COVE: CONCEPT VOCABULARY EVOLUTION FOR STREAMING CONTINUAL VISUAL INSTRUCTION TUNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Continual Visual Instruction Tuning (CVIT) aims to enable multimodal large language models to continuously acquire new vision-language knowledge while retaining previously learned knowledge. Recent studies move toward more realistic streaming settings, where heterogeneous data arrive as interleaved and dynamically evolving distributions. In such streams, both the reuse of accumulated knowledge and the demand for additional adaptation capacity vary over time, making predefined expert structures difficult to match to the current adaptation demand. To address this challenge, we propose **Concept Vocabulary Evolution (COVE)**, which organizes continual adaptation as an evolving vocabulary of reusable units discovered directly from incoming data. Specifically, COVE decomposes each incoming sample over the accumulated vocabulary, selectively refining reusable historical units when they sufficiently represent the sample, while reconstruction residuals drive the discovery of additional units when existing capacity becomes insufficient. Extensive experiments demonstrate that COVE improves continual performance and substantially reduces catastrophic forgetting, while further analyses show that the learned vocabulary dynamically evolves with the changing demands of the stream.

1 INTRODUCTION

Continual Visual Instruction Tuning (CVIT) aims to enable multimodal large language models (MLLMs) to continuously acquire new vision-language knowledge while retaining previously learned knowledge. Early CVIT methodsChen et al. (2024); Wang et al. (2025); Che et al. (2026b) often assume that data are organized into clearly separated tasks, while more recent task-free continual learning settings(Aljundi et al., 2019; Moon et al., 2023) remove the need for explicit task identities and task boundaries. Moving further toward realistic deployment, Streaming Continual Visual Instruction Tuning (StrCVIT)Che et al. (2026a) considers data arriving as a sequence of chunks, where each chunk may contain multiple interleaved task distributions and the composition of these distributions can vary over time.

Existing task-free approaches mainly address this setting through modular adaptation and dynamic expansion. Mixture-of-Experts (MoE) based(Chen et al., 2024; Wang et al., 2025) methods organize adaptation capacity by routing incoming data into multiple suitable experts without task identities. Dynamic expansion methods(Ge et al., 2025; Zhang et al., 2026; Tang et al., 2026; Zhang et al., 2025) further allow additional experts when the current model capacity becomes insufficient. These approaches improve the flexibility of continual adaptation without relying on explicit task identities.

However, Streaming CVIT introduces an additional challenge beyond handling heterogeneous data within each chunk. The task distributions are interleaved in each chunk, as well as their relative proportions vary across the stream dynamically. Consequently, the knowledge required by each incoming chunk also changed over time. Both the extent to which incoming data can rely on previously learned knowledge, and the need for additional adaptation capacity may vary substantially across chunks. This creates a mismatch between the unpredicted adaptation demand of the data stream and the predefined structure used by existing modular methods, where a fixed expert pool only redistribute samples among existing experts and predefined expansion increase capacity at

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 fixed granularity. This raises a key challenge: how to automatically discover adaptation units from streaming data while leverage the growing accumulated knowledge to evolve them continually across data stream.

Based on this observation, we argue that continual adaptation should be guided by the relationship between incoming data and accumulated adaptation knowledge. If an incoming sample can be sufficiently represented by existing adaptation units, the relevant historical knowledge should be reused and selectively refined. If the existing units are insufficient, the remaining unexplained information should indicate the need for additional capacity. More importantly, the need of new capacity should also be determined from the current data rather than fixed in advance. In this way, the model can adjust both which adaptation units are used and how many units are maintained, allowing its adaptation structure suitable for the interleaved and dynamic streaming data.

To address this challenge, we propose Concept Vocabulary Evolution (COVE), which organizes continual adaptation as an evolving vocabulary of reusable units. Instead of prescribing a fixed adaptation structure, COVE first discovers adaptation units directly from the incoming data, allowing both the content and size of the vocabulary to be determined by the current adaptation demand. As new chunks arrive, COVE evaluates how the incoming data relate to the accumulated vocabulary, where knowledge that can be represented by existing units is reused and selectively refined, while insufficiently represented knowledge drives the discovery of additional units. Importantly, the number of newly introduced units is also determined from the current data rather than fixed in advance. In this way, COVE continually adjusts both the composition and the capacity of the adaptation vocabulary to the changing requirements of the stream. Extension experiments on streaming continual visual instruction tuning setting shows that our proposed method COVE achieves the state-of-the-art (SOTA) performance among existing methods and significantly mitigates catastrophic forgetting, with further analysis revealing how the vocabulary evolves throughout the dynamic data stream. Our contributions are summarized as follows:

- We identify a key challenge in streaming CVIT: as the composition of incoming data changes across chunks, the demand of reusable knowledge and newly required adaptation capacity also varies, making predefined expert structures difficult to match to the current demand.
- We propose Concept Vocabulary Evolution (COVE), an evolving concept vocabulary framework that determines both the reuse of historical units and the introduction of new units from incoming data. COVE selectively refines reusable units and dynamically adjusts vocabulary size according to dynamical demands across data chunks.
- Extensive experiments on streaming CVIT benchmarks show that COVE improves overall continual performance while substantially reducing forgetting. Further analyses show that how the learned concept vocabulary dynamically evolves to fit the vary demand throughout the data stream.

2 Related Work

Continual Visual Instruction Tuning Continual Visual Instruction Tuning (CVIT) studies how multimodal large language models continually acquire new vision-language knowledge while retaining previously learned knowledge. Early studies(He et al., 2026; Chen et al., 2024; Wang et al., 2025) mainly formulate CVIT under a task-wise sequential protocol, where different tasks are presented to the model in separate stages. These settings generally assume a clear task-level organization, with each stage dominated by a predefined task. However, such clearly separated task stages are difficult to guarantee in realistic continual learning scenarios. Therefore, task-free continual learning(Aljundi et al., 2019; Ye & Bors, 2024; Moon et al., 2023; Wang et al., 2026) relaxes the assumption of explicit task boundaries and task identities, allowing models to learn directly from non-stationary data streams. Building on this direction, StrLoRA(Che et al., 2026a) introduces Streaming Continual Visual Instruction Tuning (StrCVIT) for MLLMs, where data arrive as a single-pass sequence of chunks and each chunk may contain a dynamically changing mixture of interleaved tasks. This setting change CVIT from task-organized sequential learning toward heterogeneous streaming adaptation. We consider this more realistic setting in our work.

Modular Adaptation and Dynamic Expansion To mitigate interference between sequentially acquired knowledge, recent CVIT methods(Chen et al., 2024; Huai et al., 2025; Wang et al., 2025; Che et al., 2026a) increasingly organize adaptation into multiple specialized modules. These approaches typically maintain a pool of parameter-efficient experts and selectively route incoming data to suitable modules. As continual learning scenarios evolve over time, fixed modular structures may constrained the model's flexibility for emerging adaptation demands. Therefore, dynamic expansion methods allow model expand capcity according to the incoming data, where VariGrow(Ardywibowo et al., 2022) introduces new experts when detecting distributional novelty, ProgLoRA(Yu et al., 2025) progressively expands a LoRA pool, and GoD-MoE(Zhang et al., 2026) determines whether additional experts are required based on existing model capabilities. Despite increasingly adaptive expansion strategies, the form and granularity of the expand modules are generally predefined. Instead, our COVE focuses on discovering adaptation modular from incoming data and evolving them along with the stream dynamically.

3 Streaming Continual Visual Instruction Tuning

Traditional continual visual instruction tuning (CVIT) typically assumes a task-wise learning protocol with explicit task boundaries, whose standard formulation is provided in Appendix B. In this work, we focus on the more realistic **Streaming Continual Visual Instruction Tuning (StrCVIT)** setting(Che et al., 2026a), where data arrive as a single-pass stream without explicit task identities or boundary information. Formally, the stream is represented as a sequence of data chunks $\mathcal{B} = \{\mathcal{B}_1, \mathcal{B}_2, \ldots, \mathcal{B}_n\}$, where each chunk $\mathcal{B}_t = \{(v_i^t, x_i^t, y_i^t)\}_{i=1}^{N_t}$ contains N_t visual-instruction samples and may comprise multiple interleaved task distributions. The data distribution at time step t can therefore be expressed as

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

where M_t denotes the number of task distributions encountered up to time t, and $\pi_t^{(m)} \in [0, 1]$ is the corresponding mixture weight satisfying $\sum_{m=1}^{M_t} \pi_t^{(m)} = 1$. Both the participating distributions and their proportions may vary across chunks, resulting in an interleaved and dynamically evolving stream. At each step, the model adapts using only the current chunk and its retained model state, without access to ground-truth task identities or replaying samples from previous chunks, and the updated state is carried forward for continual adaptation throughout the stream.

4 Proposed Method

Fig. 1 illustrate the overall framework of Continual Vocabulary Evolution (COVE).

4.1 Framework Overview

Our proposed **Concept Vocabulary Evolution (COVE)** organizes continual adaptation as an evolving vocabulary of reusable adaptation units, allowing the adaptation structure to change together with the incoming data stream. Specifically, COVE maintains a dictionary $D_t = [d_1, \ldots, d_{K_t}]$, where each atom d_j represents an adaptation pattern discovered from streaming data. Each dictionary atom is associated with a LoRA expert E_j, which provides the corresponding parameter-efficient adaptation. Together, these atom–expert pairs form the concept vocabulary:

$$\mathcal{V}_t = \{(d_j, E_j)\}_{j=1}^{K_t}, \tag{2}$$

where K_t denotes the vocabulary size at time step t.

