GEOSPEC: GEOMETRY-GUIDED EXPERT SPECIALIZATION FOR MULTIMODAL CONTINUAL INSTRUCTION TUNING

Anonymous authors
Paper under double-blind review

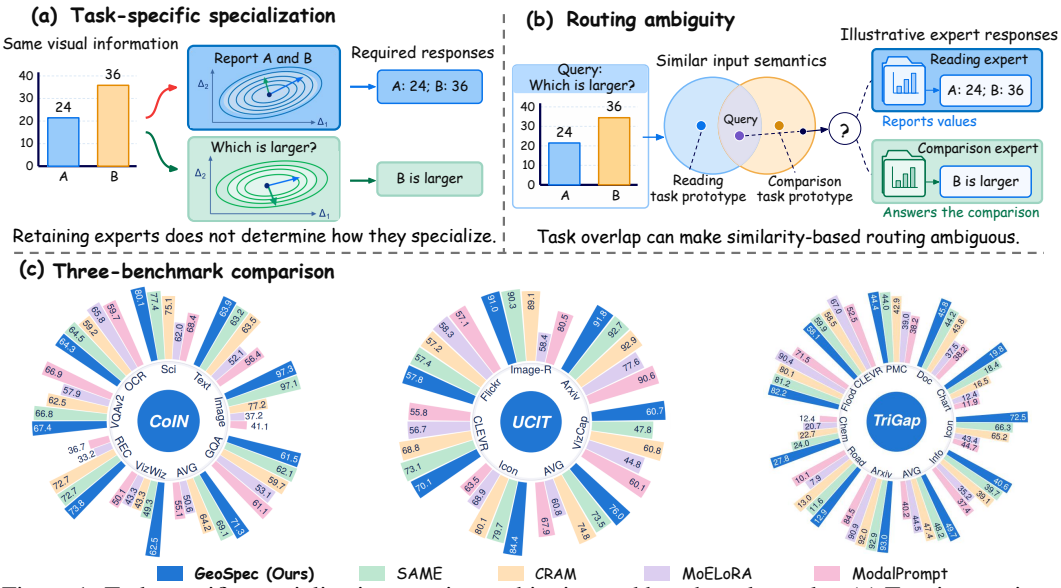

Figure 1: Task-specific specialization, routing ambiguity, and benchmark results. (a) Two instructions on the same chart require different responses; ellipses depict task-conditioned local-output geometries. (b) A query has similar affinity to two task prototypes, illustrating ambiguous prototype-based routing. (c) Task scores of GeoSpec and baselines on CoIN, UCIT, and TriGap.

ABSTRACT

Multimodal continual instruction tuning requires multimodal large language models to acquire new tasks sequentially while preserving previously learned capabilities. Progressive expert-based methods can isolate past experts from subsequent updates, but two challenges remain: how to adapt each new expert to the functional demands of the current task, and how to reliably access the accumulated experts without test-time task identities. We introduce GeoSpec, a progressive MoE-LoRA framework that addresses these challenges through geometry-guided expert adaptation and compatibility-guided routing. For each arriving task, Task-Conditioned Geometric Adaptation (TCGA) derives a task-conditioned local-output geometry from current-task activations and uses this geometry consistently to shape both expert updates and their fixed-rank realization. At inference, Compatibility-Guided Expert Routing combines task-prototype affinity with discriminative task evidence learned from answer-free, task-labeled support inputs, providing complementary signals for expert selection under task ambiguity. GeoSpec achieves the highest final AVG among the compared methods on all three evaluated MCIT benchmarks, reaching 71.34 on CoIN, 75.97 on UCIT, and 49.70 on TriGap.

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.

1 INTRODUCTION

Multimodal instruction tuning enables multimodal large language models (MLLMs) to learn tasks such as visual question answering, image recognition, and visual grounding from multimodal instruction–response data (Liu et al., 2023; Dai et al., 2023). In deployment, however, task requirements can extend beyond those covered by initial training (He et al., 2026; Cao et al., 2024; Zhang et al., 2026a). Multimodal continual instruction tuning (MCIT) addresses this need by learning new tasks in sequence while retaining performance on previous ones (Chen et al., 2024; Guo et al., 2025b; Zheng et al., 2024). This sequential adaptation is prone to catastrophic forgetting: updates for a new task can degrade performance on previous ones (Chen et al., 2025; Xie et al., 2026; Wang et al., 2025; Chen et al., 2026).

To mitigate forgetting, mixture-of-experts (MoE) approaches based on low-rank adaptation (LoRA) distribute adaptation across experts and combine their outputs through input-dependent routing (Zhang et al., 2025; Hou et al., 2026). MoELoRA and SAME maintain shared expert pools (Chen et al., 2024; Xie et al., 2026), whereas CRAM expands its pool with task-dependent ranks (Tang et al., 2026b). Beyond expert allocation, SAME uses historical activation statistics to regulate both expert and router updates, while CRAM constrains adaptation through historical expert subspaces and combines semantic grouping with visual-centroid routing (Xie et al., 2026; Tang et al., 2026b).

However, existing expert-based approaches largely formulate adaptation constraints around historical knowledge, leaving the functional effects of expert updates on current-task inputs insufficiently characterized. At inference, prototype-based routing captures resemblance to individual tasks but does not explicitly model the relative distinctions among competing tasks. These gaps give rise to two challenges in MCIT: **task-specific specialization**, which concerns shaping expert adaptation to the functional demands of each arriving task, and **routing ambiguity**, which arises when related tasks provide insufficiently separated routing evidence in the absence of test-time task identities.

Task-specific specialization concerns how adaptation should shape an expert toward the functional demands of the current task. As illustrated in Figure 1(a), the same visual content may support different operations and responses, so separating expert parameters alone does not determine how their updates should behave. At an adapted layer, a weight update affects the model through its interaction with task inputs; consequently, updates of similar parameter magnitude can induce different output changes (Xie et al., 2026; Milsom et al., 2025). Current-task activations thus define a local, task-conditioned geometry for measuring these effects and guiding specialization.

Routing ambiguity arises when multiple related tasks provide similar routing evidence for a query. In Figure 1(b), a query can receive comparable affinity to multiple task prototypes, making prototype-based expert selection ambiguous (Shi et al., 2026; Xu et al., 2026). Prototype similarity captures resemblance to individual tasks, but independently constructed prototypes do not explicitly model the relative distinctions among competing tasks. Effective routing therefore requires complementing per-task affinity with cross-task discriminative evidence.

To address these challenges, we introduce **GeoSpec**, a progressive MoE-LoRA framework that couples geometry-guided expert formation with compatibility-guided expert access. For each arriving task, **Task-Conditioned Geometric Adaptation** uses current-task activation statistics to define a local output geometry, which guides both the adaptation direction and its fixed-rank realization. The resulting expert is then frozen and retained in the growing expert pool. At inference, **Compatibility-Guided Expert Routing** combines task-prototype similarity with discriminative task scores learned from answer-free, task-labeled support inputs. Their fused scores guide a sparse mixture of the accumulated experts for each query, enabling task-specific expertise to be formed during learning and accessed without test-time task identities. We evaluate GeoSpec on CoIN, UCIT, and TriGap, where it achieves the highest final AVG among the compared methods on all three benchmarks (Figure 1(c)).

Our contributions are summarized as follows:

- We introduce Task-Conditioned Geometric Adaptation to guide the formation of each new expert through the local effects of its updates on task inputs. Our analysis relates this task-conditioned metric to the approximation used to realize the update within the expert.
- We develop Compatibility-Guided Expert Routing to supplement task-prototype similarity with discriminative task scores learned from task-labeled support inputs. The routing rule

combines per-task affinity and cross-task discriminative evidence when assigning weights to the accumulated experts for each query.

- Across CoIN, UCIT, and TriGap, GeoSpec achieves the highest AVG among the methods compared in each benchmark table. The margins over the next-highest AVG are 2.20, 1.15, and 1.48 points, respectively.

2 Related Work

Multimodal continual instruction tuning. MCIT extends multimodal instruction tuning to sequential tasks, where adaptation to new tasks may degrade performance on previous ones (Chen et al., 2024; 2025; Gao et al., 2026; Hu & Zhou, 2026; Zhang et al., 2026b). Adapt-infinity selects examples from accumulated instruction data (Maharana et al., 2025), whereas ModalPrompt uses task-specific prompts and dual-modality-guided selection (Zeng et al., 2025). PCLR compresses learned rank vectors to manage adaptation capacity (meng et al., 2026). Octopus instead constrains gradients without historical task data (Liu et al., 2026b), while DGG uses previous-task optima to guide new-task gradients (Li et al., 2026).

MoE-LoRA for continual adaptation. MoE-LoRA methods organize routed low-rank experts differently as tasks arrive. MoELoRA and SAME use shared expert pools; SAME also stabilizes routing and expert updates using historical activation statistics (Chen et al., 2024; Xie et al., 2026). PASs-MoE retains fixed capacity and uses pathway activation subspaces to coordinate routing and rank stabilization (Hou et al., 2026). ProgLoRA and BranchLoRA progressively add LoRA modules, D-MoLE allocates experts across layers under a parameter budget, and CRAM grows experts with task-dependent ranks (Yu et al., 2025; Zhang et al., 2025; Ge et al., 2025; Tang et al., 2026b). MoBLoRA and CoRe-MoE pursue more compact reuse through shared low-rank bases and coordinate experts, respectively (Chen et al., 2026; Liu et al., 2026a). At inference, BranchLoRA uses task-specific routers, SAME stabilizes a learned router, and HiDe-LLaVA and CRAM use task anchors or visual centroids (Zhang et al., 2025; Xie et al., 2026; Guo et al., 2025a; Tang et al., 2026b). Ridge-based task routing also appears in FLEX's adaptation of RanPAC (McDonnell et al., 2023; Yi et al., 2026). Within progressive MoE-LoRA, GeoSpec uses current-task activation geometry to guide the formation of each new expert. It combines prototype similarity with discriminative task scores to route queries among the learned experts.

