SHARD: SUBSPACE-CENTRIC STRUCTURED HARMONIZATION AND RELATION DISTILLATION-GUIDED ACTIVATION FOR CONTINUAL LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Multimodal Continual Instruction Tuning (MCIT) allows models to incrementally learn from task streams without retraining. However, existing approaches are hindered by cumulative knowledge conflicts between incremental knowledge from LoRA subspace and pretrained knowledge, and ineffective task-specific subspace activation. To address these challenges, we propose a novel **S**ubspace-centric Structured **H**armonization **A**nd **R**elation **D**istillation-guided Activation (**SHARD**) framework, which consists of Structure-aware Knowledge Harmonization (SKH) and Relation Distillation-guided Subspace Activation (RDSA). Specifically, SKH first identifies critical LLM substructures by learning task-specific structure importance through a continuous relaxation of L0 regularization using hard-concrete distributions. SKH subsequently employs a projection-aware mapping, driven by a semantic outer-product formulation that strictly adheres to the Transformer topology, systematically translating these 1D structural priors into precise 2D parameter-level masks. SKH ultimately utilizes these mapping masks to dynamically modulate the magnitude mask, explicitly guiding new knowledge injection toward less critical pretrained regions within the LoRA subspace for progressive harmonization with pretrained knowledge. Meanwhile, RDSA employs relation distillation to align the distance relations between old prototypes and both new task samples and their own evolving representations. This prevents representation drift, allowing the prototypes to act as stable semantic anchors that accurately route task-specific LoRA subspaces. Extensive experiments reveal the effectiveness and generality of our SHARD.

1 INTRODUCTION

Multimodal Large Language Models (MLLMs) (Chen et al., 2023; Liu et al., 2023) have recently demonstrated remarkable capabilities in multimodal understanding by extending Large Language Models (LLMs) (Touvron et al., 2023; Brown et al., 2020) with visual encoder. Despite their success, continuously emerging tasks and application domains require MLLMs to acquire new knowledge over time, making repeated retraining from scratch computationally impractical. Multimodal Continual Instruction Tuning (MCIT) (Guo et al., 2025a;b; Zhang et al., 2025) addresses this challenge by enabling MLLMs to learn from sequential tasks while preserving previously acquired capabilities.

However, continual adaptation inevitably suffers from catastrophic forgetting (French & Chater, 2002), where optimizing for new tasks progressively degrades the performance on earlier ones (He et al., 2023). To mitigate this, existing MCIT methods can be broadly categorized into two directions. Regularization-based approaches (Liang & Li, 2024; Wang et al., 2023; Li et al., 2026) preserve previously learned knowledge by constraining parameter updates during continual optimization. While regularization-based methods effectively reduce forgetting, their restricted parameter space often limits the acquisition of new knowledge, resulting in inferior continual learning performance. In contrast, dynamic architecture methods (Guo et al., 2025b; Zhang et al., 2025) expand task-specific LoRA (Hu et al., 2022) experts and modulate their outputs via a routing mechanism during inference. Existing routing mechanisms broadly fall into two categories: reconstruction loss-based routing via autoencoders (*e.g.*, MoE-CL (Kang et al., 2026), HiDe-LLaVA (Guo et al., 2025a)) and subspace routing (*e.g.*, DISCO (Guo et al., 2025b)). Given its architectural elegance

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

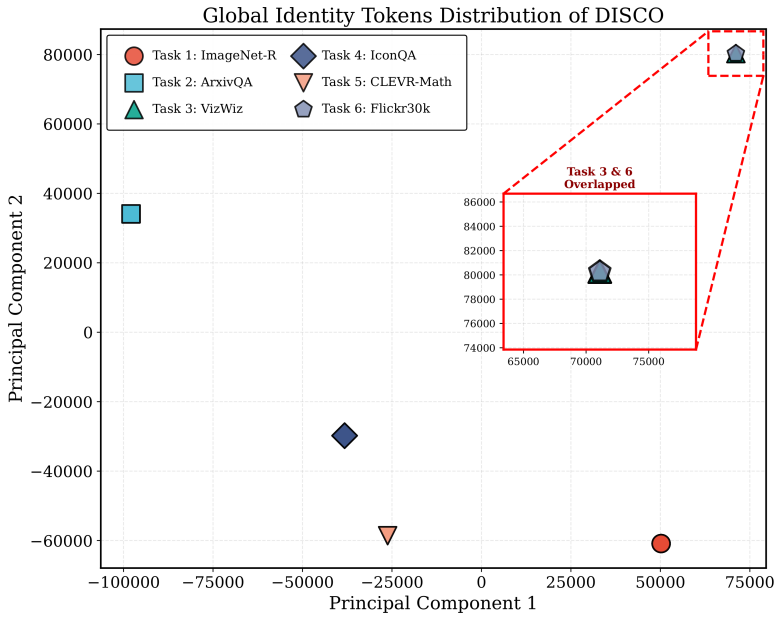

Figure 1: **t-SNE visualization of the global identity token distribution in DISCO.**

and competitive performance without requiring auxiliary autoencoders, we focus our investigation on subspace routing, where task-specific subspaces are merged according to the similarity between test samples and task identity tokens derived from the moving average of text features. However, subspace routing introduces cumulative knowledge conflicts between knowledge acquired from incremental tasks and pretrained knowledge. Moreover, as some tasks share identical instructions, subspace routing causes mismatches between the activated subspace and test samples, as proved in Figure 1.

To fill in these gaps, we propose a novel **S**ubspace-centric Structured **H**armonization **A**nd **R**elation **D**istillation-guided Activation (**SHARD**) framework for MCIT, which comprises of Structure-aware Knowledge Harmonization (SKH) and Relation Distillation-guided Subspace Activation (RDSA). Specifically, SKH first identifies critical LLM substructures (*i.e.*, attention heads, hidden and intermediate dimensions) by learning task-specific structure importance through a continuous relaxation of L0 regularization using hard-concrete distributions. The learned macroscopic structural priors are then systematically translated into microscopic parameter-level structure masks via projection-aware structure-to-parameter mapping, which employs a semantic outer-product formulation strictly adhering to the intrinsic topology of Transformer projections. Meanwhile, RDSA maintains a dynamic task prototype bank stabilized by prototype-sample relation distillation to prevent representation drift as the multimodal projector evolves. During task-agnostic inference, these prototypes serve as compact semantic anchors to accurately activate the corresponding task-specific LoRA subspaces, effectively resolving routing mismatches caused by identical instructions. Extensive experiments on two benchmarks reveal the effectiveness and generality of our SHARD.

Our contributions are summarized below: (*i*) We propose SHARD, a unified MCIT framework that jointly tackles cumulative knowledge conflict and ineffective task-specific subspace activation via subspace-centric structured harmonization during training and relation distillation-guided activation during inference. (*ii*) We devise SKH for the LoRA subspace to guide new knowledge injection toward less critical pretrained regions using learned structure importance, thereby progressively harmonizing it with pretrained knowledge. (*iii*) We propose RDSA, which uses relation distillation to stabilize task prototypes against representation drift, leveraging them as semantic anchors to accurately route LoRA subspaces and resolve mismatches from identical instructions.

2 Related Work

Multimodal Large Language Models (MLLMs) Driven by the exceptional reasoning capabilities of foundational LLMs (Chiang et al., 2023; Brown et al., 2020; Touvron et al., 2023), recent advancements in MLLMs (Chen et al., 2023; Li et al., 2023; Liu et al., 2023; Li et al., 2021; Zhu et al.,

2024; Li et al., 2022) have demonstrated remarkable proficiency in complex vision-language tasks. Beyond the core visual and textual encoders, the cross-modal alignment strategy plays a pivotal role in these architectures. While early formulations (Alayrac et al., 2022; Li et al., 2022; 2023) relied on end-to-end cross-attention mechanisms to capture fine-grained interactions, they often suffered from exorbitant computational overhead. To circumvent this, projection-based paradigm (Chen et al., 2023; Liu et al., 2023) has emerged as the prevailing standard. By employing lightweight bridging layers to map visual tokens directly into the LLM's embedding space, this paradigm achieves high parameter efficiency while preserving the pretrained structural integrity, leading to state-of-the-art performance across diverse downstream applications (Zhao et al., 2025b; Liu et al., 2024; Zhang et al., 2026).

Multimodal Continual Instruction Tuning (MCIT) To equip MLLMs with the capacity to sequentially acquire novel multimodal expertise without the prohibitive cost of retraining, MCIT has emerged as a crucial research frontier (Guo et al., 2025c; Zheng et al., 2025). The primary challenge in this domain is circumventing catastrophic forgetting, which is predominantly addressed through two distinct paradigms. The first, *regularization-based* optimization, enforces orthogonal constraints (Liang & Li, 2024; Wang et al., 2023; Li et al., 2026) to confine new parameter updates within non-interfering subspaces, thereby protecting prior knowledge. The second, *dynamic architecture* expansion (Guo et al., 2025a; Huai et al., 2025; Zhang et al., 2025; Guo et al., 2025b; Kang et al., 2026), incrementally integrates task-specific LoRA modules, with the core differentiation lying in their respective inference routing strategies. For instance, Hide-LLaVA (Guo et al., 2025a) consolidates shallow representations while selectively routing final-layer experts via reconstruction loss computed from autoencoders. Conversely, DISCO (Guo et al., 2025b) leverages Dynamic Knowledge Organization (DKO) to extract global identity tokens, facilitating Subspace Selective Activation (SSA) to achieve a dynamically weighted aggregation of task-specific LoRA subspace. Given its architectural elegance and competitive performance without requiring auxiliary autoencoders, we construct SHARD upon DISCO and replace its global identity token with task prototype, more details regarding DISCO are provided in Appendix C.

