STRLORA: TOWARDS STREAMING CONTINUAL VISUAL INSTRUCTION TUNING FOR MLLMS

Anonymous authors
Paper under double-blind review

ABSTRACT

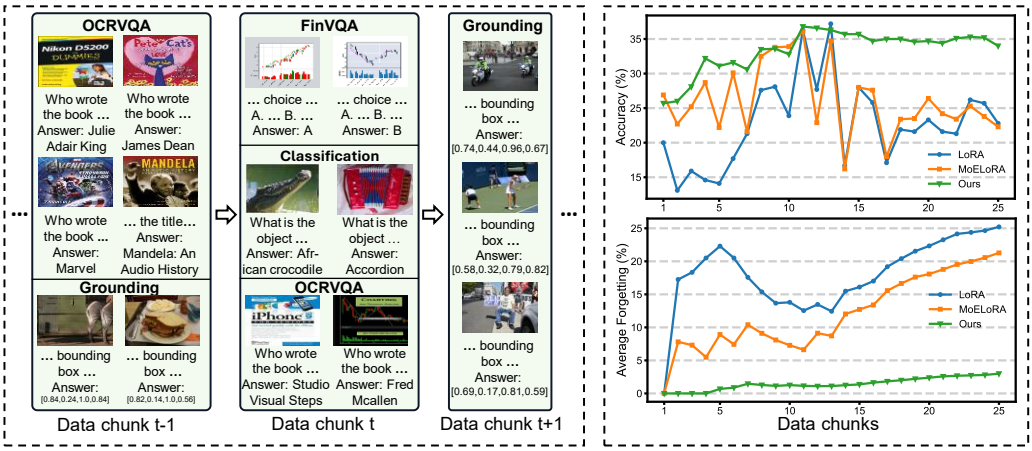
Continual Visual Instruction Tuning (CVIT) enables Multimodal Large Language Models to incrementally acquire new abilities. However, existing CVIT methods operate under a restrictive task-incremental setting, where each training phase corresponds to a single, predefined task. This does not reflect real-world conditions, where data arrives as a continuous stream of interleaved and dynamically evolving tasks. To bridge this gap, we introduce Streaming CVIT (StrCVIT), a more general and realistic setting where models learn from a stream of data chunks containing a dynamic mixture of tasks. In StrCVIT, a model must simultaneously acquire new abilities, reinforce recurring abilities, and mitigate forgetting, which requires it to identify the ability underlying each sample. Existing CVIT methods fail here as they cannot reliably distinguish or adapt to the heterogeneous task samples within each chunk. We therefore propose StrLoRA, a regularized two-stage expert routing framework. StrLoRA first performs task-aware expert selection using the textual instruction to activate a sparse subset of relevant experts, reducing cross-task interference. It then applies token-wise expert weighting within this subset, where contribution weights are computed via cross-modal attention between local visual tokens and the global instruction representation. To maintain stability across the non-stationary stream, a routing-stability regularization aligns current routing distributions with a historical exponential moving average reference. Extensive experiments on a newly developed StrCVIT benchmark show that StrLoRA consistently outperforms existing methods, effectively enhancing the model's abilities from continuously evolving data streams.

1 INTRODUCTION

Multimodal Large Language Models (MLLMs) (Bai et al., 2025; Team et al., 2025; Wang et al., 2025a) have demonstrated remarkable capabilities that extend beyond their text-only counterparts (Grattafiori et al., 2024; Yang et al., 2025). Typically developed through a multi-stage pipeline (Bai et al., 2025; Wang et al., 2025a), these models undergo initial pretraining on large-scale image-text pairs followed by visual instruction tuning to align responses with human intent. For real-world deployment, MLLMs must continuously acquire new vision-language abilities to adapt to evolving requirements. Prior work formalizes this process as Continual Visual Instruction Tuning (CVIT) (Chen et al., 2024; Wang et al., 2025b; Che et al., 2026), where data arrives in discrete phases, each assumed to correspond to a single, predefined task. However, this task-incremental setting is a key limitation: real-world data typically arrives as a continuous, interleaved stream of multiple evolving tasks, not as neatly segregated units.

To bridge this gap, we introduce Streaming CVIT (StrCVIT), a more realistic paradigm where a model learns from a single-pass stream of data chunks. As shown in Fig. 1 (a), each chunk contains a dynamic mixture of interleaved tasks; the number of active tasks, their composition, and sample proportions can all change over time. This setting requires a model to simultaneously acquire new abilities from emerging tasks, reinforce recurring abilities, and mitigate forgetting of abilities absent from the current chunk. As shown in Fig. 1 (b), this induces significant performance instability, with accuracy oscillating and suffering abrupt drops, a critical barrier to reliable, continual learning. We therefore measure forgetting continuously as the relative drop from peak historical performance, capturing the stability of acquired knowledge throughout the stream.

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.



(a) Data stream in StrCVIT (b) Performance over data stream

Figure 1: (a) In StrCVIT, data arrives as a single-pass stream of chunks, each containing a dynamic mixture of interleaved tasks whose composition and proportions shift over time. (b) Performance over the stream exhibits strong oscillations and suffers abrupt drops, while our proposed StrLoRA maintains stable accuracy and consistently low forgetting.

Existing CVIT methods are ill-suited for StrCVIT. They often employ Mixture-of-Experts (MoE) architectures (Fedus et al., 2022; Shazeer et al., 2017) with phase-specific routing (Che et al., 2026; Ge et al., 2025; Song et al., 2026), which presumes that each new phase introduces a single task and assigns samples within that phase to the same expert or expert subset. In StrCVIT, where chunks contain heterogeneous samples from multiple tasks, this strategy fails. The router cannot produce stable, discriminative routing weights, leading to homogeneous expert utilization, where diverse samples activate experts in highly similar ways (Fig. 2 (a)), undermining parameter isolation and causing cross-task interference.

We identify that textual instructions provide a stable, discriminative signal for distinguishing tasks, unlike visually entangled features (Fig. 2 (b)). Leveraging this insight, we propose StrLoRA, a regularized two-stage expert routing framework. First, it performs task-aware expert selection, using the textual instruction to activate a sparse, relevant subset of LoRA experts, thereby reducing interference. Second, within this subset, a token-wise expert weighting module computes fine-grained contribution weights by attending to both local visual tokens and the global instruction. To combat instability in the data stream, we introduce a routing-stability regularization that aligns current routing distributions with a historical exponential moving average, ensuring smooth evolution of expert contribution weights. Our design thus decouples stable, text-guided expert filtering from adaptive, cross-modal token weighting.

We construct a comprehensive benchmark to simulate the StrCVIT scenario. Extensive experiments show that StrLoRA consistently outperforms existing methods, effectively enhancing the model's abilities from continuously evolving data streams.

Our main contributions are summarized as follows:

- We introduce StrCVIT, a new and realistic continual learning setting for MLLMs that models data as a single-pass stream of interleaved and dynamically evolving tasks.
- We diagnose the homogeneous expert utilization problem in StrCVIT and propose StrLoRA, a regularized two-stage expert routing framework that decouples task-aware expert selection from token-wise expert weighting, stabilized by a regularization loss.
- We construct a StrCVIT benchmark and experimentally validate that StrLoRA achieves stable performance, effectively enhancing abilities from evolving data streams while minimizing forgetting.

2 Streaming Continual Visual Instruction Tuning

StrCVIT. Given a pre-trained MLLM f_θ, StrCVIT continually enhances its capabilities by fine-tuning on a single-pass data stream, which is formulated as a sequence of data chunks

$\mathcal{S} = (d_1, d_2, \ldots, d_t, d_{t+1}, \ldots)$. At each step t, the incoming data chunk is defined as $d_t = \{(x_i^{\text{vis}}, x_i^{\text{text}}, y_i^{\text{gt}})\}_{i=1}^{n_t}$ with n_t samples, where each sample consists of a visual input, a textual instruction, and a ground-truth response. The samples in d_t are drawn from a mixture of multiple interleaved task distributions:

$$P_t = \sum_{m=1}^{M} \pi_t^{(m)} P^{(m)}, \tag{1}$$

where M denotes the total number of task distributions observed in the stream so far, and $\pi_t^{(m)} \in [0, 1]$ denotes the mixture weight of task distribution $P^{(m)}$ at time step t, with $\sum_{m=1}^{M} \pi_t^{(m)} = 1$. The number, composition, and sample proportions of interleaved tasks can all change dynamically across chunks, and task identities are not explicitly provided to the model. Under this formulation, the overall optimization objective over the streaming process up to time step T is:

$$\min_{\theta_1,\ldots,\theta_T} \sum_{t=1}^{T} \mathbb{E}_{(x^{\text{vis}}, x^{\text{text}}, y^{\text{gt}}) \sim d_t} \left[\mathcal{L}\left(f_{\theta_t}(x^{\text{vis}}, x^{\text{text}}), y^{\text{gt}} \right) \right], \quad \text{s.t. } \theta_t \leftarrow \theta_{t-1}, \tag{2}$$

where θ_t denotes the model parameters obtained after updating θ_{t-1} on d_t.

Revisiting CVIT. The CVIT setting (Che et al., 2026; Wang et al., 2025b) presents the model with a sequence of tasks $\mathcal{T} = \{\tau_1, \tau_2, \ldots, \tau_n\}$ to acquire new vision-language abilities incrementally, where each task τ_k is associated with a dataset $\mathcal{D}_k = \{(x_i^{\text{vis}}, x_i^{\text{text}}, y_i^{\text{gt}})\}_{i=1}^{n_k}$. This setting can be regarded as a restricted form of equation 1 in which each task corresponds to a single previously unseen distribution. At step k, the mixture reduces to $\pi_k^{(k)} = 1$ and $\pi_k^{(m)} = 0$ for all $m \neq k$. The corresponding optimization objective is:

$$\min_{\theta_1,\ldots,\theta_n} \sum_{k=1}^{n} \mathbb{E}_{(x^{\text{vis}}, x^{\text{text}}, y^{\text{gt}}) \sim \mathcal{D}_k} \left[\mathcal{L}\left(f_{\theta_k}(x^{\text{vis}}, x^{\text{text}}; \tau_k), y^{\text{gt}} \right) \right], \quad \text{s.t. } \theta_k \leftarrow \theta_{k-1}, \tag{3}$$

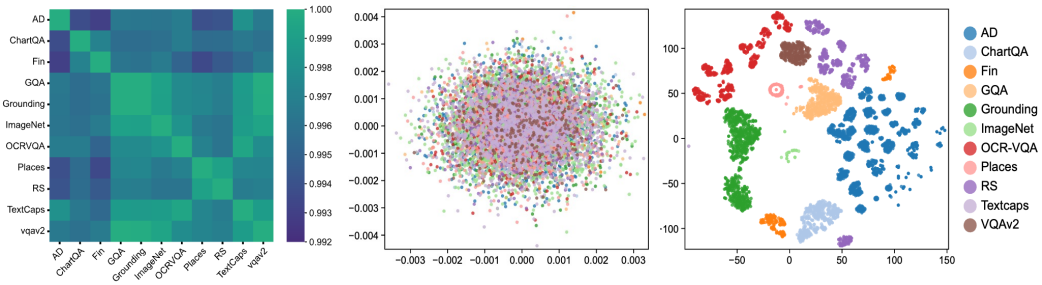
where $f_{\theta_k}(\cdot; \tau_k)$ denotes the model that can explicitly or implicitly leverage the task identity τ_k for specific adaptation.

