TRAINING-FREE CONTINUAL LEARNING FOR VISION LANGUAGE MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Continual learning (CL) for multimodal large language models is currently benchmarked with parameter-heavy mixture-of-experts and LoRA routing methods. A large part of the reported gains on popular CL suites comes from tasks being trivially separable, so a router only needs to identify the task, instead of transferring knowledge across tasks. To address this, we build a continual multimodal multiple-choice benchmark by splitting two standard evaluation suites: MMMU (6 course categories / 30 fine categories) and MMStar (6 course categories / 18 fine categories) into disjoint task streams. Following previous work, we quantify their separability with a nearest-prototype routing probe and demonstrate that our benchmark is not trivially separable. We also introduce COMEM (**Co**ntent-based **Mem**), a *training-free*, content-based k-nearest neighbor memory for a frozen Vision-Language Model (VLM). COMEM stores, per input example, the model's last-layer question representation as a key and question-conditioned embedding of the correct answer's content as a value. At test time, it interpolates the model's own answer choice distribution with a distribution obtained by matching each option to a similarity-weighted retrieved answer meaning. Because the memory is append-only and avoids modifying model weights, COMEM has essentially *zero* catastrophic forgetting by construction. Across all four evaluation streams, COMEM consistently improves over a frozen model near-zero backward transfer, as well as matches or outperforms the mixture-of-LoRA-experts current state-of-the-art method, all without updating any model weights.

1 INTRODUCTION

Multimodal large language models (MLLMs) have become remakrably capable, able to answer questions relating to almost any subject and are increasingly deployed as general-purpose assistants. This capability comes at a steep cost: training these large models requires vast amounts of data and compute. Although this cost is typically paid once, the demands on these models never stop growing as new topics, tasks, and data domains evolve constantly, and retraining these models to keep them up-to-date quickly becomes expensive. Because of this, there is a need to avoid a full retraining and update these models incrementally, introducing new data while preserving the knowledge it already has. The field of *continual learning* (CL) seeks to solve this problem, creating models that have both the plasticity to acquire new task information and the stability to retain all it has learned previously (McCloskey & Cohen, 1989). Currently, the dominant technique in multimodal CL is to keep the backbone frozen and attach small trainable adapters, such as mixtures of LoRA experts organized by a task router, that are grown or merged as tasks arrive (Chen et al., 2024a; Huai et al., 2025; Ge et al., 2025; Guo et al., 2025). These methods report strong numbers on continual multiple-choice and visual question-answering benchmarks.

However, we follow up on a previously raised concern with the standard evaluation practices for this task (Tran et al., 2026). These existing benchmarks are typically created by chaining together multiple distinct datasets (Srinivasan et al., 2022; Chen et al., 2024a; Zhao et al., 2025). For example, a multimodal continual learning stream may consist of the dataset sequence: VQAv2 (Jia et al., 2024), OCR-VQA(Mishra et al., 2019), TextVQA(Mishra et al., 2019), and ScienceQA(Lu et al., 2022). Datasets that are built for different purposes differ enormously in image domains, answer formats, and instruction styles. This has the unintended consequence of easy task separability. Just from the query alone, the task it belongs to because *trivially identifiable*, as frozen encoders can

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.

easily distinguish them in distinct clusters. When task identity is easy to recover, an expert-routing method can score well simply by training multiple experts and selecting the corresponding one for the task. This reduces the continual learning problem to simply task identification, and not of the knowledge accumulation and transfer between tasks that it should.

To address this issue, we propose a multiple-choice benchmark designed to better represent the multimodal continual learning task. Taking inspiration from traditional image classification continual learning evaluation methods Rebuffi et al. (2017), we create a task stream by splitting up existing datasets into disjoint sets of classes. We take two datasets, **MMMU** (Yue et al., 2024) and **MMStar** (Chen et al., 2024b), that were released as single-shot evaluation suites for MLLMs. We repurpose their existing category labels to form our task sequences, creating 2 coarse 6-task streams, and fine task streams consisting of 30 and 18 tasks, respectively. Because these task sequences are formed from within the same dataset, the individual tasks are semantically adjacent, and following previous work (Tran et al., 2026), we demonstrate that the task separability is not trivial as in the existing benchmark 1.

We also introduce CoMem, a training-free method that augments a *frozen* MLLM with an external key-value memory, avoiding weight updates. During adaptation, for each training example a single forward pass through the model is performed and two things are stored. The model's last-token hidden state is stored as a key, while the question and answer content embedding is stored as a value for the external memory. At inference, the model's own prediction is combined with any relevant stored answer meanings to form the final answer. We evaluate our method and the current state-of-the-art Huai et al. (2025) on our 4 continual learning streams. Notably, the design choices of our method result in approximately zero forgetting, and either match or outperform the previous work in accuracy.