3 METHOD

3.1 PROBLEM FORMULATION

Multimodal Continual Instruction Tuning (MCIT) enables an MLLM learns sequential tasks without revisiting previous data and inferences without explicit task ID. Let $\{\mathcal{D}^t\}_{t=1}^{T}$ denote a stream of T instruction-tuning tasks, where the t-th task contains N_t multimodal examples $\mathcal{D}^t = \{(\boldsymbol{x}_{t,j}^{v}, \boldsymbol{x}_{t,j}^{\text{ins}}, \boldsymbol{x}_{t,j}^{\text{ans}})\}_{j=1}^{N_t}$. Here, $\boldsymbol{x}_{t,j}^{v}$ is the visual input, $\boldsymbol{x}_{t,j}^{\text{ins}}$ is the instruction, and $\boldsymbol{x}_{t,j}^{\text{ans}}$ is the answer sequence. Given a pretrained MLLM with frozen backbone parameters $\boldsymbol{\Theta}^0$, the model is adapted to each incoming task by inserting task-specific LoRA modules into a set of language-model projection matrices. For a projection type $a \in \mathcal{A}$ at layer ℓ, where $\mathcal{A} = \{q, k, v, o, \text{gate}, \text{up}, \text{down}\}$, the pretrained weight is denoted by $\boldsymbol{W}_{\ell,a}^{0} \in \mathbb{R}^{d_{\ell,a}^{\text{out}} \times d_{\ell,a}^{\text{in}}}$, and the LoRA update for task t is

$$\Delta \boldsymbol{W}_{\ell,a}^{t} = \boldsymbol{B}_{\ell,a}^{t} \boldsymbol{A}_{\ell,a}^{t}, \quad \boldsymbol{A}_{\ell,a}^{t} \in \mathbb{R}^{r \times d_{\ell,a}^{\text{in}}}, \quad \boldsymbol{B}_{\ell,a}^{t} \in \mathbb{R}^{d_{\ell,a}^{\text{out}} \times r}, \tag{1}$$

where r is the LoRA rank. The task objective is the standard autoregressive next-token prediction loss:

$$\mathcal{L}_{\text{task}}^{t} = -\sum_{j=1}^{N_t} \sum_{i=1}^{|\boldsymbol{x}_{t,j}^{\text{ans}}|} \log p_{\boldsymbol{\Theta}^0, \Delta\boldsymbol{\Theta}^t}\left(x_{t,j,i}^{\text{ans}} \mid \boldsymbol{x}_{t,j}^{v}, \boldsymbol{x}_{t,j}^{\text{ins}}, \boldsymbol{x}_{t,j,<i}^{\text{ans}}\right), \tag{2}$$

where $\Delta\boldsymbol{\Theta}^t = \{\Delta \boldsymbol{W}_{\ell,a}^{t}\}_{\ell,a}$ denotes all LoRA updates introduced for task t.

3.2 OVERVIEW

SHARD is a unified MCIT framework that jointly tackles cumulative knowledge conflict and ineffective task-specific subspace activation via SKH and RDSA. SKH operates through a two-stage training paradigm. The first stage learns the structure importance of task-sensitive structures, while the second stage leverages this importance to facilitate knowledge harmonization between incremental and pretrained knowledge. Meanwhile, RDSA adopts relation distillation to stabilize task prototypes against representation drift, utilizing them as semantic anchors to accurately route task-specific LoRA subspaces and resolve mismatches from identical instructions.

3.3 STRUCTURE-AWARE KNOWLEDGE HARMONIZATION

Structure importance learning To identify which LLM substructures are critical for a specific task, SKH learns task-specific structure importance. Because directly optimizing discrete binary masks (*i.e.*, L0 regularization) is non-differentiable and incompatible with gradient descent, we employ a continuous relaxation using hard-concrete distributions. For each structural category $c \in \mathcal{C}$, we introduce learnable structure logits $\boldsymbol{\zeta}^{t,c}$ and sample a stochastic gate $\boldsymbol{z}^{t,c}$ via the reparameterization trick. For an element u in category c, the relaxed gate is sampled as:

$$\tilde{z}_u^{t,c} = \sigma\left(\frac{\log \epsilon_u - \log(1-\epsilon_u) + \zeta_u^{t,c}}{\tau_{\text{hc}}}\right), \epsilon_u \sim \text{Uniform}(0,1) \tag{3}$$

where σ denotes the sigmoid activation function, and τ_{hc} is the hard-concrete temperature controlling the sharpness of the distribution. To achieve exact structural sparsity, the sampled relaxed gate $\tilde{z}_u^{t,c}$ is subsequently stretched and clipped to obtain the final gate $z_u^{t,c} \in [0,1]$. During the L0 stage, these stochastic gates are multiplied with their corresponding LLM substructure, including attention heads, hidden dimensions, and FFN intermediate dimensions. Consequently, if removing a specific substructure drastically increases the task loss, gradient descent naturally pushes its logit toward a higher structure logits; conversely, the gates for redundant substructures are driven toward lower structure logits.

The L0 stage optimizes the structure logits with a sparsity-constrained objective. Let $\widehat{\rho}^t$ denote the expected sparsity induced by all sampled gates and ρ_{tar} denote the target sparsity. On the L0 1% subset $\mathcal{D}_{\text{L0}}^t$, we optimize:

$$\begin{aligned}\mathcal{L}_{\text{L0}}^t = \mathbb{E}_{\boldsymbol{z}^t \sim q(\boldsymbol{z}^t;\boldsymbol{\zeta}^t)}\left[\mathcal{L}_{\text{task}}^t\big(\boldsymbol{\Theta}^0; \boldsymbol{z}^t, \mathcal{D}_{\text{L0}}^t\big)\right] \\ + \lambda_1^t(\widehat{\rho}^t - \rho_{\text{tar}}) + \lambda_2^t(\widehat{\rho}^t - \rho_{\text{tar}})^2.\end{aligned} \tag{4}$$

where $q(\boldsymbol{z}^t;\boldsymbol{\zeta}^t)$ is the hard-concrete distribution, and λ_1^t and λ_2^t are Lagrangian coefficients. After optimization, the raw structure logits are converted into structure importance:

$$\boldsymbol{p}^{t,c} = \sigma\left(\frac{\mu_{\text{hc}}}{\tau_{\text{hc}}}\boldsymbol{\zeta}^{t,c}\right), \quad c \in \mathcal{C}, \tag{5}$$

where μ_{hc} is a scaling constant. The resulting $\{\boldsymbol{p}^{t,c}\}_{c\in\mathcal{C}}$ provides a task-specific structure importance and is saved independently for each task.

Projection-aware structure-to-parameter mapping After obtaining the 1D structure importance, they must be systematically mapped to 2D matrices, as LoRA operates on full projection weights rather than isolated architectural units. Therefore, SKH constructs a projection-aware structure mask $\boldsymbol{P}_{\ell,a}^t \in [0,1]^{d_{\ell,a}^{\text{out}} \times d_{\ell,a}^{\text{in}}}$ for every layer ℓ and projection type a. Specifically, we define an output-aligned structure importance $\boldsymbol{r}_{\ell,a}^{t,\text{out}} \in [0,1]^{d_{\ell,a}^{\text{out}}}$ and an input-aligned structure importance $\boldsymbol{r}_{\ell,a}^{t,\text{in}} \in [0,1]^{d_{\ell,a}^{\text{in}}}$, formulating the matrix via an outer product:

$$\boldsymbol{P}_{\ell,a}^t = p_\ell^{t,\text{c}} \cdot \boldsymbol{r}_{\ell,a}^{t,\text{out}}\big(\boldsymbol{r}_{\ell,a}^{t,\text{in}}\big)^\top. \tag{6}$$

This outer-product formulation operates as a probabilistic intersection: a specific entry within $\boldsymbol{W}_{\ell,a}^0$ is deemed critical if and only if both its associated input and output dimensions are critical for the task.

These dimension-aligned structure importance are strictly assigned based on the mathematical semantics and topology of each Transformer projection. For query, key, and value projections, the rows correspond to attention-head output blocks and the columns correspond to hidden input dimensions:

$$\begin{aligned}\boldsymbol{r}_{\ell,a}^{t,\text{out}} &= \text{Expand}_{\text{head}}\big(\boldsymbol{p}_\ell^{t,\text{head}}\big), \\ \boldsymbol{r}_{\ell,a}^{t,\text{in}} &= \boldsymbol{p}^{t,\text{hidden}}, \quad a \in \{q,k,v\}.\end{aligned} \tag{7}$$

For the output projection, the rows correspond to hidden output dimensions and the columns correspond to attention-head input blocks:

$$\boldsymbol{r}_{\ell,o}^{t,\text{out}} = \boldsymbol{p}^{t,\text{hidden}}, \quad \boldsymbol{r}_{\ell,o}^{t,\text{in}} = \text{Expand}_{\text{head}}\big(\boldsymbol{p}_\ell^{t,\text{head}}\big). \tag{8}$$

For FFN projections, the gate and up projections map hidden dimensions to intermediate dimensions, while the down projection maps intermediate dimensions back to hidden dimensions:

$$\boldsymbol{r}_{\ell,a}^{t,\text{out}} = \begin{cases} \boldsymbol{p}_{\ell}^{t,\text{inter}}, & a \in \{\text{gate}, \text{up}\}, \\ \boldsymbol{p}^{t,\text{hidden}}, & a = \text{down}, \end{cases}$$
$$\boldsymbol{r}_{\ell,a}^{t,\text{in}} = \begin{cases} \boldsymbol{p}^{t,\text{hidden}}, & a \in \{\text{gate}, \text{up}\}, \\ \boldsymbol{p}_{\ell}^{t,\text{inter}}, & a = \text{down}. \end{cases} \tag{9}$$

In summary, this projection-aware mapping mechanism functions as a semantic bridge. By strictly adhering to the intrinsic projection topology of the Transformer, it systematically translates macroscopic structure importance into mapping mask. This circumvents the coarse, shape-inconsistent broadcasting of structural priors, providing a precisely aligned spatial matrix to seamlessly calibrate the knowledge harmonization in the subsequent phase.

Harmonization regularization term Motivated by prior works (Frankle & Carbin, 2018; Han et al., 2015; Sun et al., 2024) suggesting that parameter magnitude can serve as a proxy for importance, we leverage this property to address the cumulative knowledge conflict. Let $\overline{\boldsymbol{W}}_{\ell,a}^{0} = \boldsymbol{W}_{\ell,a}^{0}/\|\boldsymbol{W}_{\ell,a}^{0}\|_F$ be the normalized pretrained matrix. We first compute an element-wise magnitude score:

$$G_{\ell,a,ij} = \left| \frac{1}{\log\left(|\overline{W}_{\ell,a,ij}^{0}| + \epsilon_{\text{mag}}\right)} \right|, \tag{10}$$

where ϵ_{mag} is a numerical constant. The soft magnitude mask is then

$$M_{\ell,a,ij}^{\text{mag}} = \tanh(\omega_{\text{mag}} G_{\ell,a,ij}), \tag{11}$$

where $\omega_{\text{mag}} > 0$ controls the steepness of the transition. Larger values of $M_{\ell,a,ij}^{\text{mag}}$ indicate that perturbing the corresponding pretrained entry is more likely to disrupt pretrained knowledge. However, Equation 11 only considers local importance, overlooking the macroscopic disparities in the importance of pre-trained weights across different projection types and layers within the LLM.

Therefore, we propose Sructure-aware Knowledge Harmonization (SKH) to utilize structure importance learned in L0 stage to dynamically modulate the magnitude mask. SKH does not directly perform element-wise multiplication between the mapping mask and the soft magnitude mask, an approach empirically shown to yield poor performance in Appendix Table 10, because the two quantities possess distinct statistical properties. Instead, SKH first converts structure importance $\boldsymbol{P}_{\ell,a}^{t}$ into a centered log-odds modulation matrix:

$$H_{\ell,a,ij}^{t} = \text{clip}\left(\log \frac{\widetilde{P}_{\ell,a,ij}^{t}}{1 - \widetilde{P}_{\ell,a,ij}^{t}}, -c_{\log}, c_{\log}\right),$$
$$\widetilde{P}_{\ell,a,ij}^{t} = \text{clip}\left(P_{\ell,a,ij}^{t}, \epsilon_{\text{str}}, 1 - \epsilon_{\text{str}}\right), \tag{12}$$

where ϵ_{str} prevents degenerate probabilities and $c_{\log}$ bounds extreme log-odds values. We center $H_{\ell,a}^{t}$ by subtracting its matrix mean and obtain a mean-one structural scale:

$$R_{\ell,a,ij}^{t} = \frac{\exp\left(H_{\ell,a,ij}^{t} - \overline{H}_{\ell,a}^{t}\right)}{\frac{1}{d_{\ell,a}^{\text{out}} d_{\ell,a}^{\text{in}}} \sum_{i',j'} \exp\left(H_{\ell,a,i'j'}^{t} - \overline{H}_{\ell,a}^{t}\right)}, \tag{13}$$

where $\overline{H}_{\ell,a}^{t}$ is the average of $H_{\ell,a}^{t}$.