Key Challenges in StrCVIT. In CVIT, the model focuses on incrementally learning a single new task at each phase while mitigating forgetting of prior tasks. StrCVIT, in contrast, requires the model to learn from multiple tasks whose samples are interleaved within each chunk. The model must therefore acquire new abilities from emerging tasks, reinforce existing abilities from recurring tasks, and mitigate forgetting of abilities not represented in the current chunk, which requires the model to identify the ability underlying each sample. This fundamental difference necessitates distinct evaluation paradigms. While CVIT typically measures forgetting as the performance drop on previous tasks after learning new ones, StrCVIT requires continuous assessment: because performance may fluctuate with each chunk, we measure forgetting at every step relative to the best historical accuracy, capturing the stability of all acquired abilities throughout the entire learning stream. Detailed evaluation metrics are provided in Section 4.1.

3 StrLoRA

Existing CVIT methods typically employ a LoRA-based MoE architecture to reduce interference among different tasks. Specifically, the architecture maintains N LoRA blocks as experts, each parameterized by a low-rank matrix product BA inserted into the frozen pretrained weight W_0, and a phase-specific router that assigns samples from the newly arriving task to the same expert or expert subset. However, in StrCVIT, each data chunk consists of interleaved samples from multiple tasks that dynamically shift over time. This prevents the router from relying on this phase-specific routing, making it difficult to generate stable and discriminative routing patterns for heterogeneous samples. As illustrated in Fig. 2(a), the router tends to produce highly similar expert-weight distributions across diverse samples, leading to homogeneous expert utilization that ultimately undermines effective expert specialization.

To address the unique challenges of StrCVIT, we propose StrLoRA, a regularized two-stage expert routing framework. First, it performs task-aware expert selection using the textual instruction to



(a) Router Distribution Similarity

(b) t-SNE Visualization of Visual and Textual Features

Figure 2: (a) CKA (Kornblith et al., 2019) similarity heatmap of StrCVIT, showing that standard MoE routing produces highly similar expert activation weights for samples from different tasks. (b) t-SNE visualization (Van der Maaten & Hinton, 2008) of visual and textual features, showing that textual embeddings form well-separated clusters and provide a more reliable signal for task discrimination than overlapping visual features.

activate a sparse, relevant subset of experts, reducing cross-task interference. Then, it conducts token-wise expert weighting within that subset, assigning contribution weights by attending to both local visual tokens and the global instruction. To maintain stability in streaming data, a routing-stability regularization aligns current allocations within the selected subset with a historical EMA reference. This design leverages textual stability for coarse filtering and cross-modal synergy for fine-grained adaptation, making it robust for interleaved, evolving task streams.

3.1 Task-Aware Expert Selection

In multimodal instruction data, the textual instruction typically defines the task to be performed on the accompanying image. We investigate whether this textual information can provide a robust and discriminative signal for expert routing. A dimensionality reduction analysis, shown in Fig. 2(b), reveals a clear disparity: text embeddings form well-separated clusters corresponding to different tasks, while visual features exhibit significant overlap. This critical observation suggests that the instruction text offers a stable and reliable signal for distinguishing task types, unlike the more ambiguous visual modality.

Driven by this insight, we use the text embedding to perform task-aware expert selection, which selects the subset of experts to activate for a given sample. This approach directly addresses the homogeneous routing problem by leveraging the most discriminative available signal. Let $\mathbf{x}_i^{\text{text}} = \frac{1}{L_i^{\text{text}}} \sum_{m=1}^{L_i^{\text{text}}} \mathbf{t}_i^m \in \mathbb{R}^{d_e}$ denote the global text embedding for sample i obtained by mean pooling over L_i^{text} textual token embeddings, where $\mathbf{t}_i^m \in \mathbb{R}^{d_e}$ is the m-th textual token embedding, and d_e is the embedding dimension. A linear router computes a probability distribution over all N experts:

$$p_{ij} = \frac{\exp\big((W_g \mathbf{x}_i^{\text{text}})_j\big)}{\sum_{j'=1}^{N} \exp\big((W_g \mathbf{x}_i^{\text{text}})_{j'}\big)}, \quad \mathbf{p}_i = [p_{ij}]_{j=1}^{N}, \tag{4}$$

where $W_g \in \mathbb{R}^{N \times d_e}$ is the router projection matrix. p_{ij} is the routing probability assigned to expert j for sample i, and $\mathbf{p}_i$ is the full distribution over all experts. Then, we define the expert subset $\mathcal{S}_i = \text{Top-K}(\mathbf{p}_i)$ by retaining the K experts with the highest routing probabilities for sample i. Since the router operates on the global text embedding, samples with similar intents activate consistent expert subsets across the stream, reducing interference from unrelated task intents.

3.2 Token-Wise Expert Weighting

Guided by the analysis that the textual instruction provides a clear, task-discriminative signal, we first use the text embedding to perform task-aware expert selection. This determines a relevant subset of experts $\mathcal{S}_i$ to activate for a given input sample i, effectively filtering out irrelevant experts and mitigating interference.

However, applying uniform expert weights to all tokens within a sample is suboptimal, as different visual regions correspond to distinct features whose relevance depends on the instructional con-

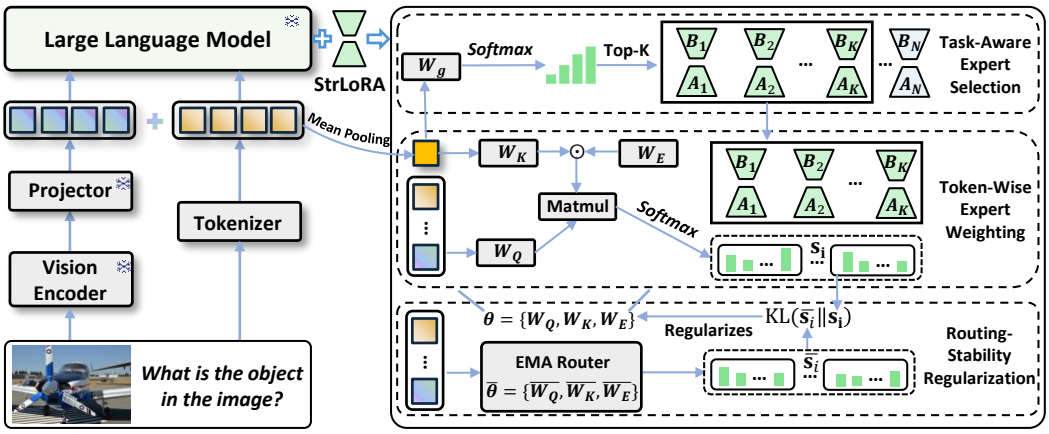


Figure 3: StrLoRA framework. StrLoRA first performs task-aware expert selection to activate a sparse subset of relevant experts. It then conducts token-wise expert weighting within the selected subset, assigning contribution weights. To maintain stability in streaming data, a routing-stability regularization aligns current allocations with a historical EMA reference.

text. Therefore, we introduce a second stage of token-wise expert weighting to compute adaptive contribution weights within the selected expert subset $\mathcal{S}_i$.

Formally, for the l-th token of sample i with hidden state $h_i^l \in \mathbb{R}^{d_{\text{in}}}$, we project it and the global text embedding $\mathbf{x}_i^{\text{text}}$ into a shared D-dimensional routing space using learnable matrices W_Q and W_K, respectively. Each expert E_j is associated with a learnable feature vector $\mathbf{e}_j \in \mathbb{R}^D$. We denote the collection of expert feature vectors by $W_E = \{\mathbf{e}_j\}_{j=1}^N$. The routing logit for expert E_j (where $j \in \mathcal{S}_i$) is calculated as:

$$z_{ij}^l = \begin{cases} \dfrac{(W_Q h_i^l) \cdot \left((W_K \mathbf{x}_i^{\text{text}}) \odot \mathbf{e}_j\right)}{\sqrt{D}}, & j \in \mathcal{S}_i, \\ -\infty, & j \notin \mathcal{S}_i. \end{cases} \tag{5}$$

where $\odot$ denotes the Hadamard product. The key term $W_K \mathbf{x}_i^{\text{text}} \odot \mathbf{e}_j$ produces an instruction-adapted expert representation, dynamically modulating the expert's profile based on the current task. The inner product then scores the relevance of this adapted expert to the specific visual token h_i^l. The logits are normalized over the selected subset $\mathcal{S}_i$ to obtain the final contribution weights:

$$s_{ij}^l = \frac{\exp(z_{ij}^l)}{\sum_{j' \in \mathcal{S}_i} \exp(z_{ij'}^l)}. \tag{6}$$

For experts not in $\mathcal{S}_i$, the weight s_{ij}^l is zero. The final adapted representation $\hat{h}_i^l$ combines the frozen pre-trained weight W_0 with a weighted sum of the activated LoRA experts (B_j, A_j):

$$\hat{h}_i^l = W_0 h_i^l + \sum_{j \in \mathcal{S}_i} s_{ij}^l \cdot B_j A_j h_i^l, \tag{7}$$

where $W_0 \in \mathbb{R}^{d_{\text{out}} \times d_{\text{in}}}$, $B_j \in \mathbb{R}^{d_{\text{out}} \times r}$ and $A_j \in \mathbb{R}^{r \times d_{\text{in}}}$ with $r \ll \min(d_{\text{in}}, d_{\text{out}})$.

