HiDe-LLaVA: Hierarchical Decoupling for Continual Instruction Tuning of Multimodal Large Language Model

Haiyang Guo[1,2*], Fanhu Zeng[2,3*], Ziwei Xiang[2,3], Fei Zhu[4], Da-Han Wang[5], Xu-Yao Zhang[1,2,3†], Cheng-Lin Liu[1,2,3]

[1]School of Advanced Interdisciplinary Sciences, UCAS [2]MAIS, CASIA

[3]School of Artificial Intelligence, UCAS [4]Centre for Artificial Intelligence and Robotics, HKISI-CAS

[5]FKLPRIU, School of Computer and Information Engineering, Xiamen University of Technology

{guohaiyang2023, zengfanhu2022, zhufei2018}@ia.ac.cn, {xyz, liucl}@nlpr.ia.ac.cn

Abstract

Instruction tuning is widely used to improve a pre-trained Multimodal Large Language Model (MLLM) by training it on curated task-specific datasets, enabling better comprehension of human instructions. However, it is infeasible to collect all possible instruction datasets simultaneously in real-world scenarios. Thus, enabling MLLM with continual instruction tuning is essential for maintaining their adaptability. However, existing methods often trade off memory efficiency for performance gains, significantly compromising overall efficiency. In this paper, we propose a task-specific expansion and task-general fusion framework based on the variations in Centered Kernel Alignment (CKA) similarity across different model layers when trained on diverse datasets. Furthermore, we analyze the information leakage present in the existing benchmark and propose a new and more challenging benchmark to rationally evaluate the performance of different methods. Comprehensive experiments showcase a significant performance improvement of our method compared to existing state-of-the-art methods. Code and dataset are released at https://github.com/Ghy0501/HiDe-LLaVA.

1 Introduction

Recent years have witnessed remarkable advancements in Multimodal Large Language Models (MLLMs) (Yin et al., 2023), which extend the capabilities of Large Language Models (Touvron et al., 2023) through sophisticated vision-text feature alignment mechanisms (Liu et al., 2024b) and autoregressive generation frameworks. The integration of large-scale training corpora and extensive model parameters (Yang et al., 2024; Kaplan et al., 2020) has enabled MLLMs to achieve state-of-the-art performance across diverse downstream applications (Zhao et al., 2025a; Lu et al., 2024), demonstrating the potential for complex world understanding and representing a significant milestone toward the realization of artificial general intelligence.

As a pivotal component for MLLMs, instruction tuning (Zhang et al., 2023b) enhances instruction-following capabilities of pre-trained models, effectively bridging general-purpose pretraining and domain-specific applications by aligning model behavior with user intent. However, in practical applications, users often perform continuous fine-tuning on diverse datasets at different times to meet specific needs. This requires the model to effectively incorporate new knowledge while overcoming catastrophic forgetting (Li and Hoiem, 2017; Kirkpatrick et al., 2017) on previous tasks.

Recent work (Chen et al., 2024a) construct a benchmark to evaluate the capability of MLLMs in continual instruction tuning (CIT) and reveal that there is a serious catastrophic forgetting phenomenon in MLLMs. In terms of methodology, they propose MoELoRA to mitigate the model's forgetting of old instructions. Based on this, ModalPrompt (Zeng et al., 2024) dynamically selects optimal prompts during inference by jointly leveraging textual and visual features, thereby enhancing model performance. However, we observe that the downstream datasets used by these methods partially overlaps with the tasks encountered during the supervised fine-tuning (SFT) phase of MLLM. Such information leakage (Kim et al., 2023) compromises the reliability of evaluating whether a method mitigates forgetting or if the model inherently retains this capability, thereby diminishing the challenge of continual instruction tuning.

In this paper, we first select the instruction tuning datasets that the model has not encountered during the SFT phase by evaluating the model's zero-shot performance on specific task. Therefore, our reconstructed continual instruction tuning dataset prioritizes instructions that are unfamiliar or un-

*Equal Contribution.

†Corresponding Author.

Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 13572–13586

known to the model, enabling a more accurate and fair comparison of the performance across different methods. To tackle the issue of catastrophic forgetting, we first investigate the CKA similarity (Kornblith et al., 2019) of the model's outputs between the same layers on different instruction tuning datasets and observe that the model exhibits a significant similarity difference between the top layer and the remaining layers. Based on this observation, our experimental analysis indicates that the model focuses more on task-specific information in the top layer, while learning more generic knowledge in the remain layers. Therefore, we propose a **Hi**erarchical **De**coupling method named **HiDe-LLaVA**, which consists of two simple yet effective strategies: task-general fusion and task-specific expansion. Comprehensive results verified that our method effectively overcoming the catastrophic forgetting during continual instruction tuning.

In summary, our main contributions include:

- We propose HiDe-LLaVA, which enhances the continual instruction tuning performance by decoupling the model layers into task-specific expansion and task-general fusion.
- We reveal information leakage in existing benchmark and propose a more fair continual instruction tuning benchmark to equitably assess the effects of different methods.
- Extensive experimental results show that our method effectively overcomes the catastrophic forgetting during continual instruction tuning.

2 Relate Work

Multimodal Large Language Models. With the predominant capability of large language models (LLMs) (Touvron et al., 2023; Zhang et al., 2023a) in handing natural language processing tasks (Wang et al., 2022a; Min et al., 2023), huge amount of efforts has been made to multimodal learning and multimodal large language models (MLLMs) (Bai et al., 2023; Zhu et al., 2024a; Dai et al., 2023) are proposed to tackle task in multimodal scenarios. In addition to a large language model, which is composed of stacks of transformer blocks (Vaswani et al., 2017), MLLMs also incorporates a vision encoder (Dosovitskiy et al., 2021) and a feature fusion module to align the cross-modality representations (usually a projection layer (Liu et al., 2024b) or cross attention (Dai et al., 2023)). Answers are then generated in response to multimodal inputs. The paradigm is of great significance and achieves impressive performance on various downstream tasks (Liu et al., 2025; Zhao et al., 2025b; Zhang et al., 2024).

Continual Instruction Tuning. Instruction tuning (Ouyang et al., 2022; Zhang et al., 2023b) aims to enpower MLLMs to better understand and follow human instructions. To adapt to dynamically changing instructions in real-world scenarios, MLLMs need the capability to learn current and retain previous learned instructions, which is named *continual instruction tuning*. In recent studies, Eproj (He et al., 2023) first constructs a benchmark in continual instruction tuning settings and reveal that catastrophic forgetting is still observed in MLLMs. However, the benchmark they proposed is insufficient in terms of the number and diversity of tasks and is not evaluated on mainstream MLLM architectures. By contrast, CoIN (Chen et al., 2024a) establishes a diversity benchmark consisting of 8 crafted datasets and proposed MoELoRA to mitigate catastrophic forgetting on various MLLMs. More recently, ModalPrompt (Zeng et al., 2024) further enhances performance in this setting by leveraging image and text modalities to guide the selection of appropriate prompts from the prompt pool. Continual-LLaVA (Cao et al., 2024) similarly proposes a pool of LoRAs (Hu et al., 2021) to select the appropriate LoRAs for training based on textual similarity, and using them during inference.