As illustrated in Fig. 1, given an incoming data chunk $\mathcal{B}_t$ and the accumulated vocabulary $\mathcal{V}_{t-1}$, COVE first decomposes each sample feature over the historical dictionary D_{t-1}. The resulting sparse coefficients characterize which historical units can be reused for the current sample, while the reconstruction residual measures the information that cannot be sufficiently represented by the accumulated vocabulary. Based on these two signals, COVE evolves the vocabulary in a data-driven manner. Samples that can be adequately represented by existing units are used to selectively refine the corresponding historical experts, whereas samples with large residuals trigger the discovery of

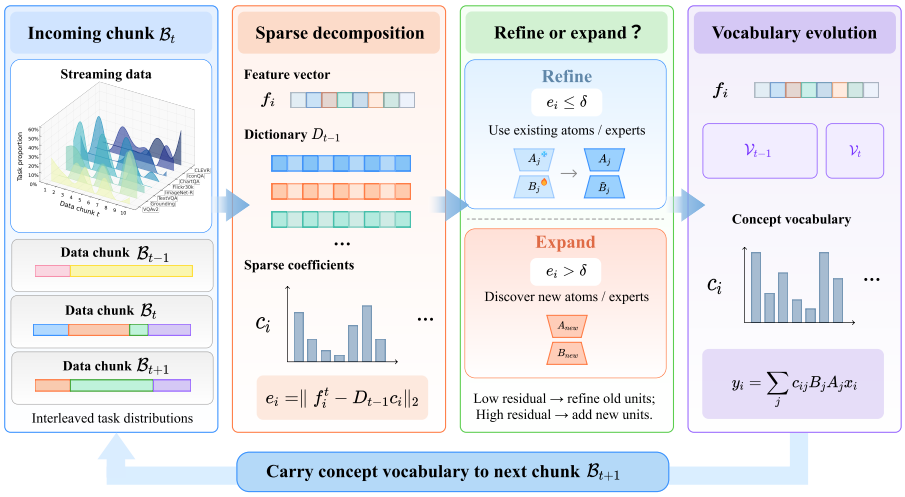


Figure 1: **Overview of Concept Vocabulary Evolution (COVE).** For each incoming data chunk, COVE sparsely decomposes sample features over the accumulated concept vocabulary. The reconstruction residual guides selective refinement of reusable historical units or data-driven expansion when existing units are insufficient. The evolved vocabulary is then carried forward for continual reuse and further evolution.

new dictionary atoms and their associated experts. Importantly, both the composition and the size of the vocabulary are determined from the incoming data rather than predefined in advance. The evolved vocabulary $\mathcal{V}_t$ is then carried forward to the next data chunk, allowing newly discovered units to become part of the accumulated knowledge and to be reused, refined, or further expanded as the stream evolves.

4.2 Vocabulary Construction and Expansion

We construct the initial vocabulary from the first data chunk $\mathcal{B}_1$. For each sample, we extract a question-conditioned multimodal feature $f_i^t = \phi_{\theta_0}(v_i^t, x_i^t) \in \mathbb{R}^d$, where ϕ_{θ_0} denotes the feature extractor of the frozen pretrained MLLM, with the detailed feature construction provided in Appendix H.2. Given the feature matrix $F_1 = [f_1^1, \ldots, f_{N_1}^1]$, we learn a candidate dictionary $\widetilde{D}_1$ through sparse dictionary learning:

$$\min_{\widetilde{D}_1, C_1} \frac{1}{2} \left\| F_1 - \widetilde{D}_1 C_1 \right\|_F^2 + \lambda_1 \left\| C_1 \right\|_1 + \lambda_g \sum_{j=1}^{K_1^{cand}} \left\| C_{1,j:} \right\|_2, \quad \text{s.t.} \left\| \tilde{d}_j \right\|_2 = 1. \tag{3}$$

The reconstruction term encourages the candidate atoms to represent the incoming data, while λ_1 and λ_g control sample-level sparsity and atom-level selection, respectively. We retain candidate atoms with nonzero coefficient and instantiate a corresponding LoRA expert for each retained atom, forming the initial vocabulary $\mathcal{V}_1$.

For each subsequent data chunk $\mathcal{B}_t$ $(t > 1)$, COVE first evaluates how well the historical vocabulary $\mathcal{V}_{t-1}$ can be used to represent the incoming samples. Specifically, we obtain the sparse coefficients of each sample over the historical dictionary D_{t-1} by solving this objective below:

$$c_i^{\text{old}} = \arg\min_c \frac{1}{2} \left\| f_i^t - D_{t-1} c \right\|_2^2 + \lambda_1 \|c\|_1. \tag{4}$$

The resulting coefficients characterize the contribution of historical atoms to reconstruct each sample. Despite it can identify reusable units, it does not determine whether the existing vocabulary sufficiently represents the incoming data. We therefore measure the unexplained component through the reconstruction residual:

$$r_i = f_i^t - D_{t-1} c_i^{\text{old}}, \qquad e_i = \|r_i\|_2. \tag{5}$$

Here, e_j quantifies the extent of how much the historical dictionary cannot explain the incoming sample, providing a signal for discovering new adaptation units.

Based on this reconstruction residual, we identify whether the samples require additional adaptation units. Specifically, we introduce a threshold δ and define a reuse indicator $m_i = \mathbb{I}(e_i \leq \delta)$ to partition the incoming samples into a reuse set and a candidate discovery set:

$$\mathcal{R}_t = \{i : m_i = 1\}, \qquad \mathcal{N}_t = \{i : m_i = 0\}. \tag{6}$$

Rather than expanding the vocabulary for each individual sample, we jointly handle the insufficiently represented samples to discover potentially reusable structures, reducing the influence of sample-specific noise. Given the selected sample features $F_t^{\text{nov}} = [f_i^t]_{i \in \mathcal{N}_t}$, we learn candidate dictionary $\widetilde{D}_t$ while preserving the historical dictionary D_{t-1}. Specifically, we construct an expanded candidate dictionary $D_t^{\text{cand}} = [D_{t-1}, \widetilde{D}_t]$, where $\widetilde{D}_t$ contains K_t^{cand} candidate atoms. We formulate the joint learning objective as follows:

$$\min_{\widetilde{D}_t,\, C_t} \ \frac{1}{2} \left\| F_t^{\text{nov}} - D_t^{cand} C_t \right\|_F^2 + \lambda_1 \left\| C_t \right\|_1 + \lambda_g \sum_{j=1}^{k_t^{cand}} \left\| \widetilde{C}_{t,j:} \right\|_2, \quad \text{s.t. } D_{t-1}^\top \widetilde{D}_t = 0, \quad \left\| \tilde{d}_j \right\|_2 = 1. \tag{7}$$

Here, $\widetilde{C}_t$ denotes the coefficient associated with the candidate atoms. The reconstruction term allows historical and candidate atoms to jointly represent the selected samples, while the orthogonality constraint aims to reduce overlap with previously learned atoms. The group sparsity term encourages atom-level selection, allowing redundant candidates to be discarded. We retain candidate atoms with nonzero coefficient and associate each retained atom with a newly instantiated LoRA expert, thereby expanding the vocabulary to $\mathcal{V}_t$. If no candidate atoms are retained, the vocabulary retain unchanged.

4.3 Expert Composition and Adaptation

Following the structural update of the vocabulary, we recompute the sparse coefficients of each incoming sample over the updated dictionary D_t:

$$c_i = \arg\min_c \frac{1}{2} \left\| f_i^t - D_t c \right\|_2^2 + \lambda_1 \|c\|_1. \tag{8}$$

The resulting sparse coefficients determine the sample-specific composition of adaptation units. And each dictionary atom is associated with a cross-layer LoRA expert $E_j = \{A_j^{(\ell)}, B_j^{(\ell)}\}_{\ell=1}^{L}$. Using the same coefficient c_{ij} across adapted layers, the sample-specific computation at layer ℓ is given by

$$y_i^{(\ell)} = W_0^{(\ell)} x_i^{(\ell)} + b^{(\ell)} + s \sum_{j=1}^{K_t} c_{ij} B_j^{(\ell)} A_j^{(\ell)} x_i^{(\ell)}, \tag{9}$$

where s denotes the LoRA scaling factor. In this way, each incoming sample combines historical and newly introduced adaptation units according to its sparse coefficients.

Having established the sample-specific composition, we next consider how to adapt these experts using incoming data. We still apply the standard supervised fine-tuning (SFT) objective to optimize the model:

$$\mathcal{L}_{\text{SFT}} = -\sum_i \log p_\theta \left(y_i^t \mid v_i^t, x_i^t \right). \tag{10}$$

However, COVE adopts different adaptation strategies for newly introduced and historical experts, because the reuse and novel samples identified in Section 4.2 provide different signals for expert adaptation. Novel samples are insufficiently represented by the historical vocabulary and therefore guide the learning of newly introduced experts. In contrast, reuse samples can be represented by existing units and are used to further refine historical experts. Therefore, for each newly introduced expert $j > K_{t-1}$, both LoRA matrices $A_j^{(\ell)}$ and $B_j^{(\ell)}$ are trained exclusively on samples from $\mathcal{N}_t$. Accordingly, for historical experts $j \leq K_{t-1}$, we freeze $A_j^{(\ell)}$ to preserve its learned input projection and selectively update $B_j^{(\ell)}$ using samples from the reuse set $\mathcal{R}_t$.

Table 1: Comparison of continual learning methods on streaming benchmarks with InternVL3.5-4B-Pretrained and InternVL3.5-8B-Pretrained as the base models. Higher AP/MAP/Final AP (↑) and lower AF/MAF (↓) indicate better performance.

