise cosine similarity matrix $\mathcal{S}$. Taking the projection vectors (matrix A) as a representative example, the entry $\mathcal{S}_{i,j}$ is defined as $\mathcal{S}_{i,j} = \frac{{a_i^m}^\top a_j^n}{\|a_i^m\| \|a_j^n\|}$ for $i, j \in [1, r]$.

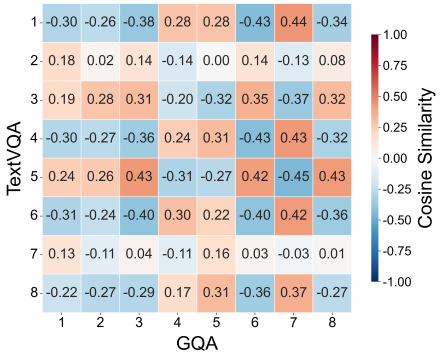


Figure 2: Visualizing the LoRA B matrix (Layer 0 query projection) reveals subspace redundancy between GQA and TextVQA.

We empirically visualize the pairwise cosine similarity of query projection vectors. Using the interaction between TextVQA and GQA in Layer 0 as a representative example, Fig. 2 reveals identifiable structural overlap in their learned subspaces despite distinct task semantics. The heatmap indicates that specific basis vectors exhibit moderate correlations (e.g., similarity > 0.4). Notably, this geometric redundancy is also observed across other layers and task pairs, suggesting that independent fine-tuning implicitly captures common functional patterns. The pervasive similarity implies that the decomposed rank-1 vectors across diverse tasks effectively reside within a common lower-dimensional subspace. This geometric insight lays the empirical foundation for our subsequent method design.

4 Methodology

4.1 Overview: From Isolated Experts to Shared Bases

Motivated by the finding that rank-1 vectors across diverse tasks reside within a shared low-dimensional subspace (Section 3.2), we propose to replace independent storage with a generative reconstruction process. By identifying a spanning basis of this subspace, we can synthesize any specific rank-1 vector through linear combinations, thereby eliminating inherent redundancies and achieving substantial parameter efficiency.

Driven by this intuition, MoBLoRA is designed

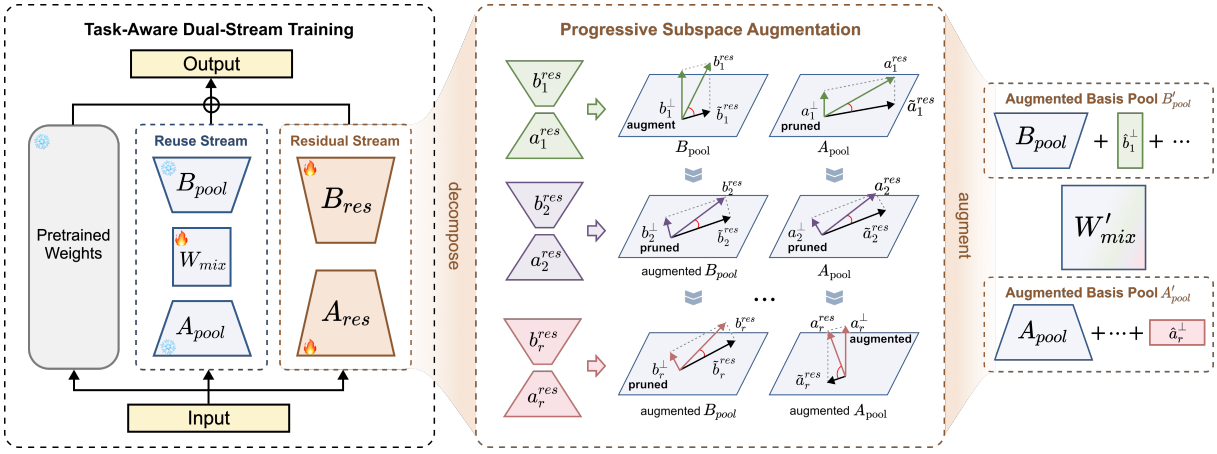


Figure 3: The MoBLoRA framework. Left: Dual-stream training decouples reuse and residual learning. Middle: Residuals are decomposed to selectively augment shared basis pools. Right: New knowledge is consolidated into the updated mixing matrix W'_{mix} with high fidelity.

to explicitly decompose the storage of information into two components: shared Basis Pools that capture the subspace directions, and task-specific Mixing Matrices that determine the reconstruction coefficients. Formally, we maintain two shared pools of orthonormal bases, denoted as $B_{pool} \in \mathbb{R}^{d_{out} \times N_B}$ and $A_{pool} \in \mathbb{R}^{N_A \times d_{in}}$, where N_B and N_A represent the current number of accumulated basis vectors in the respective pools. Under this formulation, the task-specific parameter matrices B_k and A_k are derived as approximate expansions of these shared bases:

$$B_k \approx B_{pool} C_B^k, \quad A_k \approx C_A^k A_{pool}, \tag{1}$$

where $C_B^k \in \mathbb{R}^{N_B \times r}$ and $C_A^k \in \mathbb{R}^{r \times N_A}$ denote the expansion coefficient matrices. This approximation motivates us to directly define the MoBLoRA parameterization in the factorized form:

$$\Delta W_k \triangleq B_{pool} W_{mix}^k A_{pool}, \tag{2}$$

where $W_{mix}^k = C_B^k C_A^k \in \mathbb{R}^{N_B \times N_A}$ is defined as the task-specific Mixing Matrix (Wu et al., 2024). This formulation shifts MCIT from learning isolated per-task experts to learning task-specific compositions over a shared basis set, enabling the model to efficiently approximate diverse task functions within a compact subspace. The overview of the MoBLoRA framework is shown in Fig. 3.

4.2 Task-Aware Dual-Stream Training

To balance stability and plasticity during continual adaptation, we decouple optimization into two streams:

$$\Delta W_k = \underbrace{B_{pool} W_{mix}^k A_{pool}}_{\text{Reuse Stream}} + \underbrace{\frac{\alpha}{r} B_{res} A_{res}}_{\text{Residual Stream}}, \tag{3}$$

where α and r are the scaling factor and the rank for the residual stream, respectively.

In the Reuse Stream, the global basis pools B_{pool} and A_{pool} are strictly frozen to preserve accumulated knowledge. Consequently, adaptation is achieved exclusively by optimizing the task-specific mixing matrix W_{mix}^k, which learns to reconstruct task features via linear combinations of these fixed skill primitives. To accelerate the convergence of this reuse module, we implement a Task-Aware Initialization strategy. Specifically, we identify relevant priors using non-parametric task keys $\mathcal{K}_k = \{\mu_v^k, \mu_t^k\}$, representing the average visual and textual features of task T_k respectively. We first compute the cosine similarity between the keys of the incoming task T_k and the historical ones. These similarity scores are subsequently softmax-normalized into adaptive weights γ_j to synthesize a knowledge-informed warm start (detailed in Appendix A):

$$W_{mix}^k \leftarrow \sum_{j<k} \gamma_j \cdot \text{Pad}(W_{mix}^j). \tag{4}$$

Here, $\text{Pad}(\cdot)$ zero-pads the matrix to the upper-left of $\mathbb{R}^{N_B \times N_A}$ to align with the basis pool capacities.

In parallel, the Residual Stream introduces lightweight, fully trainable low-rank matrices B_{res} and A_{res} to explicitly capture novel semantic components orthogonal to the subspace spanned by the

frozen basis pool. This structural decoupling effectively compartmentalizes novel semantics while leveraging fixed priors, streamlining the subsequent consolidation process.

