PROTOADA: PROTOTYPE-GUIDED ADAPTIVE ADAPTER EXPANSION AND GEOMETRIC CONSOLIDATION FOR MULTIMODAL CONTINUAL INSTRUCTION TUNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Multimodal Large Language Models (MLLMs) achieve strong performance through instruction tuning, but real-world deployment requires them to continually acquire new vision-language capabilities, making Multimodal Continual Instruction Tuning (MCIT) essential. To reduce inter-task interference and promote collaboration, recent methods often employ sparse architectures like Mixture of LoRA Experts with image-text similarity routing. However, tasks with distinct response structures could share highly similar visual-linguistic semantics and thus be wrongly routed to the same expert. Image-text similarity alone is insufficient for reliable task assignment and as a result, an expert in a grounding task requiring coordinate prediction may be biased toward producing short textual answers after learning semantically similar VQA tasks. This format-blind task assignment integrates heterogeneous response types into shared parameters, inducing gradient interference and ineffective expert collaboration. To address this problem, we propose PROTOADA, a prototype-guided adaptive tuning framework. PROTOADA constructs format-aware task codes that jointly characterize multimodal semantics and response-protocol structures, thereby inducing a compatibility-aware organization of the evolving task stream for task assignment and inference-time routing. It further applies geometry-aware consolidation to separate reusable directions from task-specific residuals, progressively refining shared parameters while preserving protocol-sensitive behaviors. Together, these components mitigate forgetting from both task organization and parameter consolidation perspectives. Extensive experiments on multiple benchmarks demonstrate superior performance, especially on tasks whose answer structures are vulnerable to sequential tuning.

1 INTRODUCTION

Recently, multimodal Large Language Models (MLLMs) (Bai et al., 2023; Liu et al., 2023; Zhu et al., 2023) have shown strong generalization through multimodal instruction tuning (Zhang et al., 2023a; Tong et al., 2025) on large-scale datasets, enabling a wide range of vision-language tasks (Radford et al., 2021; Dai et al., 2023; Guo et al., 2025c). In realistic scenarios, however, multimodal tasks (Hu & Singh, 2021; Yang et al., 2025b) often arrive sequentially, requiring multimodal continual instruction tuning (MCIT) (Chen et al., 2024) to continually acquire new capabilities after deployment, without access to previous training data. This setting is challenging due to catastrophic forgetting (French, 1999; French & Ferrara, 2020), where learning new tasks may overwrite not only previously learned semantic knowledge but also task-specific output conventions (Liu et al., 2025b). Recent works introduce parameter-efficient adaptation to improve the stability-plasticity trade-off (Zeng et al., 2025; Yu et al., 2025; Zhao et al., 2025; Huai et al., 2025; Wang et al., 2025b), with MoE-style parameter isolation mitigating interference by routing tasks to sparsely activated lightweight experts, typically based on image-text semantic similarity. However, semantic similarity alone cannot determine whether tasks should share the same adaptation path: tasks involving similar visual regions and instructions may require incompatible output protocols, such as bounding-box coordinates versus short textual phrases. Such mismatches can entangle heterogeneous formats within experts, resulting in severe parameter overwriting and ultimately catastrophic forgetting.

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.

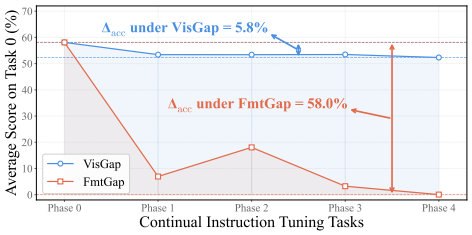
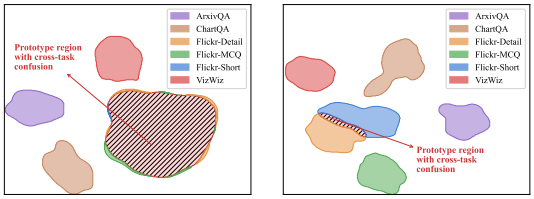


Figure 1: Accuracy of sequential finetuning under format-varied (FmtGap) and semantic-varied (VisGap) continual tuning streams.



(a) Semantic-Only. (b) Format-Aware.

Figure 2: t-SNE visualization of task representations under semantic-only and format-aware prototypes. Shaded area denotes prototype regions that will lead to incompatible task assignment.

Format-induced incompatibility suggests that the failure of semantic routing may arise from a deeper mismatch: semantically similar instructions can require fundamentally different response protocols. To verify this, we construct two controlled continual-learning streams under the same parameter-sharing setup. The first stream, termed FmtGap, uses Flickr30k (Plummer et al., 2015) throughout to fix the visual-semantic content while varying the response protocol across five phases[1]. The second, termed VisGap, fixes the output to brief descriptions while alternating between Flickr30k and VizWiz (Gurari et al., 2018) to introduce visual-semantic shifts. As shown in Fig. 1, we measure the first task's testing performance (denoted as Task-0 Accuracy) after learning each task and it drops by 58.0 under FmtGap but only 5.8 under VisGap, indicating substantially stronger interference from response-protocol shifts. This result indicates that MLLM tuning not only learns visual-linguistic associations but also aligns instructions with expected answer forms. Therefore, visual-linguistic similarity is insufficient for parameter-sharing decisions, and response-protocol compatibility should be explicitly considered.

The performance gap shown above raises a practical question: can response formats be captured in a lightweight way before deciding task sharing? For each task, we construct two representations from frozen image-text embeddings: a semantic-only prototype and a format-aware prototype augmented with simple response statistics, such as average length and token entropy. The t-SNE (Van der Maaten & Hinton, 2008) visualization in Fig. 2 shows that semantic-only embeddings poorly separate tasks with different answer protocols, while the lightweight format code makes these protocols more distinguishable. This suggests that response statistics can serve as an effective proxy for instruction-following patterns, guiding task grouping to encourage sharing among compatible protocols while reducing interference between incompatible ones.

These observations suggest that effective MCIT should both identify incompatible response protocols to avoid format-specific conflicts and mitigate parameter overwriting and catastrophic forgetting. To this end, we propose PROTOADA, a prototype-guided adaptive tuning framework that builds format-aware task prototypes from visual, language, and response-format statistics, grouping tasks by both content and expected answer protocol. To further curb the rapid growth of expert parameters, within each group, PROTOADA performs geometry-aware consolidation: group-aligned directions refine shared parameters, while residual directions are preserved to protect task-specific response conventions. Experiments on multiple benchmarks demonstrate the effectiveness of PROTOADA on complex continual data streams.

2 RELATED WORK

Multimodal Large Language Models(MLLMs) combine visual encoders with language models to support multimodal understanding and generation. Vision-language pretraining learns transferable image-text representations (Radford et al., 2021), while instruction-tuned models such as LLaVA and Qwen-VL extend them to interactive tasks (Liu et al., 2023; Bai et al., 2023). Recent work further improves visual reasoning, instruction following, and multimodal generation (Johnson et al., 2017; Zerroug et al., 2022; Zhou et al., 2023; Feng et al., 2025; Yang et al., 2025a). However, these models

[1]The five formats are brief description, detailed description, short/one-word answer, multiple choice answer, and yes/no answer. See Appendix. C for more details.

are typically trained under static or jointly sampled task distributions, leaving continual adaptation and knowledge retention insufficiently addressed.

Multimodal Continual Instruction Tuning. As MLLMs are deployed in open-world scenarios, continual instruction tuning (Chen et al., 2024; Liu et al., 2025b) becomes essential for learning new capabilities without overwriting old ones. Existing methods mainly rely on replay (Li et al., 2025; Lee et al., 2025; Yu et al., 2025), regularization or prompts (Wang et al., 2023; Cao et al., 2024; Zeng et al., 2025; Liu et al., 2025a), and parameter-efficient or expert-based adaptation (Guo et al., 2025a; Zhao et al., 2025; Huai et al., 2025; Wang et al., 2025b;a). These approaches preserve prior knowledge through sample reuse, constrained updates, or isolated lightweight modules, but their routing signals generally emphasize visual and textual semantics while leaving response protocols implicit. Thus, semantically similar tasks with different output structures may still share parameters and interfere, motivating format-aware grouping and geometry-aware consolidation for controlled knowledge sharing in MCIT.

3 PRELIMINARIES

3.1 MULTIMODAL CONTINUAL INSTRUCTION TUNING

We consider a multimodal large language model consisting of a frozen vision encoder ϕ, a multimodal projector π, and a large language model f equipped with lightweight trainable modules. Let $\{\mathcal{T}_1, \mathcal{T}_2, \ldots, \mathcal{T}_N\}$ denote a sequence of multimodal instruction-tuning tasks, where each $\mathcal{T}_i = \{(v_j, q_j, y_j)\}_{j=1}^{M_i}$ contains an image v_j, an instruction q_j, and a target response $y_j = (y_{j,1}, \ldots, y_{j,L_j})$. The frozen vision encoder extracts visual features $\tilde{\mathbf{v}} = \phi(v)$, the embedding layer $\psi(\cdot)$ produces instruction embeddings $\mathbf{u} = \psi(q)$, and the projector maps visual features into the language space as $\mathbf{w} = \pi(\tilde{\mathbf{v}})$, forming the multimodal input $\mathbf{z} = [\mathbf{w}; \mathbf{u}]$. Given a target response token sequence $y = (y_1, \ldots, y_L)$, the MLLM models the conditional distribution:

$$p_\Theta(y \mid z) = \prod_{k=1}^{L_j} p_\Theta(y_k \mid \mathbf{z}, y_{<k}). \tag{1}$$

The goal of MCIT is to obtain a unified model that performs well on all tasks observed so far. After step i, the ideal objective is

$$\Theta_i^* = \arg\min_\Theta \mathbb{E}_{(v,q,y)\sim\mathcal{T}_{\leq i}} \left[-\log p_\Theta(y \mid \mathbf{z})\right], \tag{2}$$

where $\mathcal{T}_{\leq i}$ denotes the union of all seen task distributions. In rehearsal-free MCIT, however, optimization at step i only accesses $\mathcal{T}_i$, and thus the learner must improve the current task likelihood while preserving behaviors induced by previous tasks without revisiting their samples. This mismatch of data distribution is the fundamental source of forgetting.