In summary, our work makes the following contributions:

- We propose a continual multimodal multiple choice benchmark with 4 evaluation categories and demonstrate that it consists of tasks with non-trivial separability.
- We propose CoMem, a training-free solution based on a content-based retrieval memory with minimal forgetting.
- We evaluate our method and the previous SOTA on our proposed benchmark and show comparable or better results across all evaluation schemes.

2 Related Work

Continual learning Continual learning seeks to absorb a stream of tasks without catastrophic forgetting (McCloskey & Cohen, 1989). There are three main families. *Regularisation* methods, e.g. EWC (Kirkpatrick et al., 2016), penalise changes to parameters that are deemed important for previous tasks. *Rehearsal* methods such as experience replay and GEM (Lopez-Paz & Ranzato, 2017), store and train on past samples and/or embeddings to combat forgetting. *Architecture* methods expand or isolate parameters for each task. In the era of large models, these techniques are typically instantiated on top of parameter-efficient tuning such as LoRA (Hu et al., 2021), so the underlying backbone is never fully retrained. While our method is reminiscent of rehearsal methods by leveraging a memory storage, it is distinct from all three by not updating any model parameters.

Continual learning for vision-language and multimodal LLMs Extending continual learning to VLMs and multimodal LLMs (MLLMs) is an active, fast-growing area, now with dedicated surveys (Huo & Tang, 2025; Liu et al., 2026). The current dominant solution is parameter-based expert routing, where trainable LoRA or other adapter experts are attached to a frozen backbone, and a router is learned to select or merge them as tasks arrive. MoE-Adapters (Yu et al., 2024) pioneered mixture-of-experts adapters for continual CLIP. For instruction-tuned MLLMs, CL-MoE (Huai et al., 2025) (CVPR 2025) combines a dual task and instance level router with a dual-momentum expert-merging update. D-MoLE (Ge et al., 2025) (ICML 2025) dynamically allocates LoRA experts across layers under a parameter budget with a gradient-based inter-modal curriculum. HiDe-LLaVA (Guo et al., 2025) (ACL 2025) hierarchically decouples a task-specific top layer, routed by image/text anchor similarity, from a task-general fused base. While strong in the existing benchmarks, these methods share three properties that our method avoids. (i) They *train parameters* on every task, resulting in

forgetting. (ii) Each relies on an explicit anti-forgetting mechanism (momentum merging, budgeted allocation, or anchor-based expansion) to stay stable. (iii) Their routers depend, either implicitly or explicitly, on recovering *task or modality identity* from the input in order to select the corresponding expert. We use CL-MoE as the representative SOTA and evaluate it on our new benchmark under an identical scorer for comparison.

Multimodal continual learning benchmarks To evaluate performance on this task, benchmarks chain *heterogeneous task datasets*: CLiMB (Srinivasan et al., 2022) spans distinct vision and language tasks, and CoIN (Chen et al., 2024a) chains ten datasets across eight task types (VQAv2, GQA, TextVQA, OCR-VQA, ScienceQA, and others). MLLM-CL (Zhao et al., 2025) separates IID domain streams from non-IID ability streams. These suites have two main problems in evaluating continual learning. First, each of the tasks are so distinct in input distribution, format, and instruction style, that task identity is nearly free to recover, enabling an expert-routing method to score well by *identifying* the task and switching experts, losing the need for any knowledge to cross between tasks. From the analysis of CoIN, they attribute most of the observed forgetting to instruction-alignment drift rather than to the loss of reasoning knowledge (Chen et al., 2024a). Second, several of the previously used datasets are near-saturated for modern MLLMs, so they cannot push the boundary of continual learning (Guo et al., 2026). Even single-shot multimodal evaluation is known to be contaminated by image-independent shortcuts and data leakage (Chen et al., 2024b). To address this, we build our benchmark differently, rather than chaining unrelated suites, we follow traditional image-classification continual learning paradigms (Rebuffi et al., 2017) and split existing multicategory datasets (MMMU, MMStar) into semantically adjacent tasks. Following the format of the datasets, the benchmark consists of multiple-choice questions with image components, and are scored using log-likelihood among answer choices. This holds format, instruction style, and answer space constant across the stream, so the only variation is in the task content, isolating knowledge transfer from task and format identification. As in previous work Tran et al. (2026), we utilize a nearest-prototype separability probe to quantify the difference in difficulty between task identity recovery between our benchmark and existing suites. These results are visualized in Figure 1.