3 Preliminaries

3.1 Multimodal continual instruction tuning

In MCIT, an MLLM is sequentially instruction-tuned on a stream of task datasets $\mathcal{D}_1, \ldots, \mathcal{D}_K$ (Chen et al., 2024). For task k, the model defines the autoregressive likelihood of a target response y given input z, with the corresponding token-averaged training objective:

$$p_\theta(y \mid z) = \prod_{j=1}^{|y|} p_\theta(y_j \mid z, y_{<j}), \quad \mathcal{L}_k(\theta) = -\frac{1}{\sum_{(z,y)\in\mathcal{D}_k} |y|} \sum_{(z,y)\in\mathcal{D}_k} \sum_{j=1}^{|y|} \log p_\theta(y_j \mid z, y_{<j}). \quad (1)$$

Here $\mathcal{D}_k = \{(z_i, y_i)\}_{i=1}^{N_k}$ is the dataset for task k, where $z_i = (v_i, u_i)$ pairs an optional visual input v_i with an instruction u_i, and y_i is the target answer sequence. The adapted MLLM is parameterized by θ. Let F_{θ_0} denote the frozen vision-language backbone used for answer generation. In our continual-learning setting, expert optimization at task k uses only $\mathcal{D}_k$. After each task, the router is refitted on answer-free, task-labeled support from all observed tasks. Inference is task-ID-free.

3.2 Progressive MoE-LoRA

Given a frozen backbone, MoE-LoRA represents adaptation as an input-dependent mixture of low-rank expert updates (Chen et al., 2024; Xie et al., 2026; Tang et al., 2026b; Shen et al., 2024). Suppressing the layer index, consider a weight matrix $\mathbf{W}_0 \in \mathbb{R}^{m\times d}$ with input activation $\mathbf{h} \in \mathbb{R}^d$. Each expert e is parameterized by $\mathbf{A}_e \in \mathbb{R}^{r_e\times d}$ and $\mathbf{B}_e \in \mathbb{R}^{m\times r_e}$, with fixed nonzero scale γ_e (Hu

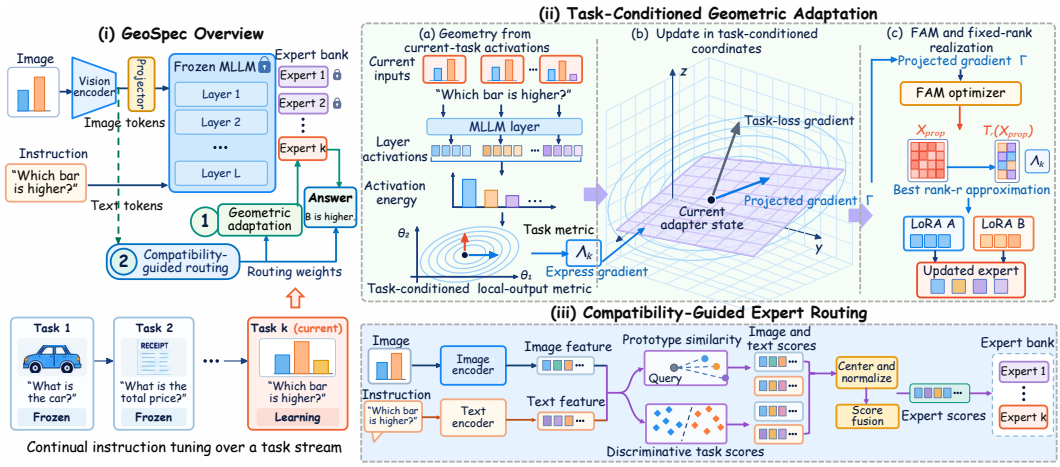

Figure 2: GeoSpec overview. (i) Language LoRA experts are trained sequentially and frozen. (ii) Current-task activations define the output metric (a); gradients are projected in its coordinates (b), and FAM updates are realized at fixed rank in the same geometry (c). (iii) Prototype similarities and discriminative task scores from frozen image and text features rank experts, with fixed weights of 0.7 and 0.3 for the top two. The geometric surface is schematic.

et al., 2021). For a pool of J_k experts, the adapted layer output is

$$\mathbf{o} = \mathbf{W}_0\mathbf{h} + \sum_{e=1}^{J_k} g_e(z)\Delta\mathbf{W}_e\mathbf{h}, \qquad \Delta\mathbf{W}_e = \gamma_e\mathbf{B}_e\mathbf{A}_e, \quad \mathrm{rank}(\Delta\mathbf{W}_e) \le r_e. \tag{2}$$

The normalized coefficients $g_e(z) \ge 0$, with $\sum_{e=1}^{J_k} g_e(z) = 1$, define the query-dependent routing weights, allowing either dense mixtures or sparse expert activation depending on the gating rule (Chen et al., 2024; Fedus et al., 2022). In progressive MoE-LoRA, the expert pool grows with the task stream, allocating new capacity while retaining previously learned experts (Yu et al., 2025; Tang et al., 2026b). Specific methods differ in how experts are allocated, updated, and routed.

4 Method

GeoSpec addresses task-specific specialization and routing ambiguity through two complementary components in a progressive MoE-LoRA framework. For each arriving task, Task-Conditioned Geometric Adaptation uses current-task activations to define a local output geometry that guides the optimization of a new low-rank expert. The trained expert is then frozen and retained in the growing expert pool. At inference, Compatibility-Guided Expert Routing combines task-prototype similarity with discriminative task scores to estimate query–task compatibility and combine the two highest-scoring experts. Together, the two components enable task-specific expertise to be formed during continual adaptation and accessed without test-time task identities (Figure 2).

4.1 Task-Conditioned Geometric Adaptation

Task-specific specialization requires adaptation to account for how parameter updates affect the current task, rather than only how large those updates are in parameter space. We therefore use current-task activations to define a task-conditioned local-output geometry over expert updates. This geometry measures updates by the output changes they induce on current-task inputs and is used consistently to guide both the optimization direction and its rank-constrained realization.

Task-conditioned local-output geometry. For an adapted layer with $\Delta\mathbf{W} = \gamma\mathbf{B}\mathbf{A}$, let $\mathbf{H} \in \mathbb{R}^{m\times d}$ denote an incremental update to the current adapter. Given an input activation $\mathbf{h}$, the corresponding first-order change in the layer output is $\mathbf{H}\mathbf{h}$. We measure the functional magnitude of this update over the current task by its expected squared output change:

$$\|\mathbf{H}\|_k^2 = \mathbb{E}\|\mathbf{H}\mathbf{h}\|_2^2 = \mathrm{tr}(\mathbf{H}\mathbf{Q}_k\mathbf{H}^\top), \qquad \mathbf{Q}_k = \mathbb{E}[\mathbf{h}\mathbf{h}^\top]. \tag{3}$$

Here, $\mathbf{Q}_k$ is the second-moment matrix of current-task activations, and $\|\cdot\|_k$ is a seminorm when $\mathbf{Q}_k$ is singular. SAME uses a related local-output measure with historical activation statistics to regulate expert updates (Xie et al., 2026). In GeoSpec, current-task activations instead define the geometry for optimizing a newly introduced expert, including its update direction and fixed-rank realization. Because $\mathbf{Q}_k$ is determined by the current task's activation distribution, the metric distinguishes parameter updates according to their functional effects on that task. Consequently, updates with similar parameter-space magnitude can induce substantially different output changes, and the same update can be weighted differently across tasks. The supervised task loss determines which behavior should be learned, while $\mathbf{Q}_k$ determines how candidate parameter updates are measured according to their effects on the current task's activation distribution. Together, the loss and the task-conditioned metric shape the optimization of the new expert.

Computing and storing the full second moment $\mathbf{Q}_k$ at every adapted layer is costly. We therefore use a normalized positive diagonal approximation $\mathbf{\Lambda} \succ 0$, estimated from current-task activation second moments accumulated over preceding optimization steps. Within each optimization step, $\mathbf{\Lambda}$ is treated as fixed. Normalization removes global activation scale, while the diagonal approximation trades cross-coordinate correlations for a lightweight per-coordinate metric (Appendix A.2). We absorb this metric into the coordinates by defining $\mathbf{X} = \Delta\mathbf{W}\mathbf{\Lambda}^{1/2}$, under which the task-conditioned output-change measure reduces to the standard Frobenius norm:

$$\|\mathbf{H}\|_\Lambda^2 := \|\mathbf{H}\mathbf{\Lambda}^{1/2}\|_F^2 = \mathrm{tr}(\mathbf{H}\mathbf{\Lambda}\mathbf{H}^\top). \tag{4}$$

This transformation lets us perform both optimization and rank-constrained approximation in coordinates aligned with the current task.

Geometry-guided adaptation. We next use the task-conditioned geometry to construct an update for the low-rank expert. Specifically, we express the task-loss gradient in the transformed coordinates and project it onto the space of first-order changes induced by the LoRA factors. With $\mathbf{\Lambda}$ fixed, let $\overline{\mathbf{A}} = \mathbf{A}\mathbf{\Lambda}^{1/2}$ and let $\mathcal{T}$ denote the space of changes $\gamma(\delta\mathbf{B}\,\overline{\mathbf{A}} + \mathbf{B}\,\delta\overline{\mathbf{A}})$ generated by varying the two LoRA factors. The transformed gradient and its projection are then

$$\mathbf{G}_X = \nabla_{\Delta\mathbf{W}}\mathcal{L}_k\,\mathbf{\Lambda}^{-1/2}, \qquad \mathbf{\Gamma} = \mathrm{Proj}_{\mathcal{T}}(\mathbf{G}_X). \tag{5}$$

The functional adaptive metric (FAM) maps $\mathbf{\Gamma}$ to direction $\mathbf{D}$ in task-conditioned coordinates using moment estimates (Kingma & Ba, 2017) and row–column statistics (Shazeer & Stern, 2018) (Appendix A.2). FAM is used as the matrix-space update mechanism within TCGA; our primary contribution concerns the task-conditioned geometry and its consistent use in update construction and rank realization.