3 Preliminary

Continual instruction tuning for MLLMs aims to address the problem of learning with continuous data and mitigate catastrophic forgetting. Assume that there are T tasks in total, and the MLLM parameterized by θ has been pre-trained on large-scale image-text data to obtain aligned multimodal features. The sequential tasks data of continual instruction tuning can be expressed as: $\mathcal{D}_t = \{(\mathbf{x}_v^{t,j}, \mathbf{x}_{ins}^{t,j}, \mathbf{x}_{ans}^{t,j})_{j=1}^{N_t}\}$, $t \in \{1, ..., T\}$, where $\mathbf{x}_v^{t,j}$, $\mathbf{x}_{ins}^{t,j}$ and $\mathbf{x}_{ans}^{t,j}$ denote the input image tokens, instruction tokens and answer tokens, N_t is the total number of image-text pairs of task t. Taking a simple image-text pair with an answer in length L, the objective of MLLM is to predict next token based on all the preceding tokens in an

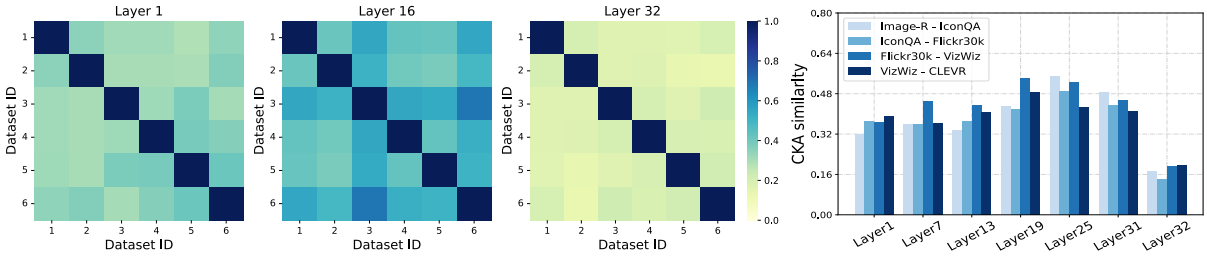


Figure 1: Left: Output CKA similarity heatmaps for different task inputs across the bottom, middle, and top layers. Overall, the output similarity across different tasks markedly decreases at the top layer. Right: Detailed similarity comparison of different task pairs. It can be seen that even for these very different pairs (IconQA and Flickr30k), the similarity differences only appear in last layers and most layers are similar.

autoregressive way:

$$\mathcal{L}^t_{MLLM} = -\sum_{l=1}^{L} \log p(\mathbf{x}^l_{ans} | \mathbf{x}^t_v, \mathbf{x}^t_{ins}, \mathbf{x}^{<l}_{ans}; \theta). \tag{1}$$

When learning task t, the goal of continual instruction tuning is to maximize the retention of knowledge from the learned task while preserving the model's generalization ability for unseen tasks.

4 Methodology

4.1 Hierarchical decoupling of MLLM

Existing research (Kornblith et al., 2019) shows that the outputs between different layers of the model exhibit notable differences, indicating that the model focuses on different patterns of the input at each layer. Inspired by this, we fine-tune LLaVA-v1.5 (Liu et al., 2024b) using LoRA on 6 instruction tuning datasets and analyze the output CKA similarity (Kornblith et al., 2019) differences across the same layers between models.[1] As illustrated in left of Fig 1, the similarity of the model outputs exhibits a clear difference between the top and remaining layers. Specifically, for two datasets that differ significantly in both input image styles and textual instructions (IconQA (Lu et al., 2021) and Flickr30k (Plummer et al., 2017)), we observe that their output similarity differs notably only in the top layer (*i.e.* the layer closest to the output), while remaining relatively high across the other layers. This suggests that for different tasks, the model primarily learns similar, generalized knowledge in the layers below the top layer, while the top layer focuses more on dissimilar, task-specific information. Therefore, we hypothesize that for continual instruction tuning, the shared information can be obtained by integrating the layers beneath

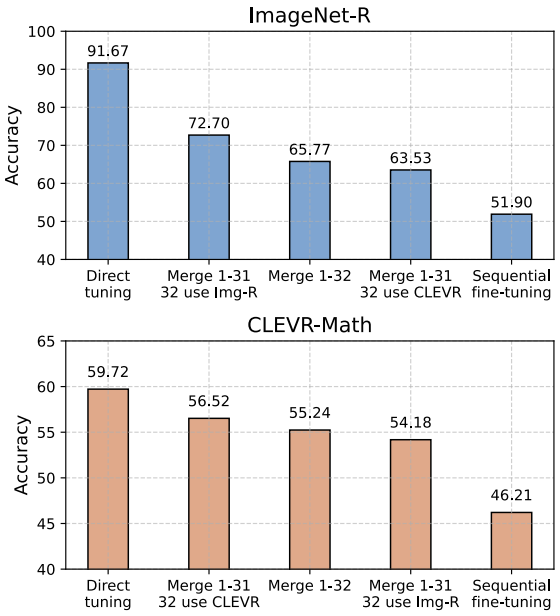
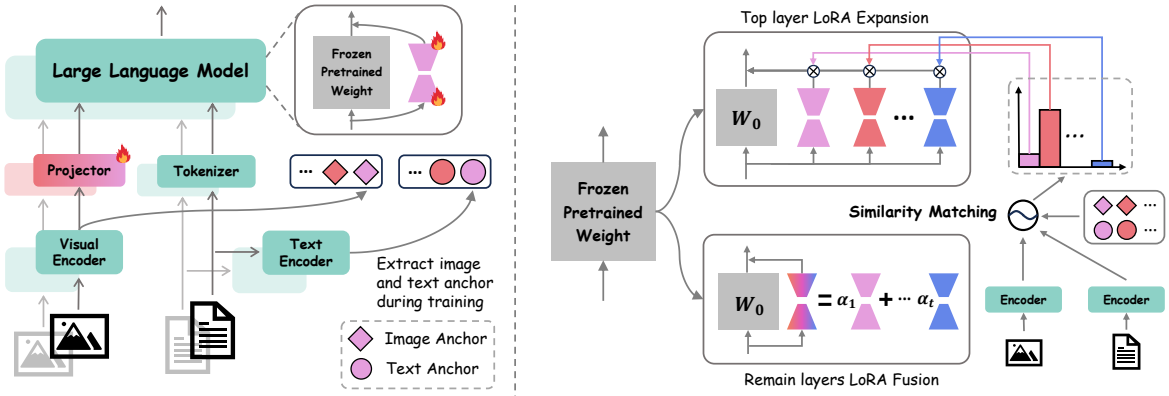


Figure 2: Impact of different LoRA operational strategies on individual task performance.

the top layer, while task-specific knowledge in the top layer should be selectively activated.

To further validate our hypothesis, we performe LoRA fine-tuning on two instruction tuning datasets, ImageNet-R (Hendrycks et al., 2021) and CLEVR-Math (Lindström and Abraham, 2022), to obtain their respective LoRA weights. We then apply the following operations to the LoRAs at different layers: ***(i)*** applying corresponding LoRAs to all layers, ***(ii)*** applying corresponding LoRAs only to the top layer (32-th layer) while merging the rest (1-31 layers), ***(iii)*** merging LoRAs across all layers, ***(iv)*** applying only the top layer with non-corresponding LoRAs while merging the rest of the layers, and ***(v)*** fine-tuning the same set of LoRAs in sequence. As shown in Fig 2, directly merging all LoRA layers for two tasks provides greater mitigation of catastrophic forgetting compared to sequential fine-tuning (*i.e. iii* and *v*), but also leads to performance degradation for the respective tasks. However, merging only the layers

[1]The details of the CKA similarity computation are presented in Appendix B.