The final structure-calibrated SKH mask is obtained by applying the structural scale to the slope of magnitude score:

$$\widehat{M}_{\ell,a,ij}^{t} = \tanh\left(\omega_{\text{mag}} R_{\ell,a,ij}^{t} G_{\ell,a,ij}\right). \tag{14}$$

To keep the overall regularization strength comparable to magnitude mask, SKH normalizes the mean of the calibrated mask to the mean of the original magnitude mask:

$$M_{\ell,a,ij}^{t} = \widehat{M}_{\ell,a,ij}^{t} \cdot \frac{\overline{M}_{\ell,a}^{\text{mag}}}{\overline{\widehat{M}}_{\ell,a}^{t} + \epsilon_{\text{norm}}}, \tag{15}$$

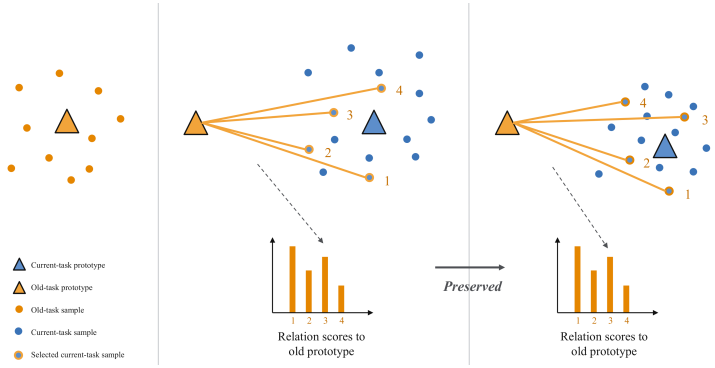

Figure 2: **Relation distillation within RDSA for task prototypes.** For each old task prototype, RDSA aligns the distance relations be- tween old prototypes and both new task samples and their own evolving representations in the mini-batch.

where $\overline{M}_{\ell,a}^{\mathrm{mag}}$ and $\overline{\widetilde{M}}_{\ell,a}^{t}$ denote matrix means, and ϵ_{norm} prevents division by zero. This formulation has two advantages: task-specific structures modulate the relative allocation of preservation strength, while pretrained weight magnitudes remain the primary parameter-level signal.

The structure-calibrated SKH mask regularizes LoRA updates by penalizing their overlap with protected pretrained regions. For task t, the SKH loss is defined as

$$\mathcal{L}_{\mathrm{SKH}}^{t} = \sum_{\ell=1}^{L} \sum_{a \in \mathcal{A}} \left\| \boldsymbol{M}_{\ell,a}^{t} \odot \Delta \boldsymbol{W}_{\ell,a}^{t} \right\|_{F}, \tag{16}$$

where $\odot$ denotes element-wise multiplication.

3.4 Relation Distillation-guided Subspace Activation

Task prototype learning and relation distillation DISCO demonstrates through its ablation studies that text features outperform image features regarding subspace routing. However, identical instructions may produce the same text features, as illustrated in Figure 1. Therefore, we adopt task prototype learning in the image modality. Meanwhile, our preliminary experiments (Table 5 in the Appendix) show that projected image features after the multimodal projector consistently outperform frozen image features, as they evolve throughout task training. Let $\phi^{t}(\cdot)$ denote the multimodal projector after the model has been adapted to task t, and let $\boldsymbol{v}_{t,j}^{\mathrm{cls}}$ be the visual CLS feature extracted by the frozen vision encoder for sample j in task t. We represent the projected image feature as

$$\boldsymbol{e}_{t,j}^{t} = \phi^{t}\left(\boldsymbol{v}_{t,j}^{\mathrm{cls}}\right) \in \mathbb{R}^{d_{\mathrm{p}}}, \tag{17}$$

where d_{p} is the prototype dimension. RDSA maintains a prototype bank $\boldsymbol{U}^{t} = \{\boldsymbol{u}_{m}^{t}\}_{m=1}^{t}$, where $\boldsymbol{u}_{m}^{t} \in \mathbb{R}^{d_{\mathrm{p}}}$ is the task prototype for task m after learning task t. At the beginning of a new task, the current prototype is random initialization for task prototype learning.

The prototype learning objective learns a representative anchor for the current task following works (Zhang et al., 2025), as shown below.

$$\mathcal{L}_{\mathrm{proto}}^{t} = \frac{1}{B} \sum_{j=1}^{B} \left(1 - \cos\left(\boldsymbol{e}_{t,j}^{t}, \boldsymbol{u}_{t}^{t}\right)\right) \tag{18}$$

where B denotes the batch size.

Given that projected image representations drift as the multimodal projector evolves during continual learning, we propose Relation Distillation-guided Subspace Activation (RDSA) for MCIT, which preserves the distance relationships between historical prototypes and both new-task samples and their evolving representations. As a result, the prototypes serve as stable semantic anchors, enabling accurate routing of task-specific LoRA subspaces, as proved in Table 4 of Appendix. When task t starts, RDSA stores the previous projector ϕ^{t-1} and the previous prototype bank $\boldsymbol{U}^{t-1}$. For a current-task sample, the teacher feature is $\boldsymbol{e}_{t,j}^{t-1} = \phi^{t-1}(\boldsymbol{v}_{t,j}^{\mathrm{cls}})$, and the student feature is $\boldsymbol{e}_{t,j}^{t} = \phi^{t}(\boldsymbol{v}_{t,j}^{\mathrm{cls}})$. For old tasks $m < t$, we define normalized prototype directions $\widehat{\boldsymbol{u}}_{m}^{t} = \boldsymbol{u}_{m}^{t} / \|\boldsymbol{u}_{m}^{t}\|_{2}$ and

$\widehat{\boldsymbol{u}}_m^{t-1} = \boldsymbol{u}_m^{t-1}/\|\boldsymbol{u}_m^{t-1}\|_2$. The student and teacher relation score vectors are

$$\begin{aligned} \boldsymbol{s}_j^{\mathrm{stu}} &= \left[\left(\boldsymbol{e}_{t,j}^{t}\right)^\top \widehat{\boldsymbol{u}}_1^{t}, \ldots, \left(\boldsymbol{e}_{t,j}^{t}\right)^\top \widehat{\boldsymbol{u}}_{t-1}^{t} \right], \\ \boldsymbol{s}_j^{\mathrm{tea}} &= \left[\left(\boldsymbol{e}_{t,j}^{t-1}\right)^\top \widehat{\boldsymbol{u}}_1^{t-1}, \ldots, \left(\boldsymbol{e}_{t,j}^{t-1}\right)^\top \widehat{\boldsymbol{u}}_{t-1}^{t-1} \right]. \end{aligned} \tag{19}$$

The relation distillation loss matches the old-task relation distribution induced by the current projector to that induced by the previous projector:

$$\mathcal{L}_{\mathrm{rel}}^{t} = \frac{\tau_{\mathrm{rel}}^2}{B} \sum_{j=1}^{B} \mathrm{KL}\left(\mathrm{softmax}\left(\boldsymbol{s}_j^{\mathrm{tea}}/\tau_{\mathrm{rel}}\right) \,\middle\|\, \mathrm{softmax}\left(\boldsymbol{s}_j^{\mathrm{stu}}/\tau_{\mathrm{rel}}\right) \right), \tag{20}$$

where τ_{rel} is the distillation temperature. When there is only one valid old task, we replace the KL term with a mean-squared relation matching loss to avoid a degenerate one-class softmax. This distillation step prevents old prototypes from becoming incompatible with the evolving projector, thereby preserving task identification signals without storing old samples.

During inference, the prototype bank produces a task-affinity distribution for selecting or weighting task-wise LoRA experts. Given a test sample with projected image feature $\boldsymbol{e}_*$, RDSA computes

$$\boldsymbol{\pi}_* = \mathrm{softmax}\left(\frac{[\cos(\boldsymbol{e}_*, \boldsymbol{u}_1^T), \ldots, \cos(\boldsymbol{e}_*, \boldsymbol{u}_T^T)]}{\tau_{\mathrm{route}}} \right), \tag{21}$$

where τ_{route} controls routing sharpness. The resulting distribution provides a lightweight task identification signal that complements the structural and parameter-level regularization mechanisms. For a detailed explanation of how the task identification signal facilitates subspace routing, please refer to the description of Subspace Selective Activation in the Appendix.

Complete training Objective The complete training objective combines task adaptation, structure-calibrated SKH, and task prototype learning:

$$\mathcal{L}_{\mathrm{SKH}}^{t} = \mathcal{L}_{\mathrm{task}}^{t} + \lambda_{\mathrm{SKH}} \mathcal{L}_{\mathrm{SKH}}^{t} + \lambda_{\mathrm{sub}} \mathcal{L}_{\mathrm{sub}}^{t} \tag{22}$$

After training task t, the task-specific LoRA parameters, and the prototype bank $\boldsymbol{U}^t$ are stored in the corresponding task checkpoint like DISCO. When task $t+1$ arrives, a new L0 stage is performed on $\mathcal{D}_{\mathrm{L0}}^{t+1}$, producing a fresh task-specific structure importance learning rather than reusing the prior from earlier tasks.

4 Experiments

4.1 Experimental Setup

Datasets. To comprehensively evaluate the continual learning dynamics of our framework, we conduct experiments across two representative MCIT benchmarks: UCIT (Guo et al., 2025a) and MLLM-DCL (Zhao et al., 2025a). Specifically, UCIT is designed to rigorously test various real-world adaptation. It comprises six datasets—ImageNet-R (Hendrycks et al., 2021), ArxivQA (Li et al., 2024), VizWiz (Gurari et al., 2018), IconQA (Lu et al., 2021), CLEVR-Math (Lindström & Abraham, 2022), and Flickr30k (Plummer et al., 2015)—that exhibit negligible overlap with the original pre-training corpora, thereby preventing models from exploiting prior knowledge (as evidenced by the low zero-shot baselines in Table 2). Conversely, MLLM-DCL is employed to assess the model's capacity for incrementally assimilating domain-specific expertise, which consists of RSVQA (Lobry et al., 2020), PathVQA (He et al., 2020), DriveLM (Sima et al., 2024), StockQA (Zhao et al., 2025a), AI2D (Kembhavi et al., 2016), SciVerse (Guo et al., 2025e), MapQA (Chang et al., 2022), and TQA (Kembhavi et al., 2017). Further dataset statistics and prompt templates are detailed in Appendix B.1.