3.2 LOW-RANK ADAPTATION IN THE LANGUAGE MODEL

We adopt a rehearsal-free parameter-efficient setting in which the vision encoder, projector, and pretrained language backbone are frozen, and trainable parameters are introduced only through LoRA modules inserted into all linear modules of the language model. For a linear module ℓ with frozen weight W_ℓ, LoRA modifies its output as

$$\mathbf{o}_\ell = \mathbf{W}_\ell \mathbf{h}_\ell + \mathbf{B}_\ell \mathbf{A}_\ell \mathbf{h}_\ell, \tag{3}$$

where $\mathbf{A}_\ell \in \mathbb{R}^{r\times d_{\text{in}}}$ and $\mathbf{B}_\ell \in \mathbb{R}^{d_{\text{out}}\times r}$ are trainable low-rank factors with rank r. Only $\Theta = \{\mathbf{A}_\ell, \mathbf{B}_\ell\}_{\ell\in\mathcal{L}}$ are updated. After learning task t, we denote the LoRA increment as $\Delta\Theta_t = \Theta_t - \Theta_{t-1}$, which must adapt to the current task while avoiding disruption to previously learned instruction-following behaviors. LoRA-based continual instruction tuning thus requires careful decisions about which parameter group receives a new task and how its update is consolidated.

Discussions. The objective in Eq. (2) implies that learning multiple tasks on the same expert can force their updates into a shared parameter space, causing parameter overwriting and, consequently, catastrophic forgetting. As discussed in Fig. 1, if task allocation and routing rely mainly on image-instruction similarity, semantically similar tasks with incompatible answer formats may be assigned

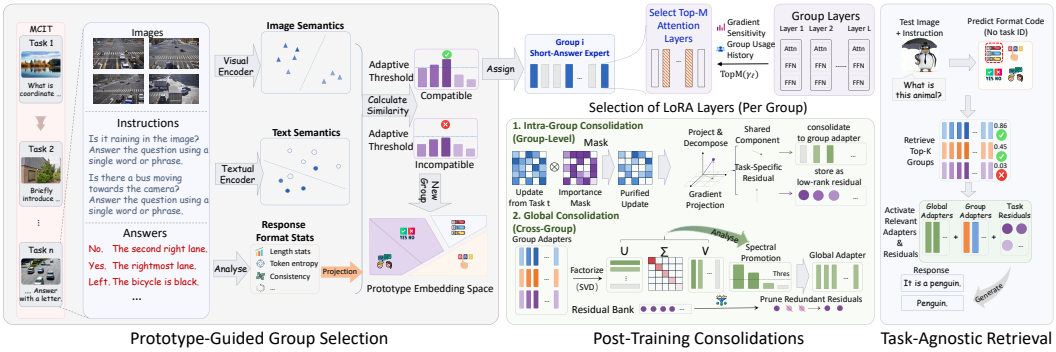


Figure 3: PROTOADA follows three stages. **Left**: Prototype-guided group selection. Visual and language semantics, together with response-format statistics, form a task prototype that assigns each task to a compatible group or a new one. Gradient sensitivity and group usage history then select the LoRA layers to adapt. **Middle**: Post-training consolidation. Importance masking and geometry-aware projection decompose each update into a reusable group component and a task-specific residual; stable directions are promoted globally, and redundant residuals are pruned. **Right**: Task-agnostic retrieval. Without task IDs, the model predicts the format code, retrieves compatible groups, activates the relevant adapters and residuals, and generates a protocol-compliant response.

to the same LoRA group, creating precisely this kind of interference. These observations call for a format-aware mechanism that distinguishes tasks by both their semantics and response protocols before sharing parameters, while consolidating compatible updates to enable reliable reuse.

4 METHOD

To preserve instruction-following ability for MLLMs, we propose PROTOADA, a prototype-guided MCIT framework that organizes LoRA parameters into multiple memory groups. The framework consists of three complementary designs followed by a unified pipeline summary. It first establishes format-aware task grouping in Sec. 4.1, constructing task prototypes from visual, language, and response-format information to assign each incoming task to a compatible memory group. Building on this assignment, Sec. 4.2 introduces prototype-conditioned adaptive growth, which selectively adds LoRA capacity according to the current task and the historical usage of each group. Once the task-specific capacity has been optimized, Sec. 4.3 develops geometry-aware consolidation and task-agnostic retrieval to separate reusable group-aligned updates from task-specific residuals, consolidate compatible directions, and retrieve the appropriate parameters during inference. The complete PROTOADA pipeline is summarized in Sec. 4.4.

4.1 FORMAT-AWARE TASK GROUPING

To better represent tasks with prototypes and more effectively identify compatibility relations among tasks, given a task $\mathcal{T}_i = \{(v_{t,j}, q_{t,j}, y_{t,j})\}_{j=1}^{M_i}$, we first compute its language and visual descriptors using frozen visual and textual encoders such as CLIP (Radford et al., 2021):

$$\mathbf{p}_t^{\text{text}} = \frac{1}{M_t}\sum_{j=1}^{M_t} E_{\text{text}}(q_{t,j}), \quad \mathbf{p}_t^{\text{vis}} = \frac{1}{M_t}\sum_{j=1}^{M_t} E_{\text{vis}}(v_{t,j}). \tag{4}$$

However, these descriptors only capture what the task is about, but not how the model should answer. As shown in Fig. 1, tasks with similar image-text semantics may require different response protocols. Routing such tasks only by image-text similarity can therefore force incompatible instruction formats into the same parameters and damage the model's instruction-following behavior. To incorporate response protocols into task grouping, we calculate the mean, variance of response length, token uncertainty, and template consistency [2] from target responses and derive a format code:

$$\mathbf{c}_t^{\text{fmt}} = [\mu_\ell, \sigma_\ell, H_{\text{tok}}, C_{\text{fmt}}], \tag{5}$$

[2]Implementation details about the calculation of these format statistics are shown in Appendix. D

This format code is mapped into the prototype space by:

$$\mathbf{p}_t^{\text{fmt}} = \mathbf{W}_{\text{fmt}}\mathbf{c}_t^{\text{fmt}} + \mathbf{b}_{\text{fmt}}. \tag{6}$$

The final task prototype combines language, vision, and format information:

$$\mathbf{q}_t = \text{Norm}\left(\mathbf{W}_q[\mathbf{p}_t^{\text{text}} \| \mathbf{p}_t^{\text{vis}} \| \mathbf{p}_t^{\text{fmt}}]\right). \tag{7}$$

Based on the format-aware task prototype, each memory group g is represented by a group prototype p_g that summarizes its assigned tasks. When a new task arrives, we compare its prototype q_t with all existing group prototypes to decide whether it can share an existing group or should initialize a new one:

$$s_{t,g} = \frac{\langle \mathbf{q}_t, \mathbf{p}_g \rangle}{\|\mathbf{q}_t\|_2 \|\mathbf{p}_g\|_2}, \qquad g_t = \arg\max_g s_{t,g}. \tag{8}$$

Since different groups may exhibit different degrees of internal diversity, we adopt a dynamic admission threshold. Specifically, for each group g, we maintain the running mean μ_g and standard deviation σ_g of the similarities between previously assigned tasks and the group prototype $\mathbf{p}_g$. The assignment rule is then defined as:

$$\text{Assign}(t) = \begin{cases} g_t, & s_{t,g_t} \geq \mu_{g_t} - \sigma_{g_t}, \\ \text{new}, & s_{t,g_t} < \mu_{g_t} - \sigma_{g_t}. \end{cases} \tag{9}$$

If the task is assigned to an existing group, its prototype is updated as:

$$\mathbf{p}_{g_t} \leftarrow \text{Norm}\big((1-\alpha)\mathbf{p}_{g_t} + \alpha \mathbf{q}_t\big). \tag{10}$$

Otherwise, a new group is initialized with $\mathbf{q}_t$. This allows compatible tasks to share LoRA parameters while separating tasks with conflicting response protocols. However, since the format is unavailable during inference, we therefore train a lightweight predictor f_{fmt} to infer it from input-side descriptors during training:

$$\hat{\mathbf{c}}_t^{\text{fmt}} = f_{\text{fmt}}(\mathbf{p}_t^{\text{text}}). \tag{11}$$

During inference, we replace $\mathbf{c}^{\text{fmt}}$ with the predicted format code $\hat{\mathbf{c}}^{\text{fmt}}$ when constructing the task prototype, thereby enabling format-aware grouping under task-agnostic retrieval.

Discussions. By incorporating response protocols into task identity, Eq. (7) fuses visual, linguistic, and format cues, while Eq. (9) admits a task into a group only when it is compatible with the group's historical prototype. The format predictor further extends this principle to task-agnostic inference, where target responses are unavailable.

4.2 PROTOTYPE-CONDITIONED ADAPTIVE GROWTH

Once a task has been assigned to a memory group, PROTOADA further decides what additional parameters should be introduced and where they should be placed. Adding Lora Modules to all layers for every task is parameter-inefficient and may perturb stable behaviors, so we selectively adapt only the layers where the current task demands the most change. For each FFN layer ℓ, we estimate task sensitivity by the gradient energy of the task loss with respect to the hidden state:

$$\gamma_\ell = \mathbb{E}_{(v,q,y)\sim\mathcal{T}_i}\left[\left\|\frac{\partial \mathcal{L}_{\text{ce}}}{\partial \hat{h}_\ell}\right\|_2\right]. \tag{12}$$

A larger γ_ℓ indicates that layer ℓ is more involved in fitting the current task and is therefore a better candidate for adaptation.

To avoid repeatedly allocating capacity to only a few high-gradient layers, we further incorporate the historical layer-usage pattern of the assigned memory group. Since tasks in the same group share compatible instruction-following behaviors, adapting a historically used layer is more likely to reuse an established adaptation path rather than open an unrelated one. We thus combine current-task sensitivity with group-level usage:

$$\tilde{\gamma}_\ell = \gamma_\ell + \eta u_{g_t,\ell}, \quad \mathcal{S}_t = \text{TopM}(\{\tilde{\gamma}_\ell\}_{\ell=1}^{L}; M), \tag{13}$$

where $u_{g_t,\ell}$ records how frequently group g_t has used layer ℓ, and $\mathcal{S}_t$ is the set of Linear layers selected for adaptation. Only the modules in $\mathcal{S}_t$ are updated for the current task.

Discussions. This mechanism refines the memory assignment made by the prototype: the prototype decides which group hosts the task, while the sensitivity-and-usage score decides where new capacity is introduced within that group. The usage term provides a soft reuse prior for compatible tasks, while the sensitivity term still allows new layers to be selected when the current task demands them, balancing parameter reuse and task-specific expansion under format-aware grouping.