4.3 Progressive Subspace Augmentation

To efficiently consolidate the residual stream, we first decompose the low-rank matrices into r pairs of rank-1 vectors $\{(b_i^{res}, a_i^{res})\}_{i=1}^{r}$, where $a_i^{res} \in \mathbb{R}^{d_{in}}$ and $b_i^{res} \in \mathbb{R}^{d_{out}}$ denote the i-th row and column vectors respectively. We then quantify the novelty of each component (e.g., input vector a_i^{res}) relative to the current basis A_{pool} via Projection Fidelity:

$$\tilde{a}_i^{res} = a_i^{res} A_{pool}^{\top} A_{pool}, \tag{5}$$

$$S(a_i^{res}) = \mathrm{CosSim}(a_i^{res}, \tilde{a}_i^{res}), \tag{6}$$

where $\tilde{a}_i^{res}$ represents the optimal reconstruction of the rank-1 vector within the current low-dimensional subspace, while S serves as a metric of projection fidelity. To regulate subspace expansion, we employ a fidelity threshold τ and execute this consolidation sequentially for each rank-1 component i from 1 to r. Specifically, components exhibiting low fidelity ($S < \tau$) imply novel semantics unexplained by prior knowledge; these are consequently orthogonalized via Gram-Schmidt and normalized to yield a new basis vector $\hat{a}_i^{\perp} = \mathrm{Normalize}(a_i^{res} - \tilde{a}_i^{res})$. Crucially, this new basis is immediately appended to A_{pool} to update the subspace for subsequent iterations. Conversely, components with high fidelity ($S \geq \tau$) are deemed redundant and explicitly pruned. An identical procedure is symmetrically applied to B_{pool}.

Finally, to achieve high-fidelity consolidation, we analytically absorb the residual weights into the mixing matrix using the augmented subspaces. We derive the coordinate vectors $p_a^i = A'_{pool}\, a_i^{res}$ and $p_b^i = (B'_{pool})^{\top} b_i^{res}$, and update the mixing matrix by explicitly integrating the LoRA scaling factor:

$$W_{mix}^k \leftarrow \mathrm{Pad}(W_{mix}^k) + \frac{\alpha}{r} \sum_{i=1}^{r} p_b^i \, (p_a^i)^{\top}. \tag{7}$$

This transformation restores the model to a unified form $\Delta W = B'_{pool} W_{mix}^k A'_{pool}$ that encodes both historical priors and newly acquired skills. We show in Appendix G that this consolidation is high-fidelity under the augmented basis representation.

4.4 Task-Agnostic Dynamic Routing

To enable inference without explicit task identifiers, we implement a non-parametric dynamic routing mechanism. We repurpose the cached task keys $\{\mathcal{K}_j\}_{j=1}^{k}$ as semantic anchors to retrieve the most relevant historical skills. For an input image V and textual instruction Q, we identify the optimal task index t^* via nearest neighbor matching based on cosine similarity (detailed in Appendix A):

$$t^* = \underset{j \in \{1,\ldots,k\}}{\mathrm{argmax}} \ \mathrm{Sim}(\mathcal{K}_j, V, Q). \tag{8}$$

Subsequently, we exclusively activate the corresponding mixing matrix $W_{mix}^{t^*}$ to reconstruct the effective adaptation weight:

$$\Delta W = B'_{pool} W_{mix}^{t^*} A'_{pool}. \tag{9}$$

Note that as the basis pools expand across tasks, any retrieved historical mixing matrix $W_{mix}^{t^*}$ is zero-padded to match the current dimensions of B'_{pool} and A'_{pool}. Since the appended entries are strictly zero, the newly added basis vectors make no contribution to the reconstructed weight, ensuring dimensional consistency without altering historical task outputs. This strategy minimizes inter-task interference, demonstrating that high-precision skill retrieval can be achieved solely through semantic anchors in the absence of explicit task identifiers.

5 Experiments

5.1 Experimental Setup

Datasets. We evaluate our method on the CoIN benchmark (Chen et al., 2024), which encompasses eight diverse multimodal datasets spanning distinct domains: 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), VQAv2 (Goyal et al., 2017), and OCR-VQA (Mishra et al., 2019). This benchmark covers a comprehensive spectrum of vision-language capabilities, ranging from multiple-choice reasoning and fine-grained classification to visual grounding and open-ended question answering. To ensure rigorous comparability, we strictly adhere to the fixed training sequence established in the original CoIN protocol.

Evaluation Metrics. Following the protocols established in recent works (Wang et al., 2022; Smith et al., 2023; Qiao et al., 2024b, 2025),

Method	Venue	Datasets								Metrics		
		ScienceQA	TextVQA	ImageNet	GQA	VizWiz	Grounding	VQAv2	OCR-VQA	Avg.ACC(↑)	Forgetting(↓)	New.ACC(↑)
Zero-shot	-	49.91	2.88	0.33	2.08	0.90	0.00	0.68	0.17	7.12	-	-
Multi-task	-	56.77	49.35	95.55	56.65	53.90	30.09	59.50	55.65	57.18	-	-
PerTaskFT	-	82.48	62.23	96.55	60.67	60.64	32.24	66.28	61.59	65.41	-	-
LoRA (Hu et al., 2022)	ICLR'22	21.26	28.74	10.25	36.78	32.45	0.83	42.50	57.08	28.74	37.29	61.36
LwF (Li and Hoiem, 2017)	TPAMI'17	63.14	39.60	8.90	34.83	14.53	2.48	40.67	62.35	33.31	22.32	52.58
EWC (Kirkpatrick et al., 2017)	PNAS'17	67.41	40.41	8.18	35.05	37.88	2.67	41.27	61.02	36.74	20.51	54.68
MT (Zhu et al., 2024)	ICML'24	79.63	55.47	35.64	58.70	44.37	32.20	62.21	61.59	53.73	14.03	66.00
PGP (Qiao et al., 2024a)	ICLR'24	**85.17**	56.85	32.26	**61.74**	49.43	32.74	**65.74**	62.20	55.77	12.94	67.09
CIA* (Qiao et al., 2025)	ICML'25	75.63	54.47	43.64	60.70	43.37	36.00	65.21	63.59	55.33	7.04	61.49
SEFE (Chen et al., 2025)	ICML'25	75.35	58.66	83.10	54.25	48.85	16.75	65.35	**66.25**	58.57	11.94	**69.02**
MoELoRA (Chen et al., 2024)	NeurIPS'24	58.92	38.59	8.85	37.10	44.25	2.45	41.40	55.35	35.86	25.71	58.36
AdaLoRA (Zhang et al., 2023)	ICLR'23	73.40	51.29	35.47	44.53	46.75	0.93	55.86	62.03	46.28	23.99	63.27
EProj (He et al., 2023)	-	78.51	57.53	92.35	55.93	44.67	**36.59**	63.74	57.00	60.79	5.42	65.54
BranchLoRA (Zhang et al., 2025)	ACL'25	68.24	40.18	24.60	41.40	49.83	15.94	51.23	62.14	44.20	23.98	65.18
ProgLoRA (Yu et al., 2025)	Findings of ACL'25	74.84	51.83	83.90	49.93	53.87	31.19	62.71	64.44	59.09	7.53	65.68
PCLR (Meng et al., 2026)	ICLR'26	78.33	58.24	86.08	58.14	57.61	33.04	64.17	61.92	62.19	3.39	65.16
MoBLoRA	-	84.96	**60.46**	**96.87**	60.07	**61.70**	32.95	64.82	60.85	**65.34**	**0.00**	65.34

Table 1: The results of performance comparisons between MoBLoRA and baselines on LLaVA-1.5-7B.

we employ three specific metrics to assess the comprehensive performance, stability, and plasticity of the model: Average Accuracy (Avg.ACC) measures the final capacity; Forgetting (FOR) quantifies the performance drop on historical tasks; New Accuracy (New.ACC) reflects plasticity. Let $A_{j,i}$ denote the accuracy on task i after training task j. The metrics are computed as: Average Accuracy $= \frac{1}{K}\sum_{i=1}^{K} A_{K,i}$, Forgetting $= \frac{1}{K-1}\sum_{i=1}^{K-1}(\max_j A_{j,i} - A_{K,i})$, and New Accuracy $= \frac{1}{K}\sum_{i=1}^{K} A_{i,i}$.