Evaluation protocol. In alignment with the standardized continual evaluation framework established by SEFE (Chen et al., 2025), we quantify the trade-off between plasticity and stability using four primary metrics. These include Mean Finetune Accuracy (MFT) for initial task mastery, Mean Final Accuracy (MFN) for absolute knowledge retention, Mean Average Accuracy (MAA) for the overall training trajectory, and Backward Transfer (BWT) to explicitly gauge catastrophic forgetting. Formally, they are defined as: 1) MFT $= \frac{1}{T}\sum_{i=1}^{T} A_{i,i}$, where $A_{i,i}$ denotes the performance on the i-th task immediately after its corresponding training phase. 2) MFN $= \frac{1}{T}\sum_{i=1}^{T} A_{T,i}$, measuring the accuracy on the i-th task after the entire sequence of T tasks is completed. 3)

Table 1: Comparison of different methods on the MLLM-DCL benchmark based on LLaVA-1.5. The best and second-best results are highlighted in **bold** and underlined, respectively.

Method	Venue	RS	Med	AD	Sci	Fin	MFT (↑)	MFN (↑)	MAA (↑)	BWT (↑)
Zero-shot	-	32.29	28.28	15.59	35.55	62.56	34.85	-	-	-
Individual	-	78.15	58.20	**52.77**	49.32	88.02	65.29	-	-	-
LoRA-FT	ICLR-22	69.65	41.59	25.43	40.88	87.45	64.98	53.00	61.13	-14.97
O-LoRA	EMNLP-23	74.64	44.42	30.02	41.47	87.15	65.16	55.54	62.12	-12.53
MoELoRA	NeurIPS-24	77.54	41.85	27.62	40.13	86.75	64.94	54.78	61.76	-12.71
ModalPrompt	EMNLP-25	53.63	45.68	40.77	41.81	87.82	53.87	53.94	49.67	**0.09**
ModalPrompt*	EMNLP-25	78.67	51.37	47.80	43.29	87.78	61.84	61.78	64.05	-0.07
CL-MoE	CVPR-25	71.34	46.84	26.33	41.17	88.74	66.06	54.88	61.79	-13.97
HiDE	ACL-25	74.31	48.95	33.21	38.54	81.55	60.77	55.31	60.68	-6.82
SEFE	ICML-25	77.26	50.37	37.21	40.87	86.82	65.01	58.51	63.63	-8.13
DGG	CVPR-26	57.76	42.24	30.27	37.53	**89.79**	66.57	51.52	57.92	-18.81
MoE-CL	WWW-26	70.69	43.11	39.36	42.24	89.73	65.27	57.03	63.39	-10.31
DISCO	ICCV-25	76.49	44.48	44.84	46.61	89.22	64.78	60.33	63.93	-5.57
Ours	-	**79.39**	**58.68**	50.44	**50.30**	89.70	**67.21**	**65.70**	**68.14**	-1.89

Table 2: Comparison of different methods on the UCIT benchmark based on LLaVA-1.5. The best and second-best results are highlighted in **bold** and underlined, respectively.

Method	Venue	ImgNet-R	ArxivQA	VizWiz	IconQA	CLEVR	Flickr30k	MFT (↑)	MFN (↑)	MAA (↑)	BWT (↑)
Zero-shot	-	16.27	53.73	38.39	19.20	20.63	41.88	31.68	-	-	-
Individual	-	**91.67**	90.83	**57.87**	**78.43**	**76.63**	**61.72**	76.19	-	-	-
LoRA-FT	ICLR-22	58.03	77.63	44.39	67.40	61.77	58.22	**76.89**	61.24	76.55	-18.78
O-LoRA	EMNLP-23	77.50	78.07	44.50	63.13	64.73	58.16	76.01	64.35	78.02	-13.99
MoELoRA	NeurIPS-24	70.07	77.70	44.69	50.03	54.03	57.34	71.17	58.98	75.08	-14.63
ModalPrompt	EMNLP-25	51.07	87.27	48.11	39.23	46.57	42.93	52.65	52.53	57.67	-0.15
ModalPrompt*	EMNLP-25	74.43	92.00	55.92	44.27	53.97	43.67	60.76	60.71	70.57	**-0.05**
CL-MoE	CVPR-25	66.33	77.00	44.78	51.87	53.53	57.42	71.46	58.49	74.19	-15.56
HiDE	ACL-25	84.03	90.73	44.43	58.93	41.37	54.25	69.96	62.29	77.32	-9.20
SEFE	ICML-25	80.83	78.00	47.01	69.63	65.83	57.92	75.98	66.54	78.76	-11.33
DGG	CVPR-26	86.33	76.57	43.20	57.37	42.33	55.17	72.40	60.16	76.66	-14.68
MoE-CL	WWW-26	84.67	94.17	46.04	71.20	56.37	56.27	75.56	68.12	80.75	-8.93
DISCO	ICCV-25	87.70	94.93	46.71	69.50	49.13	55.93	74.71	67.31	81.00	-8.87
Ours	-	87.23	**95.20**	50.15	72.40	53.50	56.51	75.18	**69.17**	**81.39**	-7.22

MAA $= \frac{1}{T}\sum_{i=1}^{T}\left(\frac{1}{i}\sum_{k=1}^{i}A_{i,k}\right)$, representing the accumulated average performance up to the i-th training stage. 4) BWT $= \frac{1}{T}\sum_{i=1}^{T}(A_{T,i} - A_{i,i})$, indicating the performance deviation on previous tasks. A detailed presentation of these metrics is provided in Appendix B.2.

Implementation details. Our framework is built upon the LLaVA-v1.5-7B architecture (Liu et al., 2023), utilizing CLIP-L/14-336 (Radford et al., 2021) as the foundational vision-language encoder. To facilitate task-specific capacity expansion akin to dynamic architecture paradigms (Guo et al., 2025a), we inject Low-Rank Adaptation (LoRA) modules (rank 16) into all projection layers of the LLM for each incoming task. The structural penalty coefficients in Equation 22 are empirically set to $\lambda_{\text{SKH}} = 0.001$ and $\omega_{\text{mag}} = 0.2$. During training, we apply a universal batch size of 16 across all tasks on NVIDIA H20 GPUs. The training schedule spans 1 epoch per task for UCIT with a learning rate of 2×10^{-4}, whereas MLLM-DCL follows a customized schedule of $\{1, 3, 1, 2, 1\}$ epochs with a finer learning rate of 2×10^{-5}. Comprehensive experimental configurations can be found in Appendix B.3.

4.2 Main Results

To rigorously validate the efficacy of SHARD, we conduct an extensive comparative analysis against a diverse spectrum of state-of-the-art methods on MLLM-DCL and UCIT. Specifically, the selected competitors encompass O-LoRA (Wang et al., 2023), DGG (Li et al., 2026), MoELoRA (Chen et al., 2024a), CL-MoE (Huai et al., 2025), ModalPrompt (Zeng et al., 2025), HiDe-LLaVA (Guo et al., 2025a), SEFE (Chen et al., 2025), DISCO (Guo et al., 2025b) and MoE-CL (Kang et al., 2026). As illustrated in Tables 1 and 2, SHARD exhibits exceptional performance on standard benchmarks. On MLLM-DCL, SHARD attains the highest MFT of **67.21**%, MFN of **65.70**%, and MAA of **68.14**%, confirming the efficacy of our framework. Significantly, SHARD yields a competitive BWT of **-1.89**, which, when considered alongside the MFT metric, indicates that our approach best mitigates catastrophic forgetting. Likewise, on UCIT, SHARD surpasses DISCO across four critical evaluation metrics. Supported by its outstanding MAA of **81.39**% and MFN of **69.17**%, SHARD displays the most balanced and resilient adaptation trajectory throughout all tasks. This excellent overall capability, paired with a highly competitive BWT of **-7.22** among the evaluated techniques, firmly confirms that SHARD prevents overfitting to particular distributions and offers a

more generalized solution for MCIT. It is worth mentioning that SHARD exhibits robust adaptability across various MLLM architectures, as elaborated in Appendix A.1.

4.3 Ablation Study

Effectiveness of each component in SHARD. Given the shorter training and evaluation time of UCIT, all ablation studies in this section are conducted on this benchmark. As reported in Appendix Table 7, removing SKH leads to a consistent decline across all evaluation metrics, with MFN decreasing by **1.23** and BWT dropping by **1.06**. These results highlight the importance of harmonizing incremental knowledge with pretrained knowledge, which effectively mitigates performance degradation throughout MCIT. Similarly, excluding RDSA reduces MFN to **68.46** and BWT to **-7.91**. Together with the subspace routing analysis in Table 4 of Appendix, these findings suggest that the performance degradation partially stems from incorrect task-specific subspace activation caused by mismatched routing during task-agnostic inference. Finally, combining SKH and RDSA achieves the best overall performance, confirming that the two modules operate synergistically to improve the learning and activation of task-specific subspaces in MCIT.

Ablation study of the SKH module. To identify the most suitable regularization strength for SKH, we evaluate different values of λ_{SKH} in Equation 22, as summarized in Appendix Table 8. We observe that $\lambda_{\mathrm{SKH}} = \mathbf{0.001}$ consistently achieves the best overall performance. Accordingly, we adopt $\lambda_{\mathrm{SKH}} = 0.001$ as the default setting in all experiments.

To thoroughly investigate the effect of structure importance on the progressive harmonization regularization, we evaluate different values of ω_{mag} in Equation 14, with the results summarized in Appendix Table 9. The results show that $\omega_{\mathrm{mag}} = 0.2$ consistently achieves the best performance across nearly all evaluation metrics. Therefore, we adopt $\omega_{\mathrm{mag}} = 0.2$ as the default setting, providing the most appropriate strength for incorporating structure importance into harmonization regularization term.

To elucidate the mechanism of projecting structure importance onto the parameter level, we introduce a naive variant for comparison, which simply broadcasts the structure importance to match the shape of the subspace and performs element-wise multiplication with magnitude mask. As shown in Appendix Table 10, direct multiplication triggers severe catastrophic forgetting, resulting in a notable decrease of **2.63** in MFN and **1.83** in BWT. We conclude that direct multiplication overlooks the significant variance in the relationships between subspaces at different projection locations and the underlying LLM substructures. This finding further highlights the elegance and necessity of our projection-aware structure-to-parameter mapping strategy.

Ablation study of the RDSA module. To reveal the effectiveness of relation distillation, we remove corresponding loss from RDSA. As shown in Appendix Table 11, the removal of relation distillation bings a drop regarding MFN and BWT. We attribute this phenomenon to the continual evolution of the projected image representations as new tasks are learned, which gradually renders historical prototypes outdated for the updated multimodal representation space. Analysis of λ_{sub} and comparison with directly learn prototype using frozen image features is provided in Appendix.