Method	VQAv2		Grounding		TextVQA		ImageNet-R		Flickr30k		ChartQA		IconQA		CLEVR		Overall		Final AP↑
	AP↑	AF↓	AP↑	AF↓	AP↑	AF↓	AP↑	AF↓	AP↑	AF↓	AP↑	AF↓	AP↑	AF↓	AP↑	AF↓	MAP↑	MAF↓	
InternVL3.5-4B-Pretrained																			
Zero-shot	69.00	–	52.60	–	20.13	–	25.80	–	6.29	–	53.60	–	64.40	–	85.80	–	47.20	–	–
Joint LoRA	70.20	–	64.00	–	74.87	–	37.00	–	7.27	–	85.20	–	70.80	–	84.60	–	61.74	–	–
Sequential-LoRA	69.92	2.07	59.68	2.28	74.33	1.63	35.20	6.67	6.87	4.12	83.66	0.89	64.94	3.07	77.48	17.02	59.01	4.72	59.66
MoELoRA	69.30	1.86	57.10	3.05	70.72	1.12	36.54	1.36	6.84	2.70	81.14	0.07	68.20	2.29	85.74	5.70	59.45	2.27	61.94
VariGrow	69.00	3.25	63.68	4.08	76.93	0.91	36.74	6.81	5.86	16.36	**83.84**	0.99	71.78	2.01	69.74	22.51	59.70	7.12	61.96
SMoLoRA	70.00	1.79	52.12	7.95	70.67	0.66	35.64	2.98	6.59	2.86	75.88	1.66	68.52	1.72	78.00	13.38	57.18	4.12	61.97
StrLoRA	**71.26**	0.70	64.34	0.64	74.17	0.28	37.08	6.89	7.16	5.55	79.42	0.21	**74.88**	**0.26**	88.88	4.61	62.15	2.39	63.94
COVE (Ours)	71.18	**0.36**	**64.76**	**0.09**	**76.85**	**0.11**	**41.46**	**1.26**	**7.33**	**1.84**	83.14	**0.72**	74.70	0.31	**91.52**	**0.55**	**63.87**	**0.66**	**65.73**
InternVL3.5-8B-Pretrained																			
Zero-shot	69.40	–	34.60	–	13.00	–	29.40	–	6.72	–	65.80	–	65.20	–	94.00	–	47.27	–	–
Joint LoRA	71.20	–	**64.80**	–	77.87	–	**43.00**	–	7.02	–	85.20	–	75.60	–	84.60	–	63.66	–	–
Sequential-LoRA	70.40	2.19	57.94	3.08	78.00	1.96	37.12	7.06	7.26	4.77	**86.66**	0.88	66.39	4.02	80.16	18.02	60.49	5.25	61.07
MoELoRA	70.54	1.67	56.29	2.61	75.30	1.20	38.30	**1.78**	7.30	2.70	85.08	**0.05**	70.56	2.26	88.54	6.76	61.49	2.38	63.70
VariGrow	69.83	3.22	61.44	4.13	80.58	1.00	38.08	7.31	6.16	15.82	87.42	0.66	74.27	2.14	72.90	22.56	61.34	7.10	63.34
SMoLoRA	70.83	1.76	49.88	8.00	74.32	0.75	36.98	3.48	6.90	2.32	79.46	1.33	71.01	1.85	81.16	13.43	58.82	4.12	63.07
StrLoRA	71.82	0.72	61.69	0.50	77.04	0.37	39.14	6.64	7.51	5.36	84.18	0.08	76.95	**0.49**	92.28	4.74	63.83	2.36	65.42
COVE (Ours)	**71.96**	**0.47**	58.48	**0.27**	**80.85**	**0.33**	41.80	2.59	**7.56**	**0.96**	84.96	0.16	**77.08**	0.65	**95.12**	**1.08**	**64.73**	**0.81**	**66.38**

Since multiple historical experts may contribute to the same reuse sample, we further control their adaptation strengths according to their relative contributions in the sparse composition. Specifically, we define the responsibility of expert j for sample i as:

$$\rho_{ij} = \frac{|c_{ij}|}{\sum_{k=1}^{K_t} |c_{ik}| + \epsilon}, \tag{11}$$

where c_{ij} denotes the sparse coefficient in Eq. 8. We then use this responsibility to modulate the updates of historical experts. For each $j \leq K_{t-1}$, the gradients of historical experts are weighted as

$$\nabla_{B_j^{(\ell)}} \mathcal{L}_i \leftarrow \rho_{ij} \nabla_{B_j^{(\ell)}} \mathcal{L}_i, \quad \nabla_{A_j^{(\ell)}} \mathcal{L}_i = 0, \qquad \text{where} \quad i \in \mathcal{R}_t, \tag{12}$$

where $\mathcal{L}_i$ denotes the SFT loss for sample i. In this way, historical experts are selectively refined according to their responsibilities for reusable samples.

After adaptation, the vocabulary incorporates newly learned experts and selectively refined historical experts. The evolved vocabulary $\mathcal{V}_t$ is then carried forward to the next data chunk, where newly introduced units become part of the historical vocabulary and can be reused, refined, or further expanded upon as the data stream evolves. The algorithm of our method is shown in the Algorithm 1

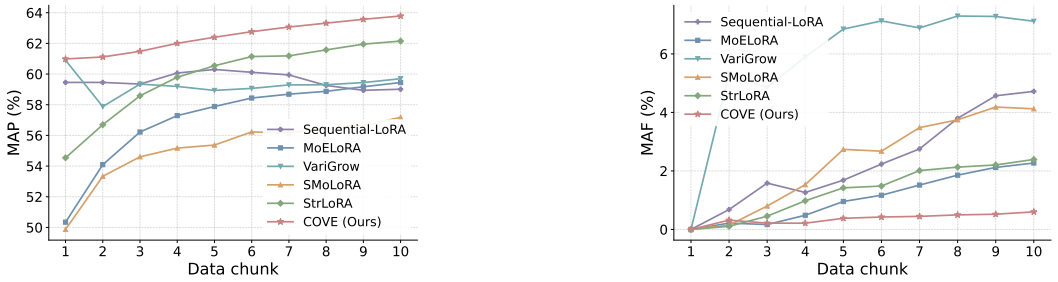
5 Experiments

In this section, we evaluate our proposed COVE on Streaming CVIT setting, and we primarily evaluate COVE on InternVL3.5-4B-Pretrained and InternVL3.5-8B-Pretrained, and further verify its generality on Gemma3-4B-PT. More Implementation Details are shown in Appendix H.

5.1 Experimental Setup

Evaluation Protocol. We follow the streaming continual visual instruction tuning setting introduced by StrCVIT, but independently construct all streaming benchmarks according to its stream-generation protocol rather than using the original benchmark data. All compared methods are reimplemented and evaluated on exactly the same constructed streams for controlled comparison.

Streaming Data Construction. Following the Streaming Continual Visual Instruction Tuning (StrCVIT)(Che et al., 2026a) setting, we construct heterogeneous data streams from a collection of vision-language datasets, including VQAv2(Goyal et al., 2017), Grounding(Mao et al., 2016), TextVQA(Singh et al., 2019), ImageNet-R(Hendrycks et al., 2021), Flickr30k(Plummer et al., 2015), ChartQA(Masry et al., 2022), IconQA(Lu et al., 2021), CLEVR(Lindström & Abraham, 2022), GQA(Hudson & Manning, 2019), Place365(Zhou et al., 2017), RSVQA-HR(Lobry et al., 2020), OCRVQA(Mishra et al., 2019) and TextCups(Sidorov et al., 2020). At each time step t, samples from multiple datasets are mixed within the same data chunk, and both the participating



(a) Mean Average Performance.

(b) Mean Average Forgetting.

Figure 2: Performance evolution over the streaming process.

distributions and their proportions vary across chunks. We construct three benchmark variants containing 8, 10, and 13 datasets, respectively. The complete stream compositions and construction details are provided in Appendix C.

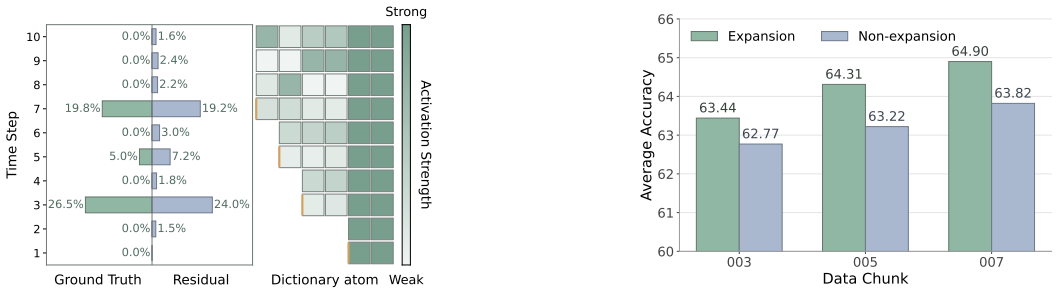
Evaluation Metrics. Following the evaluation protocol of StrCVIT (Che et al., 2026a), we evaluate continual learning performance using Average Performance (AP), Average Forgetting (AF), Mean Average Performance (MAP), and Mean Average Forgetting (MAF). Specifically, AP and AF measure the performance and forgetting of each dataset throughout the stream, while MAP and MAF report their averages across all datasets. Higher AP/MAP and lower AF/MAF indicate better continual learning performance. We additionally report Final AP, defined as the mean task accuracy of the final data chunk. Detailed metric definitions are provided in Appendix D.

Compared Methods. We compare COVE with five representative baselines. First, **Zero-shot** evaluates the pretrained model without any adaptation. **Joint LoRA**(Hu et al., 2021) jointly trains a single LoRA module using all stream data and is included as an offline reference rather than a continual-learning method, while **Sequential-LoRA** continuously fine-tunes a single shared LoRA module on incoming data chunks. As for expert-based CVIT method, **MoELoRA**(Chen et al., 2024) employs a mixture of LoRA experts for continual adaptation, whereas **SMoLoRA**(Wang et al., 2025) introduces separate routing mechanisms for visual understanding and instruction following. **VariGrow**(Ardywibowo et al., 2022) dynamically expands the model capacity based on novelty detection in task-agnostic continual learning. Finally, **StrLoRA**(Che et al., 2026a) is specifically designed for streaming continual visual instruction tuning, employing hierarchical expert routing and routing-stability regularization to handle dynamically evolving data streams. Appendix I give more details of these baselines.

5.2 Main Results

We evaluate COVE under dynamically evolving streaming data and compare it with representative continual adaptation methods across different model scales. As shown in Table 1, COVE achieves the strongest overall continual performance while substantially reducing forgetting. On InternVL3.5-4B-Pretrained, COVE obtains 63.87 MAP and 0.66 MAF, compared with 62.15 MAP and 2.39 MAF for StrLoRA. COVE also achieves the highest Final AP of 65.73, indicating that its advantage is maintained at the end of the stream. A similar trend is observed on InternVL3.5-8B-Pretrained, where COVE reaches 64.73 MAP, 0.81 MAF, and 66.38 Final AP, consistently outperforming the compared continual adaptation methods in overall and final-stage performance.