4.3 GEOMETRY-AWARE CONSOLIDATION AND TASK-AGNOSTIC RETRIEVAL

After task t is assigned to group g_t and optimized, the LoRA increment $\Delta\Theta_t$ contains two types of information: group-compatible knowledge that benefits future tasks in the same group, and task-specific corrections that only preserve the current task's response behavior. Merging the whole update into the group adapter would entangle these two components and let fragile response protocols interfere with shared abilities. We therefore decouple the reusable part of each update from its task-specific residual before consolidation.

Since not every changed parameter is equally supported by the current task, we first compute an activation-aware importance score:

$$I_t(i,j) = |\Delta\Theta_t(i,j)|\, a_j, \tag{14}$$

where $a_j = \|\mathbf{X}_{:,j}\|_2$ is the input activation norm of channel j. Keeping the top-$k\%$ entries yields a purified update (Sun et al., 2024):

$$\Delta\hat{\Theta}_t = \mathbf{M}_t \odot \Delta\Theta_t, \tag{15}$$

which is then decomposed according to the geometry of the assigned group. Let $\mathcal{U}_{g_t}$ denote the subspace spanned by the current shared memory of group g_t, and let $\mathrm{Proj}_{\mathcal{U}_{g_t}}(\cdot)$ be the orthogonal projection onto this subspace. The component aligned with the group memory is absorbed into the shared adapter, while the orthogonal component is kept as a compact task-specific residual:

$$\Delta\Theta_t^{\mathrm{share}} = \mathrm{Proj}_{\mathcal{U}_{g_t}}(\Delta\hat{\Theta}_t), \quad \Delta\Theta_t^{\mathrm{res}} = \Delta\hat{\Theta}_t - \Delta\Theta_t^{\mathrm{share}}. \tag{16}$$

Here, $\Delta\Theta_t^{\mathrm{share}}$ captures directions reusable by the assigned group, whereas $\Delta\Theta_t^{\mathrm{res}}$ preserves task-specific deviations outside the group subspace.

The shared component updates the memory of the assigned group with a prototype-dependent weight. Let $\mathbf{A}_g$ denote the shared adapter parameters maintained by group g. We update the selected group as:

$$\mathbf{A}_{g_t} \leftarrow \mathbf{A}_{g_t} + \lambda_{t,g_t}\Delta\Theta_t^{\mathrm{share}}, \quad \lambda_{t,g} = \lambda_0 \frac{\exp(\beta s_{t,g})}{\sum_{g'}\exp(\beta s_{t,g'})}. \tag{17}$$

So tasks closer to the group prototype contribute more strongly. The residual is compressed by truncated singular value decomposition (Klema & Laub, 1980) as:

$$\mathbf{R}_t = \mathrm{SVD}_{r_{\mathrm{res}}}(\Delta\Theta_t^{\mathrm{res}}), \tag{18}$$

which preserves task-specific corrections without storing a dense task copy. To avoid redundant memory growth, we periodically reorganize the memory hierarchy. After every K newly observed tasks, the group adapters at layer ℓ are concatenated as:

$$W_{\mathrm{all},\ell} = [\mathbf{A}_{\ell,1}\|\mathbf{A}_{\ell,2}\|\cdots\|\mathbf{A}_{\ell,G}], \tag{19}$$

and their dominant spectral directions are promoted into a global shared adapter:

$$\mathbf{A}_\ell^{\mathrm{shared}} \leftarrow \mathbf{A}_\ell^{\mathrm{shared}} + \Delta\mathbf{A}_\ell^{\mathrm{global}}. \tag{20}$$

Residuals already explained by shared or group memory are then removed from the residual bank according to their coverage ratio:

$$\rho_t = \frac{\left\|\mathrm{Proj}_{\mathcal{U}_{\mathrm{shared}}\cup\mathcal{U}_{g_t}}(\mathbf{R}_t)\right\|_F}{\|\mathbf{R}_t\|_F}. \tag{21}$$

During inference, since the task identity is unavailable, we construct a query prototype using the predicted format code and retrieve the top compatible groups with similarity-based weights:

$$w_g = \frac{\exp(s_g/\tau)}{\sum_{g'\in\mathrm{TopK}}\exp(s_{g'}/\tau)}, \tag{22}$$

and activates only residuals whose stored prototypes are close to the query. The prediction is produced by augmenting the frozen backbone with the global adapter, the retrieved group-level adapters, and the selected task residuals. This enables task-specific knowledge to be organized and retrieved within a unified format-aware prototype space.

Table 1: Performance on TriGap. The best and second-best results are highlighted in **bold** and underline, respectively.

Methods	PMCVQA	DocVQA	ChartQA	IconQA	InfographicVQA	ArxivQA	Roadside	ChemVQA	FloodNetVQA	CLEVR	Average
Zero-shot (Liu et al., 2023)	35.40	12.68	9.36	19.27	5.06	53.77	7.40	5.30	47.41	20.37	21.60
FT-LoRA (Hu et al., 2022)	34.20	23.32	9.84	37.07	23.53	83.83	7.00	12.70	80.31	60.27	37.21
Replay-LoRA	33.70	33.95	14.00	46.67	28.97	75.57	9.40	15.90	73.81	58.80	39.08
MoE-LoRA (Chen et al., 2024)	39.03	37.49	12.44	43.43	35.17	90.90	7.93	20.70	**90.41**	**67.00**	44.45
HiDe-LLaVA (Guo et al., 2025a)	37.00	33.20	10.52	41.97	24.09	79.20	7.73	11.17	57.39	23.00	32.53
ModalPrompt (Zeng et al., 2025)	38.23	38.23	11.92	44.73	37.37	84.47	10.13	12.43	71.52	52.50	40.15
CL-MoE (Huai et al., 2025)	40.53	36.79	13.72	52.70	32.27	93.00	7.77	18.33	80.09	65.90	44.11
SAME (Xie et al., 2026)	41.60	**43.87**	17.56	64.03	**39.57**	90.46	10.83	21.77	81.09	54.50	46.53
DISCO (Guo et al., 2025b)	42.03	43.50	18.01	63.13	38.23	91.27	11.02	22.13	80.25	55.87	46.54
PROTOADA (Ours)	**42.87**	43.01	**18.96**	**66.03**	36.31	**93.50**	**12.97**	**23.73**	74.03	57.87	**47.23**

Table 2: Average performance of different methods on the UCIT benchmark. The best and second-best results are highlighted in **bold** and underline, respectively.

Methods	ImageNet-R	ArxivQA	Vizcap	IconQA	CLEVER	Flickr30k	Average
Zero-shot (Liu et al., 2023)	18.88	52.62	38.75	21.25	21.12	41.44	32.34
FT-LoRA (Hu et al., 2022)	29.33	55.30	45.51	26.13	13.07	58.07	37.90
MoE-LoRA (Chen et al., 2024)	58.43	77.57	44.83	68.90	56.73	**58.27**	60.79
Replay-LoRA	76.93	87.07	54.31	56.43	36.40	55.94	61.18
CL-MoE (Huai et al., 2025)	64.12	78.38	44.83	62.00	50.75	58.06	59.69
HiDe-LLaVA (Guo et al., 2025a)	87.62	91.12	42.68	57.62	31.00	50.41	60.08
ModalPrompt (Zeng et al., 2025)	80.50	90.62	60.13	63.50	55.75	57.09	67.93
DISCO (Guo et al., 2025b)	88.88	**94.25**	47.52	69.50	60.75	56.32	69.54
SAME (Xie et al., 2026)	**89.91**	91.40	55.33	77.51	**68.85**	55.43	73.07
PROTOADA (Ours)	89.83	93.20	**61.49**	**81.37**	66.40	55.64	**74.66**

4.4 SUMMARY OF PROTOADA

Overall, PROTOADA runs as a unified MCIT pipeline based on a set of memory groups. For each incoming task, we first build a format-aware prototype from frozen visual and textual descriptors together with a format code (Eqs. (4)–(7)), which is extracted from target responses during training and predicted by a lightweight predictor at inference (Eq. (11)). The prototype is then compared with existing group prototypes (Eqs. (8)–(10)), so that the task either joins a compatible group or seeds a new one. Within the assigned group, LoRA modules are inserted only at layers that are both sensitive to the current task and historically used by the group (Eqs. (12)–(13)), and are optimized by a cross-entropy loss together with an MSE loss for the format predictor, while the backbone and prior adapters stay frozen. The resulting update is then purified by activation-aware masking and split into a group-compatible component and an orthogonal residual (Eqs. (14)–(16)); the former is absorbed into the group adapter, and the latter is compressed via truncated SVD(Eqs. (17)–(18)). Periodic spectral promotion further extracts common directions into the global adapter and prunes redundant residuals (Eqs. (19)–(21)). At inference, the predicted format code yields a query prototype that retrieves the most compatible groups and residuals (Eq. (22)), and the frozen backbone augmented with the global, group, and selected residual adapters produces the response in a single forward pass, all within one shared format-aware prototype space. For reference, the complete training and inference procedures are summarized in Appendix B.

5 EXPERIMENTS

5.1 EXPERIMENTAL SETUP

Benchmarks. We evaluate PROTOADA on the following two benchmarks: TriGap (Xie et al., 2026) introduces challenges through longer task sequences, broader domain shifts, and imbalanced data scales, encompassing ten tasks: PMCVQA (Zhang et al., 2023b), DocVQA (Mathew et al., 2021), ChartQA (Masry et al., 2022), IconQA (Lu et al., 2021), InfographicVQA (Mathew et al., 2022), ArxivQA (Li et al., 2024), Roadside (Guan et al., 2026), ChemVQA (Sabando et al., 2020), FloodNetVQA (Rahnemoonfar et al., 2021), and CLEVR-Math (Lindström & Abraham, 2022). UCIT (Guo et al., 2025a) employs strict filtering to mitigate pre-training contamination and comprises six tasks: ArxivQA (Li et al., 2024), CLEVR-Math (Lindström & 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).

Variant	Avg. Acc (%)
Baseline	60.79
w/ Format	68.13
w/ Format + Adaptive	72.35
w/ Format + Adaptive + Geometry (Ours)	**74.66**

Table 3: Ablation study of PROTOADA under UCIT.