5 Conclusion

In this paper, we present Subspace-centric Structured Harmonization And Relation Distillation-guided Activation (SHARD), a unified framework for Multimodal Continual Instruction Tuning (MCIT) that addresses cumulative knowledge conflicts and ineffective task-specific subspace activation. Specifically, Structure-aware Knowledge Harmonization (SKH) leverages learned structure importance together with a projection-aware mapping mechanism to harmonize incremental adaptation with pretrained knowledge in the LoRA subspace. Meanwhile, Relation Distillation-guided Subspace Activation (RDSA) employs relation distillation to stabilize task prototypes against representation drift, enabling reliable task-specific LoRA subspace routing during inference. Extensive experiments demonstrate that SHARD consistently outperforms existing MCIT methods on different Multimodal Large Language Models (MLLMs). In future work, we plan to extend SHARD to more challenging continual multimodal learning scenarios with stronger generalization and scalability.

6 AI USE STATEMENT

During the preparation of this manuscript, LLMs were used solely for language polishing, including improving grammar, clarity, and readability. LLMs were not involved in the formulation of research ideas, methodological innovations, experimental design, implementation, or interpretation and analysis of experimental results. All research contributions, experiments, and experimental analyses were conducted and verified by the authors.

REFERENCES

Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, et al. Flamingo: a visual language model for few-shot learning. *Advances in neural information processing systems*, 35:23716–23736, 2022.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

Shuaichen Chang, David Palzer, Jialin Li, Eric Fosler-Lussier, and Ningchuan Xiao. Mapqa: A dataset for question answering on choropleth maps. *arXiv preprint arXiv:2211.08545*, 2022.

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

Jinpeng Chen, Runmin Cong, Yuzhi Zhao, Hongzheng Yang, Guangneng Hu, Horace Ip, and Sam Kwong. Sefe: Superficial and essential forgetting eliminator for multimodal continual instruction tuning. In *International Conference on Machine Learning*, pp. 7982–8001. PMLR, 2025.

Jun Chen, Deyao Zhu, Xiaoqian Shen, Xiang Li, Zechun Liu, Pengchuan Zhang, Raghuraman Krishnamoorthi, Vikas Chandra, Yunyang Xiong, and Mohamed Elhoseiny. Minigpt-v2: large language model as a unified interface for vision-language multi-task learning (2023). *URL https://arxiv.org/abs/2310.09478*, 18, 2023.

Zhe Chen, Jiannan Wu, Wenhai Wang, Weijie Su, Guo Chen, Sen Xing, Muyan Zhong, Qinglong Zhang, Xizhou Zhu, Lewei Lu, et al. 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*, pp. 24185–24198, 2024b.

Wei-Lin Chiang, Zhuohan Li, Ziqing Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E Gonzalez, et al. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality. *See https://vicuna. lmsys. org (accessed 14 April 2023)*, 2(3):6, 2023.

Jonathan Frankle and Michael Carbin. The lottery ticket hypothesis: Finding sparse, trainable neural networks. *arXiv preprint arXiv:1803.03635*, 2018.

Robert M French and Nick Chater. Using noise to compute error surfaces in connectionist networks: A novel means of reducing catastrophic forgetting. *Neural computation*, 14(7):1755–1769, 2002.

Haiyang Guo, Fanhu Zeng, Ziwei Xiang, Fei Zhu, Da-Han Wang, Xu-Yao Zhang, and Cheng-Lin Liu. Hide-llava: Hierarchical decoupling for continual instruction tuning of multimodal large language model. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 13572–13586, 2025a.

Haiyang Guo, Fanhu Zeng, Fei Zhu, Wenzhuo Liu, Da-Han Wang, Jian Xu, Xu-Yao Zhang, and Cheng-Lin Liu. Federated continual instruction tuning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 1325–1335, 2025b.

Haiyang Guo, Fanhu Zeng, Fei Zhu, Jiayi Wang, Xukai Wang, Jingang Zhou, Hongbo Zhao, Wenzhuo Liu, Shijie Ma, Da-Han Wang, et al. Continual learning for generative ai: From llms to mllms and beyond. *arXiv preprint arXiv:2506.13045*, 2025c.

Haiyang Guo, Fei Zhu, Hongbo Zhao, Fanhu Zeng, Wenzhuo Liu, Shijie Ma, Da-Han Wang, and Xu-Yao Zhang. Mcitlib: Multimodal continual instruction tuning library and benchmark. *arXiv preprint arXiv:2508.07307*, 2025d.

Ziyu Guo, Renrui Zhang, Hao Chen, Jialin Gao, Dongzhi Jiang, Jiaze Wang, and Pheng-Ann Heng. Sciverse: Unveiling the knowledge comprehension and visual reasoning of lmms on multi-modal scientific problems. In *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 19683–19704, 2025e.

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

Song Han, Jeff Pool, John Tran, and William Dally. Learning both weights and connections for efficient neural network. *Advances in neural information processing systems*, 28, 2015.

Jinghan He, Haiyun Guo, Ming Tang, and Jinqiao Wang. Continual instruction tuning for large multimodal models. *arXiv e-prints*, pp. arXiv–2311, 2023.

Xuehai He, Yichen Zhang, Luntian Mou, Eric Xing, and Pengtao Xie. Pathvqa: 30000+ questions for medical visual question answering. *arXiv preprint arXiv:2003.10286*, 2020.

Dan Hendrycks, Steven Basart, Norman Mu, Saurav Kadavath, Frank Wang, Evan Dorundo, Rahul Desai, Tyler Zhu, Samyak Parajuli, Mike Guo, et al. The many faces of robustness: A critical analysis of out-of-distribution generalization. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 8340–8349, 2021.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Liang Wang, Weizhu Chen, et al. Lora: Low-rank adaptation of large language models. *International Conference on Learning Representations*, 1(2):3, 2022.

Tianyu Huai, Jie Zhou, Xingjiao Wu, Qin Chen, Qingchun Bai, Ze Zhou, and Liang He. Cl-moe: Enhancing multimodal large language model with dual momentum mixture-of-experts for continual visual question answering. In *Proceedings of the computer vision and pattern recognition conference*, pp. 19608–19617, 2025.

Jiazheng Kang, Le Huang, Cheng Hou, Zhe Zhao, ZhenXiang Yan, and Ting Bai. Self-evolving llms via continual instruction tuning. In *Proceedings of the ACM Web Conference 2026*, pp. 7724–7732, 2026.

Aniruddha Kembhavi, Mike Salvato, Eric Kolve, Minjoon Seo, Hannaneh Hajishirzi, and Ali Farhadi. A diagram is worth a dozen images. In *European conference on computer vision*, pp. 235–251. Springer, 2016.

Aniruddha Kembhavi, Minjoon Seo, Dustin Schwenk, Jonghyun Choi, Ali Farhadi, and Hannaneh Hajishirzi. Are you smarter than a sixth grader? textbook question answering for multimodal machine comprehension. In *Proceedings of the IEEE Conference on Computer Vision and Pattern recognition*, pp. 4999–5007, 2017.

Junnan Li, Ramprasaath Selvaraju, Akhilesh Gotmare, Shafiq Joty, Caiming Xiong, and Steven Chu Hong Hoi. Align before fuse: Vision and language representation learning with momentum distillation. *Advances in neural information processing systems*, 34:9694–9705, 2021.

Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In *International conference on machine learning*, pp. 12888–12900. PMLR, 2022.

Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *International conference on machine learning*, pp. 19730–19742. PMLR, 2023.

Lei Li, Yuqi Wang, Runxin Xu, Peiyi Wang, Xiachong Feng, Lingpeng Kong, and Qi Liu. Multimodal arxiv: A dataset for improving scientific comprehension of large vision-language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 14369–14387, 2024.

Songze Li, Mingyu Gao, Tonghua Su, Xu-Yao Zhang, and Zhongjie Wang. Multimodal continual instruction tuning with dynamic gradient guidance. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10820–10829, 2026.

Yan-Shuo Liang and Wu-Jun Li. Inflora: Interference-free low-rank adaptation for continual learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 23638–23647, 2024.

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

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *Advances in neural information processing systems*, 36:34892–34916, 2023.

Yuan Liu, Haodong Duan, Yuanhan Zhang, Bo Li, Songyang Zhang, Wangbo Zhao, Yike Yuan, Jiaqi Wang, Conghui He, Ziwei Liu, et al. Mmbench: Is your multi-modal model an all-around player? In *European conference on computer vision*, pp. 216–233. Springer, 2024.

Sylvain Lobry, Diego Marcos, Jesse Murray, and Devis Tuia. Rsvqa: Visual question answering for remote sensing data. *IEEE Transactions on Geoscience and Remote Sensing*, 58(12):8555–8566, 2020.

Pan Lu, Liang Qiu, Jiaqi Chen, Tony Xia, Yizhou Zhao, Wei Zhang, Zhou Yu, Xiaodan Liang, and Song-Chun Zhu. Iconqa: A new benchmark for abstract diagram understanding and visual language reasoning. *arXiv preprint arXiv:2110.13214*, 2021.

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

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pp. 8748–8763. PmLR, 2021.

Chonghao Sima, Katrin Renz, Kashyap Chitta, Li Chen, Hanxue Zhang, Chengen Xie, Jens Beißwenger, Ping Luo, Andreas Geiger, and Hongyang Li. Drivelm: Driving with graph visual question answering. In *European conference on computer vision*, pp. 256–274. Springer, 2024.

Mingjie Sun, Zhuang Liu, Anna Bair, and Zico Kolter. A simple and effective pruning approach for large language models. In *International Conference on Learning Representations*, volume 2024, pp. 4942–4964, 2024.

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

Xiao Wang, Tianze Chen, Qiming Ge, Han Xia, Rong Bao, Rui Zheng, Qi Zhang, Tao Gui, and Xuan-Jing Huang. Orthogonal subspace learning for language model continual learning. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 10658–10671, 2023.

Fanhu Zeng, Fei Zhu, Haiyang Guo, Xu-Yao Zhang, and Cheng-Lin Liu. Modalprompt: Towards efficient multimodal continual instruction tuning with dual-modality guided prompt. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 12137–12152, 2025.

Duzhen Zhang, Yong Ren, Zhong-Zhi Li, Yahan Yu, Jiahua Dong, Chenxing Li, Zhilong Ji, and Jinfeng Bai. Enhancing multimodal continual instruction tuning with branchlora. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 5743–5756, 2025.

Qiang Zhang, Fanrui Zhang, Jiawei Liu, Ming Hu, Junjun He, and Zheng-Jun Zha. Reliable lifelong multimodal editing: Conflict-aware retrieval meets multi-level guidance. *Advances in Neural Information Processing Systems*, 38:111235–111258, 2026.

Hongbo Zhao, Fei Zhu, Haiyang Guo, Meng Wang, Rundong Wang, Gaofeng Meng, and Zhaoxiang Zhang. Mllm-cl: Continual learning for multimodal large language models. *arXiv preprint arXiv:2506.05453*, 2025a.