At the dataset level, COVE also demonstrates strong performance across heterogeneous vision-language distributions. On InternVL3.5-4B, COVE achieves 41.46 AP on ImageNet-R and 91.52 AP on CLEVR, improving over StrLoRA by 4.38 and 2.64 points, respectively. Meanwhile, it substantially reduces forgetting on several datasets, with AF values of only 0.09 on Grounding, 0.11 on TextVQA, and 0.55 on CLEVR. Similar behavior can be observed on InternVL3.5-8B, where COVE achieves strong AP on TextVQA, ImageNet-R, and CLEVR while maintaining low forgetting on multiple datasets such as Grounding, Flickr30k, and ChartQA. Although the best result on individual datasets may vary across methods, the improvements of COVE are distributed across



(a) Adaptation-demand alignment.

(b) Effect of vocabulary expansion.

Figure 3: nalysis of vocabulary evolution in COVE: (a) adaptation-demand alignment through residual-guided expansion and dictionary activation; (b) performance of expansion samples before and after vocabulary evolution.

different types of visual understanding and reasoning tasks rather than being dominated by a single dataset.

Fig. 2 further illustrates the performance evolution throughout the stream. COVE maintains a consistently strong MAP trajectory as new data chunks arrive, while its MAF remains low throughout the continual adaptation process. In contrast, several compared methods accumulate increasingly pronounced forgetting in later stages of the stream. These results suggest that evolving the adaptation structure together with the incoming stream allows COVE to better accommodate changing adaptation demands while preserving previously acquired knowledge.

Additional experiments in Appendix E further evaluate COVE under different stream configurations, benchmark scales, and model architectures, providing additional evidence of its effectiveness under diverse and dynamically evolving streaming settings.

5.3 Analysis of Vocabulary Evolution

Adaptation-Demand Alignment. We first examine whether the evolution of the learned concept vocabulary follows the changing adaptation demands of the streaming data. As shown in Fig. 3(a), the proportion of samples identified as insufficiently represented by the historical vocabulary exhibits a trend consistent with the ground-truth emergence of novel distributions. This indicates that the reconstruction residual provides a meaningful signal of when the accumulated vocabulary becomes insufficient for the incoming data. Correspondingly, vocabulary expansion is triggered when such unexplained demand emerges, while the vocabulary remains unchanged when the historical units are sufficient to represent the incoming chunk.

The activation map further reveals the complementary reuse behavior of the vocabulary. Atoms discovered in early chunks remain active across later stages, showing that previously accumulated units continue to provide reusable adaptation knowledge. Meanwhile, units introduced during later expansion stages can also be reactivated by subsequent chunks, indicating that newly discovered knowledge is incorporated into the historical vocabulary and becomes reusable as the stream evolves. Together, these observations show that COVE dynamically adjusts both the reuse and expansion of adaptation units according to the changing demands of the stream.

Functional Effect of Vocabulary Evolution. We further examine whether the newly introduced units provide useful adaptation capacity for samples that are insufficiently represented by the historical vocabulary. For each chunk that triggers vocabulary expansion, we evaluate the same data chunk samples before and after incorporating the newly discovered units. As shown in Fig. 3(b), the evolved vocabulary consistently improves the average accuracy on these samples compared with using the historical vocabulary alone. This indicates that the expansion is not merely a structural change in vocabulary size, the newly introduced units capture adaptation demands that cannot be sufficiently handled by the accumulated historical units. Therefore, the residual-guided expansion provides functional capacity precisely when the existing vocabulary becomes insufficient for the incoming data.

Data-Driven Vocabulary Construction. We further examine whether constructing the adaptation vocabulary directly from incoming data leads to more effective adaptation units than using predefined expert structures. To isolate the effect of vocabulary construction from subsequent vocabulary expansion and historical-unit refinement, we conduct this analysis on the first data chunk. We vary the dictionary regularization coefficients to obtain vocabularies with different numbers of retained units and compare COVE with MoELoRA and StrLoRA under matched expert budgets.

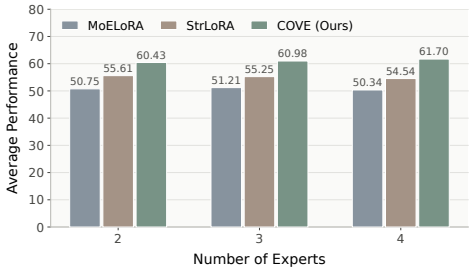


Figure 4: Performance comparison under different expert budgets.

As shown in Fig. 4, COVE consistently achieves stronger average performance across different vocabulary sizes. Notably, COVE with only two retained units already outperforms both MoELoRA and StrLoRA using four experts. This indicates that the advantage of COVE does not simply arise from maintaining more adaptation capacity. Instead, discovering adaptation units directly from the incoming data yields a more effective vocabulary, allowing the available capacity to better capture the adaptation requirements of the current data.

5.4 Ablation Study

We ablate the major components of COVE to examine their contributions to continual adaptation. As shown in Table 2, removing vocabulary expansion increases MAF from 0.66 to 1.99, despite only minor changes in MAP. While disabling historical-expert refinement raises it to 1.87, with its MAP also decreases heavily. This indicates that both adaptive expansion and selective refinement play important roles in preserving previously acquired knowledge throughout the stream. Removing the responsibility-based gradient scaling ρ leads to a smaller degradation, with MAF increasing from 0.66 to 0.70, suggesting that ρ provides an additional fine-grained control over historical expert updates. Overall, COVE achieves the best balance between continual performance and forgetting.

Table 2: Ablation study.

Setting	Ref.	Exp.	ρ	MAP↑	MAF↓
COVE	✓	✓	✓	**63.87**	**0.66**
w/o ρ	✓	✓		63.77	0.70
w/o refinement		✓		63.67	1.87
w/o expansion	✓		✓	63.84	1.99

6 Limitations and Future Work

COVE currently evolves the concept vocabulary mainly through selective refinement and vocabulary expansion. While the experts associated with historical units can be refined, the dictionary structure itself follows an incremental-only evolution: existing atoms are retained, and newly discovered atoms are added without explicitly reorganizing, merging, or replacing previous ones. Over long-term streams, this may lead to redundant units or make the accumulated vocabulary less representative under gradual concept drift. A promising direction is therefore to develop reliable vocabulary restructuring mechanisms, such as consolidation, merging, or carefully controlled atom updates, while avoiding unstable changes caused by transient streaming data.

7 Conclusion

In this work, we proposed Continual Vocabulary Evolution (COVE) for streaming continual visual instruction tuning, which organizes continual adaptation as an evolving vocabulary of reusable adaptation units. COVE discover adaptation unit form incoming data, expands the vocabulary when existing units are insufficient to capture emerging adaptation demands and then selectively refines reusable historical units. Extensive experiments demonstrate that COVE achieves strong continual performance with substantially reduced forgetting across evolving data streams.

AI USE STATEMENT

In this work, we used generative AI tools to assist with providing feedback on experiments, implementing methods, and interpreting results. We did not use generative AI tools to formulate mathematical claims or to develop theoretical models or conceptual frameworks. The following uses were not applicable to this work: proposing or refining hypotheses, generating synthetic datasets, providing critical ingredients for proving mathematical claims, assisting in the writing of proofs, assisting with translation, cleaning or reformatting datasets, and supporting qualitative or thematic data analysis.

Additionally, we used generative AI tools to polish the research paper for improved readability, and to help identify some of the relevant literatures for the paper. We reviewed all AI-assisted work. LLM-generated code was verified and tested for correctness by the authors. We take responsibility for the final content of this work, including all text, claims, and artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

For reproducibility, Sections 4.2 and 4.3 describe the core formulation of COVE, while Algorithm 1 summarizes the complete continual adaptation workflow.Appendix A provides the code implementation of the core vocabulary evolution algorithm. and Appendix H further introduce the detailed implementation and training configurations, including optimization, LoRA settings, dictionary learning, hyperparameters, and hardware. The streaming benchmark construction and evaluation metrics are documented in Appendices C and D, respectively. Together, these materials provide the information necessary to reproduce the training, evaluation, and analysis of COVE.

REFERENCES

Rahaf Aljundi, Klaas Kelchtermans, and Tinne Tuytelaars. Task-free continual learning. In *2019 IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, pp. 11246–11255. IEEE, 2019.

Randy Ardywibowo, Zepeng Huo, Zhangyang Wang, Bobak J Mortazavi, Shuai Huang, and Xiaoning Qian. Varigrow: Variational architecture growing for task-agnostic continual learning based on bayesian novelty. In *International Conference on Machine Learning*, pp. 865–877. PMLR, 2022.

Chang Che, Ziqi Wang, Hui Ma, Cheems Wang, and Zenglin Shi. Strlora: Towards streaming continual visual instruction tuning for mllms. *arXiv preprint arXiv:2605.16353*, 2026a.

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. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 19978–19986, 2026b.

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

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. *arXiv preprint arXiv:2506.11672*, 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 *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 6904–6913, 2017.

Jinghan He, Haiyun Guo, Kuan Zhu, Ming Tang, and Jinqiao Wang. Continual instruction tuning for large multimodal models. *IEEE Transactions on Image Processing*, 2026.

Dan Hendrycks, Steven Basart, Norman Mu, Saurav Kadavath, Frank Wang, Evan Dorundo, Rahul Desai, Tyler Zhu, Samyak Parajuli, Mike Guo, et al. The many faces of robustness: A critical analysis of out-of-distribution generalization. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 8320–8329. IEEE, 2021.

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

Drew A Hudson and Christopher D Manning. Gqa: A new dataset for real-world visual reasoning and compositional question answering. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6693–6702. IEEE, 2019.

Yukang Huo and Hao Tang. When continue learning meets multimodal large language model: A survey. *arXiv preprint arXiv:2503.01887*, 2025.