3.3 Routing-Stability Regularization

As the data stream evolves, rapid shifts in task distribution can cause the token-level routing module to overfit to the current chunk, deviating from previously learned allocation patterns. This instability undermines consistent expert specialization and can lead to performance fluctuations. To promote stable yet adaptive expert weight allocation, we introduce a routing-stability regularization loss. This loss encourages the token-level routing distribution to evolve smoothly within the currently activated expert subset, preserving historical allocation knowledge while remaining adaptable to new data. We maintain an exponential moving average (EMA) $\bar{\theta} = \{\bar{W}_Q, \bar{W}_K, \bar{W}_E\}$ of the routing parameters,

updated after each training step u as: $\bar{\theta}^{(u)} = \beta\,\bar{\theta}^{(u-1)} + (1-\beta)\,\theta^{(u)}$, where $\beta \in [0,1)$ is the momentum coefficient. The EMA parameters serve as a stabilized reference of past routing behavior.

For a given sample i with selected expert subset $\mathcal{S}_i$, we compute the historical routing weights $\bar{s}_{ij}^l$ using $\bar{\theta}$ via the same routing equations (equation 5 and equation 6). The regularization term is then defined as the KL divergence (Kullback & Leibler, 1951) from the historical distribution to the current one, summed over all tokens:

$$\mathcal{L}_{\text{reg}} = \sum_{l=1}^{L_i} \text{KL}\big(\bar{\mathbf{s}}_i^{\,l} \,\big\|\, \mathbf{s}_i^l\big)\,, \tag{8}$$

where L_i is the number of tokens in sample i, and $\text{KL}(\bar{\mathbf{s}}_i^{\,l} \| \mathbf{s}_i^l) = \sum_{j\in\mathcal{S}_i} \bar{s}_{ij}^l \log \frac{\bar{s}_{ij}^l}{s_{ij}^l}$.

By penalizing large deviations from the smoothed historical routing, $\mathcal{L}_{\text{reg}}$ prevents abrupt reallocation of expert contributions, thus preserving routing consistency across the learning stream. The final training objective combines this with the standard instruction-tuning loss $\mathcal{L}$ defined in equation 2:

$$\mathcal{L}_{\text{total}} = \mathcal{L} + \lambda \mathcal{L}_{\text{reg}}, \tag{9}$$

where λ balances task performance and routing stability. This design allows the model to adapt continually without forgetting how experts were productively assigned in the past, a key capability for learning robustly from non-stationary, interleaved task streams.

4 Experimental Results

4.1 Benchmark Setup

Streaming Data Construction. To approximate a realistic data stream, we integrate vision-language datasets from a range of domains, including ImageNet (Deng et al., 2009), Places365 (Zhou et al., 2017), AD (Sima et al., 2024), Fin (Zhao et al., 2025a), OCRVQA (Mishra et al., 2019), Grounding (Chen et al., 2024), TextCaps (Sidorov et al., 2020), VQAv2 (Goyal et al., 2017), RS (Lobry et al., 2020), ChartQA (Masry et al., 2022), and GQA (Hudson & Manning, 2019). At each time step t, we construct a data chunk by randomly sampling 2,000 samples from a randomly selected subset of 1 to 11 datasets, where the proportion contributed by each dataset varies and all samples are randomly shuffled. In this way, each chunk may contain samples from multiple interleaved distributions, and the stream composition changes over time. Detailed data stream statistics are presented in Appendix B.

Evaluation Metrics. Inspired by (Chaudhry et al., 2019), we measure forgetting to quantify the performance degradation of the model on each dataset over the stream. Specifically, let a_t^m denote the accuracy score on the m-th dataset after the model processes the t-th data chunk. The forgetting on dataset m at time step t is defined as:

$$\text{F}_t^m = \max\left(0, \frac{\max_{1\le j<t} a_j^m - a_t^m}{\max_{1\le j<t} a_j^m}\right). \tag{10}$$

This measures the relative performance drop on dataset m with respect to its historical best previous state. To evaluate the model's performance on each individual dataset across the entire stream, we compute Average Performance (AP) (Che et al., 2026) and Average Forgetting (AF):

$$\text{AP}_t^m = \frac{1}{t}\sum_{j=1}^{t} a_j^m, \qquad \text{AF}_t^m = \frac{1}{t}\sum_{j=1}^{t} \text{F}_j^m. \tag{11}$$

To comprehensively evaluate the model's overall performance across all datasets, we introduce Mean Average Performance (MAP) and Mean Average Forgetting (MAF):

$$\text{MAP}_t = \frac{1}{M}\sum_{m=1}^{M} \text{AP}_t^m, \qquad \text{MAF}_t = \frac{1}{M}\sum_{m=1}^{M} \text{AF}_t^m. \tag{12}$$

Table 1: Evaluation results (%) over 25 data chunks for StrCVIT across different base models. Higher AP/MAP is better, while lower AF/MAF is better.

Method	Places365		RS		AD		VQAv2		Fin		Grounding		OCRVQA		TextCaps		ImageNet		Overall	
	AP↑	AF↓	AP↑	AF↓	AP↑	AF↓	AP↑	AF↓	AP↑	AF↓	AP↑	AF↓	AP↑	AF↓	AP↑	AF↓	AP↑	AF↓	MAP↑	MAF↓
InternVL3.5-8B-Pretrained																				
Zero-shot	4.70	–	53.40	–	23.00	–	28.70	–	63.70	–	24.90	–	25.70	–	49.68	–	26.10	–	33.32	–
LoRA	22.78	25.22	68.73	1.26	32.20	4.11	69.20	1.28	74.23	0.79	57.40	3.99	**54.36**	2.13	83.89	0.96	73.48	3.19	59.59	4.77
EWC	24.45	21.62	68.36	1.00	31.19	**2.02**	70.15	0.83	72.97	0.79	**59.40**	**2.39**	52.34	3.71	83.05	1.17	66.25	3.48	58.68	4.11
Replay	21.62	28.85	67.96	2.09	31.42	4.41	**70.25**	0.81	74.31	1.22	57.34	3.32	54.25	2.03	84.25	0.91	72.75	3.49	59.35	5.24
MoELoRA	26.14	21.27	69.10	1.62	33.94	4.51	70.14	**0.72**	74.23	0.84	55.30	6.90	53.67	**1.86**	84.41	**0.65**	74.78	3.81	60.19	4.69
SMoLoRA	17.34	20.21	64.29	2.97	29.27	4.31	69.27	1.55	71.52	1.04	43.93	8.23	40.71	4.00	75.83	5.54	49.98	3.80	51.35	5.74
StrLoRA	**33.37**	**2.99**	**69.65**	**0.75**	**36.27**	3.92	70.11	1.13	**75.95**	**0.38**	57.36	2.53	53.00	3.29	**84.48**	0.74	**76.49**	**1.46**	**61.85**	**1.91**
Gemma3-4B-PT																				
Zero-shot	13.50	–	45.50	–	18.30	–	14.70	–	1.50	–	0.00	–	6.00	–	0.09	–	32.20	–	14.64	–
LoRA	32.08	6.92	58.88	4.68	24.49	20.06	55.67	2.21	67.05	3.57	5.16	17.90	56.22	4.10	70.79	3.67	63.57	**3.14**	48.21	7.36
EWC	31.67	7.17	58.89	4.80	24.57	21.72	55.53	2.50	66.85	3.85	5.00	18.37	56.42	3.74	70.84	3.14	62.82	4.31	48.07	7.73
Replay	31.36	7.12	59.96	5.41	24.31	14.76	**56.04**	1.78	68.90	1.82	4.49	30.88	57.92	**2.38**	70.77	3.17	63.93	3.16	48.63	7.83
MoELoRA	**32.70**	**5.36**	58.30	5.17	27.68	11.45	55.15	3.27	66.72	4.71	4.98	26.21	56.04	2.95	**71.26**	2.78	63.58	3.63	48.49	7.28
SMoLoRA	29.05	9.60	58.48	4.62	21.51	17.68	54.49	2.76	67.58	2.81	3.42	27.16	50.45	2.71	69.46	**2.51**	62.74	3.87	46.36	8.19
StrLoRA	32.36	6.66	**60.52**	**3.30**	**32.63**	**8.02**	55.00	**0.91**	**72.20**	**0.87**	**6.20**	**12.88**	**58.32**	3.64	68.45	8.71	**73.54**	4.15	**51.03**	**5.46**

4.2 Experimental Setup

Compared Methods. We compare our method with existing approaches applicable to StrCVIT. **LoRA** (Hu et al., 2021) fine-tunes the model using a single shared LoRA module across all tasks. We also include classical CL approaches such as **EWC** (Kirkpatrick et al., 2017) which constrains updates on parameters important to previous tasks, and **Replay** (Chaudhry et al., 2019) stores or generates past samples for rehearsal during training on new data. In addition, we evaluate methods designed for CVIT, including **MoELoRA** (Chen et al., 2024), which employs a mixture-of-experts module over LoRA experts, and **SMoLoRA** (Wang et al., 2025b), which introduces separate routing mechanisms for visual understanding and instruction following across different tasks. Finally, **Zero-shot** evaluates the base model without any fine-tuning.

Base Models. To comprehensively evaluate performance across different model architectures and scales, we conduct experiments on the StrCVIT benchmark using InternVL3.5-8B-Pretrained (Wang et al., 2025a) and Gemma3-4B-PT (Team et al., 2025) as base models, with additional results on InternVL3.5-4B-Pretrained presented in Appendix D.2.

4.3 Main Results

StrCVIT Setting. We evaluate StrLoRA against all baselines over 25 streaming data chunks across different model architectures and scales. As shown in Table 1, on InternVL3.5-8B-Pretrained, all compared methods exhibit substantial forgetting on at least one dataset. Although SMoLoRA mitigates inter-task interference in CVIT through specific routing, it struggles with mixed data streams spanning multiple tasks. In contrast, StrLoRA achieves strong accuracy on most datasets, notably improving AP on Places365 to 33.37 while reducing AF to 2.99, and reaches overall performance with an MAP of 61.85 and an MAF of 1.91. These results demonstrate that StrLoRA continuously strengthens multimodal capabilities on incoming data while maintaining consistently low forgetting in StrCVIT. In the cross-architecture setting with Gemma3-4B-PT, StrLoRA also achieves stable performance and outperforms all baselines on overall performance, highlighting the robustness of StrLoRA across different architectures.