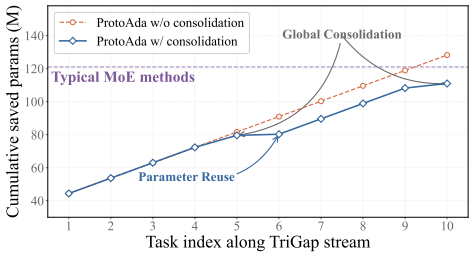


Figure 4: Parameter budget of PROTOADAover the TriGap task stream.

Evaluation metrics. Following Zhou et al. (2024); Chen et al. (2024), we denote by $\mathcal{A}_{s,t}$, the performance on task s evaluated after training up to task t, with T total tasks. We summarize the average final performance by $\bar{\mathcal{A}} = \frac{1}{T}\sum_{s=1}^{T}\mathcal{A}_{s,T}$.

Compared Methods. We compare PROTOADA with several state-of-the-art MCIT methods, including zero-shot evaluation (Liu et al., 2023), standard FineTune (Hu et al., 2022), MoELoRA (Chen et al., 2024), Replay-LoRA, HiDe-LLaVA (Guo et al., 2025a), CL-MoE (Huai et al., 2025), Modal-Prompt (Zeng et al., 2025), DISCO (Guo et al., 2025b), and SAME (Xie et al., 2026). Zero-shot results serve as a lower-bound reference. To ensure a fair comparison, all methods use an identical protocol.

Implementation details. All experiments are conducted on 4 NVIDIA RTX 5090 GPUs. We follow Prism (Tang et al., 2026) to conduct all experiments. Following Chen et al. (2024); Xie et al. (2026), we use LLaVA-v1.5-7B (Liu et al., 2023) as the backbone MLLM and CLIP-L/14-336 (Radford et al., 2021) to extract visual and textual features. The vision encoder, projector, and pretrained language model are frozen, and trainable low-rank modules are inserted into all linear layers of the language model (Wang et al., 2025a; Zhu et al., 2025). Each task is trained for 1 epoch using the AdamW optimizer with a learning rate of $2e-4$ and a linear warm-up ratio of 0.03. Unless otherwise specified, the initial rank is set to $r_0 = 8$ for each task in TriGap and $r_0 = 16$ for each task in UCIT. For PROTOADA, the residual rank is $r_{\text{res}} = 2$, the number of selected FFN layers is $M = 24$, and spectral memory maintenance is performed every $K = 5$ tasks. The format predictor is a two-layer MLP trained jointly with the task loss.

5.2 BENCHMARK COMPARISON AND ABLATION

Benchmark Comparison. We compare PROTOADA with representative continual instruction tuning baselines on TriGap and UCIT, as reported in Tab. 1 and Tab. 2. On the more challenging TriGap benchmark, PROTOADA achieves the best average accuracy of 47.23%, showing its robustness under longer task sequences and more heterogeneous response protocols. Consistent gains on UCIT further validate the effectiveness of our format-aware and consolidation-aware design, which enables compatible LoRA sharing while preserving task-specific instruction-following behaviors across benchmarks.

Ablation Study. Tab. 3 studies the contribution of each component in our method. We use MoELoRA as the baseline and fix the LoRA rank to 16 for all variants. Format-aware prototypes improve the average accuracy from 60.79% to 68.13% by grouping tasks with compatible response formats. Adaptive assignment further increases accuracy to 72.35% through more flexible memory selection. Method with Geometry-aware consolidation (full PROTOADA) achieves the best result of 74.66% by preserving task-specific update residuals while sharing common directions.

5.3 FURTHER ANALYSIS

Parameter Growth under Selective Expansion and Consolidation. A core claim of PROTOADA is that selective layer expansion in Eq. (13) and spectral consolidation in Eq. (20) keep memory growth controlled rather than letting parameters accumulate linearly. Fig. 4 records the cumulative saved adapter parameters along the TriGap task stream. Without consolidation, the parameter count

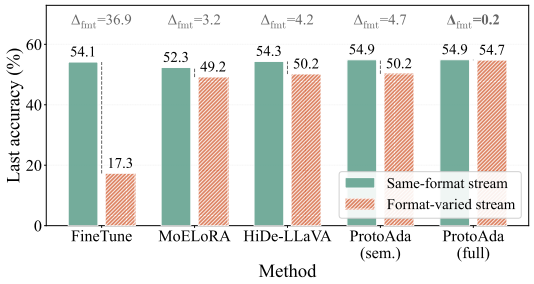


Figure 5: Performance gap between handling streams with the same format and streams with varied formats.

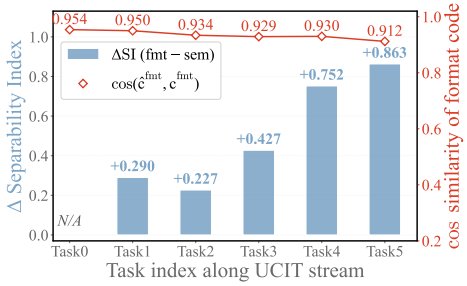


Figure 6: Separability index among task prototypes on UCIT as more tasks are learned, with the line tracking similarity between predicted and oracle format codes.

grows steadily and reaches about 128M at Task10, exceeding the typical MoE budget after Task9. In contrast, PROTOADA with consolidation grows more slowly after reuse begins around Task5 and remains below the MoE budget, ending at about 111M. The widening gap in later tasks shows that global consolidation promotes reusable directions instead of storing every task update independently. Together with the previous analyses, this confirms that PROTOADA improves continual adaptation by reusing compatible directions while preserving task-specific residuals.

Stress Test on Controlled Format Streams. To verify whether PROTOADA addresses the failure mode identified in Fig. 1, we replay the two controlled streams from the preliminary study and compare full continual learners. As shown in Fig. 5, all methods achieve similar Last accuracy on the same-format stream, mostly around 52–55%, indicating that the response protocol itself does not cause severe degradation when it remains consistent. In contrast, the format-varied stream leads to a large drop for FineTune, whose Last accuracy decreases from 54.1% to 17.3% with $\Delta_{\text{fmt}} = 36.9$. MoELoRA, HiDe-LLaVA, and the semantic-only variant of PROTOADA reduce this gap but still suffer clear format-induced forgetting, with Δ_{fmt} ranging from 3.2 to 4.7. Once format-aware prototypes are enabled, the gap is almost eliminated, with PROTOADA reaching 54.7% on the format-varied stream and reducing Δ_{fmt} to only 0.2. This shows that the observed format mismatch is not only diagnostic but can be effectively mitigated by conditioning task allocation on response protocols.

Temporal Dynamics of Format-Aware Prototypes. The visualization in Fig. 2 shows that a lightweight format code separates tasks with different response protocols at a single snapshot, and a natural follow-up question is whether this structure remains stable while new tasks are continuously inserted. Fig. 6 tracks this effect along the UCIT stream. Compared with semantic-only prototypes, format-aware prototypes consistently improve the separability index, with ΔSI increasing from $+0.290$ at Task1 to $+0.863$ at Task5. Meanwhile, the predicted format code $\hat{c}^{\text{fmt}}$ remains highly aligned with the oracle c^{fmt}, with cosine similarity staying above 0.91 throughout the stream. These results indicate that the format prototype is not only a useful one-shot descriptor as suggested in Fig. 2, but also a stable basis for continual task assignment and task-agnostic retrieval.

6 CONCLUSION

We identify format-blind task assignment as a key source of catastrophic forgetting in MCIT and propose PROTOADA, which performs explicit format-aware task grouping, selective and efficient LoRA expansion, and geometry-aware update consolidation. By reusing compatible knowledge while preserving task-specific response conventions, PROTOADA achieves stronger and more reliable continual adaptation across a diverse set of challenging multimodal benchmarks.

Limitations and Future Work. Although PROTOADA performs strongly on current MCIT benchmarks, our evaluation is limited to the present task streams and response protocols. We also do not study explicit chain-of-thought supervision, as the current format code focuses on final-response structures. Future work will investigate CoT-aware descriptors, longer streams, and broader multimodal continual-learning scenarios.

AI USE STATEMENT

In this work, we used generative AI tools for language editing and proofreading, including improving the grammar, spelling, fluency, clarity, and consistency of the manuscript, and identifying and correcting potential language and expression errors. All AI-assisted text was reviewed and verified by the authors, who take full responsibility for the final content of this work.

REFERENCES

Jinze Bai, Shuai Bai, Shusheng Yang, Shijie Wang, Sinan Tan, Peng Wang, Junyang Lin, Chang Zhou, and Jingren Zhou. Qwen-vl: A versatile vision-language model for understanding, localization, text reading, and beyond. *arXiv preprint arXiv:2308.12966*, 2023.

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

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*, pp. 57817–57840, 2024.

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

Kunyu Feng, Yue Ma, Xinhua Zhang, Boshi Liu, Yikuang Yuluo, Yinhan Zhang, Runtao Liu, Hongyu Liu, Zhiyuan Qin, Shanhui Mo, et al. Follow-your-instruction: A comprehensive mllm agent for world data synthesis. *arXiv preprint arXiv:2508.05580*, 2025.

Robert M French. Catastrophic forgetting in connectionist networks. *Trends in cognitive sciences*, 3 (4):128–135, 1999.

Robot M French and André Ferrara. Modeling time perception in rats: Evidence for catastrophic interference in animal learning. In *Proceedings of the Annual Conference of the Cognitive Science Society*, pp. 173–178. Psychology Press, 2020.

Runwei Guan, Rongsheng Hu, Shangshu Chen, Ningyuan Xiao, Xue Xia, Jiayang Liu, Beibei Chen, Ziren Tang, Ningwei Ouyang, Shaofeng Liang, et al. Roadscenevqa: Benchmarking visual question answering in roadside perception systems for intelligent transportation system. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pp. 4366–4375, 2026.

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 Annual Meeting of the Association for Computational Linguistics*, 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 international conference on computer vision*, pp. 1325–1335, 2025b.

Jiawei Guo, Tianyu Zheng, Yizhi Li, Yuelin Bai, Bo Li, Yubo Wang, King Zhu, Graham Neubig, Wenhu Chen, and Xiang Yue. Mammoth-vl: Eliciting multimodal reasoning with instruction tuning at scale. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, pp. 13869–13920, 2025c.

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/CVF Computer Vision and Pattern Recognition Conference*, pp. 3608–3617, 2018.

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 international conference on computer vision*, pp. 8340–8349, 2021.

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

Ronghang Hu and Amanpreet Singh. Unit: Multimodal multitask learning with a unified transformer. In *Proceedings of the IEEE international conference on computer vision*, pp. 1439–1449, 2021.