(a) The training stage (b) The inference stage

Figure 3: An overview of HiDe-LLaVA framework. (a) During training, we optimize the LoRA modules and projector layer with an autoregressive loss and the image-text anchors are extracted from the image and text encoders of CLIP. (b) At inference time, our method apply a MoE-like expansion on the top-layer LoRA and dynamically distribute expert weights via similarity matching with previously learned image and text anchors. For the remaining layers, general knowledge across tasks is incorporated through LoRA fusion.

below the top layer and correctly selecting task-specific LoRAs for the top layer results in further performance improvements. By contrast, selecting non-corresponding LoRAs for the top layer can degrade the model's performance on the corresponding task (*i.e. ii* and *iv*).

Building on the above analysis, we posit that catastrophic forgetting can be effectively mitigated through hierarchical decoupling, integrating task-general knowledge across tasks, and adaptively selecting task-specific information. The framework of our method is shown in Fig. 3.

4.2 Proposed Method: HiDe-LLaVA

Following the LoRA fine-tuning strategy used in LLaVA (Liu et al., 2024a), we embed LoRA modules into all linear layers of the language model. During training, LoRA modules and the projector layer are trained to align with the current input instructions. For clarity, we use E to represent all LoRA modules in the following.

4.2.1 Task-specific Expansion on Top Layer

As mentioned above, the model emphasizes task-specific knowledge at the top layer, making it important to allocate the output appropriately. Therefore, adaptively selecting the appropriate LoRA module based on the inputs, without relying on a Task-ID, becomes a critical challenge. Inspired by prototype learning (Zhu et al., 2021; Tan et al., 2022), we propose extracting high-dimensional feature representations of input data during training as an alternative to explicit Task-ID assignment. Specifically, the image features can be extracted directly by the LLaVA's visual encoder, while on the text side, we introduce CLIP's text encoder (Radford et al., 2021) to derive feature representations from the input instructions:

$$m_v^t = \frac{1}{N_t}\sum_{n=1}^{N_t} f_v(\mathbf{x}_v^t),\ m_{ins}^t = \frac{1}{N_t}\sum_{n=1}^{N_t} f_{ins}(\mathbf{x}_{ins}^t), \tag{2}$$

where $\mathbf{x}_v^t$ and $\mathbf{x}_{\text{ins}}^t$ denote the input image and instruction for task t, respectively, while m_v^t and m_{ins}^t represent the extracted image and text features, which we collectively refer to as image and text anchors. Here, f_v and f_{ins} correspond to the CLIP image encoder and text encoder, respectively.

With image and text anchors for each task, we expand the top-level LoRA of all learned tasks during inference in a manner similar to a Mixture-of-Experts (MoE) (Shazeer et al., 2017) model. Instead of using a traditional MoE Router, we select the appropriate LoRA's output based on the cosine similarity between the input test data and the anchors. Specifically, we first compute the cosine similarity between the features of the current test input $\mathbf{z}^{test}$ and each task anchor:

$$r_v^c = \text{sim}(\mathbf{z}_v^{test}, m_v^c),\ r_{ins}^c = \text{sim}(\mathbf{z}_{ins}^{test}, m_{ins}^c), \tag{3}$$

where $\text{sim}(\mathbf{a},\mathbf{b}) = \frac{\mathbf{a}\cdot\mathbf{b}}{\|\mathbf{a}\|\ \|\mathbf{b}\|}$. The similarities of the image and text are then obtained as a normalized score d_c using the softmax function:

$$d_c = \frac{e^{\bar{r^c}/T}}{\sum_{j=1}^{T} e^{\bar{r^j}/T}},\ \text{where}\ \bar{r^j} = \alpha\cdot r_v^c + \beta\cdot r_{ins}^c, \tag{4}$$

where T denotes the temperature coefficient of Softmax, α and β are the hyper-parameters that control

the fusion of image and text similarity. The resulting d_c represents the degree of match between the current test input and a particular learned task c. We use this in place of the expert weights computed via the router in traditional MoE models, multiplying the output of each top-level LoRA by the corresponding score and summing them to obtain the final output of the entire LoRA branch on the top layer:

$$O_{top} = \sum_{i=1}^{T} d_i E_i(h), \tag{5}$$

where E_i denotes the i-th LoRA expert and h represents the hidden input of the linear layer. Compared to MoELoRA, our method mitigates catastrophic forgetting by eliminating the need to train a router at each stage, leading to a more effective and stable distribution of LoRA outputs.

4.2.2 Task-general Fusion on Remain Layers

For the remaining layers outside the top layer, our CKA similarity analysis in Sec 4.1 reveals a relatively high degree of output similarity, suggesting that these layers encode more generalized knowledge shared across tasks. To integrate this generalized knowledge, we employ a simple and effective model parameter fusion strategy (Ilharco et al., 2022; Zheng et al., 2025; Zeng et al., 2025):

$$\bar{E}_T = \sum_{i=1}^{T} \epsilon_i E_i, \tag{6}$$

where T is the number of learned tasks and ϵ_i denotes the fusion coefficient of i-th LoRA module. The output of the remaining layer LoRA branches can then be represented as:

$$O_{rem} = \bar{E}_T(h). \tag{7}$$

In summary, our proposed HiDe-LLaVA hierarchically decomposes the model into a top-level LoRA extension and the fusion of LoRAs in the remaining layers. The overall framework is summarized in Algorithm 1.

Algorithm 1 Pipeline of HiDe-LLaVA

Input: $\mathcal{D}_t : \{(\mathbf{x}_v^{t,j}, \mathbf{x}_{ins}^{t,j}, \mathbf{x}_{ans}^{t,j})_{j=1}^{N_t}\}, t \in \{1, .., T\}$. LoRA modules E and Projector layer *Proj*.

for $t = 1 \rightarrow T$ **do**
  # Training stage.
  for $(\mathbf{x}_v^{t,j}, \mathbf{x}_{ins}^{t,j}, \mathbf{x}_{ans}^{t,j}) \in \mathcal{D}_t$ **do**
    Train E_t and *Proj* through Eq (1).
    Extract anchors through Eq (2).
  end for
  # Inference stage.
  for $\mathcal{D}_t^{test} \in \{\mathcal{D}_1^{test}, ..., \mathcal{D}_T^{test}\}$ **do**
    # Task-specific Expansion.
    Compute d_c through Eq (4).
    for E embedded in top layer **do**
      Expand the LoRA branch in a MoE manner and compute the output through Eq (5).
    end for
    # Task-general Fusion.
    for E embedded in remain layers **do**
      Fuse the LoRA modules of all learned tasks through Eq (6).
    end for
  end for
end for

5 Experiments

5.1 Experimental Setup

Datasets We train and evaluate the effectiveness of our method on (1) CoIN (Chen et al., 2024a) benhcmark and (2) our reconstructed benchmark. In particular, CoIN consists of datasets such as VQAv2 (Goyal et al., 2017), VizWiz (Gurari et al., 2018), ScienceQA (Lu et al., 2022), TextVQA (Singh et al., 2019), GQA (Hudson and Manning, 2019), OCR-VQA (Mishra et al., 2019), ImageNet (Deng et al., 2009) and REC-COCO (Kazemzadeh et al., 2014; Mao et al., 2016). A major limitation of CoIN is the overlap between the selected downstream datasets and the dataset used during LLaVA's pre-training phase, resulting in information leakage (Kim et al., 2023) that significantly undermines fair comparisons between different methods. To address this, we screen six datasets that are highly uncorrelated with the pre-training data based on LLaVA's zero-shot performance (See Zero-shot in Table 1), specifically: ArxivQA (Li et al., 2024), CLEVR-Math (Lindström and Abraham, 2022), IconQA (Lu et al., 2021), ImageNet-R (Hendrycks et al., 2021), VizWiz-caption (Gurari et al., 2018) and Flickr30k (Plummer et al., 2015). We term this the **U**nseen **C**ontinual **I**nstruction **T**uning (**UCIT**) benchmark. Details are provided in Appendix E and F.