Baselines. We compare MoBLoRA against an extensive array of baselines, categorized based on their learning mechanisms. To establish the performance boundaries, we report Zero-shot and Per-TaskFT results as the lower and upper bounds, alongside Multi-Task and Sequential Finetune as references. For regularization-based methods, we implement LwF (Li and Hoiem, 2017), EWC (Kirkpatrick et al., 2017), MT (Zhu et al., 2024), PGP (Qiao et al., 2024a), CIA* (w/o Instruction Grouping) (Qiao et al., 2025), and SEFE (Chen et al., 2025) within the MLLM architecture, tuning parameters to ensure effective results. For parameter isolation-based methods, we compare against two sub-categories: fixed network structures (including MoELoRA (Chen et al., 2024) and AdaLoRA (Zhang et al., 2023)) and dynamic architecture mechanisms (covering Eproj (He et al., 2023), BranchLoRA (Zhang et al., 2025), ProgLoRA (Yu et al., 2025), and PCLR) to highlight the superiority of our approach. Specific implementation details for each method can be found in Appendix C.

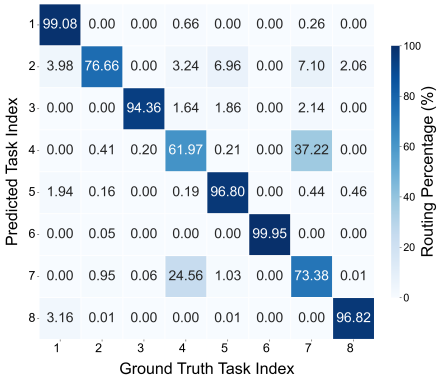


Figure 4: Confusion Matrix of Task-Agnostic Dynamic Routing.

Implementation Details. We adopt LLaVA-1.5-7B (Liu et al., 2023) as our backbone model, which utilizes a pre-trained CLIP-ViT-L/336px (Radford et al., 2021) as the visual encoder. Consistent with previous works, we freeze the vision encoder and the LLM backbone, only adapting the projector and the inserted MoBLoRA modules. Crucially, we adhere to a task-agnostic setting where the task identity is unavailable during inference, requiring the model to automatically retrieve relevant modules based on input semantics. Following LLaVA's LoRA fine-tuning strategy, we embed LoRA modules in all linear layers of the language model with the residual stream's rank set to 8 and the scaling factor set to 16. The threshold τ for projection fidelity is empirically set to 0.7. We set the training epoch for all tasks to 1 and the warm-up ratio to 0.03. The learning rates for LoRA and the projector are set to 2e-4 and 2e-5, respectively, with a cosine decay schedule. The batch size of all tasks is 128. All experiments are conducted on 8 3090 GPUs.

Method	Avg.ACC(↑)	Forgetting(↓)	New.ACC(↑)
Single LoRA	28.74	37.29	61.36
Isolated LoRA	63.21	1.19	64.25
Average Init	64.15	0.29	64.40
Kaiming Uniform	64.31	0.90	65.10
MoBLoRA	**65.34**	**0.00**	**65.34**

Table 2: Ablation studies of Dual-Stream architecture conducted on the CoIN benchmark.

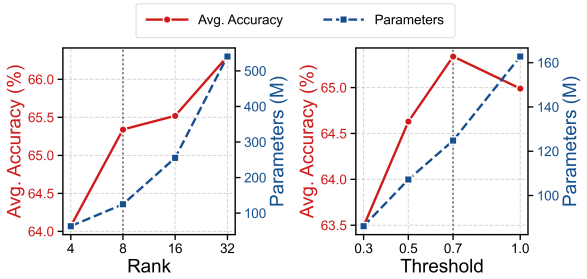


Figure 5: Parameter sensitivity analysis. Left: Impact of the rank r. Right: Influence of the projection fidelity threshold τ.

5.2 Main Results

We evaluate MoBLoRA on the CoIN benchmark, comparing it against state-of-the-art baselines in Table 1. Zero-shot (7.12% Avg.ACC) and PerTaskFT (65.41%) establish the lower and upper performance boundaries, respectively. Among standard baselines, Multi-Task learning reaches 57.18%, whereas sequential LoRA fine-tuning exhibits severe catastrophic forgetting (37.29%), yielding a low Avg.ACC of 28.74%.

Regularization-based (e.g., EWC, LwF) and fixed-structure methods (e.g., MoELoRA) suffer from significant forgetting ($>$ 7%). Even the competitive PCLR (62.19%) falls short of the theoretical limit. In contrast, MoBLoRA significantly outperforms all competing approaches. It achieves a new state-of-the-art Avg.ACC of 65.34%, with a substantial gain of 3.15% over PCLR and matches the performance of the PerTaskFT upper bound (65.41%), while maintaining compelling plasticity on new tasks.

Most notably, MoBLoRA achieves a Forgetting rate of 0.00%, demonstrating near-zero catastrophic forgetting on the CoIN benchmark. We attribute this stability in part to our task-aware initialization strategy. As visualized in the routing confusion matrix (Fig. 4), while the router exhibits high precision for distinct tasks (e.g., 99.08% for Task 1), it occasionally directs semantically related tasks to historical mixing matrices (e.g., 37.22% of Task 4 routed to Task 7). Crucially, since these experts are initialized via historical similarity, such routing leverages shared knowledge rather than causing interference. This mechanism even fosters positive backward transfer, as exemplified by improved accuracy on TextVQA after subsequent training, validating that our approach effectively converts potential ambiguity into constructive reinforcement (detailed in Appendix I).

To further investigate the marginal gap between MoBLoRA (65.34%) and the PerTaskFT upper bound (65.41%), we evaluate an oracle variant in which ground-truth task identities are explicitly provided during inference, bypassing the task-agnostic dynamic routing module. This oracle MoBLoRA achieves an Avg.ACC of 65.91%, exceeding the PerTaskFT upper bound by 0.50%. This result demonstrates that the representation capacity of our shared basis pools is sufficient to surpass the upper bound, and that the marginal gap observed in the task-agnostic setting stems from routing ambiguities rather than any limitation in the model's expressive power.

5.3 Ablation Study

Impact of Dual-Stream Architecture. To validate the necessity of our dual-stream design, we compare MoBLoRA with two variants in Table 2. First, Isolated LoRA represents the absence of the Reuse Stream, instantiating independent parameters for each task. While it exhibits competitive plasticity (New.ACC 64.25%), its failure to leverage shared knowledge results in suboptimal overall performance (Avg.ACC 63.21%) and higher parameter costs (159.91M) compared to MoBLoRA. Second, Single LoRA eliminates the Residual Stream, effectively degenerating into sequential fine-tuning. This approach succumbs to severe catastrophic forgetting (37.29% Forgetting rate), yielding the lowest Avg.ACC of 28.74%. In contrast, MoBLoRA synergizes both streams to decouple knowledge reuse from new acquisition, achieving the highest Avg.ACC (65.34%) with zero forgetting.

Impact of Task-Aware Initialization. We further investigate the efficacy of Task-Aware Initialization for the mixing matrix. Unlike agnostic approaches, our strategy exploits semantic similarity to historical subspaces to establish an informative warm start. As shown in Table 2, we benchmark this against Average Init (mean of his-

Method	Avg.ACC(↑)	Forgetting(↓)	New.ACC(↑)
LoRA†	35.68	32.90	64.47
MT†	57.47	11.26	67.32
PGP†	59.13	10.11	67.98
EProj†	61.42	5.84	65.53
CIA†	65.10	2.31	67.12
PCLR†	65.51	2.08	67.32
MoBLoRA	**68.98**	**0.02**	**69.00**

Table 3: Performance comparison on LLaVA-1.5-13B. Results marked with † are reported directly from the PCLR (Meng et al., 2026).