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 IEEE/CVF Computer Vision and Pattern Recognition Conference*, pp. 19608–19617, 2025.

Justin Johnson, Bharath Hariharan, Laurens Van Der Maaten, Li Fei-Fei, C Lawrence Zitnick, and Ross Girshick. Clevr: A diagnostic dataset for compositional language and elementary visual reasoning. In *Proceedings of the IEEE/CVF Computer Vision and Pattern Recognition Conference*, pp. 2901–2910, 2017.

Virginia Klema and Alan Laub. The singular value decomposition: Its computation and some applications. *IEEE Transactions on automatic control*, 25(2):164–176, 1980.

Minjae Lee, Minhyuk Seo, Tingyu Qu, Tinne Tuytelaars, and Jonghyun Choi. Oasis: Online sample selection for continual visual instruction tuning. *arXiv preprint arXiv:2506.02011*, 2025.

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 Annual Meeting of the Association for Computational Linguistics*, pp. 14369–14387, 2024.

Songze Li, Mingyu Gao, Tonghua Su, Xu-Yao Zhang, and Zhongjie Wang. Multimodal continual instruction tuning with dynamic gradient guidance. *arXiv preprint arXiv:2511.15164*, 2025.

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. In *Advances in Neural Information Processing Systems*, pp. 34892–34916, 2023.

Wenzhuo Liu, Fei Zhu, Haiyang Guo, Longhui Wei, and Cheng-Lin Liu. Llava-c: Continual improved visual instruction tuning. *arXiv preprint arXiv:2506.08666*, 2025a.

Yuyang Liu, Qiuhe Hong, Linlan Huang, Alexandra Gomez-Villa, Dipam Goswami, Xialei Liu, Joost van de Weijer, and Yonghong Tian. Continual learning for vlms: A survey and taxonomy beyond forgetting. *arXiv preprint arXiv:2508.04227*, 2025b.

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.

Ahmed Masry, Xuan Long Do, Jia Qing Tan, Shafiq Joty, and Enamul Hoque. Chartqa: A benchmark for question answering about charts with visual and logical reasoning. In *Findings of the association for computational linguistics*, pp. 2263–2279, 2022.

Minesh Mathew, Dimosthenis Karatzas, and CV Jawahar. Docvqa: A dataset for vqa on document images. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pp. 2200–2209, 2021.

Minesh Mathew, Viraj Bagal, Rubèn Tito, Dimosthenis Karatzas, Ernest Valveny, and CV Jawahar. Infographicvqa. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 1697–1706, 2022.

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, 2021.

Maryam Rahnemoonfar, Tashnim Chowdhury, Argho Sarkar, Debvrat Varshney, Masoud Yari, and Robin Roberson Murphy. Floodnet: A high resolution aerial imagery dataset for post flood scene understanding. *IEEE Access*, 9:89644–89654, 2021.

María Virginia Sabando, Pavol Ulbrich, Matías Selzer, Jan Byška, Jan Mičan, Ignacio Ponzoni, Axel J Soto, María Luján Ganuza, and Barbora Kozlíková. Chemva: interactive visual analysis of chemical compound similarity in virtual screening. *IEEE Transactions on Visualization and Computer Graphics*, 27(2):891–901, 2020.

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*, pp. 4942–4964, 2024.

Jun-Tao Tang, Yu-Cheng Shi, Zhen-Hao Xie, and Da-Wei Zhou. Prism: A plug-in reproducible infrastructure for scalable multimodal continual instruction tuning. *arXiv preprint arXiv:2605.26110*, 2026.

Shengbang Tong, David Fan, Jiachen Li, Yunyang Xiong, Xinlei Chen, Koustuv Sinha, Michael Rabbat, Yann LeCun, Saining Xie, and Zhuang Liu. Metamorph: Multimodal understanding and generation via instruction tuning. In *Proceedings of the IEEE international conference on computer vision*, pp. 17001–17012, 2025.

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

Runyu Wang, Peng Ping, Zhengyu Guo, Xiaoye Zhang, Quan Shi, Liting Zhou, and Tianbo Ji. Loki: Low-damage knowledge implanting of large language models. *arXiv preprint arXiv:2505.22120*, 2025a.

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*, pp. 10658–10671, 2023.

Ziqi Wang, Chang Che, Qi Wang, Yangyang Li, Zenglin Shi, and Meng Wang. Smolora: Exploring and defying dual catastrophic forgetting in continual visual instruction tuning. In *Proceedings of the IEEE international conference on computer vision*, pp. 177–186, 2025b.

Zhen-Hao Xie, Jun-Tao Tang, Yu-Cheng Shi, Han-Jia Ye, De-Chuan Zhan, and Da-Wei Zhou. Same: Stabilized mixture-of-experts for multimodal continual instruction tuning. *arXiv preprint arXiv:2602.01990*, 2026.

Jihan Yang, Shusheng Yang, Anjali W Gupta, Rilyn Han, Li Fei-Fei, and Saining Xie. Thinking in space: How multimodal large language models see, remember, and recall spaces. In *Proceedings of the IEEE/CVF Computer Vision and Pattern Recognition Conference*, pp. 10632–10643, 2025a.

Shuo Yang, Caren Han, Siwen Luo, and Eduard Hovy. Magic-vqa: Multimodal and grounded inference with commonsense knowledge for visual question answering. In *Findings of the Proceedings of the Annual Meeting of the Association for Computational Linguistics: Proceedings of the Annual Meeting of the Association for Computational Linguistics*, pp. 16967–16986, 2025b.

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

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 Conference on Empirical Methods in Natural Language Processing*, pp. 12137–12152, 2025.

Aimen Zerroug, Mohit Vaishnav, Julien Colin, Sebastian Musslick, and Thomas Serre. A benchmark for compositional visual reasoning. *Advances in Neural Information Processing Systems*, 35: 29776–29788, 2022.

Shengyu Zhang, Linfeng Dong, Xiaoya Li, Sen Zhang, Xiaofei Sun, Shuhe Wang, Jiwei Li, Runyi Hu, Tianwei Zhang, Guoyin Wang, et al. Instruction tuning for large language models: A survey. *ACM Computing Surveys*, 2023a.

Xiaoman Zhang, Chaoyi Wu, Ziheng Zhao, Weixiong Lin, Ya Zhang, Yanfeng Wang, and Weidi Xie. Pmc-vqa: Visual instruction tuning for medical visual question answering. *arXiv preprint arXiv:2305.10415*, 2023b.

Hengyuan Zhao, Ziqin Wang, Qixin Sun, Kaiyou Song, Yilin Li, Xiaolin Hu, Qingpei Guo, and Si Liu. Llava-cmoe: Towards continual mixture of experts for large vision-language models. *arXiv preprint arXiv:2503.21227*, 2025.

Da-Wei Zhou, Qi-Wei Wang, Zhi-Hong Qi, Han-Jia Ye, De-Chuan Zhan, and Ziwei Liu. Class-incremental learning: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(12):9851–9873, 2024.

Jeffrey Zhou, Tianjian Lu, Swaroop Mishra, Siddhartha Brahma, Sujoy Basu, Yi Luan, Denny Zhou, and Le Hou. Instruction-following evaluation for large language models. *arXiv preprint arXiv:2311.07911*, 2023.

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

Zhen Zhu, Yiming Gong, Yao Xiao, Yaoyao Liu, and Derek Hoiem. How to teach large multimodal models new skills. *arXiv preprint arXiv:2510.08564*, 2025.

APPENDIX OVERVIEW

The appendices are organized as follows. Appendix A provides the theoretical analysis of geometry-aware projection. Appendix B summarizes the complete training and inference procedures. Appendix C details the controlled-stream construction and statistical validation, while Appendix D explains the format statistics used to construct task codes. Appendix E studies task-order robustness, group initialization, and retrieval granularity. Appendices F and G report protocol-compliance results and the comparison with format-enhanced baselines. Appendix H presents the format-classifier and late-fusion ablations. Appendix I evaluates stronger backbones, general-purpose baselines, and grounding tasks. Appendix J analyzes protocol violations and controls for dataset identity. Finally, Appendix K briefly describes the compared methods.

A THEORETICAL ANALYSIS OF GEOMETRY-AWARE PROJECTION

In this section, we provide a theoretical justification for the geometry-aware decomposition in Eq. (16). We show that projecting the purified task update onto the group memory subspace extracts the optimal group-compatible component under the Frobenius norm, while the remaining residual captures task-specific directions that are orthogonal to the existing group memory.

A.1 PRELIMINARIES

After task t is assigned to group g_t, we obtain its purified LoRA update

$$\Delta\hat{\Theta}_t = \mathbf{M}_t \odot \Delta\Theta_t, \tag{23}$$

where $\mathbf{M}_t$ is the activation-aware binary mask. Let $\mathcal{U}_{g_t}$ denote the subspace spanned by the current shared memory of group g_t. Intuitively, $\mathcal{U}_{g_t}$ contains the parameter directions that have been consistently reused by previous tasks in the same group. Therefore, an update direction lying in $\mathcal{U}_{g_t}$ can be viewed as group-compatible, while directions outside this subspace are more likely to encode task-specific deviations.

For notational simplicity, we consider the vectorized form of the update:

$$\mathbf{x}_t = \mathrm{vec}(\Delta\hat{\Theta}_t) \in \mathbb{R}^d. \tag{24}$$

Let $\mathbf{U}_{g_t} \in \mathbb{R}^{d\times r}$ be an orthonormal basis of $\mathcal{U}_{g_t}$, where r is the dimension of the group subspace and

$$\mathbf{U}_{g_t}^\top \mathbf{U}_{g_t} = \mathbf{I}. \tag{25}$$

The orthogonal projection matrix onto $\mathcal{U}_{g_t}$ is then

$$\mathbf{P}_{g_t} = \mathbf{U}_{g_t}\mathbf{U}_{g_t}^\top. \tag{26}$$

A.2 OPTIMALITY OF THE PROJECTED SHARED UPDATE

We first show that the projected component is the closest group-compatible approximation to the purified task update.