CVIT Setting. We evaluate StrLoRA on the CoIN benchmark (Chen et al., 2024) to demonstrate that our method can also generalize to the standard CVIT setting. Following the CoIN setting, we use LLaVA-v1.5-7B (Liu et al., 2023) as the backbone. We compare StrLoRA with multiple public CVIT baselines under the original instruction setting. As shown in Table 2, StrLoRA outperforms all compared methods on the overall performance metrics.

Visualization of Overall Performance. We further visualize the evolution of MAP and MAF on InternVL3.5-8B-Pretrained as the data stream progresses. As shown in Fig. 4, StrLoRA maintains a clear lead on the accuracy curve while maintaining the lowest forgetting throughout the stream.

Table 2: Results (%) on the CoIN benchmark. MFT denotes mean fine-tuning accuracy across tasks, MFN denotes mean final accuracy across the eight tasks, MAA denotes mean average accuracy over all tasks, and BWT measures backward transfer, where negative values indicate forgetting.

Method	ScienceQA↑	TextVQA↑	ImageNet↑	GQA↑	VizWiz↑	Grounding↑	VQAv2↑	OCR-VQA↑	MFT↑	MFN↑	MAA↑	BWT↑
MoELoRA (Chen et al., 2024)	72.91	48.82	9.58	42.55	36.70	3.40	48.62	63.87	62.63	40.81	45.00	-21.83
BranchLoRA (Zhang et al., 2025)	68.24	40.18	24.60	41.40	49.83	15.94	51.23	62.14	65.18	44.20	49.94	-20.98
KSS-MoE (Song et al., 2026)	68.73	54.79	57.33	52.11	42.86	15.37	60.69	47.68	61.20	49.95	59.99	-11.25
SMoLoRA (Wang et al., 2025b)	79.75	47.99	**93.88**	42.44	46.03	17.80	53.64	57.69	63.82	54.90	63.01	-8.92
ProgLoRA (Yu et al., 2025)	74.84	51.83	83.90	49.93	**53.87**	31.19	62.71	64.44	65.68	59.09	62.38	-6.59
StrLoRA	**83.59**	**61.87**	59.49	**61.04**	38.74	**59.80**	**64.80**	**65.50**	**68.42**	**61.85**	**67.90**	**-6.57**

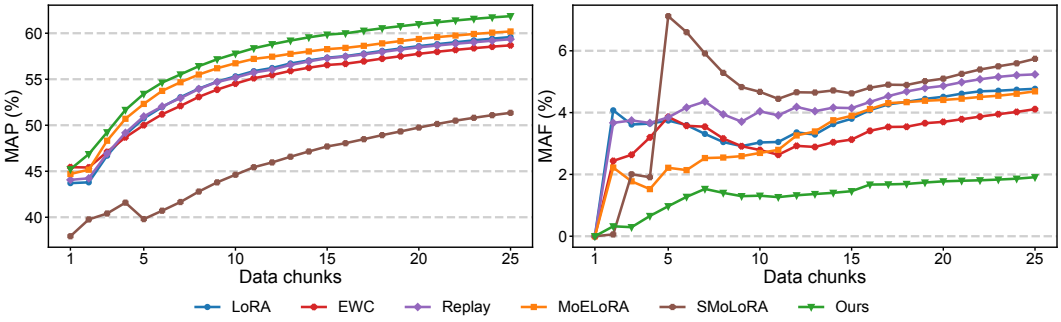


Figure 4: Overall performance over the data stream on InternVL3.5-8B-Pretrained.

This trend highlights that StrLoRA not only reaches stronger final performance, but also preserves a more stable adaptation trajectory during streaming learning.

Expert Activation Analysis. To further explore whether StrLoRA can stably distinguish heterogeneous samples and achieve expert specialization, we visualize the expert activation proportions across layers for each task in Fig. 5. The results show that samples from different tasks activate distinct subsets of experts with varying proportions, exhibiting distinct expert preferences across layers. This suggests that StrLoRA effectively partitions the parameter space to reduce interference among heterogeneous samples drawn from multiple interleaved task distributions.

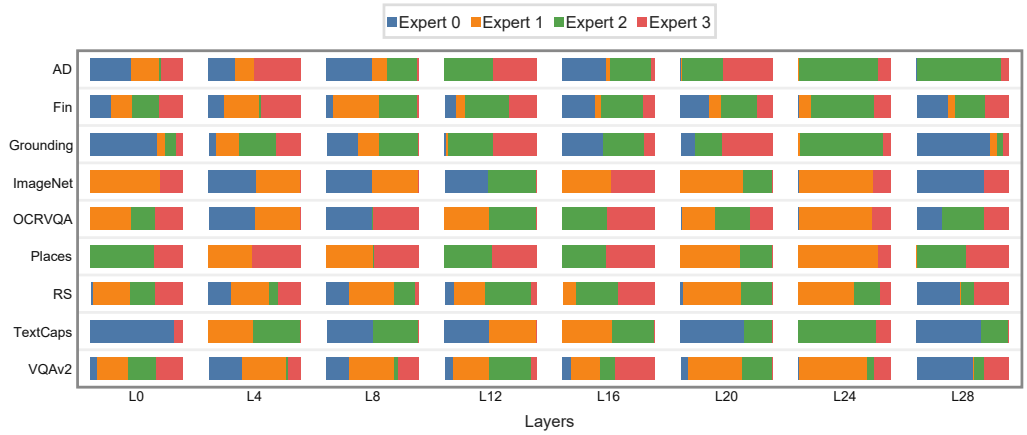


Figure 5: Expert activation proportions across tasks. Each stacked bar represents the activation proportion of each expert, averaged over all samples.

4.4 Ablation Study

We conduct ablation studies to examine the contribution of each component in StrLoRA using InternVL3.5-8B-Pretrained. Specifically, p_{ij} denotes sample-level expert weights. s_{ij}^{l} denotes token-level expert weights. $\mathcal{L}_{\text{reg}}$ denotes the routing-stability regularization term. We also include the MoELoRA setting, where none of these components are applied. As shown in Table 3, using p_{ij} alone improves the MAP from 60.19 to 60.77 and reduces MAF from 4.69 to 3.54, but the AF on Places365 remains high at 16.05, suggesting that uniform weighting is insufficient to prevent forgetting. Using s_{ij}^{l} alone also improves MAP and MAF, but the improvement is limited without p_{ij} for Top-K expert selection to separate heterogeneous samples. Combining p_{ij} and s_{ij}^{l} further improves MAP, demonstrating that sample-level expert selection and token-level weight assignment

Table 3: Ablation study of the components within StrLoRA on InternVL3.5-8B.

Component			Places365		RS		AD		VQAv2		Fin		Grounding		OCRVQA		TextCaps		ImageNet		Overall	
p_{ij}	s_{ij}^{l}	$\mathcal{L}_{\text{reg}}$	AP↑	AF↓	AP↑	AF↓	AP↑	AF↓	AP↑	AF↓	AP↑	AF↓	AP↑	AF↓	AP↑	AF↓	AP↑	AF↓	AP↑	AF↓	MAP↑	MAF↓
×	×	×	26.14	21.27	69.10	1.62	33.94	4.51	70.14	0.72	74.23	0.84	55.30	6.90	53.67	1.86	84.41	0.65	74.78	3.81	60.19	4.69
✓	×	×	28.91	16.05	70.18	2.22	34.72	3.37	70.48	0.60	76.54	0.82	57.88	3.85	49.89	2.12	83.71	1.32	74.59	1.49	60.77	3.54
×	✓	×	27.04	19.40	68.34	0.67	33.58	2.91	70.30	1.14	73.60	0.71	56.36	4.10	57.60	4.55	83.89	0.77	73.68	2.22	60.49	4.05
×	✓	✓	28.78	10.18	68.80	1.34	32.30	3.25	70.36	0.85	73.24	1.97	56.21	4.76	58.57	3.78	84.04	0.95	73.50	2.10	60.64	3.24
✓	✓	×	30.06	11.78	68.88	2.31	35.86	2.26	69.44	1.30	76.75	0.39	60.17	3.76	54.39	1.32	85.00	0.81	73.65	2.78	61.58	2.97
✓	✓	✓	33.37	2.99	69.65	0.75	36.27	3.92	70.11	1.13	75.95	0.38	57.36	2.53	53.00	3.29	84.48	0.74	76.49	1.46	61.85	1.91

are complementary. With all components enabled, StrLoRA achieves the best overall performance. Notably, on Places365, AF drops from 21.27 to 2.99, confirming that StrLoRA maintains strong accuracy with consistently low forgetting across the training process. Additional ablation studies are presented in Appendix D.4.

5 Related Work

5.1 Online Continual Learning

Beyond the traditional class-incremental continual learning setting, task-free continual learning (Aljundi et al., 2019) introduces a task-free formulation in which task boundaries are unavailable during training. DCM (Ye & Bors, 2024) further studies online task-free continual learning in a setting where class identities are unavailable and supervised signals may also be inaccessible during training. MiD-Blurry (Wang et al., 2026) introduces an online continual learning benchmark that combines multiple class distribution patterns with blurred temporal boundaries. Nevertheless, these studies are confined to classification-oriented continual learning with homogeneous vision-language tasks, and thus do not align with the practical need for MLLMs to continually acquire and reinforce diverse multimodal capabilities. This mismatch in task scope and learning objectives makes their methods not directly applicable to StrCVIT. Although OASIS (Lee et al., 2025) introduces online sample selection for CVIT, it still assumes a predefined task-incremental sequence, rather than a fully streaming scenario with continuously mixed multimodal tasks.

5.2 CVIT Methods

To mitigate task interference in CVIT, a wide range of methods have been proposed under the task-incremental setting. BranchLoRA (Zhang et al., 2025) shares LoRA A matrices, learns task-specific B branches, and uses task-specific routers. ProgLoRA (Yu et al., 2025) maintains a progressive LoRA pool and trains a new LoRA block for each new task. CL-MoE (Huai et al., 2025) distinguishes task-shared and task-specific experts selected by dual routers and updates them with different momentum weights as new tasks arrive. SMoLoRA (Wang et al., 2025b) introduces separate routing mechanisms for visual understanding and instruction following across different tasks. D-MoLE (Ge et al., 2025) dynamically allocates new task-specific LoRA experts across layers for each new task under a parameter budget. LiLoRA (Che et al., 2026) dynamically expands lightweight task-specific LoRA modules for each new task. KSS-MoE (Song et al., 2026) adds a new task-specific expert for each new task and regularizes it to be distinct from previous experts. However, these methods rely on the setting that each incoming training phase is a single predefined task to enable explicit task-specific module allocation, and thus cannot be directly applied to StrCVIT.