Order	Method	Avg.ACC(↑)	Forgetting(↓)	New.ACC(↑)
Reverse	PCLR†	62.18	3.03	**64.83**
	MoBLoRA	**64.26**	**0.04**	64.29
Alphabet	MoELoRA‡	36.32	29.65	62.26
	ProgLoRA‡	49.52	7.96	56.48
	PCLR†	60.62	4.63	**64.67**
	MoBLoRA	**64.33**	**0.29**	64.59

Table 4: Performance comparison under different task orders. Results marked with † and ‡ are reported directly from the PCLR (Meng et al., 2026) and ProgLoRA (Yu et al., 2025), respectively.

torical weights) and Kaiming Uniform (standard random initialization). Both baselines yield suboptimal results, with Average Init and Kaiming Uniform achieving 64.15% and 64.31% Avg.ACC, respectively, compared to MoBLoRA. Moreover, these variants suffer from increased forgetting rates, thereby validating the positive backward transfer facilitated by our specific initialization (as discussed in Section 5.2).

Parameter Sensitivity. We investigate the sensitivity of two pivotal hyper-parameters: the rank r and the projection fidelity threshold τ. First, regarding the rank r, which governs the capacity of the residual stream, Fig. 5 (Left) indicates that while larger ranks monotonically improve accuracy, they impose a substantial parameter burden. We adopt $r = 8$ as the optimal trade-off, securing significant performance gains with acceptable model growth. Second, we analyze the fidelity threshold τ to validate the efficacy of Progressive Subspace Augmentation. As shown in Fig. 5 (Right), performance is highly sensitive to this parameter. Stringent thresholds (e.g., $\tau = 0.3$) hinder plasticity by overly restricting expansion, whereas a threshold of 1.0 maximizes parameter cost but degrades accuracy, potentially due to the accumulation of redundant noise. Crucially, $\tau = 0.7$ attains peak accuracy with significantly reduced parameters compared to the $\tau = 1.0$ baseline, confirming that our mechanism successfully filters redundancy while preserving essential semantic knowledge. We further verify the robustness of $\tau = 0.7$ across different task orders in Appendix H.

5.4 Further Analysis

Scalability to Larger Backbones. To verify whether our conclusions generalize beyond the 7B scale, we evaluate MoBLoRA on the LLaVA-1.5-13B backbone using the CoIN benchmark. As shown in Table 3, MoBLoRA consistently achieves state-of-the-art performance, yielding the highest Avg.ACC (68.98%) and New.ACC (69.00%) while maintaining the lowest forgetting rate (0.02%). These results confirm that the orthogonal subspace mixing mechanism scales effectively with increased model capacity, and that the near-zero forgetting property of MoBLoRA is not specific to the 7B architecture.

Robustness Across Task Orders. To verify whether MoBLoRA remains effective under different task curricula, we evaluate it on two additional CoIN sequences beyond the default order: Reverse and Alphabetical. As shown in Table 4, MoBLoRA consistently achieves the lowest forgetting rate and highest Avg.ACC across both sequences. Notably, in the Reverse order, MoBLoRA achieves the highest Avg.ACC (64.26%) with a near-zero forgetting rate (0.04%), outperforming PCLR by a substantial margin. In the Alphabetical order, MoBLoRA achieves the highest Avg.ACC (64.33%) with a near-zero forgetting rate (0.29%), outperforming all baselines by a substantial margin. These results demonstrate that the stability of MoBLoRA does not depend on any specific task ordering, and that its subspace mixing mechanism converts potential inter-task interference into constructive knowledge sharing regardless of the task curriculum.

Visualization of Basis Pool and Mixing Matrix. Consistent with our previous analysis in Section 3, we utilize the B_{pool} matrix from the query projection in the first layer as a representative example. Fig. 6 visualizes the learned subspace structures to analyze the internal mechanism of MoBLoRA. Fig. 6 (Left) displays the cosine similarity matrix of the final Basis Pool. The clear diagonal pattern is consistent with the orthogonality enforced by our

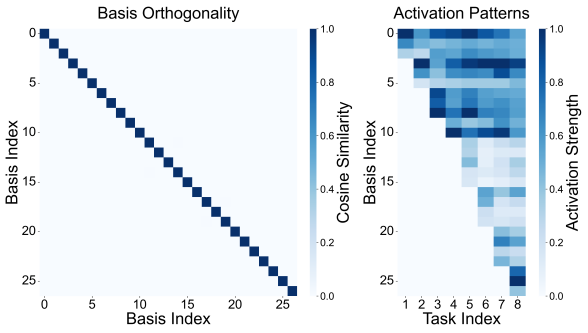


Figure 6: Visualization of the learned Basis Pool and Mixing Matrix. Left: Pairwise cosine similarity matrix of the Basis Pool. Right: Heatmap of basis activation strengths.

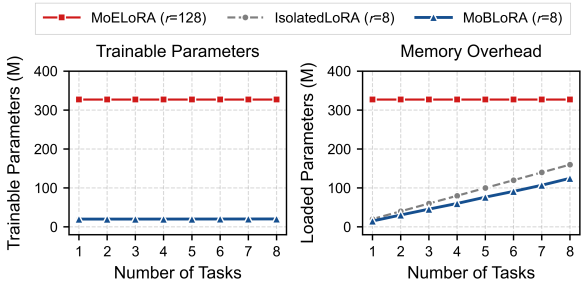


Figure 7: Efficiency comparison on the CoIN benchmark. Left: Comparison of trainable parameters. Right: Comparison of the number of parameters loaded during inference.

Progressive Subspace Augmentation. Fig. 6 (Right) shows the basis activation strengths derived from the Mixing Matrix, revealing two key patterns. First, the early basis indices (e.g., 0-3) remain highly active across subsequent tasks, demonstrating the effective reuse of historical knowledge. Second, the active region expands step-by-step as tasks increase. This verifies that our method detects new semantics and adds bases only when necessary, preventing unnecessary parameter growth.

Efficiency Analysis. We conduct a comparative analysis of parameter efficiency across training and inference phases, as illustrated in Fig. 7. First, regarding computational training costs (Fig. 7 Left), MoELoRA incurs a substantial overhead with 327.16M trainable parameters. In contrast, MoBLoRA maintains a consistently low parameter overhead (≈20M) with only marginal growth, thereby preserving the training efficiency of standard fine-tuning. Second, regarding inference memory overhead (Fig. 7 Right), MoELoRA imposes a fixed memory overhead (327.16M) due to the pre-allocation of fixed experts. Although Isolated LoRA avoids static pre-allocation, it suffers from steep linear expansion by simply concatenating independent parameters, reaching 159.91M by Task 8. MoBLoRA significantly mitigates this growth by reusing the shared Basis Pool and appending only essential orthogonal residuals, yielding a reduced memory overhead of 124.83M at the same stage. It is important to note that the current linear trend in MoBLoRA represents only the initial phase of basis accumulation. As visualized in Fig. 6, our basis pool currently contains only 26 orthonormal bases after 8 tasks, far below the intrinsic dimensionality required to span the full task space. Consequently, the system is still actively expanding its basis set to cover new semantics. Theoretically, as the number of tasks increases and the basis pool approaches completeness, the need for new bases will diminish. Thus, we hypothesize that the parameter efficiency of MoBLoRA will become increasingly pronounced as the number of tasks grows, though empirical validation on longer task sequences remains for future work. Furthermore, the post-training subspace augmentation procedure introduces negligible computational overhead, as pool expansion and residual handling are executed strictly once at the end of each task's training phase. Empirically, this process requires only 4.87s, 4.21s, and 5.29s for Tasks 1, 5, and 8, respectively, confirming an asymptotic cost of $\mathcal{O}(1)$ that never bottlenecks the continual learning pipeline.