Proposition 1. Given the purified task update $\Delta\hat{\Theta}_t$ and the group memory subspace $\mathcal{U}_{g_t}$, the projected update

$$\Delta\Theta_t^{\mathrm{share}} = \mathrm{proj}_{\mathcal{U}_{g_t}}(\Delta\hat{\Theta}_t) \tag{27}$$

is the optimal group-compatible approximation to $\Delta\hat{\Theta}_t$ under the Frobenius norm:

$$\Delta\Theta_t^{\mathrm{share}} = \arg\min_{\mathbf{Z}\in\mathcal{U}_{g_t}} \left\|\Delta\hat{\Theta}_t - \mathbf{Z}\right\|_F^2. \tag{28}$$

Proof. Using the vectorized notation, problem (28) is equivalent to

$$\min_{\mathbf{z}\in\mathcal{U}_{g_t}} \|\mathbf{x}_t - \mathbf{z}\|_2^2. \tag{29}$$

Since $\mathbf{z} \in \mathcal{U}_{g_t}$, there exists a coefficient vector $\mathbf{a} \in \mathbb{R}^r$ such that

$$\mathbf{z} = \mathbf{U}_{g_t}\mathbf{a}. \tag{30}$$

Therefore, the optimization becomes

$$\min_{\mathbf{a}} \|\mathbf{x}_t - \mathbf{U}_{g_t}\mathbf{a}\|_2^2. \tag{31}$$

Taking the derivative of Eq. (31) with respect to $\mathbf{a}$ and setting it to zero gives

$$-2\mathbf{U}_{g_t}^\top (\mathbf{x}_t - \mathbf{U}_{g_t}\mathbf{a}) = 0. \tag{32}$$

Thus,

$$\mathbf{U}_{g_t}^\top \mathbf{x}_t = \mathbf{U}_{g_t}^\top \mathbf{U}_{g_t}\mathbf{a}. \tag{33}$$

Because $\mathbf{U}_{g_t}$ is orthonormal, we have

$$\mathbf{U}_{g_t}^\top \mathbf{U}_{g_t} = \mathbf{I}. \tag{34}$$

Therefore, the optimal coefficient is

$$\mathbf{a}^* = \mathbf{U}_{g_t}^\top \mathbf{x}_t. \tag{35}$$

Substituting $\mathbf{a}^*$ back into $\mathbf{z} = \mathbf{U}_{g_t}\mathbf{a}$ yields

$$\mathbf{z}^* = \mathbf{U}_{g_t}\mathbf{U}_{g_t}^\top \mathbf{x}_t = \mathbf{P}_{g_t}\mathbf{x}_t. \tag{36}$$

Returning to the matrix form, this gives

$$\Delta\Theta_t^{\text{share}} = \text{proj}_{\mathcal{U}_{g_t}}(\Delta\hat{\Theta}_t). \tag{37}$$

Hence, the projection is the minimizer of Eq. (28). □

This result shows that the projection is not an arbitrary decomposition. Among all updates that lie in the group memory subspace, it is the one that best reconstructs the current task update.

A.3 Orthogonality of the Task-specific Residual

After extracting the group-compatible component, the residual is defined as

$$\Delta\Theta_t^{\text{res}} = \Delta\hat{\Theta}_t - \Delta\Theta_t^{\text{share}}. \tag{38}$$

We next show that this residual is orthogonal to all directions in the group subspace.

Proposition 2. The residual component satisfies

$$\langle \Delta\Theta_t^{\text{res}}, \mathbf{Z}\rangle_F = 0, \qquad \forall \mathbf{Z} \in \mathcal{U}_{g_t}. \tag{39}$$

Proof. In vectorized form, the residual is

$$\mathbf{x}_t^{\text{res}} = \mathbf{x}_t - \mathbf{P}_{g_t}\mathbf{x}_t = \mathbf{x}_t - \mathbf{U}_{g_t}\mathbf{U}_{g_t}^\top \mathbf{x}_t. \tag{40}$$

For any $\mathbf{z} \in \mathcal{U}_{g_t}$, there exists $\mathbf{b} \in \mathbb{R}^r$ such that

$$\mathbf{z} = \mathbf{U}_{g_t}\mathbf{b}. \tag{41}$$

Then,

$$\langle \mathbf{x}_t^{\text{res}}, \mathbf{z}\rangle = \left(\mathbf{x}_t - \mathbf{U}_{g_t}\mathbf{U}_{g_t}^\top \mathbf{x}_t\right)^\top \mathbf{U}_{g_t}\mathbf{b} \tag{42}$$

$$= \mathbf{x}_t^\top \mathbf{U}_{g_t}\mathbf{b} - \mathbf{x}_t^\top \mathbf{U}_{g_t}\mathbf{U}_{g_t}^\top \mathbf{U}_{g_t}\mathbf{b}. \tag{43}$$

Since $\mathbf{U}_{g_t}^\top \mathbf{U}_{g_t} = \mathbf{I}$, we have

$$\mathbf{U}_{g_t}^\top \mathbf{U}_{g_t}\mathbf{b} = \mathbf{b}. \tag{44}$$

Thus,

$$\langle \mathbf{x}_t^{\text{res}}, \mathbf{z}\rangle = \mathbf{x}_t^\top \mathbf{U}_{g_t}\mathbf{b} - \mathbf{x}_t^\top \mathbf{U}_{g_t}\mathbf{b} \tag{45}$$

$$= 0. \tag{46}$$

Therefore, the residual is orthogonal to the group memory subspace. Returning to the matrix form gives Eq. (39). □

Algorithm 1 ProtoAda Training (per task t)

Require: Task $\mathcal{T}_t$; frozen backbone Θ_0, encoders $(E_{\text{text}}, E_{\text{vis}})$; groups $\{\mathbf{p}_g, \mathbf{A}_g, \mathcal{U}_g\}$; global adapter $\mathbf{A}^{\text{shared}}$; residual bank
Ensure: Updated group adapters, residuals, and format predictor f_{fmt}
1: **Prototype construction:** Compute $\mathbf{p}_t^{\text{text}}$, $\mathbf{p}_t^{\text{vis}}$, and format code $\mathbf{c}_t^{\text{fmt}}$; fuse into task prototype $\mathbf{q}_t$ (Eqs. (4)–(7)).
2: **Format-aware assignment:** Compute similarities $s_{t,g}$ (Eq. (8)); assign to g_t or init new group via adaptive threshold (Eq. (9)); update prototype $\mathbf{p}_{g_t}$ (Eq. (10)).
3: **Adaptive growth:** Estimate layer sensitivity γ_ℓ via gradients (Eq. (12)); select top-M layers $\mathcal{S}_t$ using γ_ℓ and group usage $u_{g_t,\ell}$ (Eq. (13)).
4: **Task training:** Init LoRA on $\mathcal{S}_t$; optimize with $\mathcal{L}_{\text{ce}}$ to get update $\Delta\Theta_t$; train predictor $\hat{\mathbf{c}}_t^{\text{fmt}} = f_{\text{fmt}}(\mathbf{p}_t^{\text{text}})$ with MSE (Eq. (11)).
5: **Geometry consolidation:** Purify $\Delta\Theta_t$ with activation norm to get $\Delta\hat{\Theta}_t$ (Eqs. (14)–(15)); project onto $\mathcal{U}_{g_t}$ to decouple $\Delta\Theta_t^{\text{share}}$ and $\Delta\Theta_t^{\text{res}}$ (Eq. (16)).
6: **Memory update:** Absorb $\Delta\Theta_t^{\text{share}}$ into $\mathbf{A}_{g_t}$ with weight λ_{t,g_t} (Eq. (17)); compress $\Delta\Theta_t^{\text{res}}$ via SVD and store as $\mathbf{R}_t$ (Eq. (18)).
7: **Maintenance (every K tasks):** Promote dominant directions to $\mathbf{A}^{\text{shared}}$ (Eqs. (19)–(20)); prune redundant residuals based on coverage ratio ρ_t (Eq. (21)).

A.4 Interpretation for Continual Instruction Tuning

The above results provide a geometric explanation for the proposed consolidation strategy. The shared component

$$\Delta\Theta_t^{\text{share}} = \text{proj}_{\mathcal{U}_{g_t}}(\Delta\hat{\Theta}_t) \tag{47}$$

is the best approximation of the current task update within the group-compatible subspace. Therefore, absorbing this component into the group adapter introduces the part of the update that is most consistent with the historical memory of the assigned group.

Meanwhile, the residual

$$\Delta\Theta_t^{\text{res}} = \Delta\hat{\Theta}_t - \Delta\Theta_t^{\text{share}} \tag{48}$$

contains directions that cannot be represented by the current group memory. Because it is orthogonal to $\mathcal{U}_{g_t}$, directly merging it into the shared group adapter may introduce task-specific deviations into the group-level parameters. This is especially harmful in multimodal continual instruction tuning, where different tasks may share similar image-text semantics but require different response protocols.

Therefore, the projection-based decomposition naturally supports the design of ProtoAda:

- the group-aligned component is consolidated into the shared group adapter for parameter reuse;
- the orthogonal component is stored separately as a compact residual to preserve task-specific behavior;
- the orthogonality between the two parts reduces interference between reusable knowledge and task-specific response conventions.

In this sense, the geometry-aware decomposition provides an optimal and interpretable mechanism for extracting group-compatible updates from task-specific LoRA increments.

B Algorithmic Description of ProtoAda

Algorithm 1 summarizes the per-task training procedure of ProtoAda in the continual instruction-tuning stream, while Algorithm 2 presents the corresponding task-agnostic inference procedure. For each incoming task, ProtoAda first constructs a format-aware prototype from visual semantics, language semantics, and response-format statistics, and uses it to assign the task to a compatible memory group or initialize a new group. After assignment, the method performs prototype-conditioned adaptive growth by combining layer-wise gradient sensitivity with the historical usage pattern of the assigned group, so that LoRA capacity is introduced only at layers that are important for the current task and reusable within the group. The selected LoRA modules are then optimized with the task loss,

Algorithm 2 ProtoAda Inference (Task-agnostic)

Require: Input (v, q); frozen backbone Θ_0, encoders $(E_{\text{text}}, E_{\text{vis}})$; predictor f_{fmt}; global adapter $\mathbf{A}^{\text{shared}}$; group adapters $\{\mathbf{A}_g\}$; residual bank $\{\mathbf{R}_r\}$
Ensure: Generated response y
1: **Input processing:** Compute descriptors $\mathbf{p}^{\text{text}}$ and $\mathbf{p}^{\text{vis}}$ via frozen encoders; predict format code $\hat{\mathbf{c}}^{\text{fmt}} = f_{\text{fmt}}(\mathbf{p}^{\text{text}})$ (Eq. (11)).
2: **Query prototype construction:** Fuse visual, linguistic descriptors, and predicted $\hat{\mathbf{c}}^{\text{fmt}}$ to build the query prototype $\mathbf{q}_{\text{query}}$ (Eqs. (6)–(7)).
3: **Memory retrieval:** Compute similarities s_g between $\mathbf{q}_{\text{query}}$ and group prototypes $\mathbf{p}_g$; retrieve Top-K groups and calculate routing weights w_g (Eq. (22)).
4: **Residual activation:** Compare $\mathbf{q}_{\text{query}}$ with stored residual prototypes $\mathbf{q}_r$ to retrieve the subset of compatible task-specific residuals $\{\mathbf{R}_r\}$.
5: **Response generation:** Augment Θ_0 with $\mathbf{A}^{\text{shared}}$, retrieved group adapters (scaled by w_g), and selected residuals $\{\mathbf{R}_r\}$; generate response y in a single forward pass.

while the format predictor is trained to infer the response-format code from input-side descriptors for task-agnostic inference. After training, the learned update is purified by activation-aware masking and decomposed according to the group memory geometry: the group-compatible component is absorbed into the group adapter, whereas the remaining task-specific component is compressed and stored as a residual. Periodic maintenance further promotes dominant reusable directions into the global shared adapter and prunes redundant residuals. This procedure enables PROTOADA to reuse compatible knowledge across tasks while preserving task-specific response conventions.