6 Conclusion

In this paper, we introduce StrCVIT, a new and realistic continual learning setting for MLLMs that models data as a single-pass stream of interleaved and dynamically evolving tasks. This setting exposes a key limitation of existing CVIT methods, where standard MoE routing often leads to homogeneous expert utilization, weakened expert specialization, and increased cross-task interference. To address this challenge, we propose StrLoRA, a regularized two-stage routing framework that combines task-aware expert selection, cross-modal token-wise expert weighting, and EMA-based routing-stability regularization. Extensive experiments on the proposed StrCVIT benchmark demonstrate that StrLoRA maintains low forgetting while consistently improving the model's ability to learn from continuously evolving data streams.

AI USE STATEMENT

We used LLMs to improve wording, sentence structure, and grammar.

ETHICS STATEMENT

The StrCVIT benchmark is constructed entirely from established, publicly available academic vision-language datasets, all of which are named and cited in this paper. We create the data stream solely by rearranging existing samples and do not generate any new samples. We neither modify nor redistribute source images, and we provide links to their official sources. We obtain annotation files from the public releases of CoIN (Chen et al., 2024), SMoLoRA (Wang et al., 2025b), MLLM-CL (Zhao et al., 2025a), and ChartQA (Masry et al., 2022). The corresponding dataset cards list the licenses for these releases as CC BY 4.0, CC BY 4.0, Apache-2.0, and GPL-3.0, respectively. We will retain all required attribution, copyright, and license notices.

REPRODUCIBILITY STATEMENT

Section 4.1 describes the benchmark setup and metrics. Section 4.2 describes the compared methods. Appendix C provides the implementation details. Appendix B further describes the composition of the data chunks.

REFERENCES

Rahaf Aljundi, Klaas Kelchtermans, and Tinne Tuytelaars. Task-free continual learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 11254–11263, 2019.

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, et al. Qwen3-vl technical report. *arXiv preprint arXiv:2511.21631*, 2025.

Arslan Chaudhry, Marcus Rohrbach, Mohamed Elhoseiny, Thalaiyasingam Ajanthan, Puneet K Dokania, Philip HS Torr, and Marc'Aurelio Ranzato. On tiny episodic memories in continual learning. *arXiv preprint arXiv:1902.10486*, 2019.

Chang Che, Ziqi Wang, Pengwan Yang, Cheems Wang, Hui Ma, and Zenglin Shi. LoRA in LoRA: Towards parameter-efficient architecture expansion for continual visual instruction tuning. *Proceedings of the AAAI Conference on Artificial Intelligence*, 40(24):19978–19986, March 2026.

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

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009.

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.

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 *International Conference on Machine Learning*, pp. 19011–19033. PMLR, 2025.

Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. Making the v in vqa matter: Elevating the role of image understanding in visual question answering. In *CVPR*, 2017.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

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. *arXiv preprint arXiv:2106.09685*, 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 computer vision and pattern recognition conference*, pp. 19608–19617, 2025.

Drew A Hudson and Christopher D Manning. Gqa: A new dataset for real-world visual reasoning and compositional question answering. In *CVPR*, pp. 6700–6709, 2019.

James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, 114(13):3521–3526, 2017.

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

Solomon Kullback and Richard A Leibler. On information and sufficiency. *The annals of mathematical statistics*, 22(1):79–86, 1951.

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.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. In *NeurIPS*, pp. 34892–34916, 2023.

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

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: ACL 2022*, pp. 2263–2279, 2022.

Anand Mishra, Shashank Shekhar, Ajeet Kumar Singh, and Anirban Chakraborty. Ocr-vqa: Visual question answering by reading text in images. In *ICDAR*, pp. 947–952. IEEE, 2019.

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

Oleksii Sidorov, Ronghang Hu, Marcus Rohrbach, and Amanpreet Singh. Textcaps: a dataset for image captioning with reading comprehension. In *ECCV*, pp. 742–758. Springer, 2020.

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

Lingyun Song, Ziyao Chen, Kang Pan, Xiaolin Han, Xinbiao Gan, Yudai Pan, Xiaofan Sun, Xiaoqi Wang, and Xuequn Shang. Kss-moe: Knowledge space synergy framework in mixture of experts for continual visual instruction tuning. *Proceedings of the AAAI Conference on Artificial Intelligence*, 40(30):25536–25544, 2026.

Gemma Team, Aishwarya Kamath, Johan Ferret, Shreya Pathak, Nino Vieillard, Ramona Merhej, Sarah Perrin, Tatiana Matejovicova, et al. Gemma 3 technical report. *arXiv preprint arXiv:2503.19786*, 2025.

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

Leyuan Wang, Liuyu Xiang, Yujie Wei, Yiwei Ru, Yunlong Wang, and Zhaofeng He. Symmetric image-text tuning with entropy-guided fusion for online continual learning in non-stationary visual streams. *IEEE Transactions on Image Processing*, 35:3202–3213, 2026.

Weiyun Wang, Zhangwei Gao, Lixin Gu, Hengjun Pu, Long Cui, Xingguang Wei, Zhaoyang Liu, Linglin Jing, Shenglong Ye, Jie Shao, et al. Internvl3.5: Advancing open-source multimodal models in versatility, reasoning, and efficiency. *arXiv preprint arXiv:2508.18265*, 2025a.

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*, pp. 177–186, 2025b.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

Fei Ye and Adrian G Bors. Online task-free continual generative and discriminative learning via dynamic cluster memory. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 26202–26212, 2024.

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

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

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

Yuze Zhao, Jintao Huang, Jinghan Hu, Xingjun Wang, Yunlin Mao, Daoze Zhang, Zeyinzi Jiang, Zhikai Wu, Baole Ai, Ang Wang, et al. Swift: a scalable lightweight infrastructure for fine-tuning. *Proceedings of the AAAI Conference on Artificial Intelligence*, 39(28):29733–29735, 2025b.

Bolei Zhou, Agata Lapedriza, Aditya Khosla, Aude Oliva, and Antonio Torralba. Places: A 10 million image database for scene recognition. *IEEE transactions on pattern analysis and machine intelligence*, 40(6):1452–1464, 2017.

A LIMITATIONS

Due to the difficulty of collecting real-world MLLM data streams with naturally evolving and interleaved task distributions, we cannot directly obtain such streams at the required scale. Therefore, we use multiple existing vision-language datasets to construct sequential data chunks with varying task compositions and sample proportions to simulate the StrCVIT scenario. Although this benchmark approximates practical distribution shifts, it may not fully capture all temporal dependencies, noise patterns, and user-driven changes that occur in real-world deployment.

B DETAILED DATA STREAM COMPOSITION

B.1 DATASET STATISTICS

Table 4 summarizes the datasets used in StrCVIT. The benchmark covers eleven tasks, ranging from general capabilities such as grounding, captioning, classification, and VQA to domain-specific scenarios including autonomous driving, finance, remote sensing, OCR, and chart understanding.

Table 4: Statistics of the collected datasets in the StrCVIT benchmark.

Task	Train Dataset	Test Dataset	Test Number
Grounding	RefCOCO RefCOCO+ RefCOCOg	RefCOCO RefCOCO+ RefCOCOg	1000
Image Captioning	TextCaps	TextCaps	3166
Image Classification	ImageNet	ImageNet	1000
Scene Classification	Places365	Places365	1000
Visual Question Answering (VQA)	VQAv2	VQAv2	1000
Visual Reasoning VQA	GQA	GQA	1000
Autonomous Driving VQA (AD)	DriveLM	DriveLM	1000
Financial VQA (Fin)	StockQA	StockQA	1000
Remote Sensing VQA (RS)	RSVQA	RSVQA	1000
OCR VQA	OCRVQA	OCRVQA	1000
Chart VQA	ChartQA	ChartQA	1250

Following prior CVIT benchmark (Chen et al., 2024; Wang et al., 2025b; Zhao et al., 2025a), each textual instruction consists of two components: a task question that specifies the task to be performed on the image and a short output-format constraint that preserves instruction-following ability. As shown in Table 5, for datasets that provide an original question or referring expression, we retain it as the task description. Since heterogeneous multimodal tasks differ in their output requirements, these requirements are specified in the corresponding textual instructions (e.g., a single word or phrase for VQA, and bounding-box coordinates for grounding). To further evaluate robustness to instruction variations, we conduct an instruction robustness study in Appendix D.6.

B.2 PROXY TEST SETS

For ImageNet, Places365, AD, Fin, OCRVQA, Grounding, VQAv2, RS, and GQA, we construct proxy test sets by stratified sampling from the original test sets according to their annotation distributions, reducing evaluation cost while preserving the evaluation trends of the full test sets. To validate the reliability of these proxy sets, we evaluate LoRA on both the original and proxy test sets using InternVL3.5-8B-Pretrained. As shown in Fig. 6, the proxy and original test sets exhibit small mean accuracy differences, indicating that the proxy sets provide a faithful approximation for monitoring performance over the stream.

Table 5: Textual instruction templates for each dataset.

Dataset	Textual instruction
AD	{original question and object-encoding description} + Answer the question using a single word or phrase.
Grounding	Please provide the bounding box coordinate of the region this sentence describes: {original referring expression}.
RS	{original question} + Answer the question using a single word or phrase.
VQAv2	{original question} + Answer the question using a single word or phrase.
Fin	{original question} + Answer the question using a single word or letter directly.
ChartQA	{original question} + Answer the question using a single word or phrase based on the chart.
GQA	{original question} + Answer the question using a single word or phrase.
OCRVQA	{original question} + Answer the question using a single word or phrase.
Places365	{Question} + Answer the question using a single word or phrase.
ImageNet	{Question} + Answer the question using a single word or phrase.
TextCaps	{Question} + Please describe it in one complete sentence.