Retrieval Augmentation Retrieval-augmented models such as kNN-LM (Khandelwal et al., 2020) and RETRO (Borgeaud et al., 2022) pair a frozen model with a dictionary of key-value pairs. This is attractive for continual learning because the external storage avoids overwriting existing parameters. Yet retrieval is essentially unexplored for continual multimodal multiple choice, where the closest gradient-free relative is black-box continual learning of VLM's (Li et al., 2026). CoMem adapts the retrieval idea to this setting with three key distinctions. (i) It is *fully training-free*: there are no gradient updates or parameter changes, unlike every expert-routing method, which must deal with forgetting as a result of its continuously updating weights. (ii) Its memory is *append-only* across tasks, so forgetting is minimized by construction. (iii) the utilized storage requires *no task identity and no router*, using a single shared configuration and a growing memory. Where the current SOTA grows and routes experts by task, CoMem retrieves answer meaning by content. It is, to our knowledge, the first training-free, retrieval memory for continual multimodal multiple-choice, and a strong baseline that such methods should be required to beat.

3 A CONTINUAL MULTIPLE-CHOICE BENCHMARK

Our benchmark takes inspiration from classical continual learning evaluation schemes, specifically from class-incremental image classification. In this setting, datasets have their classes partitioned into disjoint subsets and are trained on in sequence (Rebuffi et al., 2017). Split-MNIST, Split-CIFAR-10/100, and Split-ImageNet are all examples of this, each being partitioned versions of their respective non-split counterparts (Deng, 2012; Krizhevsky, 2009; Deng et al., 2009). These tasks are related by design, sharing a domain, input format, and low-level statistics, making it difficult to identify which task a sample belongs to, even more so when the tasks are learned sequentially. In contrast, the construction of existing multimodal continual learning benchmarks results in extremely separable tasks. This is because they are comprised of chaining together multiple datasets, each corresponding to different tasks (Chen et al., 2024a; Zhao et al., 2025). Just as in classical continual learning, where task-incremental learning (when the task id is known at test time) is a significantly easier task than class-incremental learning (when no task id is provided), the high level of separability in multimodal CL has removed part of the difficulty from the task.

The two datasets chosen for our benchmark are MMMU (Yue et al., 2024) and MMStar (Chen et al., 2024b). Both datasets were released as single-shot evaluation datasets for MLLMs and contain image-based multiple choice questions. MMMU is a suite of college-level questions, each with one or more images. The dataset contains both open-ended and multiple-choice questions, spanning 30 subjects grouped into 6 disciplines (Art & Design, Business, Science, Health & Medicine, Humanities & Social Science, and Tech & Engineering). We utilize the questions in the MMMU test set, filtering out any open-ended questions and questions with multiple corresponding images, resulting in 9,873 single-image multiple-choice questions. MMStar is a curated suite of 18 tasks organized into 6 categories (Coarse Perception, Fine-grained Perception, Instance Reasoning, Logical Reasoning, Science & Technology, and Mathematics). As all of the MMStar questions are single-image and multiple-choice, we use all 1,500 samples in our benchmark.

For each dataset we have a coarse and fine task sequence, based on their predefined classes and categories, and use an 80/20 split for training and testing sets. Methods see tasks in a fixed order, and are unable to see the full training data of previously learned tasks. After finishing task k, we evaluate on the test sets of all tasks $0 \dots k$, filling an accuracy matrix $A \in \mathbb{R}^{T \times T}$ where $A_{i,k}$ is the accuracy on task i after learning task k. We report the *final accuracy* $\frac{1}{T}\sum_i A_{i,T}$ and *backward transfer* $\mathrm{BWT} = \frac{1}{T-1}\sum_{i<T-1}(A_{i,T-1} - A_{i,i})$ (Lopez-Paz & Ranzato, 2017). To match the model prediction to the multiple choice answer, we use option-letter log-likelihood. To do this, we use a single forward pass of the query, followed by an $\arg\max$ over the next-token logits of the option letters. This is done to avoid potential noise from free-form generation of the answer.

To compare the task separability between our benchmark and existing ones, we follow previous work and use a separability probe on the samples' embeddings (Tran et al., 2026). We represent each question by a frozen CLIP (Radford et al., 2021) vision feature, and a frozen LLaMA (Touvron et al., 2023) text feature. Each task is split in half, where one half is used to form per-task class-mean prototypes, and the other half is classified using the nearest cosine prototype. We calculate this classification routing accuracy on just image features, just text features, and on the concatenated features for our 4 evaluation sequences and compare with previous work in Figure 1. This probe shows a distinct gap between the separability of our benchmark and previous work Zhao et al. (2025), as the existing benchmark is perfectly separable with image embeddings alone, followed closely behind with a 97% text embedding routing accuracy. In contrast, our benchmark reaches only 77% and 71% on MMMU and 67% and 60% on MMStar using multimodal embeddings for their coarse and fine splits, respectively. In summary, our benchmark is more difficult on the basis of having tasks that are significantly more difficult to distinguish from eachother, and therefore more difficult for routing methods to select the corresponding trained expert for each question.