C CONTROLLED STREAM CONSTRUCTION AND STATISTICAL VALIDATION

This appendix specifies the controlled-stream settings used in two main-text experiments: the preliminary format-versus-semantic comparison in the Introduction (Fig. 1) and the subsequent stress test in Sec. 5.3 (Fig. 5). Both experiments use the same FmtGap and VisGap constructions: FmtGap varies response protocols while holding visual-semantic content fixed, whereas VisGap varies visual-semantic content while holding the response protocol fixed. The following sections provide the shared construction, example templates, and evaluation details.

C.1 DIFFICULTY AND FACTOR MATCHING

The two controlled streams are designed to compare format variation with visual-semantic variation under a matched parameter-sharing setup. Flickr30k and VizWiz are selected because independent LoRA fine-tuning reaches closely matched accuracy on the two datasets, and both use brief description as their native response protocol. This reduces the influence of task difficulty and native-format differences in the semantic-varied comparison.

C.2 STREAM CONSTRUCTION

In the format-varied stream, we start from VizWiz and rewrite each question–answer pair into five response protocols: brief description, detailed description, short answer, multiple-choice answer, and yes/no answer. The image and visual-semantic content remain unchanged across phases; label space, response entropy, and response length vary only as consequences of the required protocol. In the semantic-varied stream, the output protocol is fixed to brief description while the data source alternates between Flickr30k and VizWiz, introducing a visual-domain and annotation-style shift with the response protocol held constant. Each phase contains 3,000 samples, uses the same sampling procedure, and is trained with the same hyperparameters.

To make the five response protocols concrete, Tab. 4 gives one image-instruction input and output example for each protocol, using the same image while varying the instruction and expected response form.

Table 4: Template for one input-output example per response protocol in the format-varied stream.

Response Protocol	Instruction	Output Example
Brief description	What is happening in the image? Generate a brief caption for the image.	A girl is diving into a pool and is surrounded by bubbles
Detailed description	What is happening in the image? Describe the scene in detail.	A girl is diving into a pool, with bubbles surrounding her as she enters and moves through the water. The scene captures the motion of her dive and the disturbance created in the water around her.
Short/one-word answer	What is the girl doing? Answer with one word only.	Diving
Multiple-choice answer	What is the girl doing? Choose one option and answer with the option letter only. (A) Diving into a pool (B) Riding a bicycle (C) Sitting beside a pool (D) Playing basketball.	A
Yes/no answer	Is the girl diving into a pool? Answer Yes or No.	Yes

C.3 Multi-seed validation

For each stream, we repeat the experiment with five random seeds, varying both data sampling and parameter initialization. Let A_0^{before} denote the Task-0 accuracy before subsequent tasks are learned and A_0^{after} denote the accuracy after the full stream. We report the Task-0 performance drop in percentage points as

$$\Delta_0 = A_0^{\text{before}} - A_0^{\text{after}}. \tag{49}$$

The correct correspondence between the two streams is: the semantic-varied stream gives $\Delta_0 = 5.39 \pm 0.40$, whereas the format-varied stream gives $\Delta_0 = 56.94 \pm 2.01$. The format-varied stream therefore produces a much larger degradation than the semantic-varied stream, and the inter-stream gap is substantially larger than the variation across seeds. These results support the conclusion that, after controlling the data scale and visual-semantic content as described above, format variation has a pronounced effect on sequential LoRA adaptation. They do not imply that protocol variation is the only factor that changes between the streams, since label-space and response-length changes are inherent consequences of changing the required output protocol.

D Construction of Format Statistics

To incorporate response protocols into task grouping, we derive a task-level format code c_t^{fmt} from target responses. For each mini-batch, masked label positions are removed and only valid response tokens are used. The statistics are organized into four groups: mean of response length, variance of response length, token uncertainty, and template consistency. The mean of response length records the average normalized number of target tokens, which distinguishes short-answer, multiple-choice, and long-form generation protocols. The variance of response length measures the dispersion of response lengths across samples, reflecting whether the task follows a fixed output template or allows variable-length responses. Token uncertainty is measured by the normalized entropy of the empirical token distribution and the unique-token ratio, characterizing how diverse or constrained the target responses are. Template consistency is estimated from the repetition ratio and coarse token-id distribution statistics, including the normalized token-id mean and standard deviation as well as low- and middle-band token-id ratios, which serve as tokenizer-independent proxies for option markers, numbers, punctuation, and repeated answer patterns. We average these batch-level statistics over the warm-up batches to obtain a fixed-dimensional task-level descriptor $\mathbf{c}_t^{\text{fmt}}$, which is then used for format-aware prototype construction.

Table 5: Late-fusion format classification versus full format-aware integration on UCIT.

Configuration	Avg. Acc. (%)	Routing Acc. (%)
Semantic-only + late-fusion classifier	72.04	97.9
Full PROTOADA	**74.66**	**99.7**

E TASK ORDER, INITIALIZATION, AND RETRIEVAL GRANULARITY

To evaluate whether PROTOADA depends on a favorable task curriculum, we examine its robustness to task ordering on the UCIT benchmark and compare it with SAME. Specifically, we report average accuracy under the Original, Reverse, and Random sequences. The sequence definitions are:

- **Original**: the canonical task order defined by the benchmark protocol;
- **Reverse**: the exact reverse of the original sequence, where the last task is presented first;
- **Random**: a randomly shuffled task order. For reproducibility, we fix the random permutation as:

UCIT: Flickr30k → CLEVR → ArxivQA → ImageNet-R → IconQA → VizCap.

As shown in Tab. 8, PROTOADA maintains stable performance under different task orders and consistently outperforms SAME across all three orders, with mean accuracy of $72.22 \pm 1.73\%$ versus $70.70 \pm 1.76\%$. Although reversing or shuffling the task order introduces a moderate performance drop, PROTOADA still preserves competitive accuracy without relying on the canonical curriculum. This robustness mainly benefits from format-aware task grouping, prototype-conditioned adaptive growth, and geometry-aware consolidation, which respectively reduce incompatible sharing, unnecessary perturbation, and interference between reusable and task-specific directions.

Singleton-group initialization. At the beginning of the stream, the first task initializes a new group with its task prototype, so no running statistics are required. For a group containing only this task, the running mean and standard deviation in Eq. (9) are not estimated. Instead, a fixed initialization threshold τ_{init} is used to decide whether a subsequent task is admitted to the singleton group or starts another group. Once a group contains at least two tasks, the similarities of its assigned tasks are used to initialize and update μ_g and σ_g, and the running-statistics criterion in Eq. (9) is used thereafter. This procedure avoids undefined variance estimates during initialization.

Task-level supervision and sample-level retrieval. The format predictor is trained with task-level format descriptors, but inference is performed for each test sample using its own input-side visual and textual descriptors. We therefore evaluate retrieval at the sample level rather than only at the task level. The query prototype constructed from each test sample is compared with stored group prototypes, and the predicted group is checked against the sample's ground-truth task identity.

As reported in Tab. 9, the sample-level retrieval accuracy is 99.7% on UCIT and 98.0% on TriGap. These results indicate that task-level supervision transfers effectively to sample-level retrieval, while also making explicit the granularity used at training and test time.

F PROTOCOL COMPLIANCE AND TASK-LEVEL FORGETTING

The benchmark results do not imply that PROTOADA must achieve the best accuracy on every individual task. In particular, the difference between PROTOADA and CL-MoE on TriGap is affected by capacity allocation and task order: CLEVR-Math is the final task in the sequence, and without explicit parameter isolation, CL-MoE can become strongly biased toward this newest task. Although this bias can improve the final-task score, it is accompanied by more severe forgetting on earlier tasks. In contrast, PROTOADA partitions capacity across isolated groups and therefore preserves earlier-task behaviors more effectively, resulting in the higher overall average accuracy reported for PROTOADA than for CL-MoE in Tab. 1.

To directly measure the format-related component of forgetting on a real benchmark, we evaluate protocol compliance on the format-restricted UCIT tasks ArxivQA, IconQA, and CLEVR. A prediction

is counted as compliant when its externally visible response follows the answer protocol required by the task; correctness is evaluated separately using the benchmark accuracy metric. Thus, protocol compliance measures whether the model answers in the appropriate form, whereas accuracy also depends on the knowledge required to produce the correct answer. The aggregate results are reported in Tab. 6.

Table 6: Protocol compliance and average accuracy on the format-restricted UCIT tasks ArxivQA, IconQA, and CLEVR.

Method	Compliance (%)	Avg. Acc. (%)
FT-LoRA	73.0	37.90
MoE-LoRA	93.7	60.79
HiDe-LLaVA	96.7	60.08
SAME	98.3	73.07
PROTOADA	**99.1**	**74.66**

As shown in Tab. 6, PROTOADA attains the highest compliance among the compared continual learners while also achieving the best average accuracy. This supports the interpretation that its improvement is associated with reducing protocol corruption in addition to mitigating ordinary knowledge forgetting. The two metrics should nevertheless be viewed as complementary measurements rather than as a complete decomposition of every incorrect prediction into a format error or a knowledge error.