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

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

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

Junhua Mao, Jonathan Huang, Alexander Toshev, Oana Camburu, Alan L Yuille, and Kevin Murphy. Generation and comprehension of unambiguous object descriptions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 11–20, 2016.

Ahmed Masry, Jia Qing Tan, Shafiq Joty, Enamul Hoque, et al. 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 *2019 international conference on document analysis and recognition (ICDAR)*, pp. 947–952. IEEE, 2019.

Jun-Yeong Moon, Keon-Hee Park, Jung Uk Kim, and Gyeong-Moon Park. Online class incremental learning on stochastic blurry task boundary via mask and visual prompt tuning. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 11697–11707. IEEE, 2023.

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

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

Amanpreet Singh, Vivek Natarajan, Meet Shah, Yu Jiang, Xinlei Chen, Dhruv Batra, Devi Parikh, and Marcus Rohrbach. Towards vqa models that can read. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8309–8318. IEEE, 2019.

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*, 2026.

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.

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

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

Ying Zhang, Xingyue Guo, Yu Zhao, Xuhui Sui, Baohang Zhou, Xinying Qian, and Xiaojie Yuan. Grow-on-demand: Sparse and adaptive expert expansion for continual instruction tuning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 28474–28482, 2026.

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 CODE

https://anonymous.4open.science/r/COVE-6ED8/README.md

B CONTINUAL VISUAL INSTRUCTION TUNING

Continual Visual Instruction Tuning (CVIT)(Chen et al., 2024; Huo & Tang, 2025) aims to adapt a pretrained MLLMs to a sequential visual-instruction tasks while preserving the knowledge acquired from previous stages. Let $T = \{T_1, T_2, \ldots, T_n\}$ denotes a series of tasks, and $D = \{D_1, D_2, \ldots, D_n\}$ represents the corresponding instruction datasets. At each time step t, the model receives a new training set $\mathcal{D}_t = \{(v_i^t, x_i^t, y_i^t)\}_{i=1}^{N_t}$, consisting of the visual input v_i^t, the textual instruction x_i^t, and the response y_i^t. Meanwhile, the model integrates new knowledge and mitigates forgetting prior tasks, which previously data $\{D_i\}_{i=1}^{t-1}$ is generally unavailable except the replay-based method. The key challenge of CVIT is fine-tuning for new tasks with complex multimodal data while generalizing across past tasks to avoid catastrophic forgetting. Furthermore, because of the high computational cost of repeatedly fine-tuning MLLMs, existing works(Chen et al., 2024; Hu et al., 2021; Che et al., 2026a) adopted parameter-efficient fine-tuning (PEFT) to reduce the number of trainable parameters. In particular, Low-Rank Adaptation (LoRA) updates a pretrained weight matrix as $W' = W_0 + BA$, where $W_0 \in \mathbb{R}^{d\times k}$ remains frozen, and $A \in \mathbb{R}^{r\times k}$ and $B \in \mathbb{R}^{d\times r}$ are trainable low-rank matrices with $r \ll \min(d, k)$.

C DETAILED DATA STREAM COMPOSITION

Following the StrCVIT setting, we construct an evolving data stream, where each data chunk may contain multiple previously observed tasks samples while new tasks are progressively introduced. In our experiments, our construct three different scaling benchmarks, with 8-task, 10-task and 13-task corresponding. All these benchmarks covered a wide range of general capabilities, such as grounding, caption, classification, and VQA to domain-specific scenarios including chart understanding, OCR and remote sensing. We summarize the detailed task information in Table 3. These three benchmarks contains 10 chunks and each data chunk sampling 1000 samples randomly from diverse tasks datasets with different compositions, which may both from previous observed tasks and newly emerging datasets. For the inference, we use 500 samples from each tasks to evaluate the model performance of each task. Furthermore, we also construct two additional 8-tasks benchmarks with 15 data chunks and 20 data chunks, respectively.

Figures 5–9 present the detailed chunk-wise composition of the five data streams used in our experiments. Each bar corresponds to one incoming data chunk, while different colors indicate samples originating from different tasks. The height of each colored segment denotes the proportion of the corresponding task within that chunk. Across the five benchmarks, the task composition varies continuously over time, where previously observed tasks may recur in later chunks, while new tasks are progressively introduced into the stream. This results in interleaved and dynamically evolving streams rather than a sequence of isolated task blocks.

In particular, as shown in Fig 5, The stream initially contains four tasks, including VQAv2, TextVQA, ImageNet-R, and Flickr30k. New task distributions are subsequently introduced at several distinct stages, where Grounding and IconQA first appear in chunk 003, ChartQA in chunk 005, and CLEVR in chunk 007. Therefore, these chunks correspond to clear distribution-emergence points in the stream. Also Previously observed tasks repeatedly disappear and reappear across later chunks, and their relative proportions vary substantially over time. This structure provides a realistic non-stationary task setting for examining whether COVE can distinguish reusable knowledge from genuinely new adaptation demands.

D EVALUATION METRICS.

Following the evaluation protocol used in StrCVIT(Che et al., 2026a), we evaluate continual learning performance from both knowledge acquisition and forgetting perspectives. Let a_t^m denote the accuracy on the m-th dataset after the model processes the t-th data chunk. At t time step, the

Table 3: Dataset composition for different tasks.

Task	Train Dataset	Test Dataset
Grounding	RefCOCO	RefCOCO
Image Captioning	TextCaps	TextCaps
Image Classification	ImageNet-R	ImageNet-R
Scene Classification	Places365	Places365
Visual Question Answering (VQA)	VQAv2	VQAv2
Visual Reasoning VQA	GQA	GQA
Remote Sensing VQA	RSVQA-HR	RSVQA-HR
OCR VQA	OCRVQA	OCRVQA
Chart VQA	ChartQA	ChartQA
Text VQA	TextVQA	TextVQA
Image Captioning	Flickr30k	Flickr30k
Icon VQA	IconQA	IconQA
Visual Reasoning VQA	CLEVR	CLEVR

forgetting on m-th dataset is defined as:

$$F_t^m = \max\left(0, \frac{\max_{1\le j<t} a_j^m - a_t^m}{\max_{1\le j<t} a_j^m}\right). \tag{13}$$

This metric quantifies the relative performance decline on dataset m compared with its best performance across all previous stages. To evaluate the performance of each single dataset across the whole stream, we compute Average Performance (AP) [] and Average Forgetting (AF):

$$\mathrm{AP}_t^m = \frac{1}{t}\sum_{j=1}^{t} a_j^m, \qquad \mathrm{AF}_t^m = \frac{1}{t}\sum_{j=1}^{t} F_j^m. \tag{14}$$

As for the overall performance across all datasets, we also report Mean Average Performance (MAP) and Mean Average Forgetting (MAF):

$$\mathrm{MAP}_t = \frac{1}{M}\sum_{m=1}^{M} \mathrm{AP}_t^m, \qquad \mathrm{MAF}_t = \frac{1}{M}\sum_{m=1}^{M} \mathrm{AF}_t^m. \tag{15}$$

Higher AP and MAP indicate better overall continual performance, while lower AF and MAF indicate less catastrophic forgetting. In addition, we report Final AP, defined as the mean task accuracy after processing the final data chunk, to measure the model's final state performance.

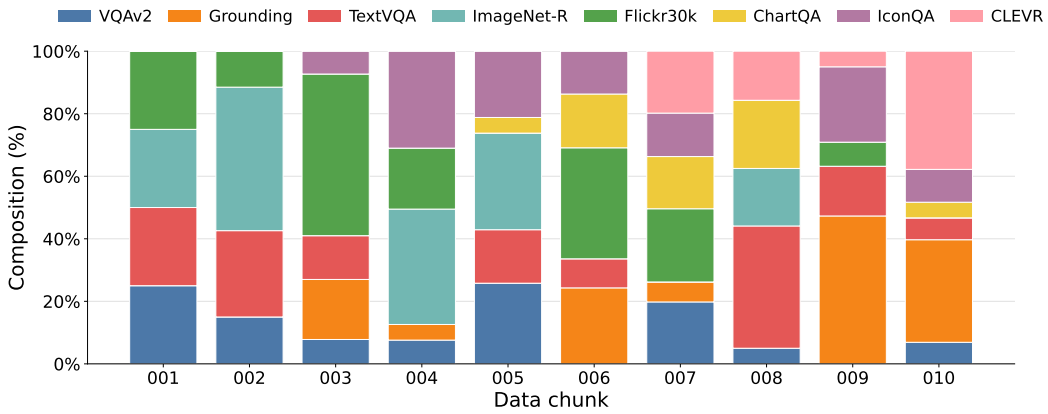


Figure 5: Composition of 10 streaming data chunks on 8 tasks.

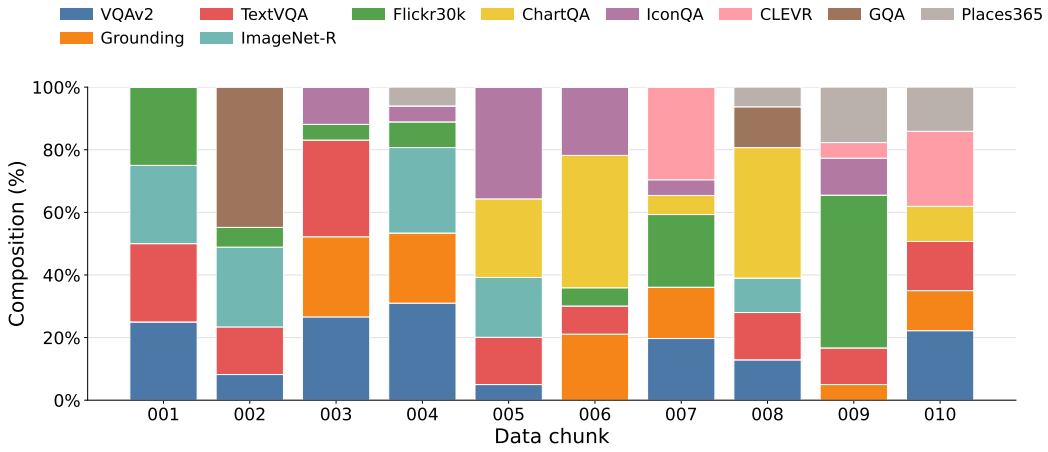


Figure 6: Composition of 10 streaming data chunks on 10 tasks.