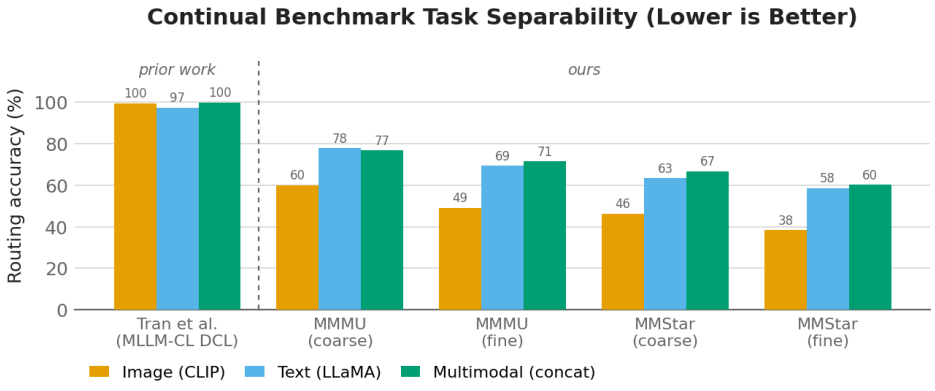


Figure 1: Task separability across benchmarks. We display the nearest-prototype routing accuracy (%) for recovering task identity from frozen CLIP image and LLaMA text features, where high accuracy means that tasks are more easily identifiable. Our benchmark suite is only moderately separable (between 60 and 77%) and leans on text, while the cross-domain stream of Zhao et al. (2025) is near-perfectly separable and leans on image embeddings.

4 COMEM: CONTENT-BASED kNN MEMORY

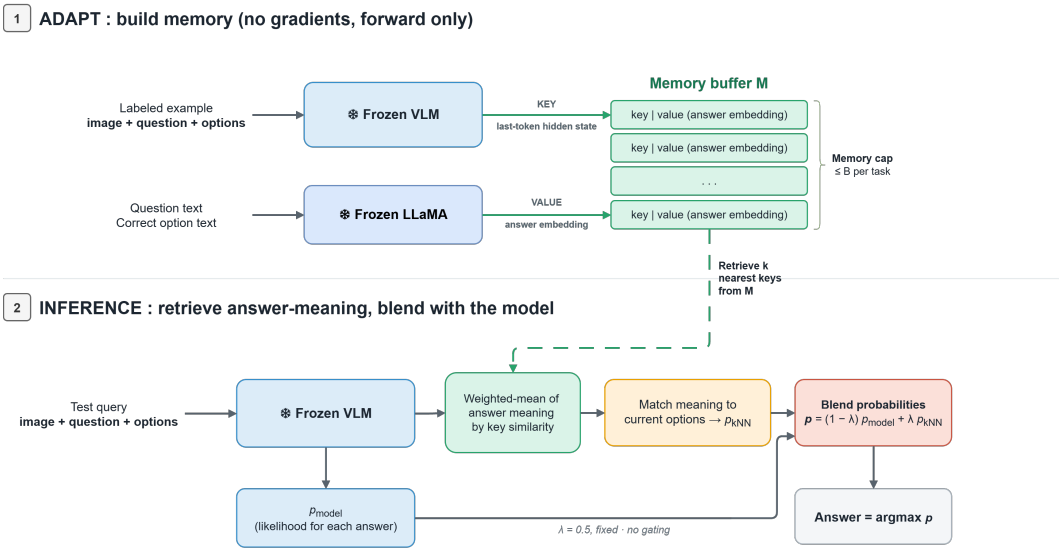


Figure 2: Task separability across benchmarks. We display the nearest-prototype routing accuracy (%) for recovering task identity from frozen CLIP image and LLaMA text features, where high accuracy means that tasks are more easily identifiable. Our benchmark suite is only moderately separable (between 60 and 77%) and leans on text, while the cross-domain stream of Zhao et al. (2025) is near-perfectly separable and leans on image embeddings.

COMEM is a multimodal continual learning method that augments a *frozen* MLLM with an external key-value memory and is training free. It has two phases, both of which are forward-only. During adaptation it writes one entry to memory per labeled example, and at inference it reads from that memory to assist the frozen model's answer. Because nothing is trained, there is no stability-plasticity trade-off to manage, resulting in minimal forgetting.

The retrieval key should capture *how the frozen model understands a question*, so that two questions it reasons about similarly become neighbors in memory. We take the model's last-token hidden state from a single multimodal forward over the image, question, and options. In a causal decoder this is the only position that has attended to the entire input, and it is the exact state from which the model reads off its answer, so retrieving on it amounts to retrieving on the model's own notion of question similarity. We write $\boldsymbol{k} \in \mathbb{R}^D$ for this L_2-normalised vector key, where the dimension D=4096.