Evaluation Metrics. Following the standard metrics in conrinual learning (Wang et al., 2022b). We report *Last*, *Avg* to evaluate the continual instruction tuning performance. *Last* is computed as the accuracy of all seen tasks after learning the final

	Method	Image-R	ArxivQA	Viz-cap	IconQA	CLEVR	Flickr30k	Average
	Zero-shot	16.27	53.73	38.39	19.20	20.63	41.88	31.68
	Multi-task	90.63	91.30	61.81	73.90	73.60	57.45	74.78
Avg	FineTune	49.31	78.40	50.48	53.44	55.53	57.95	57.52
	LwF	55.60	79.86	53.23	54.87	56.51	56.34	59.40
	EWC	54.23	80.13	53.14	55.06	57.52	55.94	59.34
	L2P	41.52	82.32	51.98	52.21	43.16	52.77	53.99
	O-LoRA	75.26	86.73	**55.86**	58.47	57.38	53.52	64.54
	MoELoRA	64.49	82.42	49.54	56.87	56.35	**58.34**	61.33
	HiDe-LLaVA	**85.70**	**92.70**	54.10	**66.87**	**59.12**	55.15	**68.94** (+4.4)
Last	FineTune	37.63	72.33	43.47	41.7	35.63	57.95	48.12
	LwF	40.27	75.93	42.76	44.38	37.43	56.34	49.52
	EWC	39.05	77.88	43.24	45.33	39.72	55.94	50.20
	L2P	32.73	80.41	43.72	42.16	39.25	52.77	48.51
	O-LoRA	69.36	82.42	48.64	53.66	42.53	53.52	58.36
	MoELoRA	49.87	77.63	43.65	46.40	36.47	**58.34**	52.06
	HiDe-LLaVA	**80.50**	**89.83**	**48.78**	**62.90**	**47.97**	55.15	**64.19** (+5.8)

Table 1: Comparison with various methods on our UCIT benchmark in terms of *Avg* and *Last*. The best and second methods are labeled with **bold** and underline styles. Our method outperforms the best previous methods by **4.4%** and **5.8%** in Avg and Last metrics, respectively.

task and *Avg* is the average accuracy of each task during the training process. These two metrics measure the model's capacity to retain learned tasks.

Baseline. We compare our HiDe-LLaVA with traditional continual learning methods and recent continual instruction tuning method. For the former, we implement LwF (Li and Hoiem, 2017), EWC (Kirkpatrick et al., 2017), L2P (Wang et al., 2022b) and O-LoRA (Wang et al., 2023) within a MLLM architecture, meticulously tuning parameters to ensure reliable and effective results. For the latter, We compare with MoELoRA (Chen et al., 2024a) to highlight the superiority of our method. The performance of zero-shot and multi-task fine-tuning is also reported to represent the lower and upper bounds for each benchmark. Specific details of each method can be found in Appendix A.

Implementation details. We use LLaVA-v1.5-7b (Liu et al., 2024b) as the base multimodal model and CLIP-L/14-336 (Radford et al., 2021) to extract visual and textual features. Following LLaVA's LoRA fine-tuning strategy, we embed LoRA modules in all linear layers of the language model with the rank set to 8. The temperature coefficient T and hyper-parameters in Eq (4) are set to 0.1, 0.5 and 0.5, respectively. The fusion coefficient in Eq (6) are uniformly set to 1.0. 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. We set the batch size to 24 for all methods and run the experiments on A800 GPUs.

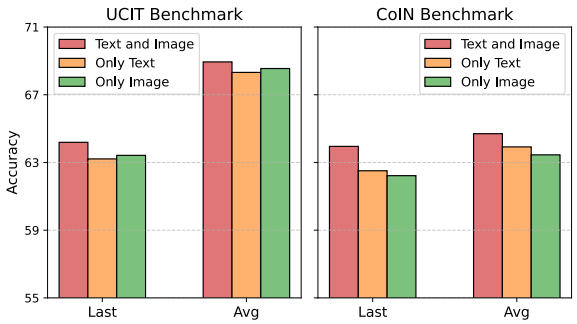


Figure 4: Ablation studies of dual-modalities similarity matching on UCIT and CoIN benchmark.

5.2 Main Results

Results on UCIT and CoIN benchmark are shown in Table 1 and 2. Our method achieves state-of-the-art performance on both benchmarks, effectively mitigating catastrophic forgetting in MLLM during continual instruction tuning. Specifically, MoELoRA alleviates forgetting by transforming LoRA fine-tuning into a Mixture-of-Experts (MoE) model, while simultaneously training a router to assign the appropriate task outputs. Traditional continual learning methods, such as LwF, EWC, and L2P, show limited success, whereas O-LoRA outperforms all other baseline methods in the comparison. However, these methods still exhibit a significant gap when compared to multi-task learning. In contrast, our method not only significantly outperforms other methods in terms of Avg and Last metrics (with average improvements of **4.4%**, **5.8%** on the UCIT benchmark, and **2.1%**, **3.2%** on the CoIN benchmark), but also leads substantially

	Method	SciQA	Image	Viz	REC	Text	GQA	VQA	OCR	Average
	Zero-shot	69.79	9.93	45.50	58.47	57.75	60.77	66.50	64.93	54.21
	Multi-task	82.36	89.63	52.51	65.83	61.27	59.93	65.67	62.03	67.40
Avg	FineTune	64.22	40.13	43.87	38.32	55.04	55.89	60.61	64.78	52.86
	LwF	65.20	40.63	43.22	40.05	56.23	54.67	60.64	65.12	53.22
	EWC	65.11	40.89	44.09	39.67	54.92	56.03	61.12	64.55	53.30
	L2P	70.52	26.89	45.53	45.21	56.84	59.03	63.52	64.11	53.96
	O-LoRA	73.32	68.37	50.26	61.12	**57.75**	60.96	65.71	63.31	62.60
	MoELoRA	68.38	48.50	44.22	40.23	55.62	57.04	62.14	**65.75**	55.24
	HiDe-LLaVA	**74.92**	**76.72**	**51.24**	**61.84**	57.13	**62.83**	**68.15**	64.76	**64.70** (+2.1)
Last	FineTune	57.43	28.90	41.88	30.05	51.39	50.76	53.28	64.78	47.31
	LwF	60.71	30.58	41.49	36.01	52.80	47.07	53.43	65.12	48.40
	EWC	59.75	31.88	42.26	34.96	51.06	51.84	55.30	64.55	48.95
	L2P	70.21	23.31	44.21	43.76	56.25	58.46	62.32	64.11	52.83
	O-LoRA	72.56	62.84	48.43	58.97	**57.66**	59.14	63.21	63.31	60.77
	MoELoRA	62.02	37.21	43.32	33.22	52.05	53.12	57.92	**65.75**	50.58
	HiDe-LLaVA	**73.20**	**69.28**	**50.76**	**59.18**	56.92	**61.33**	**67.12**	64.76	**63.95** (+3.2)

Table 2: Comparison with various methods on our CoIN benchmark in terms of *Avg* and *Last*. The best and second methods are labeled with **bold** and underline styles. Our HiDe-LLaVA outperforms the best previous methods by **2.1%** and **3.2%** in Avg and Last metrics, respectively.