Xuanle Zhao, Xianzhen Luo, Qi Shi, Chi Chen, Shuo Wang, Zhiyuan Liu, and Maosong Sun. Chartcoder: Advancing multimodal large language model for chart-to-code generation. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 7333–7348, 2025b.

Junhao Zheng, Shengjie Qiu, Chengming Shi, and Qianli Ma. Towards lifelong learning of large language models: A survey. *ACM Computing Surveys*, 57(8):1–35, 2025.

Deyao Zhu, Xiaoqian Shen, Xiang Li, Mohamed Elhoseiny, et al. Minigpt-4: Enhancing vision-language understanding with advanced large language models. In *International Conference on Learning Representations*, volume 2024, pp. 18378–18394, 2024.

Supplementary Material for:

SHARD: Subspace-centric Structured Harmonization and Relation Distillation-guided Activation for Continual Learning

Contents

Qualitative Results

Subspace Routing Comparison

Generality across Different MLLM Architectures

Additional Ablation Results

Details of Experimental Setup

Datasets

Evaluation Metrics

Implementation Details

More Baseline Details

Dynamic Knowledge Organization (DKO)

Subspace Selective Activation (SSA)

A Qualitative Results

A.1 Generality across Different MLLM Architectures

To validate the architectural generalizability of SHARD, we compare its performance against existing baselines on InternVL-Chat-7B in MLLM-DCL. As illustrated in Table 3, SHARD attains a BWT of **-4.74**, outperforming all other methods in alleviating catastrophic forgetting, excluding ModalPrompt, whose constrained MFT of **38.93** reflects inadequate capacity for new task learning. These empirical results underscore SHARD's superior proficiency in striking an optimal balance between integrating novel task information and retaining prior knowledge. Furthermore, in contrast to dynamic architecture baselines such as MoELoRA, which secures an MFT of **68.83** at the severe expense of a **-14.06** BWT, SHARD compellingly illustrates that dynamic structural modulation can fully sustain plasticity for new tasks without compromising historical knowledge.

Table 3: Performance comparison of different methods on MLLM-DCL benchmark with InternVL-Chat-7b (Chen et al., 2024b).

Method	Venue	RS	Med	AD	Sci	Fin	MFT (↑)	MFN (↑)	MAA (↑)	BWT (↑)
Zero-shot	-	31.16	29.81	14.06	33.93	64.32	34.66	-	-	-
Individual	-	81.49	66.42	54.56	54.48	91.24	69.64	-	-	-
LoRA-FT	ICLR-22	69.93	52.17	33.04	42.67	91.07	69.06	57.78	65.22	-14.11
O-LoRA	EMNLP-23	74.48	54.16	39.60	48.30	88.54	65.51	61.02	65.83	-5.62
MoELoRA	NeurIPS-24	69.90	52.08	33.17	42.19	90.58	68.83	57.58	65.97	-14.06
ModalPrompt	EMNLP-25	50.20	30.95	14.11	33.91	65.45	38.93	38.92	38.70	**-0.01**
CL-MoE	CVPR-25	78.12	52.51	35.53	42.69	91.24	69.22	60.02	67.60	-11.51
HiDE	ACL-25	75.40	**57.66**	36.73	41.48	88.59	65.26	59.97	65.94	-6.61
SEFE	ICML-25	**78.21**	57.59	51.45	44.65	**91.37**	**69.55**	64.65	**68.84**	-6.12
DISCO	ICCV-25	77.90	47.50	49.13	49.37	90.92	68.55	62.96	67.81	-6.98
Ours	-	78.07	51.07	53.43	51.63	90.54	68.74	64.95	68.40	-4.74

A.2 Comparison of Subspace Routing Results

To provide a more fine-grained analysis of subspace routing performance, we compare the Subspace Selective Activation (SSA) mechanism in DISCO with our Relation Distillation-guided Activation (RDSA) in terms of routing accuracy, which is defined as whether the selected LoRA subspace matches the ground-truth task of the test sample. As reported in Table 4, SSA achieves a routing accuracy of **0**% on Flickr30k. We attribute this failure to the fact that Flickr30k and VizWiz are both image captioning tasks sharing highly similar instructions. Consequently, routing based solely on text features is unable to distinguish between these two tasks, resulting in **0**% routing accuracy. In

Table 4: Comparison of subspace routing results for each task on UCIT benchmark.

Setting	Task	ImageNet-R	ArxivQA	VizWiz	IconQA	CLEVR-Math	Flickr30k
SSA	task1	3000/3000 (100.00%)					
	task2	3000/3000 (100.00%)	3000/3000 (100.00%)				
	task3	3000/3000 (100.00%)	3000/3000 (100.00%)	3000/3000 (100.00%)			
	task4	3000/3000 (100.00%)	2996/3000 (99.87%)	2250/2250 (100.00%)	3000/3000 (100.00%)		
	task5	3000/3000 (100.00%)	2996/3000 (99.87%)	3000/3000 (100.00%)	2943/3000 (98.10%)	2994/3000 (99.80%)	
	task6	3000/3000 (100.00%)	2996/3000 (99.87%)	3000/3000 (100.00%)	2943/3000 (98.10%)	2994/3000 (99.80%)	0/3000 (0.00%)
RDSA	task1	3000/3000 (100.00%)					
	task2	3000/3000 (100.00%)	3000/3000 (100.00%)				
	task3	3000/3000 (100.00%)	2998/3000 (99.93%)	3000/3000 (100.00%)			
	task4	3000/3000 (100.00%)	2991/3000 (99.70%)	3000/3000 (100.00%)	3000/3000 (100.00%)		
	task5	3000/3000 (100.00%)	2992/3000 (99.73%)	3000/3000 (100.00%)	3000/3000 (100.00%)	3000/3000 (100.00%)	
	task6	3000/3000 (100.00%)	2989/3000 (99.63%)	2959/3000 (98.63%)	3000/3000 (100.00%)	3000/3000 (100.00%)	3000/3000 (100.00%)

contrast, RDSA achieves nearly perfect routing accuracy across all tasks. This demonstrates that our relation distillation strategy effectively captures task-discriminative representations, substantially improving task identification under the task-agnostic inference setting and further leading to superior performance.

A.3 Design of Task Prototype Learning

We further investigate the effectiveness of the proposed task prototype learning strategy. Specifically, we consider a straightforward alternative that directly learns task prototypes from the image features extracted by the frozen vision encoder, denoted as *w/o Projected*. This setting deviates from the conventional prototype learning paradigm, where prototypes are typically constructed in the final representation space used by downstream tasks. As reported in Table 5, directly learning prototypes from the frozen visual features consistently yields inferior performance across all evaluation metrics compared with using the projected image features. We attribute this improvement to the fact that the projected features are transformed by multimodal projector and thus reside in a more suitable representation space for downstream adaptation. Consequently, they provide more discriminative task prototypes, leading to more reliable subspace routing.

Table 5: Ablation study of incorporating projected image features into task prototype learning.

Setting	**MFT**	**MFN**	**MAA**	**BWT**
w/o Projected	75.10	68.24	81.19	-8.24
w Projected	75.18	69.17	81.39	-7.22

A.4 Hyperparameter Analysis

Table 6: Sensitivity analysis of the λ_{sub} within RDSA

Setting	**MFT**	**MFN**	**MAA**	**BWT**
0.1	75.33	68.91	81.37	-7.71
0.2	75.15	69.04	81.37	-7.33
0.3	75.18	69.17	81.39	-7.22
0.4	75.12	68.33	81.24	-8.15

Given that the two losses in RDSA exhibit comparable numerical scales, we balance their contributions using a single weighting coefficient λ_{sub} in Equation 22. To investigate the impact of λ_{sub}, we conduct a series of ablation experiments with different values, as reported in Table 6. The results demonstrate that SHARD remains robust across a wide range of λ_{sub} values. We adopt $\lambda_{\text{sub}} = 0.3$ as the default setting, as it achieves the best performance in terms of both MFN and BWT metrics.

A.5 Additional Ablation Results

Table 7: Ablation study evaluating the effectiveness of individual components within the SHARD framework, which consists of the SKH and RDSA modules. The best and second-best results are highlighted in **bold** and underlined, respectively.

Variant	**MFT**	**MFN**	**MAA**	**BWT**
Baseline Model	74.71	67.31	81.00	-8.87
w/o SKH	74.84	67.94	81.00	-8.28
w/o RDSA	<u>75.05</u>	<u>68.46</u>	<u>81.28</u>	<u>-7.91</u>
Ours	**75.18**	**69.17**	**81.39**	**-7.22**

Table 8: Sensitivity analysis of the λ_{SKH} within SKH.

Setting	MFT	MFN	MAA	BWT
0.1	74.36	67.57	80.45	-8.14
0.01	74.77	68.57	81.07	-7.43
0.001	75.18	69.17	81.39	-7.22

Table 9: Sensitivity analysis of the ω_{mag} within SKH.

Setting	MFT	MFN	MAA	BWT
1.0	75.09	69.17	81.35	-7.10
0.8	75.05	68.12	81.13	-8.31
0.6	75.13	68.41	81.27	-8.06
0.4	75.11	68.60	81.34	-7.81
0.2	75.18	69.17	81.39	-7.22

Table 10: How structure importance is applied to model parameters within SKH.

Setting	MFT	MFN	MAA	BWT
directly multiple	74.08	66.54	80.52	-9.05
Ours	75.18	69.17	81.39	-7.22

Table 11: Ablation study of the effectiveness of relation distillation.

Setting	MFT	MFN	MAA	BWT
w/o Relation Distillation	75.41	67.06	81.14	-10.01
w Relation Distillation	75.18	69.17	81.39	-7.22

B Details of Experimental Setup

B.1 Datasets

To systematically validate the efficacy of SHARD in mitigating catastrophic forgetting while preserving generalization, we conduct extensive empirical evaluations on two prominent continuous learning benchmarks. Detailed dataset statistics are summarized in Table 12.

The UCIT suite Guo et al. (2025a) functions as a rigorous testbed for cross-format adaptability and distribution shift resilience. It aggregates six cognitively diverse domains, encompassing out-of-distribution visual recognition via ImageNet-R Hendrycks et al. (2021), scientific chart parsing via ArxivQA Li et al. (2024), visual question answering under degraded conditions via VizWiz Gurari et al. (2018), abstract diagrammatic reasoning via IconQA Lu et al. (2021), compositional mathematical logic via CLEVR-Math Lindström & Abraham (2022), and fine-grained image captioning via Flickr30k Plummer et al. (2015). This configuration challenges the architecture with approximately 214k training instances and 1.8k test cases partitioned into six sequential phases.

In parallel, we employ the MLLM-DCL benchmark Zhao et al. (2025a) to assess the capacity for specialized domain adaptation. This corpus spans five distinct expert spheres, integrating remote sensing via RSVQA Lobry et al. (2020), medical diagnostics via PathVQA He et al. (2020), autonomous driving via DriveLM Sima et al. (2024), financial forecasting via StockQA Zhao et al. (2025a), and an amalgamated scientific reasoning task comprising AI2D Kembhavi et al. (2016), SciVerse Guo et al. (2025e), MapQA Chang et al. (2022), and TQA Kembhavi et al. (2017). Featuring over 236k optimization samples and 48k evaluation instances, MLLM-DCL imposes stringent requirements on long-term knowledge retention owing to its pronounced domain heterogeneity.