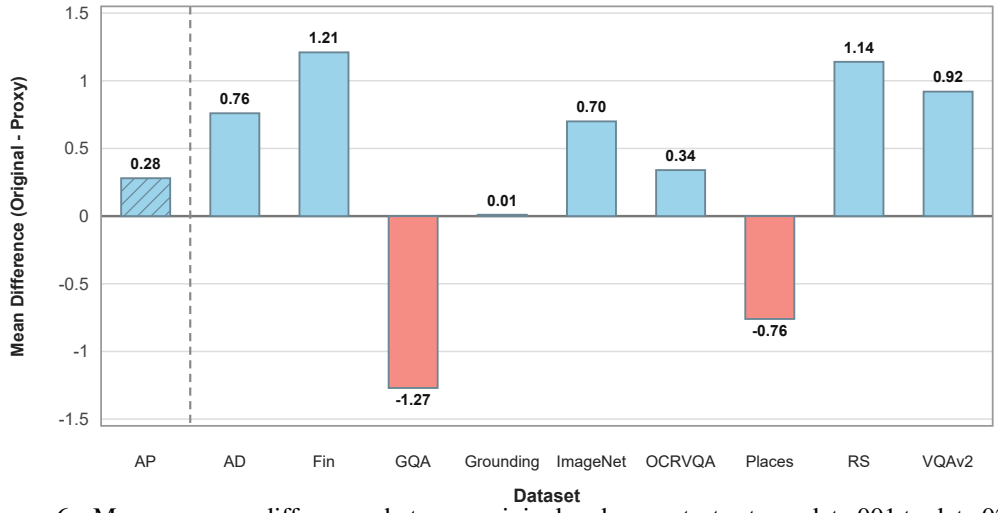


Figure 6: Mean accuracy differences between original and proxy test sets on data 001 to data 025.

B.3 Data Chunk Composition

Fig. 7 shows the composition of each data chunk in the data stream. Notably, Places365 appears only in the early stage and no longer appears after data 011, reflecting a realistic non-stationary task distribution where some tasks may disappear from the incoming stream.

C Implementation Details

C.1 Base Models

InternVL3.5-8B-Pretrained. InternVL3.5-8B-Pretrained is a multimodal InternVL model pretrained on large-scale vision-language pairs. The checkpoint used in our experiments is available at `https://huggingface.co/OpenGVLab/InternVL3_5-8B-Pretrained`.

InternVL3.5-4B-Pretrained. InternVL3.5-4B-Pretrained is a multimodal InternVL model pretrained on large-scale vision-language pairs. The checkpoint used in our experiments is available at `https://huggingface.co/OpenGVLab/InternVL3_5-4B-Pretrained`.

Gemma3-4B-PT. Gemma3-4B-PT is a multimodal Gemma 3 model pretrained on large-scale vision-language pairs. The checkpoint used in our experiments is available at `https://huggingface.co/google/gemma-3-4b-pt`.

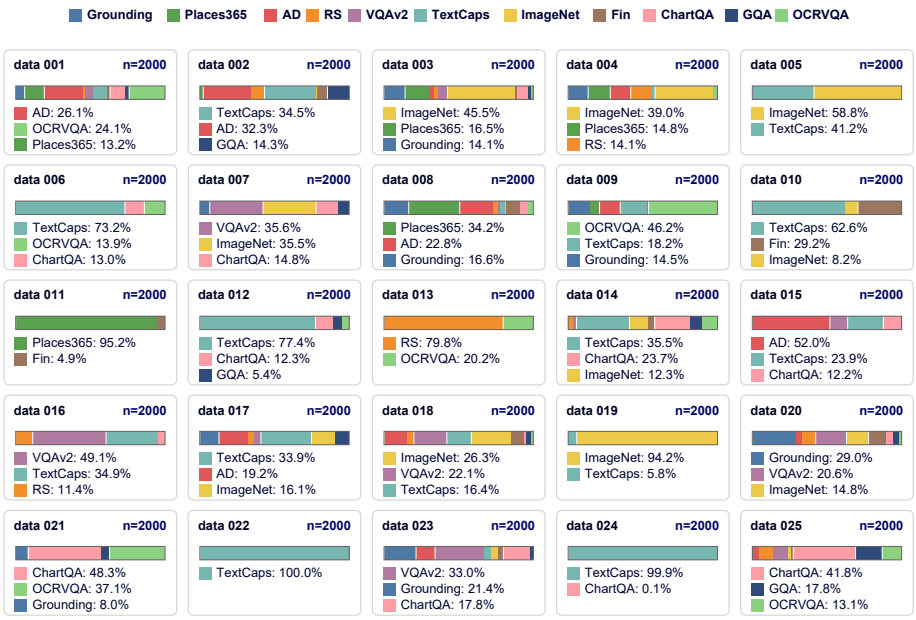


Figure 7: Composition of data chunks from data 001 to data 025. Each chunk contains 2,000 samples, and the stacked bars show the proportions of different datasets within each chunk.

LLaVA-v1.5-7B. LLaVA-v1.5-7B is a multimodal model trained by visual instruction tuning on top of a Vicuna language backbone and a CLIP vision encoder. The checkpoint used in our experiments is available at `https://huggingface.co/liuhaotian/llava-v1.5-7b`.

C.2 Training Setup

All experiments are conducted on a server with 2 Intel Xeon Platinum 8368 CPUs at 2.40GHz and 8 NVIDIA A100-SXM4-80GB GPUs. We implement training with PyTorch 2.9.1+cu128 and CUDA 12.8 under the ms-swift (Zhao et al., 2025b) 3.11.0.dev0 training framework. StrLoRA adapters are inserted into both the FeedForward Network layers and the self-attention layers of the LLM, while the projection layer between the vision encoder and the LLM is fully fine-tuned. We use the straight-through estimator to allow gradient flow through the discrete top-k selection. The hyperparameters for StrLoRA are configured as follows: the number of LoRA experts is set to 4, the Top-K selection is 2, and each LoRA expert has a rank of 16. In the routing-stability regularization loss $\mathcal{L}_{\text{reg}}$, the EMA momentum coefficient β and the loss weight λ are set to 0.99 and 0.1, respectively. We employ the Adam optimizer with a learning rate of 2×10^{-5} and a batch size of 32. For the Replay baseline, we maintain a long-term memory buffer with a capacity equal to 10% of one data chunk. When processing each new chunk, we replay samples from this buffer, which stores samples retained from previously observed chunks. For EWC, we use a regularization coefficient of $\lambda_{\text{EWC}} = 1.0$ and estimate the diagonal Fisher information from squared gradients of trainable parameters. All data chunks are trained for a single epoch.

D Additional Experiments

D.1 Evaluation on an Additional Data Stream

We conduct an experiment on a new 25-chunk data stream with different task proportions. Owing to differences in stream composition, the absolute performance of each method varies across the two streams. Nevertheless, as shown in Table 6, StrLoRA consistently achieves the highest MAP and the lowest MAF among the three methods on both streams, demonstrating its effectiveness under variations in stream composition.

Table 6: Results (%) on different data streams with different task proportions on InternVL3.5-8B.

Method	Places365		RS		AD		VQAv2		Fin		Grounding		OCRVQA		TextCaps		ImageNet		Overall	
	AP↑	AF↓	AP↑	AF↓	AP↑	AF↓	AP↑	AF↓	AP↑	AF↓	AP↑	AF↓	AP↑	AF↓	AP↑	AF↓	AP↑	AF↓	MAP↑	MAF↓
Stream 1																				
SMoLoRA	17.34	20.21	64.29	2.97	29.27	4.31	69.27	1.55	71.52	1.04	43.93	8.23	40.71	4.00	75.83	5.54	49.98	3.80	51.35	5.74
MoELoRA	26.14	21.27	69.10	1.62	33.94	4.51	**70.14**	**0.72**	74.23	0.84	55.30	6.90	**53.67**	**1.86**	84.41	**0.65**	74.78	3.81	60.19	4.69
StrLoRA	**33.37**	**2.99**	**69.65**	**0.75**	**36.27**	**3.92**	70.11	1.13	**75.95**	**0.38**	**57.36**	**2.53**	53.00	3.29	**84.48**	0.74	**76.49**	**1.46**	**61.85**	**1.91**
Stream 2																				
SMoLoRA	20.00	23.25	67.48	1.25	30.34	3.26	70.20	2.29	80.06	0.77	56.53	**2.93**	46.95	**3.32**	76.28	3.41	53.16	3.23	55.67	4.86
MoELoRA	30.85	17.66	68.51	1.24	33.35	**1.52**	69.70	2.10	81.11	0.59	56.97	6.53	47.29	5.24	78.08	1.64	59.16	2.73	58.34	4.36
StrLoRA	**33.60**	**3.27**	**69.45**	**0.85**	**36.68**	1.82	**70.58**	**1.16**	**81.42**	**0.49**	**57.96**	3.59	**48.67**	3.47	**81.61**	**0.85**	**62.04**	**1.43**	**60.23**	**1.88**

Table 7: Evaluation results (%) over 15 and 20 data chunks across different base models.