The projection places $\mathbf{\Gamma}$ in the space of first-order LoRA-factor changes, but the finite matrix step formed from $\mathbf{D}$ need not have rank at most r. We therefore impose the rank constraint on the proposal in the same task-conditioned coordinates. Let $\mathscr{T}_r$ denote the best rank-at-most-r Frobenius approximation (Bogachev et al., 2026), and write $\mathbf{X}_{\mathrm{prop}} = \mathbf{X} - \eta\mathbf{D}$ and $\mathbf{W}_{\mathrm{prop}} = \mathbf{X}_{\mathrm{prop}}\mathbf{\Lambda}^{-1/2}$. For task k, layer ℓ, and step s, the update is

$$\Delta\mathbf{W}_{k,\ell,s+1} = \mathscr{T}_r(\mathbf{X}_{k,\ell,s} - \eta_{k,s}\mathbf{D}_{k,\ell,s})\,\mathbf{\Lambda}_{k,\ell,s}^{-1/2}, \qquad \mathrm{rank}(\mathscr{T}_r(\cdot)) \le r. \tag{6}$$

Here, $\eta_{k,s}$ denotes the step size at optimization step s of task k. Because the task-conditioned metric becomes Euclidean in $\mathbf{X}$-coordinates, $\mathscr{T}_r(\mathbf{X}_{\mathrm{prop}})$ gives the best rank-r approximation under that metric. In exact arithmetic,

$$\mathscr{T}_r(\mathbf{X}_{\mathrm{prop}})\mathbf{\Lambda}^{-1/2} \in \arg\min_{\mathrm{rank}(\mathbf{Z})\le r} \|(\mathbf{Z} - \mathbf{W}_{\mathrm{prop}})\mathbf{\Lambda}^{1/2}\|_F^2. \tag{7}$$

This establishes optimality of the rank-r realization under the task-conditioned metric, not monotonic decrease of the task loss. To express the resulting matrix in the original LoRA parameterization, let $\mathbf{U}_r$ contain the retained orthonormal left singular vectors of $\mathbf{X}_{\mathrm{prop}}$. Suppressing indices, we obtain

$$\mathbf{B}^+ = \mathbf{U}_r, \qquad \mathbf{A}^+ = \frac{1}{\gamma}\mathbf{U}_r^\top\mathbf{X}_{\mathrm{prop}}\mathbf{\Lambda}^{-1/2}, \qquad \gamma\mathbf{B}^+\mathbf{A}^+ = \mathscr{T}_r(\mathbf{X}_{\mathrm{prop}})\mathbf{\Lambda}^{-1/2}. \tag{8}$$

We then refactorize the projected matrix into the standard LoRA form. Recomputing both factors after each step allows the expert's rank-r subspace to evolve throughout adaptation. After adaptation, the resulting task-specific expert is frozen and added to the growing expert pool. The task-conditioned geometry is used only to shape expert learning and is not required once the expert has been trained.

4.2 Compatibility-Guided Expert Routing

To address routing ambiguity when related tasks provide insufficiently separated evidence, Compatibility-Guided Expert Routing augments task-prototype similarity with discriminative task scores. Prototype similarity captures how strongly a query resembles each task individually, whereas the discriminative scores are learned jointly across task labels and capture relative distinctions among competing tasks. Their fusion encodes within-task affinity and cross-task discrimination.

Prototype similarity and discriminative task scores. After completing task k, we fit the router for the current expert pool using support from tasks $1, \ldots, k$. We write $J = J_k$ and suppress the stage index below. For each task e, we construct an answer-free support set $\mathcal{S}_e$ from its training data while retaining the task label. For modality $m \in \{v, u\}$, let z_m denote the corresponding visual or textual input. For routing, E_v and E_u are the image and text encoders of a separate frozen CLIP Base model. The encoder E_m produces the normalized feature $\mathbf{r}^m(z) = E_m(z_m)/\|E_m(z_m)\|_2$. The text-only case follows the same construction and is detailed in Appendix B.2. Following task-anchor matching (Guo et al., 2025a), we use normalized means of support features as task prototypes and score query affinity by cosine similarity:

$$\mathbf{c}_e^m = \frac{\sum_{z_i \in \mathcal{S}_e} \mathbf{r}^m(z_i)}{\left\|\sum_{z_i \in \mathcal{S}_e} \mathbf{r}^m(z_i)\right\|_2}, \qquad [\mathbf{p}^m(z)]_e = \mathbf{r}^m(z)^\top \mathbf{c}_e^m. \tag{9}$$

The vector $\mathbf{p}^m(z) \in \mathbb{R}^J$ measures the query's resemblance to each task independently. Because task prototypes are constructed independently, prototype similarity does not explicitly encode how a query should be distinguished among related tasks. We therefore learn an additional discriminative score from the same support features by fitting a class-balanced ridge classifier jointly over the task labels for each modality. The resulting vector $\mathbf{d}^m(z) \in \mathbb{R}^J$ provides discriminative evidence across tasks, complementing the per-task affinity encoded by $\mathbf{p}^m(z)$. Appendix B.2 gives the fitting details. These task-level scores parameterize routing over the accumulated expert pool. We combine the resulting discriminative scores with task-prototype similarities to form the routing signal.

Score fusion and expert weighting. Because prototype and classifier scores can differ in scale and offset, we normalize each task-score vector by mean-centering it across tasks and scaling it to unit ℓ_2 norm before fusion. For a nonconstant $\mathbf{a} \in \mathbb{R}^J$, normalization and fusion are defined as

$$\mathsf{N}(\mathbf{a}) = \frac{\mathbf{a} - \bar{a}\mathbf{1}_J}{\|\mathbf{a} - \bar{a}\mathbf{1}_J\|_2}, \qquad \bar{a} = J^{-1}\mathbf{1}_J^\top \mathbf{a}, \tag{10}$$

$$\mathbf{s}(z) = \sum_{m \in \{v,u\}} \left[\omega_{m,p}\mathsf{N}(\mathbf{p}^m(z)) + \omega_{m,d}\mathsf{N}(\mathbf{d}^m(z))\right]. \tag{11}$$

The fused vector $\mathbf{s}(z)$ combines independent task affinity with cross-task discriminative evidence and serves as our compatibility-aware routing signal. The nonnegative fusion coefficients sum to one and are fixed from the start of the task stream. For $J \geq 2$, let $e_{(1)}(z)$ and $e_{(2)}(z)$ index the highest and second-highest compatibility scores. We use a top-2 gate with fixed mixing weights:

$$g_e(z) = 0.7\,\mathbb{I}[e = e_{(1)}(z)] + 0.3\,\mathbb{I}[e = e_{(2)}(z)]. \tag{12}$$

The scores determine expert ranking, while the two selected experts receive weights of 0.7 and 0.3, respectively; all others receive zero. When the pool contains only one expert, its weight is one. These query-dependent weights are shared across the adapted layers in the MoE-LoRA forward computation of Equation (2). At each stage, prototype and discriminative components are fitted on the same support features from the observed tasks and held fixed for that stage's inference. The final expert pool is the case $k = K$. Fitting procedures and score-fusion settings appear in Appendix B.2.

5 Experiments

5.1 Experimental setup

Datasets. We evaluate GeoSpec on three established MCIT benchmarks implemented in Prism (Tang et al., 2026a): CoIN (Chen et al., 2024), UCIT (Guo et al., 2025a), and TriGap (Xie et al., 2026). Together, they cover heterogeneous task requirements, including recognition, grounding, captioning, and multiple forms of visual question answering. CoIN contains eight tasks, UCIT contains six tasks,

Table 1: Performance on the TriGap benchmark. Results for methods marked † were obtained from our reproductions; unmarked baseline results are from the cited papers. Boldface and underlining indicate the best and second-best scores in each column, respectively.

Method	PMC VQA	Doc VQA	Chart QA	Icon QA	Info VQA	Arxiv QA	Road	Chem VQA	Flood VQA	CLEVR	AVG
Zero-shot	35.40	12.68	9.36	19.27	5.06	53.77	7.40	5.30	47.41	20.37	21.60
FT-LoRA (ICLR'22)	34.20	23.32	9.84	37.07	23.53	83.83	7.00	12.70	80.31	60.27	37.21
MoELoRA (NeurIPS'24)	39.03	37.49	12.44	43.43	35.17	90.90	7.93	20.70	**90.41**	**67.00**	44.45
CL-MoE (CVPR'25)	40.53	36.79	13.72	52.70	32.27	93.00	7.77	18.33	80.09	65.90	44.11
HiDe-LLaVA (ACL'25)†	36.40	32.93	10.52	40.30	25.04	79.10	7.60	11.83	58.10	26.53	32.84
ModalPrompt (EMNLP'25)	38.23	38.23	11.92	44.73	37.37	84.47	10.13	12.43	71.52	52.50	40.15
SAME (ICML'26)†	44.00	44.20	18.36	66.30	39.66	92.90	11.57	24.00	81.23	59.93	48.22
SiGMA (ECCV'26)†	39.77	35.72	9.44	37.93	30.00	80.33	8.23	9.43	71.30	40.07	36.22
CRAM (EMNLP'26)†	42.90	43.77	16.48	65.17	39.09	92.00	**13.03**	22.73	80.14	58.53	47.38
ProtoAda (arXiv'26)	42.87	43.01	18.96	66.03	36.31	**93.50**	12.97	23.73	74.03	57.87	47.23
GeoSpec (Ours)	**44.37**	**45.78**	**19.76**	**72.53**	**40.55**	92.97	12.93	**27.83**	82.21	58.10	**49.70**

Table 2: Performance on the UCIT benchmark. Results for methods marked † were obtained from our reproductions; unmarked baseline results are from the cited papers. Boldface and underlining indicate the best and second-best scores in each column, respectively.