Methods	Last	Δ	Avg	Δ	Param. (M)	Δ
HiDe-LLaVA	64.19	0.0	68.94	0.0	44.27	×1
Merge (1-32)	61.26	-2.93	65.43	-3.51	38.27	×0.86
Merge (1-31)	60.64	-3.55	63.28	-5.66	44.27	×0.84
Expand (1-32)	67.62	+3.43	70.91	+1.97	229.62	×5.20
Expand (1-31)	66.84	+2.65	70.13	+1.19	222.42	×5.02

Table 3: Ablation studies of different fusion and expansion strategies conducted on the UCIT benchmark.

in training and inference efficiency (See Sec 5.4).

Notably, by comparing the Zero-shot and Multi-task performance on the UCIT and CoIN benchmarks in Table 1 and 2, we observe serious information leakage in CoIN. Specifically, the average performance difference between Multi-task and Zero-shot in CoIN is only 13.19%, while in our UCIT benchmark, the difference is significantly higher at **43.10%**. Additionally, some datasets in CoIN even performed worse than Zero-shot after fine-tuning (*e.g.,* GQA, OCRVQA and VQAv2), suggesting that LLaVA may have encountered similar tasks during pre-training. The continued reuse of these datasets could lead to overfitting, thereby impacting the fairness of comparisons between different methods. Additional experimental results are provided in Appendix D.

5.3 Ablation Study

Our HiDe-LLaVA comprises two main components: the expansion of task-specific knowledge in the top layer and the fusion of task-general information in the remaining layers. We explore their individual impacts on the results in this section.

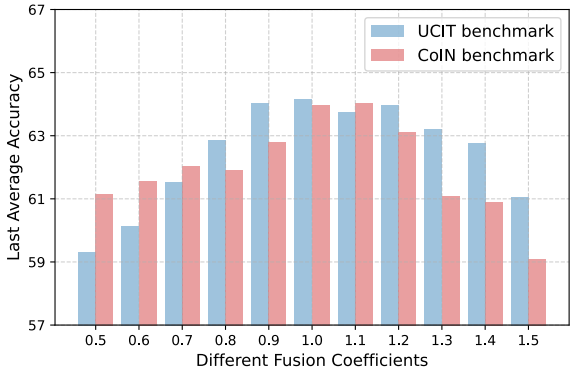


Figure 5: Ablation studies of the fusion coefficient on UCIT and CoIN benchmark.

The effect of dual-modality similarity matching. At the top layer, we compute the expert weights for each LoRA output by matching the input features of the two modalities, image and text, with the mean values of the features extracted during training. In Fig 4, we compare the results of calculating similarity using only a single modality and observe that the results for individual modalities are consistently lower than those for dual-modality matching. This is primarily because a single modality cannot fully capture the differences between tasks. For instance, the image inputs for some tasks may all come from the same dataset, yet the textual instructions differ significantly. In contrast, some tasks may all involve question-answering, but the input images vary greatly in style. Therefore, considering both image and text similarities is crucial.

The impact of fusion and expansion strategies. Based on the CKA similarity analysis presented in

Metrics \ Order	RAVICF	AIRFCV	IFRCAV
Last	64.19	63.56	64.87
Avg	68.94	68.02	69.68

Table 4: Results of different task orders on UCIT benchmark. We adopt an abbreviation scheme to simplify the representation of task sequence notation. For example, **RAVICF** represents the order of ImgNet-**R** → **A**rxivQA → **V**izWiz-cap → **I**conQA → **C**LEVR-Math → **F**lickr30k.

Sec 4.1, our HiDe-LLaVA expands LoRA at the top layer and fuse it in the remaining layers. In this section, we compare our method with the results of expanding or fusing LoRA modules across all layers. As shown in Table 3, the model's performance significantly degrades when all layers are fused, or when all layers except the top layer are fused with LoRA. Although expanding the top layer operation to all layers or the remaining layers yields some performance improvement, it leads to a considerable increase in parameters during inference (*e.g.,* ×**5.20**), which is unreasonable for long-term continual instruction tuning tasks. Our HiDe-LLaVA, therefore, strikes an optimal balance between performance and parameter efficiency.

The impact of fusion coefficient. In order to integrate generalized knowledge across different tasks, we propose a simple yet effective model fusion strategy in Sec 4.2.2, where the choice of parameter fusion coefficients plays a crucial role in the results. In Fig 5, we present a comprehensive results of the impact of different fusion coefficients. Overall, larger fusion coefficients contribute positively to the model's performance, with a coefficient of 1.0 yielding the best average results across both benchmarks.

5.4 Further Analysis

Analysis of Different task order. To evaluate the robustness of our method across diverse scenarios, we conduct additional experiments with different task sequences on UCIT benchmark. The results, as presented in Table 4, demonstrate that our proposed HiDe-LLaVA maintains consistent and stable performance, regardless of the order in which the tasks are presented. This suggests that our method is resilient to task ordering and can effectively handle variations in task sequences during continuous instruction fine-tuning.

Analysis of the efficiency of the training and inference phases. In Fig. 6, we compare the efficiency in terms of both training and inference times across different methods. Specifically, MoELoRA requires the most memory occupied at inference time, as it transforms all embedded LoRAs into the form of Mixture-of-Experts and necessitates prior knowledge of the number of tasks to learn. This limitation is problematic in real-world applications, where predicting the number of future tasks is often infeasible. O-LoRA, on the other hand, concatenates the LoRAs of all learned tasks with the current LoRA, resulting in a substantial increase in memory usage as the number of tasks increases. In contrast, our method only retains the LoRAs of all learned tasks at the top level, which significantly reduces the memory overhead associated with additional parameters.

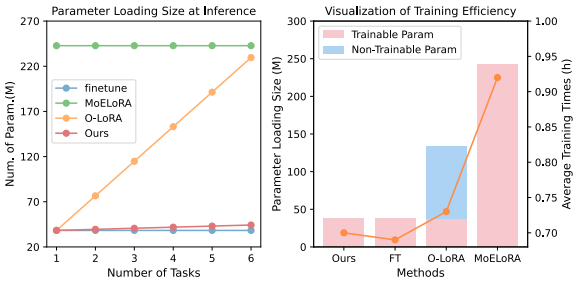


Figure 6: Efficiency analysis on UCIT benchmark. Left: Comparison of the number of parameters loaded during inference. Right: Comparison of training time and loaded parameters.

As for the overhead during training, MoELoRA introduces a large number of trainable parameters, leading to a **20%** increase in training time compared to standard fine-tuning. O-LoRA, on the other hand, need to compute the orthogonal loss between the parameter space of the current task and those of previously learned tasks, requiring the inclusion of LoRAs for all past tasks during training. In contrast, our method achieves optimal performance while maintaining training times and efficiency nearly on par with fine-tuning, further demonstrating the effectiveness of our approach.