Table 12: Statistics of the training datasets and test datasets for UCIT and MLLM-DCL.

Task	Train Dataset	Test Dataset	Train Number	Test Number
UCIT Benchmark				
ImgNet-R	ImageNet-R	ImageNet-R	24k	3k
ArxivQA	ArxivQA	ArxivQA	40k	3k
VizWiz	VizWiz	VizWiz	40k	3k
IconQA	IconQA	IconQA	30k	3k
CLEVR	CLEVR-Math	CLEVR-Math	40k	3k
Flickr30k	Flickr30k	Flickr30k	40k	3k
MLLM-DCL Benchmark				
RS	RSVQA	RSVQA	60k	10k
Med	PathVQA	PathVQA	22.8k	9.8k
AD	DriveLM	DriveLM	60k	10k
Sci	AI2D, SciVerse, MapQA, TQA	AI2D, SciVerse, MapQA, TQA	33.4k (12.4k, 0.9k, 9.6k, 7.8k)	8.2k (3.1k, 0.2k, 2.4k, 1.9k)
Fin	StockQA	StockQA	60k	10k

B.2 Evaluation Metrics

Adhering to the standard continuous evaluation paradigm established in previous literature (Chen et al., 2025), we quantify the learning trajectory using four holistic metrics. Assuming a task sequence of length T, we denote $A_{i,k}$ as the accuracy achieved on task k immediately following the optimization phase of task i.

Mean Finetune Accuracy estimates the immediate acquisition of novel tasks by averaging the diagonal elements of the performance matrix. This metric serves as an empirical upper bound of plasticity before any interference occurs:

$$\text{MFT} = \frac{1}{T}\sum_{i=1}^{T} A_{i,i}. \tag{23}$$

Mean Final Accuracy captures the ultimate knowledge retention upon the completion of the entire task sequence. The discrepancy between the initial finetune accuracy and the final accuracy acts as a direct indicator of catastrophic interference:

$$\text{MFN} = \frac{1}{T}\sum_{i=1}^{T} A_{T,i}. \tag{24}$$

Mean Average Accuracy provides a temporal perspective on the stability and plasticity balance by aggregating the intermediate accuracies across all sequential stages. Unlike metrics restricted to the final model state, this measurement effectively encapsulates learning efficiency and temporal knowledge preservation:

$$\text{MAA} = \frac{1}{T}\sum_{i=1}^{T}\left(\frac{1}{i}\sum_{k=1}^{i} A_{i,k}\right). \tag{25}$$

Backward Transfer explicitly quantifies representational degradation on historical tasks. Negative values indicate a regression in previously acquired capabilities, whereas values approaching zero reflect optimal robustness against forgetting:

$$\text{BWT} = \frac{1}{T}\sum_{i=1}^{T}(A_{T,i} - A_{i,i}). \tag{26}$$

B.3 Implementation Details

Our framework incorporates Low-Rank Adaptation modules across all projection layers of the core language model. For the optimization phase, we restrict the training duration to a single epoch per task on UCIT and apply a tailored epoch schedule of $\{1, 3, 1, 2, 1\}$ on MLLM-DCL. When utilizing the LLaVA-v1.5-7b backbone, the adaptation learning rate is configured at $2\text{e}{-4}$ for UCIT and $2\text{e}{-5}$ for MLLM-DCL. For the InternVL-Chat-7b architecture, the learning rate remains constant at $1\text{e}{-4}$ across both experimental setups. All computations are executed with a uniform global batch size of 16 utilizing NVIDIA H20 hardware.

To ensure an equitable baseline comparison and isolate algorithmic contributions from raw capacity scaling, we meticulously synchronize the trainable parameter budgets adhering to the MCITlib conventions (Guo et al., 2025d). Single-branch architectures operate with a base rank of 16 on UCIT and 32 on MLLM-DCL. Although SEFE leverages auxiliary large-scale reconstruction datasets and multi-format conversions, its adaptation rank is strictly capped to align with baseline models.

For dynamic expansion paradigms allocating E task-specific experts, we ensure that each individual expert initializes with the identical base rank as the single-branch methods. As detailed in Table 13, this allocation yields a cumulative rank capacity of $E \times 16$ for models including O-LoRA, CL-MoE, HiDe, DISCO, and our proposed SHARD. This constraint places each parametric branch on equal footing at initialization. MoELoRA operates under its native configuration with a marginally reduced per-expert rank. Prompt-based techniques including ModalPrompt utilize a prefix length of 10 for UCIT and 20 for MLLM-DCL to maintain comparable capacity constraints.

C More Baseline Details

Multimodal Continual Instruction Tuning (MCIT) aims to enable Multimodal Large Language Models (MLLMs) to progressively acquire new instruction-following capabilities from a stream of tasks

Table 13: Detailed parameter settings of different continual learning methods across LLaVA and InternVL on the UCIT and MLLM-DCL benchmarks.

Methods	LLaVA				InternVL			
	LR	Epoch	PEFT	Parameter setting	LR	Epoch	PEFT	Parameter setting
UCIT Benchmark								
LoRA-FT	2e-4	{1,1,1,1,1,1}	LoRA	rank = 16	1e-4	{1,1,1,1,1,1}	LoRA	rank = 16
O-LoRA	2e-4	{1,1,1,1,1,1}	LoRA	rank = 96, expert num = 6	1e-4	{1,1,1,1,1,1}	LoRA	rank = 96, expert num = 6
MoELoRA	2e-4	{1,1,1,1,1,1}	LoRA	rank = 18, expert num = 6	1e-4	{1,1,1,1,1,1}	LoRA	rank = 18, expert num = 6
ModalPrompt	2e-4	{1,1,1,1,1,1}	Prompt	prefix len = 10, expert num = 6	1e-4	{1,1,1,1,1,1}	Prompt	prefix len = 10, expert num = 6
CL-MoE	2e-4	{1,1,1,1,1,1}	LoRA	rank = 96, expert num = 6	1e-4	{1,1,1,1,1,1}	LoRA	rank = 96, expert num = 6
HiDe	2e-4	{1,1,1,1,1,1}	LoRA	rank = 96, expert num = 6	1e-4	{1,1,1,1,1,1}	LoRA	rank = 96, expert num = 6
SEFE	2e-4	{1,1,1,1,1,1}	LoRA	rank = 16	1e-4	{1,1,1,1,1,1}	LoRA	rank = 16
DISCO	2e-4	{1,1,1,1,1,1}	LoRA	rank = 96, expert num = 6	1e-4	{1,1,1,1,1,1}	LoRA	rank = 96, expert num = 6
SHARD	2e-4	{1,1,1,1,1,1}	LoRA	rank = 96, expert num = 6	1e-4	{1,1,1,1,1,1}	LoRA	rank = 96, expert num = 6
MLLM-DCL Benchmark								
LoRA-FT	2e-5	{1,3,1,2,1}	LoRA	rank = 32	1e-4	{1,3,1,2,1}	LoRA	rank = 32
O-LoRA	2e-5	{1,3,1,2,1}	LoRA	rank = 160, expert num = 5	1e-4	{1,3,1,2,1}	LoRA	rank = 160, expert num = 5
MoELoRA	2e-5	{1,3,1,2,1}	LoRA	rank = 35, expert num = 5	1e-4	{1,3,1,2,1}	LoRA	rank = 35, expert num = 5
ModalPrompt	2e-4	{1,3,1,2,1}	Prompt	prefix len = 20, expert num = 5	1e-4	{1,3,1,2,1}	Prompt	prefix len = 20, expert num = 5
CL-MoE	2e-5	{1,3,1,2,1}	LoRA	rank = 160, expert num = 5	1e-4	{1,3,1,2,1}	LoRA	rank = 160, expert num = 5
HiDe	2e-5	{1,3,1,2,1}	LoRA	rank = 160, expert num = 5	1e-4	{1,3,1,2,1}	LoRA	rank = 160, expert num = 5
SEFE	2e-5	{1,3,1,2,1}	LoRA	rank = 32	1e-4	{1,3,1,2,1}	LoRA	rank = 32
DISCO	2e-5	{1,3,1,2,1}	LoRA	rank = 160, expert num = 5	1e-4	{1,3,1,2,1}	LoRA	rank = 160, expert num = 5
SHARD	2e-5	{1,3,1,2,1}	LoRA	rank = 160, expert num = 5	1e-4	{1,3,1,2,1}	LoRA	rank = 160, expert num = 5

while preserving previously learned knowledge. Since continually updating the same parameter space can easily cause catastrophic forgetting, a common strategy is to introduce task-specific adaptation modules to reduce direct parameter interference among different tasks. Among such dynamic architecture methods, DISCO provides a representative task-subspace organization paradigm, where task-specific LoRA parameters are organized into multiple separated subspaces and selectively activated during inference. Although DISCO is originally introduced under a federated continual instruction tuning setting, its core mechanism can be more generally interpreted as a task-level LoRA organization and activation strategy for MCIT.

C.1 Dynamic Knowledge Organization (DKO)

Given a pretrained weight matrix $\boldsymbol{W}^0 \in \mathbb{R}^{d^{\text{out}} \times d^{\text{in}}}$, LoRA parameterizes the task-specific update with a low-rank decomposition:

$$\begin{aligned} \boldsymbol{W} &= \boldsymbol{W}^0 + \Delta \boldsymbol{W} \\ &= \boldsymbol{W}^0 + \boldsymbol{B}\boldsymbol{A}, \quad \boldsymbol{B} \in \mathbb{R}^{d^{\text{out}} \times r},\ \boldsymbol{A} \in \mathbb{R}^{r \times d^{\text{in}}}, \end{aligned} \tag{27}$$

where $r \ll \min(d^{\text{out}}, d^{\text{in}})$ denotes the LoRA rank. The key idea of LoRA is to approximate the full parameter update $\Delta \boldsymbol{W}$ with the product of two low-rank matrices, thereby enabling parameter-efficient adaptation.

However, in continual instruction tuning, if all tasks share or are merged into the same LoRA update space, the update for a new task may overwrite adaptation directions required by previous tasks, leading to task interference and forgetting. To mitigate this issue, DISCO decomposes the single LoRA update space into multiple task-specific subspaces:

$$\Delta \boldsymbol{W} = \boldsymbol{B}\boldsymbol{A} \Longleftrightarrow \{\boldsymbol{B}_1\boldsymbol{A}_1, \boldsymbol{B}_2\boldsymbol{A}_2, \cdots, \boldsymbol{B}_T\boldsymbol{A}_T\}, \tag{28}$$

where $\boldsymbol{B}_i\boldsymbol{A}_i$ denotes the LoRA subspace associated with the i-th task or a group of semantically similar tasks. In this way, continual learning is no longer performed by repeatedly writing all task knowledge into a single shared LoRA module; instead, it maintains a dynamic collection of task-related LoRA subspaces. Different task knowledge is assigned to different adaptation spaces, which reduces direct parameter conflicts between new and old tasks.