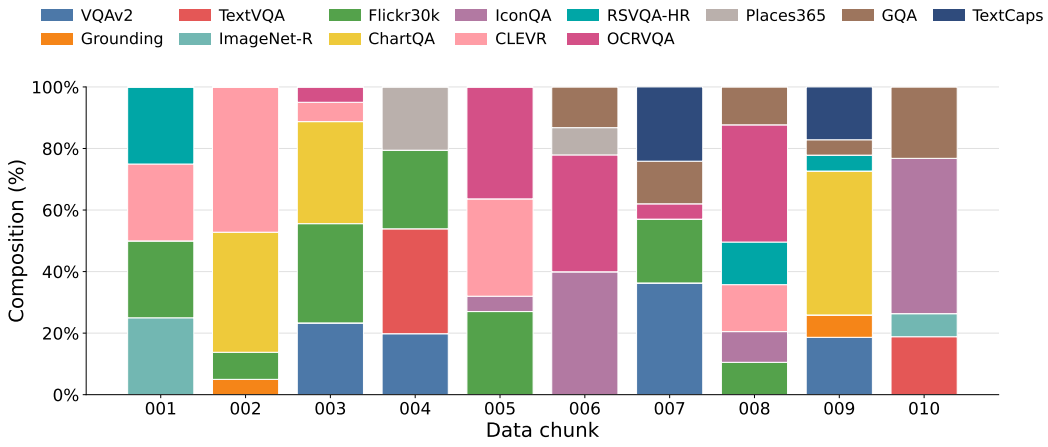


Figure 7: Composition of 10 streaming data chunks on 13 tasks.

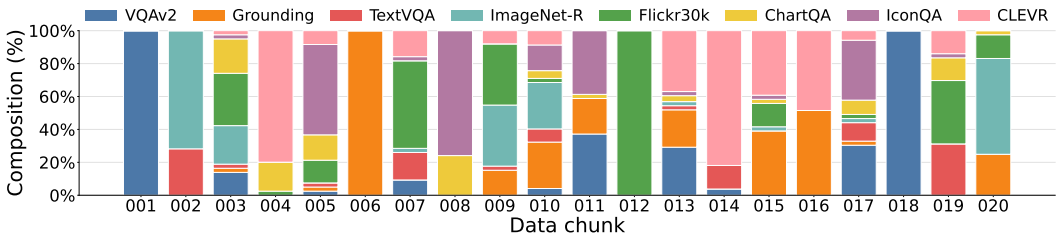


Figure 8: Composition of 15 streaming data chunks on 8 tasks.

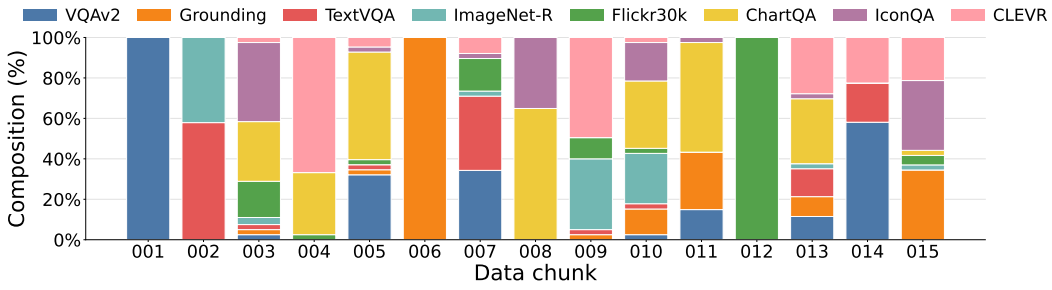


Figure 9: Composition of 20 streaming data chunks on 8 tasks.

E Additional Experiments

To further examine the robustness of COVE beyond the streaming data experimental setting, we evaluate it under different chunk scalings and streams containing a larger number of tasks.

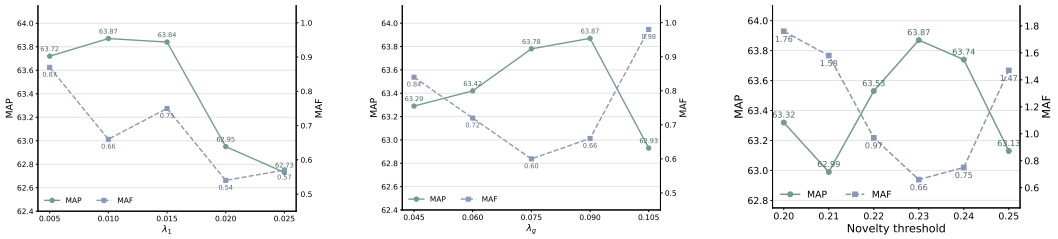
Different data chunk scaling. We further evaluate COVE under finer stream granularities by dividing the 8-task stream into 15 and 20 data chunks. As shown in Table 4, COVE consistently achieves the highest MAP and substantially lower MAF across both stream granularity and model scales. With InternVL3.5-4B, COVE obtains 63.47/0.91 and 63.70/0.77 MAP/MAF under 15 and 20 chunks, respectively. As for InternVL3.5-8B, the corresponding results are 65.74/0.77 and 65.95/0.63. These results show that COVE remains effective in more data chunk setting, suggesting that its continual adaptation does not depend on a particular stream granularity.

Scaling to more tasks. We also examine whether COVE remains effective as the number of tasks increasing. Table 5 reports the results on the 10-task benchmark with InternVL3.5-4B-Pretrained. COVE achieves an overall MAP of 58.74, outperforming Sequential-LoRA, MoELoRA, and StrLoRA, while reducing MAF to 1.54. In comparison, the competing continual adaptation methods StrLoRA(Che et al., 2026a) raises MAF to 3.28. Meanwhile, the results shown in Table 6 also show that our method achieves the highest MAP 60.8 and lower MAF 1.78 compared with other methods. Therefore, the improvement of our COVE is particularly success in forgetting, indicating that the evolving vocabulary can accommodate a larger set of task distributions while preserving previously acquired knowledge.

Generalization to different base models. We further evaluate COVE with Gemma3-4B-PT to examine whether its effectiveness generalizes beyond InternVL3.5. As shown in Table 7, COVE achieves an overall MAP of 48.22, compared with 46.43 for StrLoRA, while substantially reducing MAF from 5.75 to 1.95. COVE also improves performance on most individual tasks. These results demonstrate that the effectiveness of COVE is not specific to the InternVL3.5 backbone and transfers consistently to a different pretrained vision-language model.

F Sensitive Analysis

Firstly, We investigate the sensitivity of COVE to the two hyperparameters λ_1 and λ_g, as shown in Fig. **??**. For λ_1, COVE maintains comparable MAP when λ_1 varies from 0.005 to 0.015, with the best MAP of 63.87 achieved at $\lambda_1 = 0.01$. Increasing λ_1 further reduces MAF and leads to a noticeable drop in MAP, which indicates an increasing trade-off between knowledge retention and adaptation. Meanwhile, we observe the similar trend for λ_g. The performance remains stable over a moderate range, where reaching 63.87 MAP and 0.66 MAF at $\lambda_g = 0.09$, whereas an excessively large value ($\lambda_g = 0.105$) degrades both metrics. As for δ, which controls the criterion for identifying novel samples. As δ approaches 0.23, COVE achieves the highest MAP of 63.87 and the lowest MAF of 0.66. Deviating from this value in either direction degrades performance, indicating that an appropriate novelty threshold is important for reliably distinguishing novel samples from those that can be explained by the existing vocabulary. Overall, COVE is relatively insensitive within reasonable parameter ranges, while extreme values may disturb the balance between adaptation and forgetting. Overall, COVE is robust to moderate variations in these hyperparameters, whereas extreme settings may lead to suboptimal expert refinement or inaccurate novel-sample identification, thereby affecting subsequent vocabulary expansion.



(a) Sensitivity analysis of λ_1. (b) Sensitivity analysis of λ_g. (c) Sensitivity analysis of δ.

Figure 10: Sensitivity analysis of λ_1, λ_g, and the novelty threshold.

Table 4: Performance comparison under different stream granularities (15 and 20 chunks). Higher AP/MAP (↑) and lower AF/MAF (↓) indicate better performance.