Method	ImageNet -R	Arxiv QA	VizWiz Caption	Icon QA	CLEVR Math	Flickr30k	AVG
Zero-shot	18.88	52.62	38.75	21.25	21.12	41.44	32.34
FT-LoRA (ICLR'22)	29.33	55.30	45.51	26.13	13.07	58.07	37.90
MoELoRA (NeurIPS'24)	58.43	77.57	44.83	68.90	56.73	**58.27**	60.79
CL-MoE (CVPR'25)	64.12	78.38	44.83	62.00	50.75	58.06	59.69
HiDe-LLaVA (ACL'25)†	89.13	86.40	42.87	63.07	47.90	51.27	63.44
ModalPrompt (EMNLP'25)	80.50	90.62	60.13	63.50	55.75	57.09	67.93
Octopus (CVPR'26)	88.20	93.03	58.46	60.50	69.00	56.87	71.01
SAME (ICML'26)†	90.33	92.73	47.84	79.67	**73.10**	57.39	73.51
SiGMA (ECCV'26)†	66.63	**93.83**	56.27	68.60	57.80	56.05	66.53
CRAM (EMNLP'26)†	89.10	92.90	60.79	80.13	68.80	57.21	74.82
ProtoAda (arXiv'26)	89.83	93.20	**61.49**	81.37	66.40	55.64	74.66
GeoSpec (Ours)	**90.97**	91.83	60.71	**84.37**	70.13	57.80	**75.97**

and TriGap extends the sequence to ten tasks spanning document, chart, scientific, and general visual reasoning. Task boundaries are known during training; inference is task-ID-free.

Compared methods. We compare GeoSpec primarily with recent expert-based MCIT methods, including MoELoRA, CL-MoE, SAME, CRAM, and ProtoAda (Chen et al., 2024; Huai et al., 2025; Xie et al., 2026; Tang et al., 2026b; Shi et al., 2026), which provide the closest comparisons in expert adaptation and routing. We additionally include HiDe-LLaVA and ModalPrompt as task-structured parameter-efficient approaches (Guo et al., 2025a; Zeng et al., 2025), and PCLR, SiGMA, and Octopus as representative continual-adaptation methods (meng et al., 2026; Park & Kim, 2026; Liu et al., 2026b). FT-LoRA and zero-shot evaluation serve as single-adapter and frozen-backbone references, respectively. Further details are provided in the supplementary material.

Implementation. We use LLaVA-v1.5-7B as the frozen backbone (Liu et al., 2024) and train rank-8 language LoRA experts as tasks arrive. The adapters are applied to the gate, up, and down projections of the feed-forward layers. Each expert is trained for one epoch with an effective batch size of 32 using bf16 precision, a base learning rate of 2×10^{-4}, cosine decay, and a 0.03 warm-up ratio, then frozen. Loss normalization and hardware details are provided in Appendix B.1. After each task, we refit the routing components from answer-free, task-labeled training support of all tasks observed so far. Task-prototype similarities and discriminative task scores are computed from frozen image and text features and fused according to Section 4.2 to obtain the task-ID-free routing signal used during

Table 3: Performance on the CoIN benchmark. Results for methods marked † were obtained from our reproductions; unmarked baseline results are from the cited papers. Boldface and underlining indicate the best and second-best scores in each column, respectively.

Method	Science QA	Text VQA	Image Net	GQA	VizWiz	REC	VQAv2	OCR VQA	AVG
MoELoRA (NeurIPS'24)	62.02	52.05	37.21	53.12	43.32	33.22	57.92	**65.75**	50.58
HiDe-LLaVA (ACL'25)†	61.73	50.05	4.42	31.32	47.07	19.57	35.44	35.41	35.63
ModalPrompt (EMNLP'25)	68.42	56.40	41.13	61.11	50.13	36.69	<u>66.90</u>	59.68	55.06
PCLR (ICLR'26)	<u>78.33</u>	58.24	86.08	58.14	<u>57.61</u>	33.04	64.17	61.92	62.19
SAME (ICML'26)†	77.39	63.15	<u>97.13</u>	**62.13**	49.25	72.68	66.82	<u>64.54</u>	<u>69.14</u>
SiGMA (ECCV'26)†	0.00	54.11	5.37	42.43	0.00	25.67	0.00	0.02	15.95
CRAM (EMNLP'26)†	75.06	<u>63.48</u>	77.21	59.73	43.32	<u>72.74</u>	62.45	59.21	64.15
GeoSpec (Ours)	**80.11**	**63.87**	**97.29**	<u>61.47</u>	**62.45**	**73.82**	**67.42**	64.31	**71.34**

generation. The top-2 expert weights are fixed at 0.7 and 0.3 in descending score order. We report average final performance as $\mathrm{AVG} = \frac{1}{K}\sum_{k=1}^{K} \mathsf{M}_{K,k}$, where $\mathsf{M}_{K,k}$ is the task-ID-free score on task k after learning all K tasks.

5.2 Benchmark comparison

Tables 1–3 evaluate whether GeoSpec's two design principles—task-conditioned expert formation and task-ID-free expert access—translate into end-to-end gains across heterogeneous continual task streams. GeoSpec achieves the highest final AVG among the compared methods on all three benchmarks, reaching 49.70 on TriGap, 75.97 on UCIT, and 71.34 on CoIN, with margins of 1.48, 1.15, and 2.20 points over the next-highest AVG, respectively.

TriGap provides the longest and most heterogeneous stream, with ten tasks spanning document, chart, scientific, domain-specific, and general visual reasoning. GeoSpec improves over SAME on nine of the ten tasks, including +6.23 on IconQA, +3.83 on ChemVQA, and +1.58 on DocVQA. Such gains across substantially different task demands are consistent with the motivation of TCGA: instead of applying a task-agnostic adaptation geometry, each arriving expert is shaped according to the functional effects of its updates on the current task.

The same pattern extends to the shorter UCIT and CoIN streams. On UCIT, GeoSpec reaches 75.97 AVG, compared with 74.82 for CRAM, with its largest gain on IconQA (+4.24). On CoIN, it obtains 71.34 AVG, exceeding SAME by 2.20 points, with notable gains on VizWiz (+13.20) and ScienceQA (+2.72). These improvements span different visual conditions, reasoning requirements, and response formats, suggesting that the advantage is not tied to a single task family. Overall, the benchmark results support the combined design of forming experts in a task-conditioned geometry and accessing them through compatibility-guided routing.

5.3 Ablation studies

We evaluate adaptation and routing under matched conditions: training variants reuse the fitted router, while routing variants share the same trained expert pool. We denote the full adaptation procedure as Task-Conditioned Geometric Adaptation (TCGA) and the fused router as Compatibility-Guided Expert Routing (CGER).

Contributions of adaptation and routing. Table 4 separates the effects of expert training and routing. With CGER fixed, TCGA improves AVG by 3.77 points on UCIT and 3.51 on TriGap. With TCGA-trained experts fixed, adding discriminative task scores to prototype similarity yields gains of 7.62 and 5.64 points, respectively. TCGA also remains beneficial under prototype-only routing.

Table 4: Component-wise ablation under task-ID-free inference. PS: task-prototype similarity; DTS: <u>discriminative task scores</u>.

Variant	TCGA	PS	DTS	UCIT↑	TriGap↑
Base MoE-LoRA	×	✓	×	60.15	42.44
w/o TCGA	×	✓	✓	72.20	46.19
w/o DTS	✓	✓	×	68.35	44.06
GeoSpec	✓	✓	✓	75.97	49.70

Table 5: TCGA metric ablation with the fitted CGER router fixed. AVG is measured under task-ID-free inference.

Metric	UCIT AVG↑	TriGap AVG↑
Identity	73.15	46.85
Permuted	72.08	46.64
TCGA	75.97	49.70

Table 6: CGER ablation on TriGap with TCGA-trained experts fixed. Macro match is the task-averaged top-2 source-expert hit rate (%).

Router	AVG↑	Macro match↑
Prototype-only	44.06	86.15
Classifier-only	46.39	89.72
CGER	49.70	93.40

Role of task-conditioned geometry. Table 5 isolates the metric while retaining FAM, fixed-rank realization, and the fitted CGER router. Replacing the task-conditioned metric with Identity reduces AVG by 2.82 points on UCIT and 2.85 on TriGap. Permuting the same diagonal entries reduces performance further, indicating that the gain depends on their alignment with task activations rather than on nonuniform scaling alone.

Complementarity of routing evidence. With the TCGA-trained expert pool fixed, CGER improves TriGap AVG to 49.70, compared with 44.06 for prototype-only and 46.39 for classifier-only routing (Table 6). Macro top-2 source-expert match likewise increases to 93.40%, supporting the complementary roles of task affinity and cross-task discrimination.

5.4 Computational cost

TCGA changes expert training through metric estimation and fixed-rank realization, whereas CGER requires router fitting, CLIP feature extraction, and per-query scoring. Table 7 pairs final TriGap AVG with per-task trainable parameters, retained model memory, complete-stream training time, peak GPU memory, and inference throughput that includes feature extraction, routing, and generation. GeoSpec attains 49.70 AVG with 11.64M trainable parameters and 25.1 GiB peak GPU memory, compared with 48.22 AVG, 122.14M, and 32.5 GiB for SAME, and 47.38 AVG, 57.65M, and 29.5 GiB for CRAM. Its 3.12 samples/s throughput also exceeds SAME (2.54) and HiDe-LLaVA (2.25). The combined adaptation-and-routing design thus pairs its TriGap result with a smaller trainable and peak-GPU footprint than these expert-based comparators.

Table 7: Resource comparison and final AVG on TriGap. Continual state is the retained expert bank and fitted routing parameters after the task sequence. Bold and underline denote the best and second-best values in each column.