Method	Places365		RS		AD		VQAv2		Fin		Grounding		OCRVQA		TextCaps		ImageNet		Overall	
	AP↑	AF↓	AP↑	AF↓	AP↑	AF↓	AP↑	AF↓	AP↑	AF↓	AP↑	AF↓	AP↑	AF↓	AP↑	AF↓	AP↑	AF↓	MAP↑	MAF↓
InternVL3.5-8B-Pretrained 15 Chunks																				
LoRA	22.81	16.10	65.66	1.66	31.99	4.11	67.90	1.53	72.49	0.54	53.85	4.57	53.03	1.02	82.40	0.94	65.91	3.74	57.34	3.80
EWC	25.37	13.56	65.67	1.08	30.82	**1.82**	69.55	0.68	71.41	0.65	**56.09**	2.41	51.07	3.75	81.53	0.87	57.43	3.34	56.55	3.13
Replay	22.05	20.49	64.38	2.80	31.26	4.10	**69.97**	**0.66**	72.84	0.69	54.09	3.60	52.96	**0.85**	82.64	**0.54**	65.35	3.57	57.28	4.15
MoELoRA	27.70	12.72	65.83	2.20	33.90	3.81	69.53	0.72	72.54	0.75	52.17	7.51	51.94	2.42	82.88	0.63	67.95	4.29	58.27	3.90
SMoLoRA	17.44	6.25	60.25	3.08	28.08	3.36	68.53	1.54	68.92	1.05	36.38	11.43	37.36	2.82	71.11	7.91	41.14	4.16	47.69	4.62
Ours	**32.42**	**1.37**	**67.45**	**0.93**	**34.87**	4.20	69.35	1.40	**73.52**	**0.27**	54.78	**1.82**	**53.07**	1.46	**82.97**	0.63	**70.23**	**1.04**	**59.85**	**1.46**
InternVL3.5-8B-Pretrained 20 Chunks																				
LoRA	22.60	22.33	67.60	1.35	32.12	4.10	68.72	1.36	73.54	0.55	55.81	4.56	**53.93**	**1.57**	83.24	0.96	69.85	3.83	58.60	4.51
EWC	24.95	18.16	67.34	1.03	31.13	**1.63**	70.01	**0.66**	72.26	0.80	**57.88**	**2.56**	51.97	3.54	82.41	1.13	61.94	3.81	57.76	3.70
Replay	21.50	26.50	66.68	2.27	31.54	3.74	**70.18**	0.70	73.79	0.99	55.92	3.55	53.75	1.57	83.57	0.74	69.31	3.73	58.47	4.86
MoELoRA	26.72	18.07	67.90	1.77	34.12	3.71	69.93	0.67	73.52	0.91	53.88	7.51	53.06	1.87	83.78	0.64	71.46	4.54	59.37	4.41
SMoLoRA	17.73	13.76	62.71	3.10	28.99	3.37	69.06	1.45	70.17	1.27	40.25	9.41	39.88	2.60	74.12	6.31	44.84	4.62	49.75	5.10
Ours	**33.02**	**2.38**	**68.77**	**0.76**	**35.78**	3.87	69.79	1.25	**74.96**	**0.29**	56.07	2.61	53.11	2.46	**83.90**	**0.60**	**73.48**	**1.76**	**60.98**	**1.78**
InternVL3.5-4B-Pretrained 15 Chunks																				
LoRA	29.32	5.48	63.64	4.77	34.47	1.68	68.70	1.75	74.07	0.98	58.20	2.09	56.09	3.09	81.38	0.51	63.28	2.93	58.79	2.59
EWC	30.21	3.00	66.21	**2.32**	33.64	3.27	68.67	1.17	73.87	0.98	59.44	2.82	**57.80**	2.07	80.78	0.55	64.47	**2.45**	59.46	2.07
Replay	29.96	4.34	65.05	4.69	**34.62**	1.88	68.41	1.69	74.47	0.87	56.42	5.81	57.55	**1.09**	81.81	0.44	63.57	3.38	59.09	2.69
MoELoRA	30.04	4.92	65.56	2.89	34.45	2.19	68.73	0.89	73.18	0.80	58.97	2.23	57.26	2.82	80.91	0.87	**66.29**	2.55	59.49	2.24
SMoLoRA	25.33	5.81	60.33	7.17	30.14	3.70	**68.83**	0.80	73.06	0.68	51.99	2.10	55.56	2.44	79.43	**0.12**	65.17	3.62	56.65	2.94
Ours	**31.21**	**2.99**	**66.29**	2.56	33.61	**1.36**	68.58	**0.59**	**75.72**	**0.62**	**60.05**	**1.23**	57.09	1.42	**82.00**	0.33	65.39	5.33	**59.99**	**1.82**
InternVL3.5-4B-Pretrained 20 Chunks																				
LoRA	29.40	6.86	66.37	3.74	34.58	1.93	68.82	1.68	75.29	1.01	59.82	2.44	56.60	2.94	82.41	0.54	67.24	3.85	60.06	2.78
EWC	30.28	5.34	**68.41**	**1.93**	33.76	3.56	68.90	1.14	75.18	0.98	60.62	3.37	57.88	2.58	81.70	0.81	68.78	2.75	60.61	2.50
Replay	29.97	6.13	67.40	3.67	**34.93**	1.70	68.62	1.64	75.60	1.01	58.47	5.22	**57.90**	1.73	82.70	0.47	67.40	4.24	60.33	2.87
MoELoRA	29.70	8.34	67.71	2.37	34.58	2.18	68.84	0.90	74.57	0.77	60.37	2.79	57.81	3.26	81.77	1.12	**70.72**	**2.53**	60.67	2.70
SMoLoRA	25.89	9.05	63.75	5.57	31.45	3.26	**68.98**	1.09	74.29	0.72	54.16	2.26	55.97	3.00	80.71	**0.33**	69.22	3.92	58.27	3.24
Ours	**31.64**	**4.24**	67.91	2.30	34.07	**1.47**	68.80	**0.61**	**76.95**	**0.66**	**60.80**	**1.82**	57.58	**1.45**	**82.93**	0.38	70.38	4.30	**61.22**	**1.91**
Gemma3-4B-PT 15 Chunks																				
LoRA	**32.49**	**2.95**	55.55	6.44	19.62	24.46	53.53	2.58	62.11	3.91	4.23	**4.29**	52.92	6.00	66.48	4.31	54.17	**3.10**	44.57	6.45
EWC	32.01	3.78	55.75	6.34	19.27	25.14	53.57	2.91	61.78	4.30	4.21	5.08	53.17	5.56	66.24	4.39	52.48	6.29	44.28	7.09
Replay	31.73	4.23	55.44	7.94	19.56	18.14	**54.03**	2.26	64.67	2.12	3.11	34.68	55.45	3.58	67.23	**3.05**	54.42	3.14	45.07	8.79
MoELoRA	32.37	3.55	54.81	7.05	23.27	8.20	52.97	4.53	61.70	6.40	**4.47**	13.21	52.96	3.86	67.03	3.42	53.25	4.77	44.76	6.11
SMoLoRA	29.26	3.59	55.05	6.49	15.75	15.01	52.07	3.36	62.77	3.36	3.06	14.15	44.81	**3.36**	64.28	3.27	53.45	3.60	42.28	6.24
Ours	31.97	5.44	**58.11**	**4.87**	**33.10**	**6.00**	53.99	**0.96**	**70.73**	**0.71**	3.73	15.69	**56.46**	4.34	**68.69**	7.17	**67.37**	5.03	**49.35**	**5.58**
Gemma3-4B-PT 20 Chunks																				
LoRA	32.61	4.39	57.52	5.40	22.57	20.73	54.95	2.14	65.31	3.56	4.06	16.80	54.81	4.96	68.91	3.93	58.61	3.61	46.59	7.28
EWC	32.33	4.40	57.68	5.34	22.94	21.03	54.86	2.51	65.00	3.95	4.05	17.40	54.96	4.49	68.93	3.57	57.73	5.20	46.50	7.54
Replay	32.01	4.59	58.24	6.19	22.59	14.23	**55.21**	1.99	67.34	1.87	4.30	27.95	56.82	2.93	69.12	3.30	58.89	**3.52**	47.17	7.40
MoELoRA	**33.10**	**3.25**	56.86	5.99	26.85	7.56	54.35	3.49	64.86	5.29	4.04	26.01	54.78	3.39	**69.42**	3.05	58.65	3.90	46.99	6.88
SMoLoRA	29.50	6.29	57.19	5.33	19.99	13.58	53.55	2.87	65.62	3.20	2.88	26.17	48.10	**2.75**	67.44	**2.75**	57.56	4.33	44.65	7.47
Ours	32.27	6.07	**59.60**	**3.89**	**32.93**	**6.92**	54.65	**0.89**	**71.62**	**0.84**	**5.23**	**12.96**	**57.53**	4.01	68.47	8.22	**70.71**	4.98	**50.33**	**5.42**

D.2 Evaluation Under Different Stream Lengths

We further present evaluation results on the StrCVIT benchmark under the 15- and 20-chunk settings. As shown in Table 7, StrLoRA achieves consistently superior overall performance while maintaining low forgetting across all models.

D.3 Evaluation on Datasets with Potential Pretraining Overlap

ChartQA may overlap with the pretraining data of both InternVL3.5 and Gemma3-4B, and GQA may also overlap with the pretraining data of InternVL3.5, as suggested by the consistently high

zero-shot scores. To isolate this factor, we present additional evaluations on these two datasets separately. As shown in Table 8, StrLoRA still maintains stable performance across the data stream. Since these datasets already exhibit strong zero-shot performance, the results mainly reflect whether continual tuning can preserve pretrained capabilities while adapting to the stream. StrLoRA remains competitive in AP and maintains low forgetting.

Table 8: Additional evaluation on GQA and ChartQA across different base models.

Method	Gemma3-4B-PT				InternVL3.5-4B-Pretrained				InternVL3.5-8B-Pretrained			
	GQA		ChartQA		GQA		ChartQA		GQA		ChartQA	
	AP↑	AF↓	AP↑	AF↓	AP↑	AF↓	AP↑	AF↓	AP↑	AF↓	AP↑	AF↓
Zero-shot	4.00	–	32.80	–	57.50	–	54.32	–	60.40	–	**57.76**	–
LoRA	43.84	4.23	33.27	1.53	60.84	**0.92**	53.98	0.55	61.88	2.09	57.43	0.76
EWC	43.84	4.23	33.29	1.08	60.83	1.16	53.98	0.55	61.96	**0.80**	57.33	0.42
Replay	43.62	3.88	33.44	1.22	60.95	1.14	53.94	0.46	61.96	1.96	57.38	**0.39**
MoELoRA	43.92	4.86	32.62	1.55	60.89	1.43	54.27	0.69	61.78	1.38	57.50	0.43
SMoLoRA	42.67	3.93	32.70	1.23	60.81	1.93	53.79	0.68	61.16	1.73	56.24	0.97
Ours	**44.38**	**3.05**	**34.15**	**0.47**	**61.02**	1.47	**54.57**	**0.35**	**61.99**	2.02	57.61	0.51

D.4 Additional Ablation Studies

We further analyze the sensitivity of StrLoRA to its main hyperparameters, including the number of LoRA experts n, the number of selected experts k, the expert rank r, and the regularization weight λ. As shown in Table 9, the default setting $(n = 4, k = 2, r = 16, \lambda = 0.1)$ achieves the best overall balance, obtaining the highest MAP and the lowest MAF.

Table 9: Additional ablation study of StrLoRA hyperparameters on InternVL3.5-8B.

Hyperparameter				Places365		RS		AD		VQAv2		Fin		Grounding		OCRVQA		TextCaps		ImageNet		Overall	
n	k	r	λ	AP↑	AF↓	AP↑	AF↓	AP↑	AF↓	AP↑	AF↓	AP↑	AF↓	AP↑	AF↓	AP↑	AF↓	AP↑	AF↓	AP↑	AF↓	MAP↑	MAF↓
4	2	16	0.1	33.37	2.99	69.65	0.75	36.27	3.92	70.11	1.13	75.95	0.38	57.36	2.53	53.00	3.29	84.48	0.74	76.49	1.46	61.85	1.91
4	2	16	0.01	28.28	6.43	67.77	2.80	35.20	1.51	70.38	0.52	75.99	0.73	58.11	3.67	56.19	4.18	84.27	0.86	74.43	2.95	61.18	2.63
4	2	8	0.1	26.28	11.01	67.17	2.55	31.47	2.51	70.14	0.87	72.42	0.53	55.90	4.05	57.38	4.78	83.44	1.02	66.28	4.81	58.94	3.57
6	2	16	0.1	35.41	5.84	68.06	0.82	34.96	2.24	70.56	1.05	76.32	0.56	53.81	2.93	53.58	3.01	81.78	3.56	77.22	2.39	61.30	2.49
6	3	16	0.1	33.11	6.21	67.79	0.82	34.96	2.24	70.36	1.05	75.66	0.57	55.97	2.83	57.12	2.83	82.37	3.54	74.82	2.52	61.35	2.51

To further verify the contributions of $W_Q h_i^l$ and $W_K x_i^{\text{text}}$ in the token-wise expert weighting, we remove each term individually while keeping all other components unchanged and evaluate the resulting accuracy. As shown in Table 10, removing either component degrades the average accuracy. We evaluate the model on the final chunk after training on the entire data stream.