For the corresponding value, storing the correct option's letter and voting on letters captures a positional prior rather than any knowledge, and it collapses the moment the options are reordered. We instead store the meaning of the correct answer by computing an embedding of the correct option's text. This makes retrieval invariant to option order and lets the memory represent notions such as "questions like this one tend to have answers about X", rather than "questions like this one tend to answer B." The content embedding is produced by the frozen model itself. We encode the option text and mean-pool a layer's hidden states, which contextualises subwords so that numbers and formulas ("1.2×10^3") get a meaningful vector instead of an average of digit embeddings. For some questions with minimal answers, such as single numbers, the answer content alone might not be enough. A bare option such as "67" is nearly meaningless out of context. To remedy this, we additionally compute a *question-conditioned* embedding. We combine one forward of the question together with all options, mean-pooling each option's own token span from that shared context, so the option is represented in the presence of its question. The *hybrid* value we use keeps both an isolated and a question-conditioned channel, combined at match time by a weight β. This question-conditioned embedding contributes positively for quantitative subjects with minimal answer context.

4.1 ADAPTATION

Adaptation is a forward-only sweep. For each labelled example we run the multimodal forward once, take $\boldsymbol{k}$ as above, form the value $\boldsymbol{v}$ as the hybrid content embedding of the *correct* option, and append the pair. The actual multiple-choice letter is only used to identify the correct answer, and is discarded after use. After tasks $0 \dots k$ the memory is $\mathcal{M} = \{(\boldsymbol{k}_n, \boldsymbol{v}_n)\}_{n=1}^{N}$. When a fixed storage budget is used we cap the memory at B entries per task by reservoir sampling. All key-value pairs are kept until a total of B is reached, then admit each new item with probability B/n and evict a uniformly-random slot.

4.2 INFERENCE

At inference the frozen model runs once on the test question, yielding both its own distribution p_{model} over the option letters and the query key $\boldsymbol{q}$. We retrieve the k nearest stored keys, take a similarity-weighted mean of their answer-meanings to form a single retrieved meaning $\bar{\boldsymbol{v}}$, and score each of the *current* question's options by how well its content matches $\bar{\boldsymbol{v}}$, giving a second distribution p_{kNN} that we linearly interpolate with the model's own:

$$p_{\text{model}} = \text{softmax}_{L \in \mathcal{L}}\big(\text{logit}(L)\big), \tag{1}$$

$$w_j = \text{softmax}\big(\text{sim}(\boldsymbol{q}, \boldsymbol{k}_j)/\tau\big) \text{ over top-}k \text{ neighbours}, \quad \bar{\boldsymbol{v}} = \textstyle\sum_j w_j \, \boldsymbol{v}_j, \tag{2}$$

$$p_{\text{kNN}} = \text{softmax}_{L \in \mathcal{L}}\big(\text{sim}(\text{emb}(L), \bar{\boldsymbol{v}})/\tau_c\big), \tag{3}$$

$$p = (1 - \lambda)\, p_{\text{model}} + \lambda\, p_{\text{kNN}}, \qquad \hat{L} = \arg\max_{L} p. \tag{4}$$

Here sim is cosine similarity, τ and τ_c are temperatures, and $\text{emb}(L)$ is the content embedding of the current option L. For the hybrid value the option-answer match is computed per channel and combined by β. We deliberately keep the mixing weight λ a single fixed constant. We have tested *gating* it per query on the model's confidence or on retrieval similarity, but found that such gating produces either negligible gains or harms performance, so the final method uses a constant λ and needs neither task identity nor a router at test time. We detail the gating experiments as part of the ablations in Section 6.

5 EXPERIMENTS

5.1 EXPERIMENTAL SETUP

We evaluate on the four streams of our benchmark: MMMU as coarse (6 tasks) and fine (30 tasks), MMStar as coarse (6 tasks) and fine (18 tasks), using a fixed $80/20$ per-task split with the seed 42, and use a common option-letter log-likelihood scorer.

All methods share a frozen LLaVA-v1.5-7B backbone (Liu et al., 2023), consisting of a CLIP ViT-L/14-336 vision tower, a LLaMA-7B language model, and an MLP projector. The pretrained checkpoint used was `liuhaotian/llava-v1.5-7b`. For the COMEM hyperparameters, we use k=16 neighbours, interpolation weight λ=0.5, the hybrid content value with isolated-channel weight β=0.25, retrieval and content softmax temperatures τ=τ_c=0.1, and no per-query gating. For the main results, we use an unbounded memory buffer. We run all experiments on a single 24 GB GPU.

For a lower bound we use the frozen LLaVA-v1.5-7B model and the aforementioned pretrained checkpoint in a zero shot fashion. For the oracle upper bound, we conduct a 1 epoch LoRA (Hu et al., 2021) finetuning across the entire shuffled dataset in a non-continual fashion, where all classes are visible at the same time during training. For LoRA baselines, we use rank-64 LoRA with α=128, dropout 0.05, and learning rate 5×10^{-5}. The current SOTA, CL-MoE (Huai et al., 2025), follows its published configuration: four LoRA experts with a dual router and momentum merging, trained one epoch per task and scored identically.