To determine which subspace an incoming task update should be assigned to, DISCO introduces an *identity token* as a semantic descriptor of the task. For task t, suppose a task partition k contains n_k^t instruction texts:

$$\left\{\boldsymbol{x}_{t,j}^{\text{ins}}\right\}_{j=1}^{n_k^t}. \tag{29}$$

A frozen text encoder $f_{\text{ins}}(\cdot)$ is used to extract instruction features, and their mean vector is taken as the local identity token:

$$\boldsymbol{\mu}_k^t = \frac{1}{n_k^t} \sum_{j=1}^{n_k^t} f_{\text{ins}}\left(\boldsymbol{x}_{t,j}^{\text{ins}}\right). \tag{30}$$

The intuition is that the instruction distribution often reflects the semantic type and response format of a task. For example, image classification tasks usually contain short-answer instructions such as "what is the object", while image captioning tasks often contain generative instructions such as "describe the image". Chart question answering, mathematical reasoning, and OCR tasks also exhibit different textual instruction patterns. Therefore, the mean instruction feature $\boldsymbol{\mu}_k^t$ can be viewed as a semantic prototype of the current task.

A dynamic cache maintains a set of global identity tokens:

$$\{\tilde{\boldsymbol{z}}_i\}_{i=1}^{M}, \tag{31}$$

where each global identity token $\tilde{\boldsymbol{z}}_i$ is associated with a LoRA subspace:

$$\boldsymbol{\theta}_i = \boldsymbol{B}_i \boldsymbol{A}_i. \tag{32}$$

When a new local identity token $\boldsymbol{\mu}_k^t$ arrives, its similarity to an existing global identity token $\tilde{\boldsymbol{z}}_i$ is measured by cosine similarity:

$$\cos\left(\boldsymbol{\mu}_k^t, \tilde{\boldsymbol{z}}_i\right) = \frac{(\boldsymbol{\mu}_k^t)^\top \tilde{\boldsymbol{z}}_i}{\|\boldsymbol{\mu}_k^t\|_2 \, \|\tilde{\boldsymbol{z}}_i\|_2}. \tag{33}$$

If the similarity exceeds a predefined threshold δ_{sim},

$$\cos\left(\boldsymbol{\mu}_k^t, \tilde{\boldsymbol{z}}_i\right) \geq \delta_{\text{sim}}, \tag{34}$$

the incoming task update is considered semantically matched with the i-th subspace. The corresponding LoRA output is then assigned to that subspace. Meanwhile, the global identity token can be updated by a sample-size-weighted average:

$$\tilde{\boldsymbol{z}}_i^* = \frac{n_{\tilde{\boldsymbol{z}}_i} \tilde{\boldsymbol{z}}_i + \sum_{j=1}^{p} n_j^t \boldsymbol{\mu}_j^t}{n_{\tilde{\boldsymbol{z}}_i} + \sum_{j=1}^{p} n_j^t}, \tag{35}$$

where p denotes the number of local identity tokens matched to this subspace, $n_{\tilde{\boldsymbol{z}}_i}$ denotes the number of samples previously used to form $\tilde{\boldsymbol{z}}_i$, and n_j^t denotes the number of samples associated with the j-th matched update. This weighted update allows the global identity token to be progressively refined as more task samples arrive, thereby providing a more stable semantic center for the corresponding subspace.

If the new identity token cannot match any existing global identity token, i.e., no existing subspace satisfies Equation 34, a new global identity token and a new LoRA subspace are initialized:

$$\tilde{\boldsymbol{z}}_{\text{new}} = \frac{\sum_{j=1}^{p} n_j^t \boldsymbol{\mu}_j^t}{\sum_{j=1}^{p} n_j^t}, \qquad \boldsymbol{\theta}_{\text{new}} = \boldsymbol{B}_{\text{new}} \boldsymbol{A}_{\text{new}}. \tag{36}$$

The LoRA parameters assigned to the same subspace are then updated by a generic aggregation function:

$$\boldsymbol{\theta}_i^* = \mathcal{F}\left(\boldsymbol{\theta}_1^t, \boldsymbol{\theta}_2^t, \cdots, \boldsymbol{\theta}_p^t\right), \qquad \boldsymbol{\theta}_i = \boldsymbol{B}_i \boldsymbol{A}_i. \tag{37}$$

Here, $\mathcal{F}(\cdot)$ can be understood as a general function for aggregating multiple LoRA outputs within the same semantic subspace, such as average aggregation or weighted aggregation.

Therefore, dynamic knowledge organization prevents task knowledge from being indiscriminately merged into a single LoRA module. Instead, task updates are routed to semantically matched subspaces according to instruction-level identity tokens. By explicitly decoupling different instruction distributions into separated adaptation spaces, this mechanism alleviates task interference in continual instruction tuning.

C.2 Subspace Selective Activation

After task-specific LoRA subspaces have been organized, a key question remains during inference: given a test input, how should the model use the accumulated task subspaces?

A straightforward solution is to concatenate all LoRA subspaces along the low-rank dimension. Specifically, all $\boldsymbol{B}$ matrices can be horizontally concatenated as

$$\overline{\boldsymbol{B}} = [\boldsymbol{B}_1, \boldsymbol{B}_2, \cdots, \boldsymbol{B}_T] \in \mathbb{R}^{d^{\text{out}} \times rT}, \tag{38}$$

and all $\boldsymbol{A}$ matrices can be vertically concatenated as

$$\overline{\boldsymbol{A}} = \begin{bmatrix} \boldsymbol{A}_1 \\ \boldsymbol{A}_2 \\ \vdots \\ \boldsymbol{A}_T \end{bmatrix} \in \mathbb{R}^{rT \times d^{\text{in}}}. \tag{39}$$

This yields an expanded LoRA output:

$$\Delta \boldsymbol{W}_{\text{concat}} = \overline{\boldsymbol{B}}\overline{\boldsymbol{A}} = \sum_{i=1}^{T} \boldsymbol{B}_i \boldsymbol{A}_i. \tag{40}$$

This formulation activates all task subspaces simultaneously, and every subspace directly contributes to the current parameter update. However, direct concatenation does not consider the relevance between the current input and different task subspaces. In other words, even if some task subspaces are unrelated to the current test instruction, they can still affect the output.

For example, when the current instruction only requires a short class name, LoRA subspaces learned from image captioning, long-form generation, or complex reasoning tasks may induce overly long or format-mismatched responses. Therefore, merely accumulating and concatenating all subspaces cannot ensure appropriate task knowledge utilization during inference.

To address this issue, Subspace Selective Activation (SSA) introduces an input-dependent activation factor for each task subspace. A standard LoRA output can first be rewritten by inserting an intermediate mixing matrix between the two low-rank matrices:

$$\Delta \boldsymbol{W} = \boldsymbol{B}\boldsymbol{\Lambda}\boldsymbol{A}, \qquad \boldsymbol{\Lambda} \in \mathbb{R}^{r\times r}. \tag{41}$$

In vanilla LoRA, this intermediate matrix is usually equivalent to an identity matrix and can thus be omitted:

$$\Delta \boldsymbol{W} = \boldsymbol{B}\boldsymbol{I}_{r\times r}\boldsymbol{A} = \boldsymbol{B}\boldsymbol{A}. \tag{42}$$

The key idea of SSA is to reinterpret this originally fixed identity matrix as an input-dependent activation matrix. For the i-th task subspace, the LoRA output is written as

$$\Delta \boldsymbol{W}_i = \boldsymbol{B}_i \left(\pi_{*,i}\boldsymbol{I}_{r\times r}\right) \boldsymbol{A}_i, \tag{43}$$

where $\pi_{*,i}$ is the activation factor of the i-th subspace. When $\pi_{*,i}$ is large, the corresponding subspace contributes more strongly to the current output; when $\pi_{*,i}$ is close to zero, the subspace is significantly suppressed. Thus, SSA introduces a lightweight gating mechanism inside LoRA subspaces to control the participation of different task knowledge during inference.

Given a test instruction $\boldsymbol{x}_*^{\text{ins}}$, SSA first extracts its feature with the same frozen text encoder:

$$\boldsymbol{h}_* = f_{\text{ins}}\left(\boldsymbol{x}_*^{\text{ins}}\right). \tag{44}$$

Then, the cosine similarity between the test instruction feature and each global identity token is computed as

$$s_{*,i} = \cos(\tilde{\boldsymbol{z}}_i, \boldsymbol{h}_*) = \frac{\tilde{\boldsymbol{z}}_i \cdot \boldsymbol{h}_*}{\|\tilde{\boldsymbol{z}}_i\|_2 \, \|\boldsymbol{h}_*\|_2}. \tag{45}$$

Here, $s_{*,i}$ measures the semantic similarity between the test instruction and the i-th task subspace. A larger similarity indicates that the subspace is more likely to be relevant to the current input.

The activation factors are then obtained by softmax normalization:

$$\pi_{*,i} = \frac{\exp(s_{*,i}/\tau_{\text{route}})}{\sum_{j=1}^{T} \exp(s_{*,j}/\tau_{\text{route}})}, \tag{46}$$

where τ_{route} is a temperature coefficient controlling the sharpness of the distribution. A smaller τ_{route} makes the activation weights more concentrated on the most relevant subspaces, while a larger τ_{route} produces a smoother distribution across multiple subspaces. In this way, subspaces that are semantically closer to the current instruction receive larger activation weights, while irrelevant subspaces are naturally suppressed.

The complete SSA update can be written as

$$\Delta \boldsymbol{W}_{\text{SSA}} = \overline{\boldsymbol{B}}\,\overline{\boldsymbol{\Lambda}}\,\overline{\boldsymbol{A}}, \tag{47}$$

where $\overline{\boldsymbol{B}}$ and $\overline{\boldsymbol{A}}$ are the concatenated matrices defined in Equations 38 and 39, and $\overline{\boldsymbol{\Lambda}}$ is a block-diagonal activation matrix:

$$\overline{\boldsymbol{\Lambda}} = \text{diag}\left(\pi_{*,1}\boldsymbol{I}_{r\times r}, \pi_{*,2}\boldsymbol{I}_{r\times r}, \cdots, \pi_{*,T}\boldsymbol{I}_{r\times r}\right). \tag{48}$$

Equivalently, Equation 47 can be expanded as

$$\Delta \boldsymbol{W}_{\text{SSA}} = \sum_{i=1}^{T} \pi_{*,i} \boldsymbol{B}_i \boldsymbol{A}_i. \tag{49}$$

Compared with direct concatenation in Equation 40, SSA replaces uniform activation over all subspaces with input-dependent weighted activation. As a result, the model can preserve all accumulated task subspaces while dynamically adjusting the contribution of each subspace according to the semantic type of the current instruction. This helps reduce the interference of irrelevant task knowledge during inference.