G COMPARISON AGAINST BASELINE METHODS WITH FORMAT CODE

To isolate the contribution of the format code from the rest of the proposed framework, we augment SAME with the same format-aware code used for task assignment while keeping SAME's spectral consolidation unchanged. All configurations use the same UCIT backbone, data, task order, and training protocol, with the UCIT initial LoRA rank set to $r_0 = 16$. The comparison is reported in Tab. 7.

Table 7: Comparison between SAME, SAME with format-code task assignment, and PROTOADA on UCIT.

Method	Avg. Acc. (%)	Params (M)	Train Time (h)
SAME	73.07	281	4.24
SAME + format code	**74.92**	281	4.31
PROTOADA	74.66	**235**	**3.35**

As shown in Tab. 7, adding the format code improves SAME by 1.85 percentage points, showing that response-format information can improve task assignment beyond a particular consolidation implementation. SAME with the format code slightly exceeds PROTOADA in average accuracy, but requires $1.29\times$ the training time and $1.20\times$ the stored parameters. PROTOADA therefore provides a more favorable accuracy–efficiency trade-off under the same benchmark setting, and this difference becomes increasingly relevant as the task stream grows.

H FORMAT CLASSIFIER AND LATE-FUSION ABLATION

As shown in Fig. 6, the predicted format code has cosine similarity above 0.91 with the oracle code, indicating that response-format information is already decodable from the semantic-only instruction representation. This observation does not make the code unnecessary; it instead raises the question of whether the information should be appended only after semantic routing or integrated throughout the parameter-sharing process. To examine this distinction, we compare full PROTOADA with a

Table 8: Task-order robustness on **UCIT**. We report average accuracy (%) under three task sequences and their mean ± standard deviation.

Method	Original	Reverse	Random	Mean ± Std.
PROTOADA	74.66	70.83	71.17	72.22 ± 1.73
SAME	73.07	70.19	68.84	70.70 ± 1.76

Table 9: Sample-level group retrieval accuracy under task-level predictor supervision.

Benchmark	Sample-level Retrieval Acc. (%)
UCIT	99.7
TriGap	98.0

Table 10: General-purpose zero-shot backbones and the continual-tuning result of PROTOADA on TriGap and UCIT. Zero-shot and continual-tuning results are shown as separate regimes.

Method	TriGap Avg. (%)	UCIT Avg. (%)
LLaVA-v1.5-7B zero-shot	21.60	32.34
Qwen3-VL-4B zero-shot	42.63	46.61
Qwen3-VL-8B zero-shot	43.85	53.70
GLM-4.1V-9B zero-shot	42.14	39.14
PROTOADA (LLaVA-v1.5-7B)	**47.23**	**74.66**

late-fusion baseline that adds a format classifier after the semantic-only router while keeping the remaining settings unchanged. The results are reported in Tab. 5.

According to the results shown in Tab. 5, full PROTOADA improves the average accuracy by 2.62 percentage points and the routing accuracy by 1.8 percentage points. The result shows that format information is more effective when integrated into the task prototype and used throughout task grouping, capacity allocation, and inference than when appended as a late-fusion signal. Accordingly, our contribution does not rely on claiming that routing, consolidation, or masking is individually unrelated to prior work. Rather, the central contribution is a unified response-protocol-aware design that uses format information to separate incompatible tasks while safely reusing compatible updates, together with an empirical validation of this mechanism in MCIT.

I STRONGER BACKBONES, GENERAL-PURPOSE BASELINES, AND GROUNDING

To examine whether the observed failure mode is tied to the LLaVA-v1.5-7B backbone or to VQA-style tasks, we provide additional comparisons with stronger general-purpose MLLMs, a cross-backbone controlled stream, and a grounding-extended UCIT stream. These experiments are intended to separate zero-shot generalization from continual adaptation and to test format-aware routing on a structured output protocol.

I.1 GENERAL-PURPOSE ZERO-SHOT BASELINES

We first evaluate several modern general-purpose MLLMs in zero-shot mode on TriGap and UCIT. The results are reported in Tab. 10. Zero-shot models and continual-tuning methods occupy different regimes: the former measure the capabilities already present in a pretrained model, whereas the latter measure the ability to acquire and retain new task knowledge sequentially. We therefore report them as separate references rather than treating them as a single directly comparable ranking.

As shown in Tab. 10, the stronger zero-shot models substantially improve over the LLaVA-v1.5-7B zero-shot reference, but still underperform continual tuning, especially on UCIT. This gap is expected because a pretrained model cannot anticipate all post-training knowledge and task requirements: new events, scientific and medical findings, regulations, and industrial standards continue to emerge after deployment. Such capabilities must be acquired and retained as new tasks arrive, which is precisely the setting addressed by MCIT.

Table 11: Task-0 accuracy drop under semantic and format variation for two MLLM backbones.

Backbone	Semantic Variation	Format Variation
LLaVA-v1.5-7B	5.8	58.0
Qwen3-VL-8B	3.8	36.6

Table 12: Results on the grounding-extended seven-task UCIT stream.

Method	Grounding (Acc@0.5 IoU)	7-task Avg. Acc. (%)
MoE-LoRA	35.21	64.98
SAME	75.03	72.96
PROTOADA	**79.26**	**74.63**

I.2 CROSS-BACKBONE CONTROLLED VALIDATION

We then rerun the controlled continual-tuning experiment from Fig. 1 on Qwen3-VL-8B using the same task orders and training protocol as the LLaVA-v1.5-7B experiment. The results are summarized in Tab. 11.

As reported in Tab. 11, Qwen3-VL-8B shows a smaller absolute degradation, consistent with stronger pretrained instruction-following priors. Nevertheless, format variation still causes substantially greater degradation than semantic variation on both backbones. The controlled phenomenon is therefore observed in at least two MLLM families, while the evidence does not warrant extending the claim to every newer model without further experiments.

I.3 GROUNDING-EXTENDED UCIT

To test whether the method extends beyond VQA-style answer formats, we construct an extended UCIT stream by inserting a RefCOCO grounding task at the beginning of the original sequence. The task requires structured bounding-box coordinate outputs, which are fundamentally different from the natural-language responses in the remaining tasks. We evaluate grounding using accuracy at 0.5 intersection-over-union (Acc@0.5 IoU) and report the average accuracy over all seven tasks. The results are shown in Tab. 12.

As shown in Tab. 12, PROTOADA outperforms SAME by 4.23 percentage points on grounding and by 1.67 percentage points on the seven-task average. This result shows that format-aware routing remains effective for structurally distinct output spaces, rather than only for variations among natural-language VQA answer formats.

J PROTOCOL VIOLATION AND DATASET IDENTITY CONTROL

Our controlled stream establishes the existence of format-induced interference. To quantify how this failure mode appears on real benchmark data, we further analyze forgotten samples in UCIT. We call a sample forgotten when it is answered correctly immediately after its task is learned but answered incorrectly after the complete task stream. Among these forgotten samples, we measure the fraction whose final output violates the protocol required by the corresponding task.

Table 13: Protocol-violation rates among forgotten samples in UCIT.

Method	Violation Rate (%)
FT-LoRA	23.1
MoE-LoRA	14.5
SAME	3.1
PROTOADA (semantic-only)	3.9
PROTOADA (full)	**2.2**

Under this definition, the protocol-violation rates reported in Tab. 13 are 23.1% for FT-LoRA, 14.5% for MoE-LoRA, 3.1% for SAME, 3.9% for the semantic-only PROTOADA variant, and 2.2% for full PROTOADA. Thus, up to 23.1% of forgetting events under format-blind tuning manifest as protocol violations, whereas format-aware grouping reduces this rate to 2.2%. This analysis separates protocol corruption from knowledge forgetting more directly than accuracy alone.

We also examine whether the format code could be identifying dataset identity instead of response protocol. When different response formats are generated from the same VizWiz data, the format-aware code separates them despite identical visual content and task semantics. Conversely, Flickr30k and VizWiz produce similar format codes when both use the same brief-description protocol. These controls indicate that the code captures response-protocol information rather than dataset-specific cues.

K Brief Description of Compared Methods

Zero-shot. This baseline directly evaluates the frozen pre-trained LLaVA model on the complete task sequence without updating any parameters. It reflects the intrinsic cross-task generalization ability of the original multimodal backbone.

FT-LoRA. FT-LoRA performs continual instruction tuning with standard LoRA adapters. For each incoming task, the model is trained sequentially by updating only the LoRA parameters, while the pre-trained backbone remains fixed. No explicit mechanism is introduced to mitigate forgetting across tasks.

Replay-LoRA. Replay-LoRA extends sequential LoRA fine-tuning with an episodic memory buffer. A small number of samples from previous tasks are stored and mixed with the current task data during training, so that the model can rehearse earlier knowledge while adapting to new instructions.

MoE-LoRA Chen et al. (2024). MoE-LoRA equips the backbone with multiple LoRA experts and learns a routing function to combine their outputs. Instead of relying on a single adapter, the model dynamically aggregates expert-specific adaptations, providing additional capacity for heterogeneous tasks in continual tuning.

HiDe-LLaVA Guo et al. (2025a). HiDe-LLaVA adopts a task-aware expert decomposition strategy for multimodal continual instruction tuning. It maintains task-associated LoRA experts and activates the corresponding expert during training. At test time, the task identity is estimated by matching CLIP-based visual and textual anchors, which are then used to select the most relevant expert.

CL-MoE Huai et al. (2025). CL-MoE introduces an input-dependent mixture-of-experts mechanism for continual multimodal instruction tuning. It performs fine-grained routing at the layer and token levels, allowing different tokens to access different experts without requiring explicit task identifiers. Combined with memory replay, it serves as a strong task-agnostic continual learning baseline.

DISCO Guo et al. (2025b). DISCO builds task prototypes from CLIP image and text representations and uses them to guide expert aggregation. During inference, the similarity between the current input and stored task prototypes determines diagonal mask weights over LoRA experts, enabling prototype-based routing in the absence of explicit task IDs.

ModalPrompt Zeng et al. (2025). ModalPrompt is a prompt-based continual learning method that assigns learnable soft prompts to different tasks. In inference, the model retrieves the most relevant prompts according to dual-modal guidance from image and text features. A balancing coefficient controls the relative contribution of visual and textual cues during prompt selection.

SAME Xie et al. (2026). SAME exploits spectral information from LoRA parameter updates to identify stable task-related directions. It incrementally performs singular value decomposition over recent LoRA statistics and retains dominant spectral anchors for knowledge consolidation. A curvature-aware importance measure is further used to regulate parameter updates and reduce interference among tasks.