6 Conclusion

In this paper, we introduce MoBLoRA, a novel framework that aims to address the stability-plasticity dilemma in MCIT. Motivated by the geometric observation of subspace redundancy across tasks, MoBLoRA shifts the paradigm from discrete expert selection to orthogonal subspace mixing. By decomposing adaptation weights into shared basis pool and task-specific mixing matrices, our approach effectively decouples the accumulation of general knowledge from task-specific reconstruction. Extensive experiments on the CoIN benchmark demonstrate that MoBLoRA achieves state-of-the-art performance, effectively mitigating catastrophic forgetting while maintaining superior parameter efficiency through progressive subspace augmentation. We hope this subspace-centric perspective offers new insights for developing scalable and sustainable continual learning systems.

Limitations

Despite the promising results of MoBLoRA in mitigating catastrophic forgetting and enhancing parameter efficiency, several limitations remain to be addressed.

First, while our experiments across LLaVA-1.5-7B and LLaVA-1.5-13B demonstrate that MoBLoRA scales effectively with increased model capacity, our evaluation remains confined to the LLaVA architecture. The applicability of our orthogonal subspace mixing mechanism to heterogeneous MLLM architectures (e.g., Qwen-VL) remains to be verified, and we leave a comprehensive cross-architecture evaluation to future work.

Second, although the proposed mechanism effectively handles the sequence length of the current CoIN benchmark (8 tasks), its behavior under more demanding settings remains to be investigated. Under extremely long task sequences, the basis pool may face saturation or aggressive expansion depending on the degree of shared structure across tasks. Furthermore, significant distribution shifts at inference time may impair the routing mechanism, as semantic keys derived from training data may not accurately reflect shifted test inputs. We leave these directions to future work.

Additionally, the projection fidelity threshold τ is fixed (0.7) and selected via ablation. While reasonably robust, a fixed threshold may be suboptimal in open-world settings, as it cannot reflect varying task novelty across domains. A natural extension is to adapt τ based on cross-task semantic similarity derived from the task keys $\mathcal{K}$. For a new task k, we compute $S_{max} = \max_{j<k} \text{Sim}(\mathcal{K}_k, \mathcal{K}j)$ as an estimate for its similarity to prior tasks. A higher $Smax$ suggests using a stricter threshold to avoid redundancy, while a lower value calls for a more permissive threshold to improve plasticity. This can be implemented via a linear interpolation: $\tau_k = \tau_{min} + (\tau_{max} - \tau_{min}) \cdot S_{max}$. We leave empirical validation of this adaptive strategy to future work.

Finally, we address potential risks stemming from the enhanced stability and efficiency of MoBLoRA. From a fairness perspective, the framework's robust retention capabilities may inadvertently entrench societal biases within the shared basis pool, thereby complicating the "unlearning" of toxic content. This resistance to forgetting also raises privacy concerns; unlike methods that naturally decay historical information, MoBLoRA's shared subspace risks stubbornly preserving sensitive data points from early tasks. Consequently, the model may become more susceptible to membership inference attacks and training data extraction.

Acknowledgments

This research was supported by the National Natural Science Foundation of China (No.62502436) and the Zhejiang Provincial Natural Science Foundation of China under Grant No.LMS26F020004.

References

Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, Roman Ring, Eliza Rutherford, Serkan Cabi, Tengda Han, Zhitao Gong, Sina Samangooei, Marianne Monteiro, Jacob L. Menick, Sebastian Borgeaud, and 8 others. 2022. Flamingo: a visual language model for few-shot learning. *Advances in Neural Information Processing Systems*, 35:23716–23736.

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*, 1(2):3.

Cheng Chen, Junchen Zhu, Xu Luo, Heng T Shen, Jingkuan Song, and Lianli Gao. 2024. Coin: A benchmark of continual instruction tuning for multimodel large language models. *Advances in Neural Information Processing Systems*, 37:57817–57840.

Jinpeng Chen, Runmin Cong, Yuzhi Zhao, Hongzheng Yang, Guangneng Hu, Horace Ho Shing Ip, and Sam Kwong. 2025. Sefe: Superficial and essential forgetting eliminator for multimodal continual instruction tuning. In *ICML*.

Wenliang Dai, Junnan Li, Dongxu Li, Anthony Tiong, Junqi Zhao, Weisheng Wang, Boyang Li, Pascale N Fung, and Steven Hoi. 2023. Instructblip: Towards general-purpose vision-language models with instruction tuning. *Advances in Neural Information Processing Systems*, 36:49250–49267.

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.

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 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, yelong shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. Lora: Low-rank adaptation of large language models. *ICLR*, 1(2):3.

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.

Aaron Hurst, Adam Lerer, Adam P. Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, Aleksander Madry, Alex Baker-Whitcomb, Alex Beutel, Alex Borzunov, Alex Carney, Alex Chow, Alex Kirillov, Alex Nichol, Alex Paino, and 79 others. 2024. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*.

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 C. Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A. Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, Demis Hassabis, Claudia Clopath, Dharshan Kumaran, and Raia Hadsell. 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 Hoi. 2023. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *International conference on machine learning*, pages 19730–19742. PMLR.

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

Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. 2024. Improved baselines with visual instruction tuning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 26296–26306.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. 2023. Visual instruction tuning. *Advances in Neural Information Processing Systems*, 36:34892–34916.

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.

Weicheng Meng, Jingyang Qiao, Zhizhong Zhang, Shaohui Liu, and Yuan Xie. 2026. Pclr: Progressively compressed lora for multimodal continual instruction tuning. In *The Fourteenth International Conference on Learning Representations*.

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.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul F. Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35:27730–27744.

Jingyang Qiao, Zhizhong Zhang, Xin Tan, Chengwei Chen, Yanyun Qu, Yong Peng, and Yuan Xie. 2024a. Prompt gradient projection for continual learning. In *The Twelfth International Conference on Learning Representations*.

Jingyang Qiao, Zhizhong Zhang, Xin Tan, Yanyun Qu, Shouhong Ding, and Yuan Xie. 2025. Large continual instruction assistant. In *International conference on machine learning*.

Jingyang Qiao, Zhizhong Zhang, Xin Tan, Yanyun Qu, Wensheng Zhang, and Yuan Xie. 2024b. Gradient projection for parameter-efficient continual learning. *arXiv e-prints*, pages arXiv–2405.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. 2021. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR.

Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. 2017. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. *arXiv preprint arXiv:1701.06538*.

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.

James Seale Smith, Leonid Karlinsky, Vyshnavi Gutta, Paola Cascante-Bonilla, Donghyun Kim, Assaf Arbelle, Rameswar Panda, Rogerio Feris, and Zsolt Kira. 2023. Coda-prompt: Continual decomposed attention-based prompting for rehearsal-free continual learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11909–11919.

Gemini Team. 2025. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *arXiv preprint arXiv:2507.06261*.

Liyuan Wang, Xingxing Zhang, Hang Su, and Jun Zhu. 2024. A comprehensive survey of continual learning: Theory, method and application. *IEEE transactions on pattern analysis and machine intelligence*, 46(8):5362–5383.

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

Taiqiang Wu, Jiahao Wang, Zhe Zhao, and Ngai Wong. 2024. Mixture-of-subspaces in low-rank adaptation. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 7880–7899.

Lingling Xu, Haoran Xie, Si-Zhao Joe Qin, Xiaohui Tao, and Fu Lee Wang. 2023. Parameter-efficient fine-tuning methods for pretrained language models: A critical review and assessment. *arXiv preprint arXiv:2312.12148*.

Shukang Yin, Chaoyou Fu, Sirui Zhao, Ke Li, Xing Sun, Tong Xu, and Enhong Chen. 2024. A survey on multimodal large language models. *National Science Review*, 11(12):nwae403.