Method	VQAv2		Grounding		TextVQA		ImageNet-R		Flickr30k		ChartQA		IconQA		CLEVR		Overall	
	AP↑	AF↓	AP↑	AF↓	AP↑	AF↓	AP↑	AF↓	AP↑	AF↓	AP↑	AF↓	AP↑	AF↓	AP↑	AF↓	MAP↑	MAF↓
InternVL3.5-4B-Pretrained, 20 chunks																		
Sequential-LoRA	69.62	2.48	59.03	2.75	73.48	2.09	34.50	7.70	6.81	4.85	83.21	1.30	64.24	3.90	76.18	20.60	58.38	5.71
MoELoRA	69.12	2.10	56.85	3.35	70.37	1.30	36.29	**1.75**	6.80	3.25	80.84	**0.22**	67.95	2.65	85.09	7.20	59.16	2.73
StrLoRA	71.23	0.85	64.09	0.95	73.95	0.42	36.66	7.55	7.14	6.10	79.32	0.40	**74.66**	0.45	88.33	5.75	61.92	2.81
COVE (Ours)	**71.40**	**0.38**	**64.36**	**0.72**	**77.07**	**0.17**	**40.72**	2.00	**7.36**	**1.30**	**83.39**	0.81	74.38	**0.25**	**90.90**	**0.55**	**63.70**	**0.77**
InternVL3.5-4B-Pretrained, 15 chunks																		
Sequential-LoRA	69.27	2.73	58.18	3.10	72.28	2.74	33.50	8.90	6.71	5.45	82.66	1.65	63.44	4.35	74.63	23.40	57.58	6.54
MoELoRA	69.00	2.30	56.53	3.65	69.82	1.55	35.79	2.35	6.73	3.65	80.54	**0.37**	67.35	3.00	84.14	8.60	58.74	3.18
StrLoRA	71.29	0.95	63.87	1.20	73.55	0.54	36.21	8.00	7.10	6.75	79.17	0.49	**74.34**	0.65	87.41	6.85	61.62	3.18
COVE (Ours)	**71.45**	**0.42**	**64.18**	**0.84**	**76.89**	**0.23**	**40.50**	**2.32**	**7.32**	**1.46**	**83.21**	0.91	74.13	**0.33**	**90.08**	**0.77**	**63.47**	**0.91**
InternVL3.5-8B-Pretrained, 20 chunks																		
Sequential-LoRA	70.45	2.30	61.25	3.50	77.15	2.10	37.05	7.20	7.22	5.20	88.10	1.20	66.30	4.20	78.80	19.50	60.79	5.65
MoELoRA	70.65	1.70	59.80	2.95	74.70	1.20	38.45	**1.60**	7.27	3.05	86.60	**0.20**	70.70	2.20	87.50	7.20	61.96	2.51
StrLoRA	72.00	0.70	65.20	0.85	77.50	0.35	39.20	6.60	7.49	5.70	85.80	0.30	77.10	0.35	91.30	5.00	64.45	2.48
COVE (Ours)	**72.35**	**0.28**	**65.75**	**0.65**	**80.00**	**0.15**	**42.35**	1.85	**7.55**	**1.10**	**88.30**	0.40	**77.60**	**0.18**	**93.70**	**0.45**	**65.95**	**0.63**
InternVL3.5-8B-Pretrained, 15 chunks																		
Sequential-LoRA	70.05	2.55	60.40	4.00	75.95	2.60	36.35	8.30	7.14	5.65	87.60	1.60	65.30	4.90	77.20	22.60	60.00	6.53
MoELoRA	70.30	1.80	59.35	3.25	74.12	1.40	37.90	2.15	7.22	3.50	86.25	**0.30**	70.32	2.40	86.50	8.70	61.50	2.94
StrLoRA	71.96	0.78	64.90	1.08	77.00	0.48	38.76	7.20	7.46	6.15	85.55	0.42	76.82	0.47	90.55	5.90	64.13	2.81
COVE (Ours)	**72.39**	**0.34**	**65.57**	**0.73**	**79.80**	**0.18**	**42.15**	**2.05**	**7.54**	**1.30**	**88.10**	0.53	**77.45**	**0.27**	**92.90**	**0.73**	**65.74**	**0.77**

Table 5: Performance comparison on the 10-task benchmark with InternVL3.5-4B-Pretrained. Higher AP/MAP (↑) and lower AF/MAF (↓) indicate better performance.

Method	VQAv2		Grounding		TextVQA		ImageNet-R		Flickr30k		ChartQA		IconQA		CLEVR		GQA		Places365		Overall	
	AP↑	AF↓	AP↑	AF↓	AP↑	AF↓	AP↑	AF↓	AP↑	AF↓	AP↑	AF↓	AP↑	AF↓	AP↑	AF↓	AP↑	AF↓	AP↑	AF↓	MAP↑	MAF↓
Zero-shot	69.00	–	52.60	–	20.13	–	25.80	–	6.29	–	53.60	–	64.40	–	85.80	–	68.20	–	9.40	–	45.52	–
Sequential-LoRA	69.10	1.90	58.80	2.70	72.40	1.55	34.80	7.20	6.85	4.80	78.10	1.30	66.90	3.80	76.20	17.50	67.90	2.10	9.10	15.00	54.02	5.79
MoELoRA	69.40	1.40	60.30	2.50	72.90	1.10	36.10	2.70	6.98	3.40	79.30	0.70	69.70	2.40	83.20	6.50	69.10	1.20	9.50	10.20	55.65	3.21
StrLoRA	**71.20**	**0.67**	64.26	**0.50**	74.07	0.28	37.10	6.89	**7.15**	5.55	79.54	**0.21**	**74.88**	**0.26**	88.80	4.61	70.60	0.90	9.74	12.98	57.73	3.28
COVE (Ours)	**71.20**	1.22	**64.76**	0.63	**76.93**	**0.21**	**39.80**	**1.04**	6.92	**2.75**	**82.78**	0.45	74.16	0.28	**89.04**	**3.19**	**71.18**	**0.36**	**10.58**	**5.26**	**58.74**	**1.54**

Table 6: Performance comparison on the 13-task benchmark with InternVL3.5-4B-Pretrained. Higher AP/MAP (↑) and lower AF/MAF (↓) indicate better performance.

Method	VQAv2		Grounding		TextVQA		ImageNet-R		Flickr30k		ChartQA		IconQA	
	AP↑	AF↓	AP↑	AF↓	AP↑	AF↓	AP↑	AF↓	AP↑	AF↓	AP↑	AF↓	AP↑	AF↓
Zero-shot	69.00	–	52.60	–	20.13	–	25.80	–	6.29	–	53.60	–	64.40	–
Sequential-LoRA	68.85	2.30	58.22	3.25	71.65	2.10	33.90	8.30	6.72	5.50	77.45	1.85	65.88	4.70
MoELoRA	69.12	1.70	59.85	2.85	72.48	1.48	35.42	3.45	6.91	4.05	78.88	1.05	69.10	2.90
StrLoRA	70.82	**0.95**	63.75	0.82	73.62	0.46	36.55	7.45	**7.06**	6.10	79.12	**0.42**	73.95	0.53
COVE (Ours)	**71.08**	1.08	**64.28**	**0.78**	**76.15**	**0.38**	**39.14**	**1.42**	6.88	**3.15**	**82.10**	0.62	**74.25**	**0.48**

Method	CLEVR		GQA		Places365		RSVQA-HR		OCRVQA		TextCaps		Overall	
	AP↑	AF↓	AP↑	AF↓	AP↑	AF↓	AP↑	AF↓	AP↑	AF↓	AP↑	AF↓	MAP↑	MAF↓
Zero-shot	85.80	–	68.20	–	9.40	–	48.70	–	22.40	–	46.10	–	44.03	–
Sequential-LoRA	74.92	19.20	67.35	2.75	8.86	16.50	62.35	4.90	48.62	6.80	78.42	5.30	55.63	6.42
MoELoRA	81.90	8.20	68.75	1.65	9.20	12.40	65.20	3.75	51.28	4.80	80.45	3.10	57.58	3.95
StrLoRA	87.95	5.50	70.05	1.25	9.56	14.20	66.80	2.85	52.05	3.60	81.32	2.70	59.43	3.60
COVE (Ours)	**88.65**	**3.75**	**70.92**	**0.55**	**10.20**	**6.30**	**68.40**	**1.60**	**54.70**	**1.85**	**83.65**	**1.20**	**60.80**	**1.78**

Table 7: Performance comparison on the 8-task benchmark with Gemma3-4B-PT. Higher AP/MAP (↑) and lower AF/MAF (↓) indicate better performance.

Method	VQAv2		Grounding		TextVQA		ImageNet-R		Flickr30k		ChartQA		IconQA		CLEVR		Overall	
	AP↑	AF↓	AP↑	AF↓	AP↑	AF↓	AP↑	AF↓	AP↑	AF↓	AP↑	AF↓	AP↑	AF↓	AP↑	AF↓	MAP↑	MAF↓
Gemma3-4B-PT																		
Zero-shot	14.70	–	0.00	–	5.80	–	15.20	–	0.90	–	23.50	–	28.00	–	38.80	–	15.86	–
StrLoRA	55.00	6.10	6.20	13.20	58.30	3.42	34.10	7.18	7.10	5.98	69.80	2.65	64.50	3.15	76.40	4.35	46.43	5.75
COVE (Ours)	57.30	1.15	7.55	5.60	61.20	0.55	36.90	2.05	6.95	1.40	72.40	2.92	64.35	0.65	79.10	1.30	48.22	1.95

Algorithm 1 Continual Vocabulary Evolution (COVE)

Require: Data stream $\{\mathcal{B}_t\}_{t=1}^{T}$, frozen pretrained model θ_0, threshold δ
Ensure: Evolved vocabulary $\mathcal{V}_T$

for $t = 1, \ldots, T$ **do**
  Extract multimodal features F_t using ϕ_{θ_0}.
  if $t = 1$ **then**
    Learn the initial candidate dictionary $\widetilde{D}_1$ using Eq. 3.
    Retain atoms with nonzero coefficient rows to obtain D_1.
    Instantiate a LoRA expert for each retained atom to form $\mathcal{V}_1$.
  else
    Compute historical sparse coefficients $\{c_i^{\text{old}}\}$ using Eq. 4.
    Compute reconstruction residuals $\{r_i, e_i\}$ using Eq. 5.
    Partition samples into $\mathcal{R}_t$ and $\mathcal{N}_t$ using Eq. 6.
    $D_t \leftarrow D_{t-1}$
    if $\mathcal{N}_t \neq \emptyset$ **then**
      Construct the candidate dictionary $D_t^{\text{cand}} = [D_{t-1}, \widetilde{D}_t]$.
      Learn candidate atoms using Eq. 7.
      Retain candidate atoms with nonzero coefficient rows.
      Expand D_t with the retained atoms.
      Instantiate corresponding LoRA experts.
    end if
    Form the updated vocabulary $\mathcal{V}_t$.
  end if
  Compute sparse coefficients $\{c_i\}$ over D_t using Eq. 8.
  Compose sample-specific experts using Eq. 9.
  Optimize the SFT objective in Eq. 10:
  if $t = 1$ **then**
    Train both LoRA matrices of all initialized experts.
  else
    Train newly introduced experts using only samples in $\mathcal{N}_t$.
    Freeze $A_j^{(\ell)}$ for all historical experts.
    Compute responsibilities ρ_{ij} using Eq. 11.
    Refine historical experts using samples in $\mathcal{R}_t$ with Eq. 12.
  end if
  Carry forward the adapted vocabulary $\mathcal{V}_t$.
end for
return $\mathcal{V}_T$

G Algorithm

The complete algorithm is shown in the Algorithm 1.