6 Conclusion

In this paper, we investigate how to equip MLLM with the ability to continuously follow user input instructions. Our method first decouples the MLLM into task-specific knowledge layer and task-general knowledge layer based on the differences in CKA similarity of the outputs at each layer between tasks. Then, a task-specific expansion and task-general fusion framework is proposed to mitigate catastrophic forgetting during continual instruction tuning. Ad-

ditionally, we identify and analyze the information leakage in existing benchmark and introduce more challenging benchmark to ensure a fair evaluation of different methods. Extensive experiments demonstrate that our method not only achieves optimal performance but also maintains high efficiency, making it a well-balanced solution.

Limitations

In this work, our proposed HiDe-LLaVA introduces a post-processing framework designed for task-general knowledge fusion and task-specific knowledge expansion. While it achieves state-of-the-art performance, the framework is constrained by performance degradation caused by model fusion operations. We argue that investigating methods to reduce inter-task conflicts during training could further alleviate the issue of catastrophic forgetting.

Moreover, while this paper and existing research primarily address the forgetting phenomenon of old tasks in the context of continual instruction tuning, we emphasize the importance of preserving the original capabilities of large multimodal models like LLaVA, which exhibit strong zero-shot generalization abilities to unseen tasks. Ensuring that such models retain their foundational strengths under continuous instruction fine-tuning remains a critical challenge, and we plan to explore this direction in future work.

Ethical Impact

We are committed to upholding intellectual property rights and adhering to all applicable laws and regulations. The images and text instructions included in our benchmark are sourced from publicly available materials, and we have implemented rigorous measures to ensure that no personally sensitive information is present. Furthermore, our efforts are solely dedicated to research purposes and are not intended for commercial use.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (62222609, 62320106010), National Science and Technology Major Project (2022ZD0116500), CAS Project for Young Scientists in Basic Research (YSBR-083), and Key Research Program of Frontier Sciences of CAS (ZDBS-LY-7004), Unveiling and Leading Projects of Xiamen (3502Z20241011) , Major Science and Technology Plan Project on the Future Industry Fields of Xiamen City (3502Z20241027) and the InnoHK program.

References

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*.

Meng Cao, Yuyang Liu, Yingfei Liu, Tiancai Wang, Jiahua Dong, Henghui Ding, Xiangyu Zhang, Ian Reid, and Xiaodan Liang. 2024. Continual llava: Continual instruction tuning in large vision-language models. *arXiv preprint arXiv:2411.02564*.

Cheng Chen, Junchen Zhu, Xu Luo, Hengtao Shen, Lianli Gao, and Jingkuan Song. 2024a. Coin: A benchmark of continual instruction tuning for multimodel large language model. *arXiv preprint arXiv:2403.08350*.

Zhe Chen, Jiannan Wu, Wenhai Wang, Weijie Su, Guo Chen, Sen Xing, Muyan Zhong, Qinglong Zhang, Xizhou Zhu, Lewei Lu, et al. 2024b. Internvl: Scaling up vision foundation models and aligning for generic visual-linguistic tasks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 24185–24198.

Wenliang Dai, Junnan Li, Dongxu Li, Anthony Meng Huat Tiong, Junqi Zhao, Weisheng Wang, Boyang Li, Pascale Fung, and Steven Hoi. 2023. Instructblip: Towards general-purpose vision-language models with instruction tuning. *Preprint*, arXiv:2305.06500.

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.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. 2021. An image is worth 16x16 words: Transformers for image recognition at scale. *ICLR*.

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*.

Dan Hendrycks, Steven Basart, Norman Mu, Saurav Kadavath, Frank Wang, Evan Dorundo, Rahul Desai, Tyler Zhu, Samyak Parajuli, Mike Guo, Dawn Song, Jacob Steinhardt, and Justin Gilmer. 2021. The many faces of robustness: A critical analysis of out-of-distribution generalization. *ICCV*.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*.

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, Marco Tulio Ribeiro, Mitchell Wortsman, Suchin Gururangan, Ludwig Schmidt, Hannaneh Hajishirzi, and Ali Farhadi. 2022. Editing models with task arithmetic. *arXiv preprint arXiv:2212.04089*.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. 2020. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*.

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.

Gyuhak Kim, Changnan Xiao, Tatsuya Konishi, and Bing Liu. 2023. Learnability and algorithm for continual learning. In *International Conference on Machine Learning*, pages 16877–16896. PMLR.

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.

Simon Kornblith, Mohammad Norouzi, Honglak Lee, and Geoffrey Hinton. 2019. Similarity of neural network representations revisited. In *International conference on machine learning*, pages 3519–3529. PMLR.

Lei Li, Yuqi Wang, Runxin Xu, Peiyi Wang, Xiachong Feng, Lingpeng Kong, and Qi Liu. 2024. Multimodal arxiv: A dataset for improving scientific comprehension of large vision-language models. *arXiv preprint arXiv:2403.00231*.

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

Adam Dahlgren Lindström and Savitha Sam Abraham. 2022. Clevr-math: A dataset for compositional language, visual and mathematical reasoning. *arXiv preprint arXiv:2208.05358*.

Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. 2024a. 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. 2024b. Visual instruction tuning. *Advances in neural information processing systems*, 36.

Yuan Liu, Haodong Duan, Yuanhan Zhang, Bo Li, Songyang Zhang, Wangbo Zhao, Yike Yuan, Jiaqi Wang, Conghui He, Ziwei Liu, et al. 2025. Mmbench: Is your multi-modal model an all-around player? In *European Conference on Computer Vision*, pages 216–233. Springer.

Pan Lu, Hritik Bansal, Tony Xia, Jiacheng Liu, Chunyuan Li, Hannaneh Hajishirzi, Hao Cheng, Kai-Wei Chang, Michel Galley, and Jianfeng Gao. 2024. Mathvista: Evaluating mathematical reasoning of foundation models in visual contexts. In *The Twelfth International Conference on Learning Representations*.

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.

Pan Lu, Liang Qiu, Jiaqi Chen, Tony Xia, Yizhou Zhao, Wei Zhang, Zhou Yu, Xiaodan Liang, and Song-Chun Zhu. 2021. Iconqa: A new benchmark for abstract diagram understanding and visual language reasoning. In *The 35th Conference on Neural Information Processing Systems (NeurIPS) Track on Datasets and Benchmarks*.

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.

Bonan Min, Hayley Ross, Elior Sulem, Amir Pouran Ben Veyseh, Thien Huu Nguyen, Oscar Sainz, Eneko Agirre, Ilana Heintz, and Dan Roth. 2023. Recent advances in natural language processing via

large pre-trained language models: A survey. *ACM Computing Surveys*, 56(2):1–40.

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 Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744.

Bryan A Plummer, Liwei Wang, Chris M Cervantes, Juan C Caicedo, Julia Hockenmaier, and Svetlana Lazebnik. 2015. Flickr30k entities: Collecting region-to-phrase correspondences for richer image-to-sentence models. In *Proceedings of the IEEE international conference on computer vision*, pages 2641–2649.

Bryan A. Plummer, Liwei Wang, Christopher M. Cervantes, Juan C. Caicedo, Julia Hockenmaier, and Svetlana Lazebnik. 2017. Flickr30k entities: Collecting region-to-phrase correspondences for richer image-to-sentence models. *IJCV*, 123(1):74–93.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. 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.