Yahan Yu, Duzhen Zhang, Yong Ren, Xuanle Zhao, Xiuyi Chen, and Chenhui Chu. 2025. Progressive lora for multimodal continual instruction tuning. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 2779–2796.

Yuexiang Zhai, Shengbang Tong, Xiao Li, Mu Cai, Qing Qu, Yong Jae Lee, and Yi Ma. 2023. Investigating the catastrophic forgetting in multimodal large language models. *arXiv preprint arXiv:2309.10313*.

Duzhen Zhang, Yong Ren, Zhong-Zhi Li, Yahan Yu, Jiahua Dong, Chenxing Li, Zhilong Ji, and Jinfeng Bai. 2025. Enhancing multimodal continual instruction tuning with branchlora. *arXiv preprint arXiv:2506.02041*.

Qingru Zhang, Minshuo Chen, Alexander Bukharin, Nikos Karampatziakis, Pengcheng He, Yu Cheng, Weizhu Chen, and Tuo Zhao. 2023. Adalora: Adaptive budget allocation for parameter-efficient fine-tuning. *arXiv preprint arXiv:2303.10512*.

Didi Zhu, Zhongyi Sun, Zexi Li, Tao Shen, Ke Yan, Shouhong Ding, Kun Kuang, and Chao Wu. 2024. Model tailor: Mitigating catastrophic forgetting in multi-modal large language models. *arXiv preprint arXiv:2402.12048*.

Yue Zhu, Haiwen Diao, Shang Gao, Jiazuo Yu, Jiawen Zhu, Yunzhi Zhuge, Shuai Hao, Xu Jia, Lu Zhang, Ying Zhang, and Huchuan Lu. 2025. Regularizing subspace redundancy of low-rank adaptation. In *Proceedings of the 33rd ACM International Conference on Multimedia*, pages 1666–1675.

A Definition of Task Keys

To enable task-aware initialization and identifier-free inference, we introduce the Task Key, a semantic prototype derived from pre-trained encoders (f_v, f_t). For the k-th task, we randomly sample N instances and compute modality-specific centroids:

$$\begin{aligned} \mu_v^k &= \frac{1}{N}\sum_{i=1}^{N} f_v(V_i), \\ \mu_t^k &= \frac{1}{N}\sum_{i=1}^{N} f_t(Q_i), \end{aligned} \tag{10}$$

where V_i and Q_i denote the input image and instruction. The Task Key is formally defined as $\mathcal{K}_k = \{\mu_v^k, \mu_t^k\}$.

Task-Aware Initialization. To transfer knowledge from historical priors $\{\mathcal{K}_j\}_{j=1}^{k-1}$ to the current task, we quantify their correlation as a weighted cosine similarity between task-level prototypes:

$$\bar{r}^j = \lambda_v \cdot \mathrm{CS}(\mu_v^k, \mu_v^j) + \lambda_t \cdot \mathrm{CS}(\mu_t^k, \mu_t^j), \tag{11}$$

where $\mathrm{CS}(\cdot,\cdot)$ denotes cosine similarity. We normalize these scores via a Softmax function with temperature T to obtain mixing coefficients γ_j, which guide the initialization of the new mixing matrix (see Section 4.2).

Task-Agnostic Dynamic Routing. During inference, no task prototype is available for the incoming sample. Instead, we derive a sample-level query by passing the input image V and instruction Q through the same encoders, and compute its similarity to each historical task key:

$$\begin{aligned} \mathrm{Sim}(\mathcal{K}_j, V, Q) = \lambda_v \cdot \mathrm{CS}(f_v(V), \mu_v^j) \\ + \lambda_t \cdot \mathrm{CS}(f_t(Q), \mu_t^j), \end{aligned} \tag{12}$$

where $\mathrm{CS}(\cdot,\cdot)$ denotes cosine similarity, λ_v and λ_t are the same modality weights used in the initialization stage. The task index with the highest similarity score is selected to retrieve the corresponding mixing matrix (see Section 4.4).

Task	Dataset	Instruction	Train Number	Test Number
Grounding	RefCOCO RefCOCO+ RefCOCOg	Please provide the bounding box coordinate of the region this sentence describes: <description>	55k	31k
Classification	ImageNet	What is the object in the image? Answer the question using a single word or phrase	129k	5k
Image Question Answering (IQA)	VQAv2	Answer the question using a single word or phrase	82k	107k
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
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
OCR IQA	OCR-VQA	Answer the question using a single word or phrase	165k	100k

Table 5: The statistics of collected datasets and instructions in the CoIN benchmark.

B Dataset

The detailed statistics for the eight multimodal datasets included in the CoIN benchmark (Chen et al., 2024) are presented in Table 5. The CoIN benchmark is distributed under the Creative Commons Attribution 4.0 International (CC-BY 4.0) license. We strictly adhere to the terms of this license, utilizing the data for continual fine-tuning tasks, which aligns with the benchmark's original intent. In response to ethical considerations regarding data collection, we emphasize that all data used in our experiments are sourced from well-established, publicly available academic datasets. These datasets have been widely vetted by the research community and do not contain any information that identifies individual people or any offensive content.

C Details of the comparison method

In this section, we outline the principles of the baseline methods used in our experiments:

LoRA introduces low-rank decomposition matrices into the weight matrices of pretrained models to achieve parameter-efficient fine-tuning. By freezing the original model weights and optimizing only rank-decomposition matrices, it significantly reduces the trainable parameter count while preserving generalization capabilities.

Learning Without Forgetting (LwF) mitigates catastrophic forgetting via knowledge distillation. It utilizes the pretrained model to guide the current model, enforcing alignment between their output distributions through a weighted sum of prediction and distillation losses, thereby retaining historical knowledge without accessing old datasets.

Elastic Weight Consolidation (EWC) protects critical parameters from previous tasks by introducing a regularization term based on the Fisher Information Matrix (FIM). It computes the importance of each parameter (diagonal elements of FIM) and penalizes significant changes to these weights during new task training.

Model Tailor (MT) adopts a parameter-efficient strategy that restricts training to a subset of critical parameters while compensating for variations in the trainable weights to balance stability and plasticity.

Prompt Gradient Projection (PGP) employs a gradient projection approach to preserve historical knowledge. It enforces model parameter updates to be orthogonal to the feature subspaces of previous tasks, thereby minimizing interference while enabling adaptation to new data.

Dynamic EMA (CIA*) derives optimal balance weights based on a stability-plasticity tradeoff premise and Exponential Moving Average (EMA) updates. The weights are adaptively determined by gradients and learned parameters to satisfy continual learning conditions.

SEFE mitigates superficial forgetting via Answer Style Diversification (ASD) to unify data formats, and addresses essential forgetting via RegLoRA by regularizing critical elements in historical weight update matrices.

MoELoRA integrates the Mixture-of-Experts (MoE) mechanism with LoRA to enhance adaptability in dynamic environments. It transforms a single LoRA layer into multiple experts and trains a router to dynamically allocate expert resources for specific modalities or tasks. In our setting, we configure it with a static structure of 2 experts per layer.

AdaLoRA adaptively allocates parameter budgets based on importance. It uses SVD to parameterize

updates and prunes less critical singular values to concentrate resources on key components.

Eproj utilizes a dynamic model adaptation strategy involving task grouping. It explicitly identifies and groups high-conflict tasks for separate handling while managing low-conflict tasks via regularization to optimize the continual learning process.

BranchLoRA adopts an asymmetric architecture with a shared input matrix and multiple experts for efficiency. It mitigates forgetting via a flexible tuning-freezing mechanism and employs task-specific routers with an automatic selector to ensure accurate parameter allocation.

ProgLoRA dynamically expands model capacity by instantiating a new LoRA block for each incoming task to minimize interference. It employs task-aware allocation to selectively leverage relevant historical knowledge and utilizes task recall to realign the model with previously learned distributions.