H Implementation Details

H.1 Training and Optimization Setup

We implement COVE on InternVL3.5-4B-Pretrained and InternVL3.5-8B-Pretrained, using LoRA modules as the basic reusable adaptation units. All methods are trained under the same streaming protocol and backbone configuration for a fair comparison. For each incoming data chunk, the model is optimized only using the currently available samples, without revisiting data from the previous chunks. We use Adam with a learning rate 2×10^{-5}, a batch size of 32, and train for 1 epoch/ 32 optimizer step for each chunk. Each LoRA is set to 4 rank, with a scaling factor of 32. and LoRA modules 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.

For COVE, we initialize the dictionary from 64 candidate atoms selected via farthest-point sampling. Sparse coefficients are estimated independently for each sample using batched FISTA. We use 80 FISTA iterations within each of 20 alternating optimization rounds during dictionary construction, and 40 FISTA iterations for sample-wise coefficient estimation during training and interference. The resulting coefficients are shared across all LoRA-augmented layers within the same forward pass. Then the vocabulary refined and expanded according to the optimization described in Sec. 4.2. The hyperparameter λ_1 and λ_g controlling historical unit refinement and vocabulary expansion are set to 0.01 and 0.09, respectively, and are kept fixed across all datasets. All experiments are conducted using NVIDIA A800 GPUs.

H.2 Question-conditioned Feature Extraction

We use the frozen base MLLM as the feature extractor throughout all the experiments. Rather than directly averaging all multimodal tokens, we construct a question-conditioned representation that preserves the textural semantics while emphasizing relevant visual information to corresponding question.

Textural Representation. Let $\mathbf{H}^q = \{\mathbf{h}_j^q\}_{j=1}^{N_q}$ denote the hidden states of the textual tokens extracted from a selected intermediate layer of the frozen MLLM. We first apply token-wise ℓ_2 normalization:

$$\hat{\mathbf{h}}_j^q = \frac{\mathbf{h}_j^q}{\|\mathbf{h}_j^q\|_2}. \tag{16}$$

We then average the normalized token representations and normalize the result again:

$$\mathbf{z}_q = \mathrm{Norm}\left(\frac{1}{N_q}\sum_{j=1}^{N_q}\hat{\mathbf{h}}_j^q\right), \tag{17}$$

where $\mathrm{Norm}(\cdot)$ denotes ℓ_2 normalization. The resulting $\mathbf{z}_q$ provides a compact representation of the semantic content of the question.

Question-Weighted Visual Pooling. Let $\mathbf{H}^v = \{\mathbf{h}_i^v\}_{i=1}^{N_v}$ denote the visual-token hidden states from the same intermediate layer. Similarly, each visual token is first normalized as

$$\hat{\mathbf{h}}_i^v = \frac{\mathbf{h}_i^v}{\|\mathbf{h}_i^v\|_2}. \tag{18}$$

Then, we compute the relevance of each visual token to the question representation using cosine similarity:

$$s_i = (\hat{\mathbf{h}}_i^v)^\top \mathbf{z}_q. \tag{19}$$

The relevance scores are converted into normalized weights using a temperature-scaled softmax:

$$\alpha_i = \frac{\exp(s_i/\tau)}{\sum_{k=1}^{N_v}\exp(s_k/\tau)}, \tag{20}$$

where τ denotes the temperature parameter. Therefore, the question-guided visual representation is obtained as

$$\mathbf{z}_v = \mathrm{Norm}\left(\sum_{i=1}^{N_v}\alpha_i\hat{\mathbf{h}}_i^v\right). \tag{21}$$

Finally, we concatenate the textual representation and the corresponding question-weighted visual representation:

$$\mathbf{f}(x,q) = \mathrm{Norm}\left([\mathbf{z}_q;\mathbf{z}_v]\right), \tag{22}$$

where x and q denote the image and question, respectively. This feature is subsequently used for vocabulary construction and sample-wise routing in COVE.

H.3 Implementation of Expert Adaptation

We provide further implementation details for the expert adaptation strategy introduced in Sec. 4.3. Importantly,the sparse coefficients determine how experts contribution to forward computation to get the final response, while how these experts are updated are determined by the reuse and novel samples during backpropagation.

For each incoming sample i, the contribution of expert j at layer ℓ is

$$\Delta y_{ij}^{(\ell)} = s\, c_{ij} B_j^{(\ell)} A_j^{(\ell)} x_i^{(\ell)}, \tag{23}$$

where c_{ij} is the sparse coefficient defined in the main text. Here, We use stop-gradient to control the backward update while preserving this forward contribution. And we denote the stop-gradient operator by $\mathrm{sg}(\cdot)$, which satisfies $\mathrm{sg}(z) = z$ in the forward pass but has zero gradient during back-propagation.

Historical experts. For each historical expert $j \le K_{t-1}$, we keep $A_j^{(\ell)}$ fixed and only allow $B_j^{(\ell)}$ to be refined. For a reuse sample $i \in \mathcal{R}_t$, we modulate the expert contribute as:

$$\widetilde{\Delta y}_{ij}^{(\ell)} = \rho_{ij} \Delta y_{ij}^{(\ell)} + (1 - \rho_{ij})\, \mathrm{sg}\left(\Delta y_{ij}^{(\ell)}\right), \tag{24}$$

where ρ_{ij} is the responsibility defined in Eq. 11. This formulation leaves the forward contribution unchanged, while scaling the gradient received by $B_j^{(\ell)}$ by ρ_{ij}.

For a novel sample $i \in \mathcal{N}_t$, the historical expert is kept in the forward computation but receives no update:

$$\widetilde{\Delta y}_{ij}^{(\ell)} = \mathrm{sg}\left(\Delta y_{ij}^{(\ell)}\right). \tag{25}$$

Thus, novel samples can still use previously learned experts during inference, but cannot overwrite their parameter.

Newly introduced experts. For each newly introduced expert $j > K_{t-1}$, both $A_j^{(\ell)}$ and $B_j^{(\ell)}$ are trainable. A novel sample $i \in \mathcal{N}_t$ uses the original expert contribution,

$$\widetilde{\Delta y}_{ij}^{(\ell)} = \Delta y_{ij}^{(\ell)}, \tag{26}$$

so both LoRA matrices are optimized normally. In contrast, for a reuse sample $i \in \mathcal{R}_t$, we apply

$$\widetilde{\Delta y}_{ij}^{(\ell)} = \mathrm{sg}\left(\Delta y_{ij}^{(\ell)}\right), \tag{27}$$

such that the newly introduced expert remains available in the forward composition but does not receive gradients from reusable samples.

Unified formulation. The above cases can be summarized using a coefficient:

$$\gamma_{ij} = \mathbb{I}[i \in \mathcal{R}_t,\, j \le K_{t-1}]\, \rho_{ij} + \mathbb{I}[i \in \mathcal{N}_t,\, j > K_{t-1}], \tag{28}$$

where $\mathbb{I}[\cdot]$ is the indicator function. The effective expert contribution can then be written uniformly as

$$\widetilde{\Delta y}_{ij}^{(\ell)} = \gamma_{ij} \Delta y_{ij}^{(\ell)} + (1 - \gamma_{ij})\, \mathrm{sg}\Big(\Delta y_{ij}^{(\ell)}\Big). \tag{29}$$

The stop-gradient operation preserves the numerical value of $\Delta y_{ij}^{(\ell)}$, Eq. 29 and does not change the forward computation in Eq 9, but it only controls the backward adaptation of experts determined by reuse and novel samples.

I Baselines

MoELoRA(Chen et al., 2024). To mitigate catastrophic forgetting across different tasks during continual instruction tuning, MoELoRA(Chen et al., 2024) replaces a single LoRA module with a fixed set of multiple LoRA experts. Then it uses a trainable gating function to assign different weights to the experts for each input. Therefore, different incoming samples can activate different combinations of experts, which encourages expert specialization. However, MoELoRA uses fixed experts, which may limit adaptation when new patterns emerge in the data stream.

SMoLoRA(Wang et al., 2025). Standard MoELoRA uses a single router to select LoRA blocks based on the input. However, visual understanding and instruction requirements may change in different ways across evolving tasks, so one router may not handle both well. Therefore, SMoLoRA(Wang et al., 2025) introduces two separate routing modules. The visual understanding (VU) router uses the average feature of the current instance to select LoRA blocks, while the instruction following (IF) router uses the embedding of the current task instruction. Then, SMoLoRA uses adaptive fusion to assign appropriate weights to the outputs of the two modules. However, SMoLoRA still relies on a predefined set of LoRA blocks. This fixed structure may not fully match the continuously changing demands of real-world data streams.

VariGrow(Ardywibowo et al., 2022). Ardywibowo et al. (2022) introduce VariGrow to handle continual learning without clear task boundaries or task-change information. For each incoming sample, it uses a novelty score to evaluate how familiar the sample is to existing experts. A low novelty score means that an expert is familiar with the sample, while a high score indicates that the sample is unfamiliar. Therefore, if the sample is not detected as novel, VariGrow will assign it to the existing expert with the lowest novelty score. Otherwise, it will create a new expert module to learn the new data. However, VariGrow assigns each sample to a single expert, which may be insufficient when a complex sample requires knowledge from multiple experts.

StrLoRA(Che et al., 2026a). In real-world streaming scenarios, each data chunk may contain different types of tasks, and their proportions can change continuously. To better adapt to such dynamic streaming data, Che et al. (2026a) introduce StrLoRA, a two-stage expert routing framework. By comparing textual instructions with visual features, they find that textual instructions provide a clearer signal for distinguishing task types. Therefore, StrLoRA first uses the instruction embedding to select a subset of relevant experts. Then it assigns weights to the selected experts for each visual token, allowing each token to rely on a different expert. In addition, to avoid large routing changes as the data stream evolves, StrLoRA uses a historical EMA routing reference to keep the current routing more stable. However, StrLoRA operates with a fixed expert pool, meaning that it can only select and weight existing experts rather than introduce new ones when new patterns emerge.