Yue Tan, Guodong Long, Lu Liu, Tianyi Zhou, Qinghua Lu, Jing Jiang, and Chengqi Zhang. 2022. Fedproto: Federated prototype learning across heterogeneous clients. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 8432–8440.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in Neural Information Processing Systems*.

Haifeng Wang, Hua Wu, Zhongjun He, Liang Huang, and Kenneth Ward Church. 2022a. Progress in machine translation. *Engineering*, 18:143–153.

Xiao Wang, Tianze Chen, Qiming Ge, Han Xia, Rong Bao, Rui Zheng, Qi Zhang, Tao Gui, and Xuanjing Huang. 2023. Orthogonal subspace learning for language model continual learning. *arXiv preprint arXiv:2310.14152*.

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.

Shijia Yang, Bohan Zhai, Quanzeng You, Jianbo Yuan, Hongxia Yang, and Chenfeng Xu. 2024. Law of vision representation in mllms. *arXiv preprint arXiv:2408.16357*.

Shukang Yin, Chaoyou Fu, Sirui Zhao, Ke Li, Xing Sun, Tong Xu, and Enhong Chen. 2023. A survey on multimodal large language models. *arXiv preprint arXiv:2306.13549*.

Fanhu Zeng, Haiyang Guo, Fei Zhu, Li Shen, and Hao Tang. 2025. Parameter efficient merging for multimodal large language models with complementary parameter adaptation. *arXiv preprint arXiv:2502.17159*.

Fanhu Zeng, Fei Zhu, Haiyang Guo, Xu-Yao Zhang, and Cheng-Lin Liu. 2024. Modalprompt: Dual-modality guided prompt for continual learning of large multimodal models. *arXiv preprint arXiv:2410.05849*.

Duzhen Zhang, Yahan Yu, Jiahua Dong, Chenxing Li, Dan Su, Chenhui Chu, and Dong Yu. 2024. Mm-llms: Recent advances in multimodal large language models. *arXiv preprint arXiv:2401.13601*.

Renrui Zhang, Jiaming Han, Chris Liu, Peng Gao, Aojun Zhou, Xiangfei Hu, Shilin Yan, Pan Lu, Hongsheng Li, and Yu Qiao. 2023a. Llama-adapter: Efficient fine-tuning of language models with zero-init attention. *arXiv preprint arXiv:2303.16199*.

Shengyu Zhang, Linfeng Dong, Xiaoya Li, Sen Zhang, Xiaofei Sun, Shuhe Wang, Jiwei Li, Runyi Hu, Tianwei Zhang, Fei Wu, et al. 2023b. Instruction tuning for large language models: A survey. *arXiv preprint arXiv:2308.10792*.

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*.

Hongling Zheng, Li Shen, Anke Tang, Yong Luo, Han Hu, Bo Du, Yonggang Wen, and Dacheng Tao. 2025. Learning from models beyond fine-tuning. *Nature Machine Intelligence*, pages 1–12.

Deyao Zhu, Jun Chen, Xiaoqian Shen, Xiang Li, and Mohamed Elhoseiny. 2024a. Minigpt-4: Enhancing vision-language understanding with advanced large language models. In *The Twelfth International Conference on Learning Representations*.

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

Fei Zhu, Xu-Yao Zhang, Chuang Wang, Fei Yin, and Cheng-Lin Liu. 2021. Prototype augmentation and self-supervision for incremental learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5871–5880.

A Details of the comparison method

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

LwF mitigates catastrophic forgetting by using knowledge distillation loss during training. Specifically, LwF processes current task inputs through both the current and old models, using the old model's outputs as "soft labels" to constrain learning via knowledge distillation loss, preserving past knowledge while training on new tasks.

EWC mitigates catastrophic forgetting by restricting updates to important weights. It computes parameter importance via the Fisher information matrix and penalizes significant changes, preserving knowledge from previous tasks.

L2P introduces a dynamic prompts pool, allowing the model to select and optimize specific prompts during training. Besides, a regularization loss is proposed to encourage task-specific prompt selection, thereby reducing catastrophic forgetting.

O-LoRA enforces an orthogonality loss in parameter space to encourage the current task to optimize in a direction orthogonal to previous tasks, thereby reducing conflicts of different tasks. During inference, it integrates learned knowledge by concatenating the LoRAs of all tasks along the dimension.

MoELoRA transforms a single LoRA into a Mixture-of-Experts (MoE) model with multiple LoRAs based on the number of tasks and trains a router to dynamically assign the appropriate LoRA for each task.

In our experiments, we standardize all methods to the LoRA fine-tuning framework and carefully optimized their hyperparameters to ensure fair and optimal comparisons.

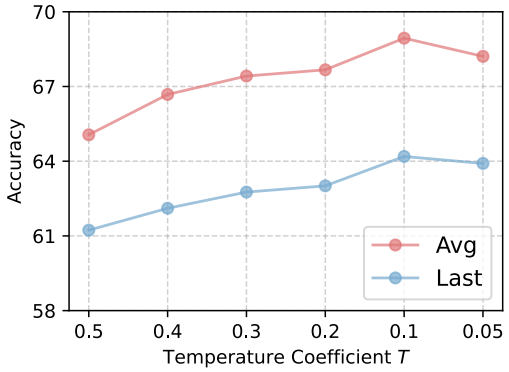


Figure 7: Ablation study of temperature coefficient.

B Details of CKA similarity.

The CKA similarity computation process in Section 4.1 can be summarized as:

(1) Assuming that $X_i \in \mathbb{R}^{n \times p}$ and $Y_i \in \mathbb{R}^{n \times p}$ represent the output features of two different task inputs at layer i, where n is the number of samples and p denotes the dimension of the feature. The first step is to compute the linear kernel matrix: $K_{X_i} = \hat{X}_i \hat{X}_i^T$, $K_{Y_i} = \hat{Y}_i \hat{Y}_i^T$, where $\hat{X}_i, \hat{Y}_i$ denotes the feature after decentralization.

(2) Next, compute the Hilbert-Schmidt Independence Criterion (HSIC) using linear kernel matrices: $\text{HSIC}(K_{X_i}, K_{Y_i}) = \frac{1}{(n-1)^2}\text{tr}(K_{X_i} K_{Y_i})$, where $\text{tr}(\cdot)$ denotes the trace of a matrix.

(3) Finally, calculate the CKA similarity by normalizing the HSIC values: $\text{CKA}(X_i, Y_i) = \frac{\text{HSIC}(K_{X_i}, K_{Y_i})}{\sqrt{\text{HSIC}(K_{X_i}, K_{X_i})\ \text{HSIC}(K_{Y_i}, K_{Y_i})}}$.

The value of CKA similarity ranges from 0 to 1, where a higher value indicates greater similarity between the two feature representations. In our work, we compute and compare the CKA similarity for each layer individually using inputs from different datasets.

C Ablation study of hyper-parameters

We perform ablation experiments on the temperature coefficient T in Eq (4) in this section. Specifically, we evaluate the impact of different temperature coefficient T on the Avg and Last metrics in the UCIT benchmark. As shown in Fig. 7, smaller tem-

perature coefficients lead to sharper distributions of regularization scores d_c, increasing the proportion of LoRA outputs corresponding to the target task and improving overall performance. The chosen value of 0.1 achieves the best overall performance in our experiments.