5.2 RESULTS

Our main result is that we do comparable or better than the current SOTA over our benchmark, all without updating any model weights. As shown in Table 5.2, COMEM beats CL-MoE by 1-3% in

all splits except MMMU (fine), where it lags behind by only .2%. In Table 5.2, we compare the Backward transfer (BWT) as a measure of forgetting. In addition to comparing with the previous SOTA, we use a sequential LoRA finetuning scheme as a naive baseline comparison, where the LoRA is continuously learned on all tasks in sequence with no safeguard for forgetting. As expected, the baseline shows signs of forgetting, even up to a drop of 11.2%, while CL-MoE shows significant forgetting on the coarse splits. CoMem, in contrast, only forgets slightly on MMStar (coarse), and has zero or positive BWT on all other splits.

Method	**MMMU** course (6 cat.)	**MMMU** fine (30 cat.)	**MMStar** course (6 cat.)	**MMStar** fine (18 cat.)
Frozen (zero-shot)	35.5	35.6	37.3	37.9
CL-MoE (SOTA)	33.1	**37.2**	38.7	38.1
CoMem (continual, ours)	**36.0**	37.0	**41.4**	**39.0**
LoRA-FT (upper bound)	48.6	40.4	41.0	41.5

Table 1: Final accuracy (%) on the four continual streams of our benchmark. Joint fine-tuning in a non-continual fashion (grey) serves as the oracle upper bound. Our method achieves comparable or superior results to the previous SOTA across all splits.

	MMMU		**MMStar**	
Method	coarse	fine	coarse	fine
LoRA-FT (sequential)	-1.4	$+0.0$	-11.2	-4.3
CL-MoE (SOTA)	-4.7	$+0.0$	-2.4	$+\mathbf{1.3}$
CoMem (ours)	$+\mathbf{0.0}$	$+\mathbf{0.2}$	$-\mathbf{0.4}$	$+0.8$

Table 2: Backward transfer (higher is better, 0 = no forgetting) on all benchmark splits. We use sequential LoRA-FT as a naive continual baseline comparison. Our method achieves little to no forgetting across all splits.

6 Ablation and analysis

In this section, we evaluate the impact of capping the retrieval memory, varying how many neighbors to retrieve, and how to weight the memory against the model. All values are the continual final accuracy under the configuration of Section 5.1.

6.1 Changing the buffer size

Table 6.3 varies the storage budget. We choose 200 as a per-task cap as it is typical in rehearsal-based continual learning work when building a rehearsal buffer. The main takeaway is that bounding the memory is nearly free: a fixed per-task reservoir cap of 200 entries costs at most 0.5% relative to unbounded, append-only memory, and on MMMU-coarse it is in fact slightly *higher*.The adapt sets in that split are large and highly redundant, so the random 15% subset loses nothing and denoises slightly. On both MMStar streams the capped and unbounded rows are *identical*. This is because the MMStar task has fewer than 200 eligible examples, so the cap never triggers and must return exactly the unbounded result. What does hurt is bounding the *total* budget rather than the per-task budget: a single global reservoir of 500 entries starves the many-task MMMU splits. 500 samples shared across 30 subjects is roughly 17 examples per task, dragging fine-grained MMMU back toward the frozen model. Across every budget, backward transfer stays at approximately zero, so the storage saving comes at no cost in forgetting.

6.2 Impact of per-query gating

Table 6.3 ablates the per-query gate, which is the rule that scales the memory weight λ for each individual question. Every gate that *down-weights* memory when the model looks confident hurts.

Hard abstention when the top option's probability exceeds a threshold (*abstain*) is middling, and the soft confidence gates (*conf*, *both*) are the worst of all, costing up to 3–4 points on MMStar. The reason is that the frozen model's confidence is poorly calibrated on these tasks. Many of its confident answers are wrong, and gating discards memory on exactly those queries, forfeiting the cases where retrieval would have corrected a confident mistake. Doing nothing and applying a constant λ (*none*) is best, tied by the similarity gate (*sim*), which multiplies λ by the top-neighbor cosine, because the nearest neighbor is almost always close, that multiplier is near one and *sim* reduces to *none* in practice, so the 0.1-point edge it shows on MMMU-fine is within reasonable noise. We accordingly use no gating, which is also the simplest choice and removes any need for a task-specific threshold.