Table 10: Ablation study of the terms in token-wise expert weighting on InternVL3.5-8B.

Setting	Places365	RS	AD	VQAv2	Fin	Grounding	OCRVQA	TextCaps	ImageNet	Average
w/o $W_K x_i^{\text{text}}$	30.50	71.50	33.30	70.30	79.70	59.50	47.90	85.42	83.70	62.42
w/o $W_Q h_i^l$	32.50	73.90	35.00	70.80	79.20	60.50	51.70	84.99	86.10	63.85
Full Design	34.00	73.20	37.40	71.20	80.10	61.80	53.70	86.05	88.40	65.09

To disentangle the source of the reduced forgetting, we compare adjacent training checkpoints with and without EMA regularization. For each pair, we measure the performance change and compute the JS divergence of the routing distributions, the consistency of expert selection, and the magnitude of parameter updates, which quantify router drift, expert specialization, and plasticity, respectively. As shown in Table 11, EMA regularization alleviates forgetting on ImageNet while achieving higher accuracy on the remaining tasks. It also yields more consistent routing, reflected in lower router drift and higher expert consistency, together with larger parameter updates to both the router and the LoRA modules. These results indicate that EMA regularization alleviates forgetting by stabilizing the routing distribution rather than by restricting the model's plasticity.

D.5 Computational Efficiency

The Overhead of the Token-Wise Router. StrLoRA introduces two additional trainable routing components: the task-aware router and the token-wise router. The EMA router acts only as a regularizer during training, and it contributes no trainable parameters and incurs no computation at inference. The number of parameters is summarized in Table 12. As shown in Table 13,

Table 11: Ablation of EMA regularization on forgetting on InternVL3.5-8B.

Setting	ImageNet AP	MAP Change on Other Tasks	Router Drift	Expert Consistency	Router Update Magnitude	LoRA Update Magnitude
Without EMA Regularization	27.70 → 19.60	+3.59	0.0697	90.48%	0.976%	2.753%
With EMA Regularization	27.70 → **26.90**	**+3.80**	0.0387	93.65%	1.900%	7.437%

although the token-wise router introduces slight additional computational overhead by computing routing weights for each token, it consistently achieves better performance and substantially reduces forgetting compared with task-level routing.

Table 12: The number of parameters in the StrLoRA components on InternVL3.5-8B.

Stage	LoRA Experts	Task-Aware Router	Token-Wise Router	Full Projector
Trainable Parameters	174.588M	4.129M	37.765M	33.563M
Inference Active Parameters	87.294M	4.129M	37.765M	33.563M

Table 13: Training and inference cost and performance of StrLoRA on InternVL3.5-8B.

Setting	Training Time (s/step) ↓	Inference Time (s/sample) ↓	MAP ↑	MAF ↓
Task-aware	**35.67**	**7.81**	60.77	3.54
Task-aware + Token-wise	37.46	9.83	61.58	2.97
Full Design	41.11	9.83	**61.85**	**1.91**

Computational Efficiency Comparison. We present an efficiency comparison in Table 14. StrLoRA introduces additional routing modules and therefore has more total trainable parameters. However, compared with the soft routing in MoELoRA, it activates fewer parameters per sample. Compared with SMoLoRA, which has nearly identical training and inference costs, StrLoRA achieves better MAP and MAF. Overall, StrLoRA provides more stable performance with modest additional computational overhead.

D.6 Evaluation on Instruction Robustness

We vary each dataset's instruction style while preserving the original semantics. For each sample, we construct the same set of five unseen instructions with different styles. We evaluate SMoLoRA, MoELoRA, and StrLoRA under these instructions; for StrLoRA, we additionally compare the selected experts with those obtained using the original instruction. As shown in Tables 15 and 16, StrLoRA maintains high routing consistency and achieves the highest average accuracy among the compared methods under the same five unseen instruction styles.

We further examine tasks whose textual instructions share a nearly identical instruction style but differ in semantics: ImageNet ("What is the object in the image? Answer the question using a single word or phrase.") and Places365 ("What is the background in the image? Answer the question using a single word or phrase."). As shown in Fig. 5, the two tasks activate different subsets of experts and yield different expert proportions across layers. This indicates that StrLoRA can distinguish tasks even when their instruction styles are highly similar.

E Case Study

We track the same validation samples across the streaming sequence. As shown in Fig. 8, StrLoRA keeps the correct prediction from $t = 1$ to $t = 25$ on diverse tasks, while MoELoRA and SMoLoRA often predict the sample correctly in early chunks but produce incorrect responses after later stream updates. These examples show that forgetting is not only reflected in aggregate accuracy drops, but also appears as concrete answer drift on individual samples.

Table 14: Computational efficiency and performance comparison on InternVL3.5-8B.

Method	Trainable Parameters ↓	Active Parameters ↓	Training Time (s/step) ↓	Inference Time (s/sample) ↓	MAP ↑	MAF ↓
MoELoRA	**213.46M**	213.46M	**39.24**	**7.62**	60.19	4.69
SMoLoRA	213.80M	**126.51M**	39.74	9.30	51.35	5.74
StrLoRA	250.04M	162.75M	41.11	9.83	**61.85**	**1.91**

Table 15: Accuracy (%) evaluated using the five unseen instructions.

Method	AD	Fin	Grounding	ImageNet	OCRVQA	Places365	RS	TextCaps	VQAv2	Average
SMoLoRA	28.00	72.00	**65.00**	41.00	50.00	18.00	70.00	73.65	65.00	53.63
MoELoRA	28.00	78.00	60.00	55.00	**56.00**	16.00	68.00	77.36	69.00	56.37
StrLoRA	**35.80**	**78.60**	59.30	**73.20**	51.50	**28.40**	**70.90**	**78.38**	**69.10**	**60.58**

Table 16: Routing consistency of StrLoRA under unseen instruction styles.

Metric	AD	Fin	Grounding	ImageNet	OCRVQA	Places365	RS	TextCaps	VQAv2	Average
Top-1 Consistency (%)	100.00	95.43	89.57	98.64	97.71	90.14	92.43	89.35	90.71	93.78
Top-2 Consistency (%)	94.12	92.43	88.71	89.57	93.29	89.86	86.86	82.71	87.57	89.46

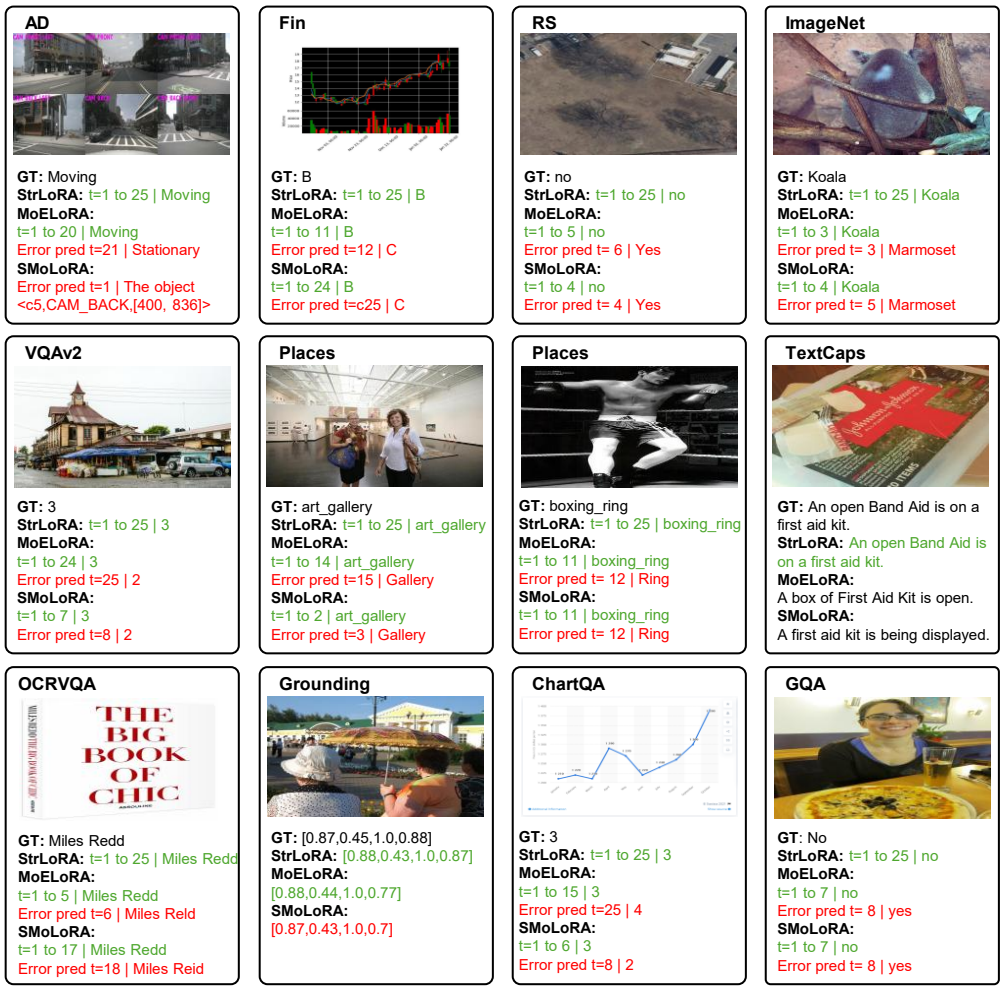


Figure 8: Qualitative temporal forgetting examples across the streaming process. Each panel tracks one fixed validation sample. Green text indicates correct predictions, and red text indicates the first observed erroneous prediction.