PCLR introduces a LoRA Rank Pool (LRP) and a Compression-Integration-Learning (CIL) pipeline. It decomposes weights to enable fine-grained rank control, then balances plasticity and stability by pruning rank experts (Compression), fusing similar experts via distillation (Integration), and training new experts in the released space (Learning).

The baseline results reported in Table 1 are cited from BranchLoRA (Zhang et al., 2025) and PCLR. To ensure a fair comparison, we strictly adhere to the same experimental settings for MoBLoRA.

D Details of evaluation

Following the protocols in CoIN (Chen et al., 2024), we evaluate Image Question Answering tasks by calculating the accuracy of predicted answers against the ground truth. For classification tasks, the metric relies on the accuracy of predicted labels compared to the ground truth. For referring expression comprehension, we adopt the standard Intersection over Union (IoU) metric, considering a prediction correct if the IoU between the predicted and ground-truth bounding boxes exceeds 0.5. The specific prompts utilized for evaluation across these tasks are presented in Table 5.

E Visualization of CoIN benchmark

To demonstrate the effectiveness of our method across the diverse tasks of the CoIN benchmark, we provide visualization examples in Table 10. We select representative samples from ScienceQA, TextVQA, ImageNet, GQA, VizWiz, Grounding, VQAV2, and OCR-VQA to qualitatively illustrate our model's robust performance in each domain.

F Information About Use Of AI Assistants

During the preparation of this work, we used Gemini 3 in order to improve the language and readability of the manuscript. After using this tool, we reviewed and edited the content as needed and take full responsibility for the content of the publication.

G High-Fidelity Consolidation

We formally establish that the Progressive Subspace Augmentation procedure in Section 4.3 consolidates the residual stream into the shared basis pools with high-fidelity approximation, where the fidelity is explicitly controlled by the projection fidelity threshold τ.

Let $B'_{pool} \in \mathbb{R}^{d_{out} \times N'_B}$ and $A'_{pool} \in \mathbb{R}^{N'_A \times d_{in}}$ denote the augmented basis pools after consolidation in Equation (7), where N'_B and N'_A are the updated pool sizes. The original residual stream $\frac{\alpha}{r} B_{res} A_{res}$ is decomposed into r rank-1 components, where the i-th component is $\Delta W^i = \frac{\alpha}{r} b_i^{res} (a_i^{res})^\top$, with $b_i^{res} \in \mathbb{R}^{d_{out}}$ and $a_i^{res} \in \mathbb{R}^{d_{in}}$ denoting the i-th column and row vectors of B_{res} and A_{res}, respectively. The coordinate vectors $p_b^i = (B'_{pool})^\top b_i^{res}$ and $p_a^i = A'_{pool}\, a_i^{res}$ are the projections of the residual vectors onto the augmented pools.

By linearity, it suffices to analyze each rank-1 component individually. For the i-th component, the consolidation step yields:

$$\begin{aligned} \Delta W_{rec}^i &= B'_{pool} \left(\frac{\alpha}{r} p_b^i (p_a^i)^\top \right) A'_{pool} \\ &= \frac{\alpha}{r} \Pi_B b_i^{res} \cdot (a_i^{res})^\top \Pi_A \\ &= \frac{\alpha}{r} (\Pi_B b_i^{res})(\Pi_A a_i^{res})^\top, \end{aligned} \tag{13}$$

where $\Pi_B = B'_{pool}(B'_{pool})^\top \in \mathbb{R}^{d_{out} \times d_{out}}$ and $\Pi_A = (A'_{pool})^\top A'_{pool} \in \mathbb{R}^{d_{in} \times d_{in}}$ denote the projection matrices onto the column space of B'_{pool} and the row space of A'_{pool}, respectively. Thus, the mixing matrix obtained from Equation (7) exactly reconstructs the projections of b_i^{res} and a_i^{res} onto the augmented pools.

According to the Progressive Subspace Augmentation procedure, any component with $S(a_i^{res}) < \tau$ is orthogonalized and appended to the pool, ensuring $\Pi_A a_i^{res} = a_i^{res}$ exactly. For retained

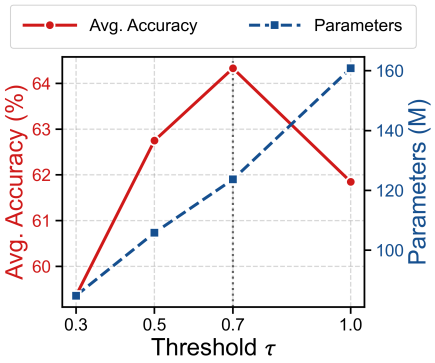


Figure 8: Influence of the projection fidelity threshold τ on Avg.ACC and parameter overhead under the alphabetical task order.

components with $S(a_i^{res}) \geq \tau$, the fidelity condition implies $\|\hat{a}_i^{\perp}\| \leq \sqrt{1-\tau^2}\,\|a_i^{res}\|$, where $\hat{a}_i^{\perp} = a_i^{res} - \Pi_A a_i^{res}$, and symmetrically for $\hat{b}_i^{\perp} = b_i^{res} - \Pi_B b_i^{res}$. Therefore, ΔW_{rec}^i constitutes a high-fidelity approximation of ΔW^i, with the consolidation error vanishing as $\tau \to 1$. Summing over all r components and incorporating the frozen reuse stream, the complete dual-stream update is recovered with high fidelity:

$$B'_{pool} {W^k_{mix}}' A'_{pool} \approx B_{pool} W^k_{mix} A_{pool} + \frac{\alpha}{r} B_{res} A_{res}. \quad (14)$$

This establishes that MoBLoRA achieves high-fidelity weight consolidation controlled by τ, providing a theoretical guarantee of knowledge retention consistent with the near-zero catastrophic forgetting observed.

H Parameter Sensitivity Across Task Orders

We further investigate whether the default projection fidelity threshold $\tau = 0.7$ remains robust when the task order is permuted. Specifically, we evaluate MoBLoRA under $\tau \in \{0.3, 0.5, 0.7, 1.0\}$ on the Alphabetical sequence of the CoIN benchmark, and report the results in Fig. 8.

As shown, the trend is consistent with the default order analysis in Section 5.3: performance is sensitive to τ, and $\tau = 0.7$ again achieves the best Avg.ACC (64.33%) with a reasonable parameter overhead (123.68M). A stringent threshold ($\tau = 0.3$) over-restricts subspace expansion, limiting plasticity and yielding the lowest Avg.ACC (59.36%). Conversely, $\tau = 1.0$ maximizes parameter growth (160.81M) but degrades accuracy (61.85%), likely due to the accumulation of redundant bases. These results confirm that $\tau = 0.7$ remains the optimal trade-off under a different task curriculum, demonstrating the robustness of our threshold selection across task orders.

I Supplementary Results of Continual Instruction Tuning

Due to space constraints, the tables and figures in the main text report only the final performance metrics obtained after the completion of the entire training sequence. In this section, we provide comprehensive experimental results to facilitate a more granular analysis of the learning dynamics. For each entry in the following tables, we adopt a dual-row format to explicitly demonstrate the trade-off between learning and retention:

- The **first row** reports the accuracy for task i evaluated immediately after tuning on that specific task (denoted as $A_{i,i}$), reflecting the model's initial learning capability.
- The **second row** reports the accuracy for task i after the model has finished fine-tuning on the final task K (denoted as $A_{K,i}$), reflecting the model's ability to retain knowledge over time.

Table 6 presents detailed performance comparisons on LLaVA-1.5-7B under the default task order, covering ablations on the dual-stream architecture (Isolated LoRA), task-aware initialization strategies (Average Init, Kaiming Uniform), residual stream rank ($r \in \{4, 16, 32\}$), and projection fidelity threshold ($\tau \in \{0.3, 0.5, 1.0\}$). Table 7 reports the detailed per-task results of MoBLoRA on the larger LLaVA-1.5-13B backbone, confirming that the near-zero forgetting property generalizes beyond the 7B scale. Table 8 and Table 9 provide comprehensive results under alternative task orderings: the Reverse sequence and the Alphabetical sequence, respectively. For the Alphabetical order, we additionally report results across varying τ values to verify the robustness of our threshold selection under a different task curriculum.