6.3 Changing the number of neighbors

Table 6.3 varies the number of retrieved neighbors k. For this ablation, we show results on the MMMU (fine) split. The accuracy rises from 36.7% at k=4 to a peak of 37.6% at k=16, then falls, and for very large k returns to the frozen level (35.6% at k=128). The shape is intuitive. Too few neighbor give a noisy, high-variance estimate of the retrieved answer meaning, while too many pull in distant, unrelated questions whose answers dilute the similarity-weighted mean until it carries no task-specific signal. At the limit, the retrieved meaning is just the average answer of the whole memory, which is uninformative. The peak is broad, with any $k \in \{8, 16, 32\}$ within a point of the best, so the method is insensitive to the exact value. For the main experiments, we fix k=16 throughout.

Memory Budget	**MMMU fine**	**MMMU coarse**	**MMStar coarse**	**MMStar fine**
Unbounded (append-only)	36.8	35.5	39.4	37.2
$\leq$ 200 / task (reservoir)	36.3	35.8	39.4	37.2
500 total (global reservoir)	35.8	35.5	39.0	37.1

Table 3: Ablation: Changing the buffer size. A fixed per-task reservoir cap costs at most 0.5 points versus unbounded memory (and raises MMMU-coarse), whereas a small *global* cap starves the many-task MMMU splits. BWT stays ≈ 0 for all budgets.

Gate	**MMMU coarse**	**MMMU fine**	**MMStar coarse**	**MMStar fine**
none	36.0	37.0	41.4	40.6
sim	36.0	37.1	41.4	40.6
abstain	36.0	37.2	40.7	39.0
conf	35.5	36.8	39.4	37.2
both	35.5	36.6	39.4	37.2

Table 4: Ablation: Impact of per-query gating. Every gate that down-weights memory when the model looks confident (conf/abstain/both) hurts. No gating, or the near-identical similarity gate performs best.

k	4	8	**16**	32	64	128	256
Accuracy	36.7	36.7	**37.6**	36.6	36.5	35.6	35.9

Table 5: Ablation: Changing the number of neighbors k. We report results are from the MMMU (fine) split. Accuracy peaks at k=16. Too few neighbors are not enough and very large k washes the retrieval out to the frozen level.

6.4 Storage and compute

Each memory entry consists of a handful of half/single-precision vectors: a 4096-d key plus a two-channel 4096-d content value. This is approximately 24 KB per entry in `fp16` (≈ 48 KB in `fp32`),

and *no raw image or text*. Under a fixed budget of 200 entries per task, the total memory is small and grows only linearly in the number of tasks: on our streams the capped memory holds 5,519 entries for the 30-task MMMU split ($\approx$ 130 MB `fp16`) and $\approx$ 1,200 entries for each six-task split ($\approx$ 28 MB). This is competitive with, and often smaller than, a rehearsal buffer of the same example count, which must store raw images ($\sim$30–80 KB each as JPEG, $\sim$340 KB as a raw preprocessed tensor) *and* take gradient steps over them at every subsequent task; CoMem instead does a single forward per stored example, once, and thereafter only requires cosine look-ups. As previously mentioned, the capped memory costs little: relative to unbounded memory under the identical configuration, the 200-per-task budget loses at most 0.5% (and evicts up to 85% of MMMU-discipline examples with no loss), while backward transfer stays $\approx$ 0, so the fixed-storage variant preserves both the accuracy while minimizing forgetting.

7 Conclusion

In this paper we explore the usefulness of existing benchmarks in continual learning for VLMs. To further research in this area, we built a controlled benchmark by separating two standard evaluation suites, MMMU and MMStar, each into disjoint continual streams at a coarse and a fine granularity, under one fixed multiple-choice format and a common option-log-likelihood scorer. Our separability probe shows our intra-suite streams are not trivially separable with a routing accuracy between 60 and 77%, far below the essentially 100% separability of the existing cross-domain benchmarks.

With this benchmark we introduced CoMem , a training-free, content-based, k-nearest-neighbour memory over a frozen MLLM. CoMem stores, per labelled example, a retrieval key and an embedding of the correct answer's *content*, and at inference blends a retrieved answer-meaning with the model's own option distribution. It touches no weights, needs no task identity or router at test time, forgets essentially nothing by construction, and under a fixed per-task budget, bounds its storage to a few tens of kilobytes per example with no raw data retained. Across all four streams CoMem improves over the frozen model with backward transfer approximately 0 at worst, and matches or exceeds the mixture-of-LoRA-experts current SOTA system that trains and forgets. While joint fine-tuning remains the ceiling, a transparent, non-forgetting, fixed-storage memory is a strong baseline that future continual methods should be required to beat. We hope the benchmark, separability analysis, and this baseline push multimodal continual-learning evaluation toward rewarding genuine transfer rather than just task identification.

AI use statement

(This section is **required** and does not count toward the page limit.)

In this work, we used generative AI tools for no tasks with required disclosure. We have not used generative AI tools for the development of theoretical models or conceptual frameworks, proposal or refinement of hypotheses, the design or to provide feedback on research methodology or experiments, to implement methods, to assist with translation, to clean and reformat dataset, to support qualitative and thematic data analysis, to interpret results, and generation of synthetic data, formulation and proof of mathematical claims are not applicable to this work.