D Further Experiments

Comparison with Model Tailor. Model Tailor (Zhu et al., 2024b) addresses catastrophic forgetting of generic capabilities by identifying and enhancing critical model parameters during downstream task adaptation. However, unlike our setting, which considers continual learning across multiple tasks, Model Tailor only focuses on mitigating forgetting after single-task fine-tuning. To ensure a fair comparison, we preserve Model Tailor's original methodology while evaluating it on our UCIT benchmark under multi-task continual learning conditions. The comparative results are shown in Table 7. While Model Tailor excels on the first task, its performance declines markedly on subsequent tasks, likely due to its need for repeated parameter recalibration. Our method overcomes this limitation through robust knowledge fusion and expansion, maintaining stable performance across tasks.

	Method	R	A	V	I	C	F	Avg
Avg	Model Tailor	**89.90**	71.30	15.66	39.34	29.50	44.63	48.39
	HiDe-LLaVA	85.70	**92.70**	**54.10**	**66.87**	**59.12**	**55.15**	**68.94**
Last	Model Tailor	**88.37**	69.00	22.79	38.27	29.60	44.63	48.78
	HiDe-LLaVA	80.50	**89.83**	**48.78**	**62.90**	**47.97**	**55.15**	**64.19**

Table 7: Comparison with Model Tailor on the UCIT benchmark. `R`: `ImageNet-R`, `A`: `ArxivQA`, `V`: `VizWiz`, `I`: `Iconqa`, `C`: `CLEVR-Math`, `F`: `Flickr30k`.

Experiment on other MLLM backbone. To further evaluate the scalability of our proposed method, we conduct experiments on different MLLM architectures. Specifically, we adopt InternVL-Chat-7B (Chen et al., 2024b) as our base model, which integrates a 6B-parameter Vision Transformer (ViT) as the visual encoder with a 7B-parameter LLM (*i.e.* Vicuna) via a multimodal projection layer. The experimental results, presented in Fig. 8, demonstrate consistent performance improvements, confirming our method's architecture-agnostic scalability.

E Details of UCIT benchmark

In this paper, we evaluate our method on the CoIN benchmark and the proposed UCIT benchmark. The training and testing scales of these benchmarks, along with the task-specific instruction samples, are detailed in Table 5 and Table 6.

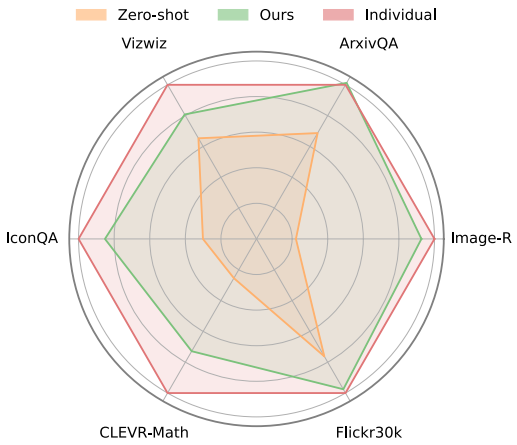


Figure 8: Experiment on InternVL-Chat-7B.

It is important to note that our proposed UCIT benchmark comprises six instruction fine-tuning datasets, selected based on LLaVA's zero-shot performance, which were not included in LLaVA's pre-training phase. This ensures the avoidance of information leakage, a common issue in CoIN. Additionally, we introduce image description tasks to diversify the input instruction types, further enriching the problem formulation.

F Visualization of UCIT benchmark

As shown in Table 8, we present examples from various tasks in the UCIT benchmark, where the input domains of most of these datasets differ significantly from LLaVA's original training data domain. Regarding performance, LLaVA's zero-shot performance on several of our proposed datasets is generally low, making them well-suited as benchmarks for continual instruction tuning tasks. In contrast, the datasets proposed by CoIN exhibit substantial overlap with LLaVA's original training data, which fails to effectively capture the catastrophic forgetting across different methods.

G Details of evaluation

In our benchmark, the output forms of tasks are different, so it is necessary to design appropriate way to calculate accuracy for each task. Specifically, for tasks that answer a single option or word, we use `pred.upper() in ground_truth.upper()` to determine whether the answer is correct. For caption sentences, we adopt metrics commonly used in image caption tasks to measure the accuracy of the response. Specifically, we use `Bleu_1`, `Bleu_2`, `Bleu_3`, `Bleu_4`, `METEOR`, `ROUGE_L`, and `CIDEr` to

Dataset	Train	Test	Task-specific Instruction
ArxivQA	40000	3000	Answer with the option's letter from the given choices directly.
ImageNet-R	23998	3000	Answer the question using a single word or phrase.
VizWiz-cap	40000	3000	Generate a brief caption for the image.
IconQA	29859	3000	Answer with the option's letter from the given choices directly.
CLEVR-Math	40000	3000	Answer the question using a single word or phrase.
Flickr30k	40000	3000	Generate a brief caption for the image.

Table 5: Details of datasets used in UCIT benchmark.

Dataset	Train	Test	Task-specific Instruction
GQA	20000	3000	Answer the question using a single word or phrase.
ImageNet	20000	3000	Answer the question using a single word or phrase.
TextVQA	34602	3000	Answer the question using a single word or phrase.
OCRVQA	20000	3000	Answer the question using a single word or phrase.
REC-COCO	20000	3000	Please provide the bounding box coordinate of the region this sentence describes:
VizWiz	20523	3000	Answer the question using a single word or phrase.
VQAv2	20000	3000	Answer the question using a single word or phrase.
ScienceQA	20000	3000	Answer with the option's letter from the given choices directly.

Table 6: Details of datasets used in CoIN benchmark.

comprehensively evaluate the consistency between the ground truth and the responses. For simplicity, we report the average of the seven metrics.

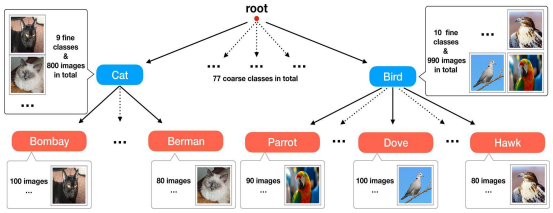
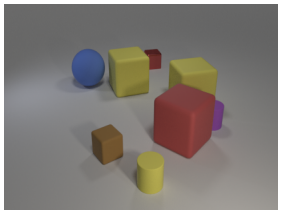
	**Question:** What time is shown? Answer by typing a time word, not a number. It is (_) to ten. **Answer:** quarter
	**Question:** What is the object in the image? answer the question using a single word or phrase. **Answer:** Binoculars
	**Question:** What is happening in the image? Generate a brief caption for the image. **Answer:** A green and white plastic condiment bottle containing Basil leaves.
	**Question:** How many coarse classes are represented in the figure? A) Less than 50 B) Exactly 77 C) More than 100 D) Exactly 99 Answer with the option's letter from the given choices directly. **Answer:** B
	**Question:** Subtract all brown matte objects. Subtract all blue cylinders. How many objects are left? Answer the question using a single word or phrase. **Answer:** quarter
	**Question:** What is happening in the image? Generate a brief caption for the image. **Answer:** A man in a suit walks along s large building.

Table 8: Task data with images across *IconQA*, *ImageNet-R*, *VizWiz-caption*, *ArxivQA*, *CLEVR-Math* and *Flickr30k*.