Method	Trainable (M)↓	Continual state (MiB)↓	Train (h)↓	Peak GPU (GiB)↓	Samples/s ↑	AVG↑
SAME	122.14	233.0	13.0	32.5	2.54	<u>48.22</u>
CRAM	57.65	**201.5**	**6.20**	<u>29.5</u>	**3.22**	47.38
HiDe-LLaVA	<u>19.99</u>	403.6	<u>6.61</u>	<u>29.5</u>	2.25	32.84
GeoSpec	**11.64**	<u>223.4</u>	8.1	**25.1**	<u>3.12</u>	**49.70**

6 Conclusion

We presented GeoSpec, a progressive MoE-LoRA framework for multimodal continual instruction tuning that addresses both expert formation and task-ID-free expert access. For each arriving task, TCGA uses current-task activation statistics to define a local output metric and applies this metric consistently when constructing the expert update and realizing it under a fixed-rank constraint. At inference, CGER complements task-prototype affinity with discriminative task evidence to route queries among the accumulated experts without requiring task identities. Across CoIN, UCIT, and TriGap, GeoSpec achieves the highest final AVG among the compared methods. Ablations show that the task-conditioned metric improves expert performance, while complementary routing signals improve expert access. Overall, these results support task-conditioned functional geometry and complementary routing evidence as effective design principles for progressive expert-based MCIT.

AI USE STATEMENT

During manuscript preparation, we used generative AI to organize related literature and experimental results, assist with tables and figures, and revise the writing. We also sought its feedback on method formulation, experimental design, and interpretation of results. The authors assessed the AI-assisted suggestions and retain responsibility for the final methods, experiments, results, and manuscript.

ETHICS STATEMENT

This work evaluates continual multimodal instruction tuning on the existing CoIN, UCIT, and TriGap benchmarks. We did not recruit human participants or collect new personal data. We cite the model and dataset sources and follow the released evaluation protocols.

REPRODUCIBILITY STATEMENT

Sections 3–5 specify the learning setting, GeoSpec method, benchmark setup, and evaluation metrics. Supplementary Sections A–D provide the geometric derivation and optimizer details, expert and router configurations, scoring and ablation protocols, and computational-complexity analysis. Together, these sections document the procedures used for the reported comparisons.

REFERENCES

Vladimir Bogachev, Vladimir Aletov, Alexander Molozhavenko, Denis Bobkov, Vera Soboleva, Aibek Alanov, and Maxim Rakhuba. Lora meets riemannion: Muon optimizer for parametrization-independent low-rank adapters. In C. Vondrick, B. Hariharan, C. Raffel, L. Pinto, D. Yang, and A. Faust (eds.), *International Conference on Learning Representations*, volume 2026, pp. 82190–82215, 2026.

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 Tao Shen, Jingkuan Song, and Lianli Gao. Coin: A benchmark of continual instruction tuning for multimodel large language models. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 57817–57840. Curran Associates, Inc., 2024. doi: 10.52202/079017-1844.

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 Aarti Singh, Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien, Felix Berkenkamp, Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu (eds.), *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 7982–8001. PMLR, 13–19 Jul 2025.

Pei Chen, Xilai Wang, Shiqixu, Zejian Li, and Lingyun Sun. From experts to bases: Orthogonal subspace mixture for continual multimodal instruction tuning. In Maria Liakata, Viviane P. Moreira, Jiajun Zhang, and David Jurgens (eds.), *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 10545–10561, San Diego, California, United States, July 2026. Association for Computational Linguistics. ISBN 979-8-89176-390-6. doi: 10.18653/v1/2026.acl-long.481.

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. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 49250–49267. Curran Associates, Inc., 2023. doi: 10.52202/075280-2142.

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

Zijian Gao, Wangwang Jia, Xingxing Zhang, Pengfei Qian, Tao Sun, Bo Ding, Yong Dou, Huaimin Wang, and Kele Xu. Many: Merge anything for multimodal continual instruction tuning. *arXiv preprint arXiv:2604.14016*, 2026.

Chendi Ge, Xin Wang, Zeyang Zhang, Hong Chen, Jiapei Fan, Longtao Huang, Hui Xue, and Wenwu Zhu. Dynamic mixture of curriculum LoRA experts for continual multimodal instruction tuning. In Aarti Singh, Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien, Felix Berkenkamp, Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu (eds.), *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 19011–19033. PMLR, 13–19 Jul 2025.

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 Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 13572–13586, Vienna, Austria, July 2025a. Association for Computational Linguistics. ISBN 979-8-89176-251-0. doi: 10.18653/v1/2025.acl-long.666.

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*, 2025b.

Vineet Gupta, Tomer Koren, and Yoram Singer. Shampoo: Preconditioned stochastic tensor optimization. In Jennifer Dy and Andreas Krause (eds.), *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 1842–1850. PMLR, 10–15 Jul 2018.

Jinghan He, Haiyun Guo, Kuan Zhu, Ming Tang, and Jinqiao Wang. Continual instruction tuning for large multimodal models. *IEEE Transactions on Image Processing*, 35:2699–2713, 2026. ISSN 1941-0042. doi: 10.1109/tip.2026.3671616.

ZhiYan Hou, Haiyun Guo, Haokai Ma, Yandu Sun, Yonghui Yang, and Jinqiao Wang. PASs-MoE: Mitigating misaligned co-drift among router and experts via pathway activation subspaces for continual learning. In Maria Liakata, Viviane P. Moreira, Jiajun Zhang, and David Jurgens (eds.), *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 31959–31972, San Diego, California, United States, July 2026. Association for Computational Linguistics. ISBN 979-8-89176-390-6. doi: 10.18653/v1/2026.acl-long.1474.

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

Tao Hu and Da-Wei Zhou. Dynamic cross-modal prompt generation for multimodal continual instruction tuning. *arXiv preprint arXiv:2605.10765*, 2026.

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 Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 19608–19617, June 2025.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. 2017.

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 (CVPR)*, pp. 10820–10829, June 2026.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 34892–34916. Curran Associates, Inc., 2023. doi: 10.52202/075280-1516.

Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction tuning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 26296–26306, June 2024.

Runze Liu, Naibin Gu, Mingxu Ai, Yuqing Li, Peng Fu, Zheng Lin, and Weiping Wang. Coremoe: Compact reusable moe for continual multimodal instruction tuning. *arXiv preprint arXiv:2608.27867*, 2026a.

Yuehao Liu, Shanyan Guan, Weijia Zhang, Xuanming Shang, Yanhao Ge, Wei Li, and Chao Ma. Octopus: History-free gradient orthogonalization for continual learning in multimodal large language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3898–3907, June 2026b.

Adyasha Maharana, Jaehong Yoon, Tianlong Chen, and Mohit Bansal. Adapt-\infty: Scalable continual multimodal instruction tuning via dynamic data selection. In Y. Yue, A. Garg, N. Peng, F. Sha, and R. Yu (eds.), *International Conference on Learning Representations*, volume 2025, pp. 66453–66473, 2025.

Mark D. McDonnell, Dong Gong, Amin Parvaneh, Ehsan Abbasnejad, and Anton van den Hengel. Ranpac: Random projections and pre-trained models for continual learning. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 12022–12053. Curran Associates, Inc., 2023. doi: 10.52202/075280-0526.

weicheng meng, Jingyang Qiao, Zhizhong Zhang, Shaohui Liu, and Yuan Xie. Pclr: Progressively compressed lora for multimodal continual instruction tuning. In C. Vondrick, B. Hariharan, C. Raffel, L. Pinto, D. Yang, and A. Faust (eds.), *International Conference on Learning Representations*, volume 2026, pp. 55054–55089, 2026.

Edward Milsom, Ben Anson, and Laurence Aitchison. Function-space learning rates. In Aarti Singh, Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien, Felix Berkenkamp, Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu (eds.), *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 44225–44251. PMLR, 13–19 Jul 2025.

Keonhee Park and Gunhee Kim. Sigma: Sign-guided merging and adaptation for multimodal continual instruction tuning. *arXiv preprint arXiv:2607.20511*, 2026.

Noam Shazeer and Mitchell Stern. Adafactor: Adaptive learning rates with sublinear memory cost. In Jennifer Dy and Andreas Krause (eds.), *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 4596–4604. PMLR, 10–15 Jul 2018.

Ying Shen, Zhiyang Xu, Qifan Wang, Yu Cheng, Wenpeng Yin, and Lifu Huang. Multimodal instruction tuning with conditional mixture of LoRA. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 637–648, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.38.

Yu-Cheng Shi, Zhen-Hao Xie, Jun-Tao Tang, and Da-Wei Zhou. Protoada: Prototype-guided adaptive adapter expansion and geometric consolidation for multimodal continual instruction tuning. *arXiv preprint arXiv:2606.02576*, 2026.

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*, 2026a.

Jun-Tao Tang, Zhen-Hao Xie, Yu-Cheng Shi, and Da-Wei Zhou. Cram: Centroid-routing and adaptive moe for multimodal continual instruction tuning. *arXiv preprint arXiv:2606.02502*, 2026b.

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/CVF International Conference on Computer Vision (ICCV)*, pp. 177–186, October 2025.

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.

Kunlun Xu, Yanqin Zhang, Wenwen Qiang, and Jiahuan Zhou. Hyper-llava: Hyperbolic uncertainty-aware modality-balanced routing for multimodal continual instruction tuning. *arXiv preprint arXiv:2609.13742*, 2026.

Huiyu Yi, Yongqi Xu, Bogang Zhang, Dunwei Tu, Xu Zhiming, Zhen-Hao Xie, Baile Xu, and Furao Shen. Beyond routing saturation: A long-horizon class-incremental perspective on expert routing in multimodal continual instruction tuning. *arXiv preprint arXiv:2608.01437*, 2026.

Yahan Yu, Duzhen Zhang, Yong Ren, Xuanle Zhao, Xiuyi Chen, and Chenhui Chu. Progressive LoRA for multimodal continual instruction tuning. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 2779–2796, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-256-5. doi: 10.18653/v1/2025.findings-acl.143.