Method	Datasets								Metrics		
	ScienceQA	TextVQA	ImageNet	GQA	VizWiz	Grounding	VQAv2	OCR-VQA	Avg.ACC(↑)	Forgetting(↓)	New.ACC(↑)
MoBLoRA (Ours)	84.96	60.16	97.05	60.14	61.77	32.95	64.82	60.85	65.34	0.00	65.34
	84.96	60.46	96.87	60.07	61.70	32.95	64.82	60.85			
Isolated LoRA	84.70	60.56	96.91	60.45	61.38	28.93	64.27	56.78	63.21	1.19	64.25
	84.58	59.42	93.35	57.08	61.24	28.93	64.27	56.78			
Average Init	84.63	60.66	97.01	60.30	61.94	32.56	64.82	53.30	64.15	0.29	64.40
	84.60	60.63	96.67	58.73	61.89	32.56	64.82	53.30			
Kaiming Uniform	85.00	59.84	97.03	59.95	60.75	32.42	64.30	61.49	64.31	0.90	65.10
	84.79	59.02	95.92	55.86	60.66	32.42	64.30	61.49			
rank = 4	84.34	60.58	96.85	59.84	61.66	28.20	64.84	56.58	64.07	0.05	64.11
	84.34	60.50	96.65	59.80	61.63	28.20	64.84	56.58			
rank = 16	86.11	60.62	96.69	59.86	61.91	37.60	64.82	56.64	65.52	0.01	65.53
	86.11	60.66	96.55	59.97	61.82	37.60	64.82	56.64			
rank = 32	85.88	60.71	96.57	60.45	61.43	41.28	64.43	59.73	66.30	0.01	66.31
	85.88	60.78	96.53	60.48	61.31	41.28	64.43	59.73			
$\tau = 1$	85.76	60.47	97.03	60.14	61.77	35.41	64.64	54.61	64.99	-0.01	64.98
	85.76	60.79	96.87	60.09	61.75	35.41	64.64	54.61			
$\tau = 0.5$	83.73	60.41	97.05	59.72	61.50	30.04	64.88	60.10	64.63	0.06	64.68
	83.73	60.51	96.79	59.45	61.50	30.04	64.88	60.10			
$\tau = 0.3$	84.56	59.83	96.69	58.21	61.36	23.48	64.70	59.64	63.49	0.08	63.56
	84.56	59.82	96.48	57.86	61.38	23.48	64.70	59.64			

Table 6: Comprehensive performance comparisons on the CoIN benchmark using LLaVA-1.5-7B. For each method, the first row reports per-task accuracy evaluated immediately after training on that task ($A_{i,i}$), and the second row reports per-task accuracy after completing all tasks ($A_{K,i}$).

Method	Datasets								Metrics		
	ScienceQA	TextVQA	ImageNet	GQA	VizWiz	Grounding	VQAv2	OCR-VQA	Avg.ACC(↑)	Forgetting(↓)	New.ACC(↑)
MoBLoRA (Ours)	88.96	65.69	96.73	63.19	62.65	40.30	67.68	66.78	68.98	0.02	69.00
	88.94	65.63	96.61	63.22	62.68	40.30	67.68	66.78			

Table 7: Comprehensive performance of MoBLoRA on the CoIN benchmark using LLaVA-1.5-13B. The first row reports per-task accuracy evaluated immediately after training on that task ($A_{i,i}$), and the second row reports per-task accuracy after completing all tasks ($A_{K,i}$).

Method	Datasets (Reverse)								Metrics		
	OCR-VQA	VQAv2	Grounding	VizWiz	GQA	ImageNet	TextVQA	ScienceQA	Avg.ACC(↑)	Forgetting(↓)	New.ACC(↑)
MoBLoRA (Ours)	63.60	66.28	28.45	62.10	57.07	93.05	60.61	83.16	64.26	0.04	64.29
	63.61	66.16	28.43	62.05	57.08	93.05	60.50	83.16			

Table 8: Comprehensive performance of MoBLoRA on the CoIN benchmark in reverse order using LLaVA-1.5-7B. The first row reports per-task accuracy evaluated immediately after training on that task ($A_{i,i}$), and the second row reports per-task accuracy after completing all tasks ($A_{K,i}$).

Method	Datasets (Alphabet)								Metrics		
	GQA	Grounding	ImageNet	OCR-VQA	ScienceQA	TextVQA	VizWiz	VQAv2	Avg.ACC(↑)	Forgetting(↓)	New.ACC(↑)
$\tau = 1$	59.97	36.24	96.50	57.51	59.11	59.85	61.45	64.76	61.85	0.08	61.92
	60.10	36.21	96.26	56.83	59.14	60.05	61.47	64.76			
$\tau = 0.7$	59.97	29.83	96.51	59.63	84.53	59.12	62.24	64.90	64.33	0.29	64.59
	60.03	29.80	96.20	57.61	84.56	59.36	62.21	64.90			
$\tau = 0.5$	59.48	32.61	96.59	45.41	84.13	59.03	61.66	64.83	62.75	0.25	62.97
	59.48	32.59	96.30	43.87	84.15	59.17	61.61	64.83			
$\tau = 0.3$	58.13	23.84	96.44	45.00	68.71	59.26	60.34	64.64	59.36	0.21	59.55
	58.08	23.83	96.14	43.88	68.73	59.22	60.34	64.64			

Table 9: Comprehensive performance comparisons on the CoIN benchmark in alphabetical order using LLaVA-1.5-7B. For each method, the first row reports per-task accuracy evaluated immediately after training on that task ($A_{i,i}$), and the second row reports per-task accuracy after completing all tasks ($A_{K,i}$).

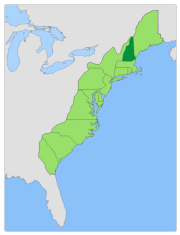
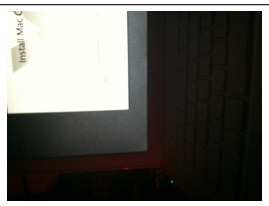
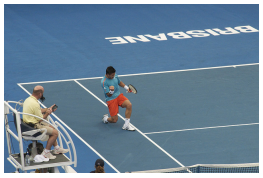
	**Dataset:** ScienceQA **Question:** What is the name of the colony shown? A. Maryland B. New Hampshire C. Rhode Island D. Vermont Answer with the option's letter directly. **Answer:** B
	**Dataset:** TextVQA **Question:** what number is on the player's jersey? Reference OCR token:22 Answer the question using a single word or phrase. **Answer:** 22
	**Dataset:** ImageNet **Question:** What is the object in the image? Answer the question using a single word or phrase. **Answer:** Cabbage butterfly.
	**Dataset:** GQA **Question:** Is it overcast? Answer the question using a single word or phrase. **Answer:** No
	**Dataset:** VizWiz **Question:** Is there anything on the screen? When the provided information is insufficient, respond with 'Unanswerable'. Answer the question using a single word or phrase. **Answer:** yes
	**Dataset:** Grounding **Question:** Please provide the bounding box coordinate of the region this sentence describes: tennis player. **Answer:** [0.32,0.33,0.58,0.74]
	**Dataset:** VQAv2 **Question:** Where is he looking? Answer the question using a single word or phrase. **Answer:** down
	**Dataset:** OCR-VQA **Question:** Who wrote this book? When the provided information is insufficient, respond with 'Unanswerable'. Answer each question using a single word or phrase. **Answer:** Amy Knapp

Table 10: Task data with images across ScienceQA, TextVQA, ImageNet, GQA, VizWiz, Grounding, VQAV2, and OCR-VQA.