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 Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng (eds.), *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 12126–12141, Suzhou, China, November 2025. Association for Computational Linguistics. ISBN 979-8-89176-332-6. doi: 10.18653/v1/2025.emnlp-main.609.

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 Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 5743–5756, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-251-0. doi: 10.18653/v1/2025.acl-long.287.

Duzhen Zhang, Yahan Yu, Qiaoyi Su, Jiahua Dong, and Tielin Zhang. Progressive multimodal alignment for continual instruction tuning. *arXiv preprint arXiv:2607.26947*, 2026a.

Hongsheng Zhang, Zhong Ji, Di Wang, and Yanwei Pang. Interleaved prompt generation for continual instruction tuning of multimodal large language model. *IEEE Transactions on Image Processing*, 35:9935–9948, 2026b. ISSN 1941-0042. doi: 10.1109/tip.2026.3732640.

Junhao Zheng, Qianli Ma, Zhen Liu, Binquan Wu, and Huawen Feng. Beyond anti-forgetting: Multimodal continual instruction tuning with positive forward transfer. *arXiv preprint arXiv:2401.09181*, 2024.

GeoSpec: Supplementary Material

Geometry-Guided Expert Specialization for Multimodal Continual Instruction Tuning

SUPPLEMENTARY OVERVIEW

The supplement follows the two method components from their mathematical basis to the completed evaluation. The four parts answer different questions; notation and task-score conventions follow the main paper.

Part	What the reader finds here
A. Geometry	The local-output measure, factor-change projection, and metric-consistent fixed-rank realization.
B. Protocol	Router fitting, task-score definitions, result provenance, and benchmark examples.
C. Evidence	Matched conditions behind the ablations and conditional AVG changes computed from the completed component table.
D. Cost	Analytical arithmetic and memory for adaptation and routing.

A GEOMETRIC ADAPTATION: DERIVATION AND REALIZATION

A.1 PARAMETER AND LOCAL-OUTPUT ORTHOGONALITY

The full activation second moment $\mathbf{Q}_k$ induces the bilinear form $\langle \mathbf{H}_1, \mathbf{H}_2 \rangle_k = \mathrm{tr}(\mathbf{H}_1 \mathbf{Q}_k \mathbf{H}_2^\top)$. Parameter-space orthogonality transfers to this local-output measure for every pair of matrices only under an isotropic activation moment.

Proposition 1. *For $\mathbf{Q}_k \succeq 0$, the implication $\langle \mathbf{H}_1, \mathbf{H}_2 \rangle_F = 0 \Rightarrow \langle \mathbf{H}_1, \mathbf{H}_2 \rangle_k = 0$ holds for all $\mathbf{H}_1, \mathbf{H}_2 \in \mathbb{R}^{m \times d}$ if and only if $\mathbf{Q}_k = c\mathbf{I}_d$ for some $c \geq 0$.*

If $\mathbf{Q}_k = c\mathbf{I}_d$, the bilinear form is $c\langle \mathbf{H}_1, \mathbf{H}_2 \rangle_F$. Conversely, if $\mathbf{Q}_k$ has orthonormal eigenvectors $\mathbf{u}, \mathbf{v}$ with distinct eigenvalues λ_u, λ_v, choose matrices whose first rows are $(\mathbf{u}+\mathbf{v})^\top$ and $(\mathbf{u}-\mathbf{v})^\top$, with other rows zero. Their Frobenius inner product vanishes, whereas their local-output inner product is $\lambda_u - \lambda_v \neq 0$. Thus every eigenvalue must coincide; the scalar-input case follows directly.

A.2 TASK METRIC AND UPDATE INTERFACE

Metric construction. For task k, layer ℓ, and step s, the diagonal statistic $\widehat{\mathbf{q}}_{k,\ell,s}$ estimates the coordinate-wise second moment of layer-input activations from the current task. It uses attention-mask-valid positions in the expanded multimodal sequence, including image tokens, and accumulates statistics through step $s-1$. The statistic is detached from the task-loss computation. With $\bar{q} = d^{-1}\mathbf{1}_d^\top \widehat{\mathbf{q}} > 0$, the metric is

$$\mathbf{\Lambda}_{k,\ell,s} = \mathrm{Diag}\left(\max\left\{ \frac{\widehat{\mathbf{q}}_{k,\ell,s}}{\bar{q}}, 32\epsilon_{\mathrm{fp32}} \right\} \right) \succ 0, \tag{13}$$

where the maximum is elementwise. The first update of a task uses the identity metric; subsequent updates use the preceding statistics. The metric remains fixed within a step and is reset at a task boundary. When the floor is inactive, the unnormalized diagonal output measure equals $\bar{q}\|\mathbf{H}\|_\Lambda^2$.

Gradient projection. For fixed $\mathbf{\Lambda}$, let $\overline{\mathbf{A}} = \mathbf{A}\mathbf{\Lambda}^{1/2}$. The orthogonal projectors onto the row space of $\overline{\mathbf{A}}$ and column space of $\mathbf{B}$ are

$$\mathbf{\Pi}_{\mathrm{in}} = \overline{\mathbf{A}}^\top (\overline{\mathbf{A}}\overline{\mathbf{A}}^\top)^\dagger \overline{\mathbf{A}}, \qquad \mathbf{\Pi}_{\mathrm{out}} = \mathbf{B}(\mathbf{B}^\top\mathbf{B})^\dagger \mathbf{B}^\top, \tag{14}$$

where $\dagger$ denotes the Moore–Penrose inverse. The projection of $\mathbf{G}_X = \nabla_{\Delta\mathbf{W}} \mathcal{L}_k \, \mathbf{\Lambda}^{-1/2}$ onto the first-order factor-change space is

$$\mathbf{\Gamma} = \mathbf{G}_X \mathbf{\Pi}_{\mathrm{in}} + \mathbf{\Pi}_{\mathrm{out}} \mathbf{G}_X (\mathbf{I}_d - \mathbf{\Pi}_{\mathrm{in}}). \tag{15}$$

The implementation recovers this projected gradient from the two LoRA factor gradients using thin factor-space decompositions. It does not form a dense $m \times d$ task-loss gradient; pseudoinverses use

the retained numerical singular directions. The FAM optimizer receives $\mathbf{\Gamma}$ and returns a direction $\mathbf{D}$ in the same coordinates; the learning rate enters the proposed matrix update of Equation (6), not $\mathbf{\Gamma}$.

FAM direction. The moment estimates use exponential averaging and bias correction as in Adam (Kingma & Ba, 2017). Row–column energy summaries are related to Adafactor's factored second-moment estimates (Shazeer & Stern, 2018). Suppress task and layer indices, and let t count updates within the current task. FAM retains a first moment $\mathbf{M}_t$ in weight coordinates, a scalar second moment v_t in task-conditioned coordinates, and row and column energy profiles $\boldsymbol{\rho}_t$ and $\boldsymbol{\kappa}_t$ in weight coordinates. With $\mathbf{g}_W = \mathbf{\Gamma}\mathbf{\Lambda}^{-1/2}$, the zero-initialized states follow

$$\mathbf{M}_t = \mathscr{T}_{4r}(\beta_1 \mathbf{M}_{t-1} + (1-\beta_1)\mathbf{g}_W), \qquad v_t = \beta_2 v_{t-1} + (1-\beta_2)\|\mathbf{\Gamma}\|_F^2, \tag{16}$$

$$\boldsymbol{\rho}_t = \beta_2 \boldsymbol{\rho}_{t-1} + \frac{1-\beta_2}{d}\operatorname{diag}(\mathbf{g}_W \mathbf{g}_W^\top), \qquad \boldsymbol{\kappa}_t = \beta_2 \boldsymbol{\kappa}_{t-1} + \frac{1-\beta_2}{m}\operatorname{diag}(\mathbf{g}_W^\top \mathbf{g}_W). \tag{17}$$

Here $\beta_1 = 0.9$, $\beta_2 = 0.999$, and $\mathbf{M}_t$ is retained at rank at most $4r$. Bias-corrected row and column profiles are normalized by their respective means, clipped below $\sqrt{\epsilon_{\mathrm{fp32}}}$, and raised to the power $-1/2$ to form diagonal preconditioners $\mathbf{R}_t$ and $\mathbf{C}_t$. This two-sided scaling is related to matrix preconditioning in Shampoo (Gupta et al., 2018); here the factors are diagonal inverse-square-root energy scalings rather than inverse fourth roots of full Gram matrices. Defining $\widetilde{\mathbf{M}}_t = \mathbf{R}_t \mathbf{M}_t \mathbf{C}_t \mathbf{\Lambda}^{1/2}$, FAM returns

$$\mathbf{D}_t = \frac{\|\mathbf{M}_t \mathbf{\Lambda}^{1/2}\|_F}{\|\widetilde{\mathbf{M}}_t\|_F} \frac{\widetilde{\mathbf{M}}_t}{(1-\beta_1^t)\left(\sqrt{v_t/(1-\beta_2^t)} + 10^{-8}\right)}. \tag{18}$$

The norm ratio preserves the unpreconditioned direction's magnitude before scalar second-moment scaling. A zero-energy profile uses unit preconditioning. States are reset at each task boundary, and the current $\mathbf{\Lambda}$ maps the retained weight-coordinate moment into the task-conditioned direction at each step. Equation (6) then applies the scheduled step size to $\mathbf{D}_t$ and realizes the result at rank r.

A.3 First-order factor changes

The space induced by varying both factors at the current point is

$$\mathcal{T} = \{\gamma(\mathbf{Z}_1 \overline{\mathbf{A}} + \mathbf{B}\mathbf{Z}_2) : \mathbf{Z}_1, \mathbf{Z}_2\} = \{\mathbf{Z} : (\mathbf{I}_m - \mathbf{\Pi}_{\mathrm{out}})\mathbf{Z}(\mathbf{I}_d - \mathbf{\Pi}_{\mathrm{in}}) = 0\}. \tag{19}$$

For the second equality, decompose any matrix in the right-hand set as $\mathbf{Z}\mathbf{\Pi}_{\mathrm{in}} + \mathbf{\Pi}_{\mathrm{out}}\mathbf{Z}(\mathbf{I}_d - \mathbf{\Pi}_{\mathrm{in}})$. The first term lies in the row space of $\overline{\mathbf{A}}$; the second lies in the column space of $\mathbf{B}$. The complementary map $\mathbf{G} \mapsto (\mathbf{I}_m - \mathbf{\Pi}_{\mathrm{out}})\mathbf{G}(\mathbf{I}_d - \mathbf{\Pi}_{\mathrm{in}})$ is an orthogonal projection onto $\mathcal{T}^\perp$, which gives Equation (15). For every factor-induced change $\mathbf{H}$, the chain rule yields

$$\mathrm{D}\mathcal{L}_k(\Delta\mathbf{W})[\mathbf{H}] = \langle \mathbf{G}_X, \mathbf{H}\mathbf{\Lambda}^{1/2}\rangle_F = \langle \mathbf{\Gamma}, \mathbf{H}\mathbf{\Lambda}^{1/2}\rangle_F. \tag{20}$$

Thus the projection preserves the first-order task-loss variation attainable by changing the current LoRA factors before optimizer preconditioning. At a rank-deficient point, $\mathcal{T}$ denotes the factor-differential image; at a regular rank-r point it is the rank-r tangent space.

A.4 Metric-consistent rank realization

For a fixed positive metric, the invertible map $\mathbf{Z} \mapsto \mathbf{Z}\mathbf{\Lambda}^{1/2}$ preserves matrix rank and converts the approximation objective into the Frobenius norm:

$$\|(\mathbf{Z} - \mathbf{W}_{\mathrm{prop}})\mathbf{\Lambda}^{1/2}\|_F^2 = \|\mathbf{Z}\mathbf{\Lambda}^{1/2} - \mathbf{X}_{\mathrm{prop}}\|_F^2. \tag{21}$$

The best rank-at-most-r approximation is therefore $\mathscr{T}_r(\mathbf{X}_{\mathrm{prop}})$ in transformed coordinates. Returning through $\mathbf{\Lambda}^{-1/2}$ proves Equation (7). Equation (8) then expresses the realized matrix in standard LoRA factors. A rank-revealing basis of the compact proposal and a small-core eigendecomposition recover its leading directions without a dense $m \times d$ SVD. Recomputing both factors after each step leaves the rank-r subspace free to change during adaptation.

B Implementation and evaluation protocol

B.1 Expert configuration

The shared LLaVA-v1.5-7B backbone remains frozen. The trainable language LoRA parameters are attached to the feed-forward gate, up, and down projections, with rank 8. At each arriving task, the adapted parameters are optimized for one epoch and then frozen. The supervised loss averages over valid target tokens, excluding padding and ignored positions. Training uses two NVIDIA A100 GPUs; batch size, precision, learning rate, and schedule are specified in Section 5.1.

B.2 Router fitting

Support and features. After completing task k, we refit the router using $\mathcal{S}_{\le k} = \bigcup_{e=1}^{k} \mathcal{S}_e$, where each $\mathcal{S}_e$ is drawn from task e's training source with answers removed and task labels retained. A separate frozen CLIP Base model supplies image and text features for both support inputs and test queries. We use its projected image embedding and pooled text representation, and normalize the features to unit length. As each task arrives, we retain its support features, task labels, and group identifiers for subsequent router refitting. Image prototypes use support examples with images; text prototypes use the available instructions. The two class-balanced ridge classifiers ($\alpha = 0.01$) are fitted across task labels, whereas each prototype is formed from one task's support features. Thus prototype scores measure per-task affinity and classifier scores supply relative evidence across the tasks observed so far. At each task boundary, prototypes and classifiers are refitted on the cached features of $\mathcal{S}_{\le k}$ and frozen for inference over the current expert pool. Earlier-task support is used for router fitting, not for expert optimization.

Fusion and inference. The image-prototype, text-prototype, image-classifier, and text-classifier fusion weights are fixed at $(0.20, 0.30, 0.25, 0.25)$, respectively, from the start of each task stream and are shared across CoIN, UCIT, and TriGap. They remain unchanged as prototypes and classifiers are refitted after each task. Each component's task-score vector is centered across tasks and normalized before fusion to obtain the compatibility scores. The highest-scoring and second-highest-scoring experts receive fixed weights of 0.7 and 0.3, respectively, and all other weights are zero, following Equation (12). These expert-mixing weights are distinct from the four score-fusion coefficients above. For a text-only input, routing averages the text-prototype and text-classifier scores and applies the same top-2 gate. At the first stage, the sole expert receives unit weight without score normalization or top-2 selection. The inference model at each stage retains the fitted prototypes, classifier parameters, fusion coefficients, and task-to-expert mapping alongside the expert bank; support examples are not required for generation.

B.3 Scores and result provenance

Task scores. CoIN evaluates ScienceQA on image-containing examples, uses soft VQA accuracy for TextVQA, and counts a grounding prediction as correct when its box has intersection-over-union above 0.5 for REC. UCIT's two captioning tasks report the mean of BLEU-1–4, METEOR, ROUGE-L, and CIDEr after percentage scaling. TriGap uses ANLS for DocVQA and InfographicVQA. The other task scores follow their configured benchmark evaluators.

The three benchmark tables report the final task-ID-free score of each task under its benchmark evaluator. As defined in the main text, AVG assigns equal weight to the final task scores within a benchmark. A dagger marks a project reproduction, while unmarked baseline numbers come from the cited papers.

For baseline numbers not reproduced in the project, CoIN uses the published values collected in SAME Table 2, with PCLR taken from its Table 1 (Xie et al., 2026; meng et al., 2026). PCLR's Grounding score is shown under REC, and its Avg.ACC denotes the final task mean. UCIT and TriGap use the corresponding published comparisons in CRAM Tables 1 and 2 (Tang et al., 2026b). The UCIT Octopus row uses the *Last* block of its Table 1, which denotes the final checkpoint rather than the average over training stages (Liu et al., 2026b). ProtoAda's UCIT and TriGap rows use its Tables 2 and 1, respectively; its AVG is the final task mean (Shi et al., 2026).

B.4 BENCHMARK VISUALIZATIONS

Figures 3–5 illustrate the input and target formats across the three task streams. Each panel pairs an image and instruction with a reference target marked GT.

C EXTENDED ABLATION EVIDENCE

C.1 MATCHED ABLATION CONDITIONS

Adaptation and routing components. Table 4 crosses base language-adapter training with Task-Conditioned Geometric Adaptation (TCGA), and prototype-only routing with Compatibility-Guided Expert Routing (CGER). All conditions share the frozen backbone, task sequence, training examples, adapter placement, and rank. Base training updates the LoRA factors without the task metric, geometric projection, FAM direction, or metric-consistent rank realization. Within each trained expert pool, the routing conditions share support, frozen CLIP features, the top-2 gate with fixed 0.7/0.3 weights, decoding, and evaluator. Because CLIP feature extraction is independent of the trained experts, fitted routing components are reused across expert-training conditions. The full GeoSpec entries correspond to the benchmark results reported in Tables 1 and 2.

Task-metric controls. In Table 5, Identity replaces $\mathbf{\Lambda}$ by $\mathbf{I}$. Permuted reassigns the estimated diagonal entries by a fixed layer-wise permutation, preserving their values while breaking their correspondence to activation coordinates. Each condition uses its assigned metric for the coordinate transform, gradient projection, FAM direction, rank approximation, and factor recovery. The training data, optimizer, rank, and fitted CGER router otherwise remain fixed. Because routing features do not depend on the trained experts, the same query follows the same route in these conditions; AVG compares the answer quality of the resulting expert pools under task-ID-free inference.

Routing components. Table 6 holds the TCGA-trained expert pool fixed. Prototype-only removes the discriminative scores, and classifier-only removes prototype scores. Both variants use fixed score-fusion coefficients throughout the task stream. All three rules share the frozen query features, top-2 gate with fixed 0.7/0.3 weights, decoder, and evaluator. Macro match averages per-task top-2 hit rates with equal task weights: a query matches if its source-task expert is among the top two. AVG measures generated-answer quality and is the primary end-to-end comparison. The CGER entry corresponds to the full-model result in Table 1.

C.2 CONDITIONAL COMPONENT DIFFERENCES

Table 8 reports the four component contrasts implicit in the 2×2 design of Table 4. Each entry subtracts two observed AVG values while holding the other component fixed. TCGA has a positive difference under either routing rule on both benchmarks; the DTS difference is likewise positive with either training procedure.

Table 8: Conditional AVG differences (points) derived from Table 4. Each column changes one component while fixing the other.

	Add TCGA		Add DTS	
Benchmark	PS fixed	CGER fixed	Base fixed	TCGA fixed
UCIT	8.20	3.77	12.05	7.62
TriGap	1.62	3.51	3.75	5.64

D ANALYTICAL COST

D.1 COMPUTATIONAL COMPLEXITY

Let an adapted layer ℓ have weight shape $m_\ell \times d_\ell$, expert rank r, and n_ℓ activation vectors contributing to one metric update. Let q_ℓ denote the proposal's represented rank, K the number of completed tasks, and d_v, d_u the image and text feature dimensions. Table 9 summarizes arithmetic and memory

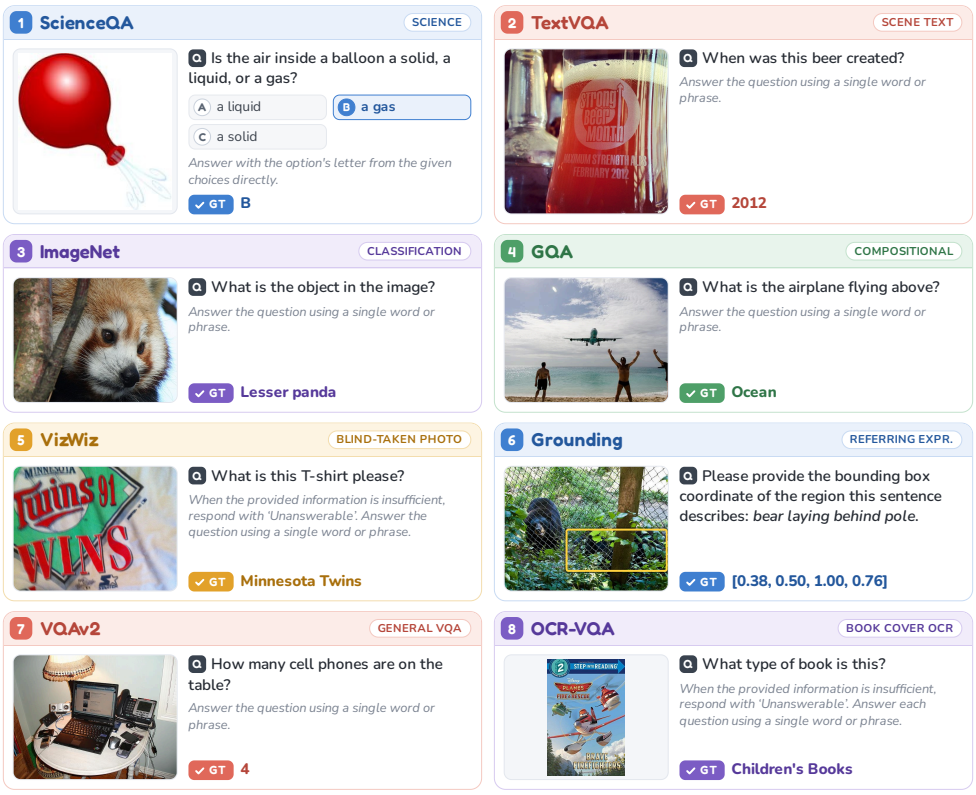

Figure 3: CoIN benchmark visualization. The eight panels illustrate the image-instruction pairs and reference targets of its task stream.

costs. Top-2 mixing evaluates two rank-r experts per adapted layer; identifying them requires an $O(K)$ scan.

Table 9: Analytical arithmetic and memory for adaptation and routing. Training arithmetic is per step; expert-bank memory and routing cost are evaluated after K tasks.

Operation	Arithmetic	Memory
Diagonal metric	$O(\sum_\ell n_\ell d_\ell)$	$O(\sum_\ell d_\ell)$, task-local
Factor-space projection	$O(\sum_\ell (m_\ell + d_\ell) r^2)$	No cross-task memory
FAM direction	$O(\sum_\ell (m_\ell + d_\ell) r^2)$	$O(\sum_\ell r(m_\ell + d_\ell))$, task-local
Rank realization	$O(\sum_\ell (m_\ell + d_\ell) q_\ell^2)$	No cross-task memory
Expert bank	$O(\sum_\ell r(m_\ell + d_\ell))$ per token (two active experts)	$O(K \sum_\ell r(m_\ell + d_\ell))$
Router encoding and scoring	$C_{\text{CLIP}}(z) + O(K(d_v + d_u))$ per query	$O(K(d_v + d_u))$

The projection and FAM bounds use factorized rank-$O(r)$ operations; the rank-realization bound uses a compact rank-q_ℓ proposal and a small-core eigendecomposition. $C_{\text{CLIP}}(z)$ denotes the frozen image and/or text encoder forward pass for query z; its cost depends on the available modalities. The support-feature cache used for stage-wise refitting occupies $O(N_v d_v + N_u d_u)$ memory, where N_v and N_u count the retained image and text support features.

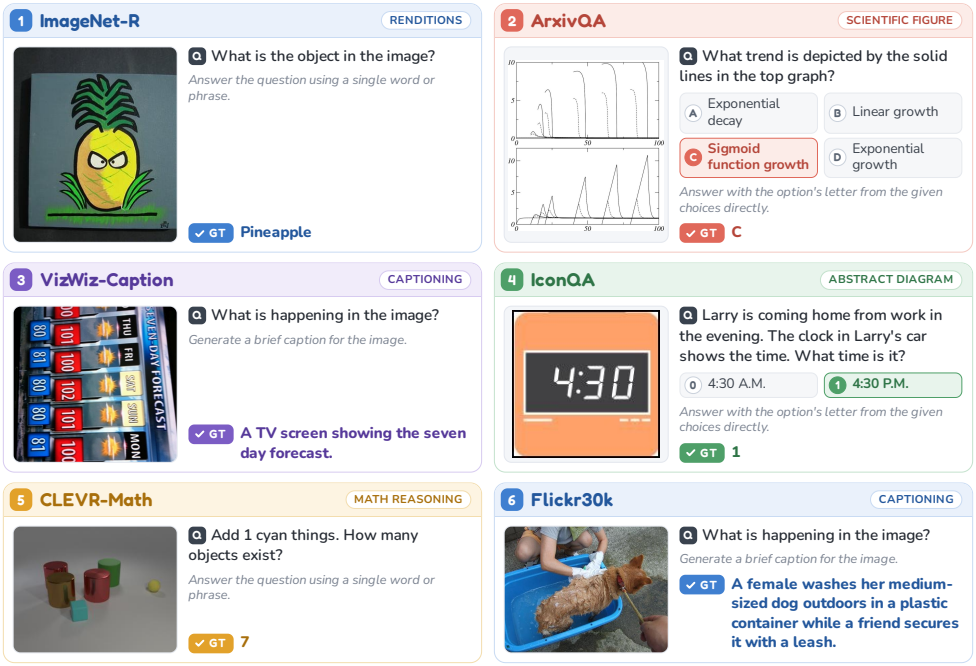

Figure 4: UCIT benchmark visualization. The six panels illustrate the image-instruction pairs and reference targets of its task stream.

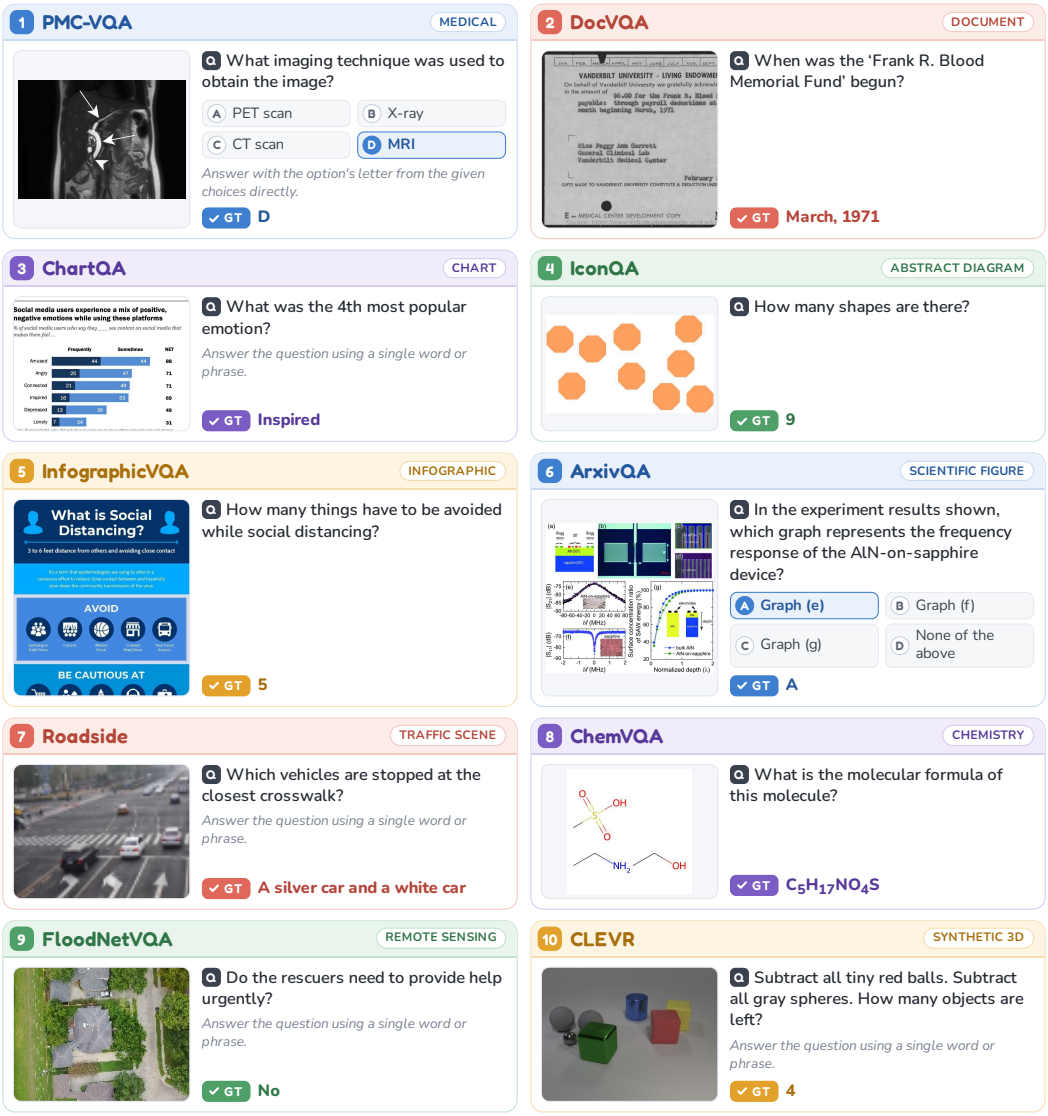

Figure 5: TriGap benchmark visualization. The ten panels illustrate the image-instruction pairs and reference targets of its task stream.