References

Sebastian Borgeaud, Arthur Mensch, Jordan Hoffmann, Trevor Cai, Eliza Rutherford, Katie Millican, George van den Driessche, Jean-Baptiste Lespiau, Bogdan Damoc, Aidan Clark, Diego de Las Casas, Aurelia Guy, Jacob Menick, Roman Ring, Tom Hennigan, Saffron Huang, Loren Maggiore, Chris Jones, Albin Cassirer, Andy Brock, Michela Paganini, Geoffrey Irving, Oriol Vinyals, Simon Osindero, Karen Simonyan, Jack W. Rae, Erich Elsen, and Laurent Sifre. Improving language models by retrieving from trillions of tokens, 2022. URL `https://arxiv.org/abs/2112.04426`.

Cheng Chen, Junchen Zhu, Xu Luo, Hengtao Shen, Lianli Gao, and Jingkuan Song. Coin: A benchmark of continual instruction tuning for multimodel large language model, 2024a. URL `https://arxiv.org/abs/2403.08350`.

Lin Chen, Jinsong Li, Xiaoyi Dong, Pan Zhang, Yuhang Zang, Zehui Chen, Haodong Duan, Jiaqi Wang, Yu Qiao, Dahua Lin, and Feng Zhao. Are we on the right way for evaluating large vision-language models?, 2024b. URL https://arxiv.org/abs/2403.20330.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 248–255, 2009. doi: 10.1109/CVPR.2009.5206848.

Li Deng. The mnist database of handwritten digit images for machine learning research [best of the web]. *IEEE Signal Processing Magazine*, 29(6):141–142, 2012. doi: 10.1109/MSP.2012.2211477.

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, 2025. URL https://arxiv.org/abs/2506.11672.

Haiyang Guo, Fanhu Zeng, Ziwei Xiang, Fei Zhu, Da-Han Wang, Xu-Yao Zhang, and Cheng-Lin Liu. Hide-llava: Hierarchical decoupling for continual instruction tuning of multimodal large language model, 2025. URL https://arxiv.org/abs/2503.12941.

Haiyun Guo, Zhiyan Hou, Yandu Sun, Jinghan He, Yu Chen, Yuzhe Zhou, Yuheng Jia, Jinqiao Wang, and Tat-Seng Chua. Mllm-ctbench: A benchmark for continual instruction tuning with reasoning process diagnosis, 2026. URL https://arxiv.org/abs/2508.08275.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models, 2021. URL https://arxiv.org/abs/2106.09685.

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, 2025. URL https://arxiv.org/abs/2503.00413.

Yukang Huo and Hao Tang. When continue learning meets multimodal large language model: A survey, 2025. URL https://arxiv.org/abs/2503.01887.

Ziheng Jia, Zicheng Zhang, Jiaying Qian, Haoning Wu, Wei Sun, Chunyi Li, Xiaohong Liu, Weisi Lin, Guangtao Zhai, and Xiongkuo Min. Vqa^2: Visual question answering for video quality assessment, 2024. URL https://arxiv.org/abs/2411.03795.

Urvashi Khandelwal, Omer Levy, Dan Jurafsky, Luke Zettlemoyer, and Mike Lewis. Generalization through memorization: Nearest neighbor language models, 2020. URL https://arxiv.org/abs/1911.00172.

James Kirkpatrick, Razvan Pascanu, Neil C. Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A. Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, Demis Hassabis, Claudia Clopath, Dharshan Kumaran, and Raia Hadsell. Overcoming catastrophic forgetting in neural networks. *CoRR*, abs/1612.00796, 2016. URL http://arxiv.org/abs/1612.00796.

Alex Krizhevsky. Learning multiple layers of features from tiny images. 2009. URL https://api.semanticscholar.org/CorpusID:18268744.

Yuting Li, Weihang Fang, Haoyuan Gao, Linghe Kong, Yexin Li, Lichao Sun, and Weiran Huang. Black-box continual learning for vision-language models, 2026. URL https://arxiv.org/abs/2606.22999.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning, 2023. URL https://arxiv.org/abs/2304.08485.

Yuyang Liu, Qiuhe Hong, Linlan Huang, Alexandra Gomez-Villa, Dipam Goswami, Tiantian Peng, Xialei Liu, Joost van de Weijer, and Yonghong Tian. Continual learning for vlms: A survey and taxonomy beyond forgetting, 2026. URL https://arxiv.org/abs/2508.04227.

David Lopez-Paz and Marc'Aurelio Ranzato. Gradient episodic memory for continuum learning. *CoRR*, abs/1706.08840, 2017. URL http://arxiv.org/abs/1706.08840.

Pan Lu, Swaroop Mishra, Tony Xia, Liang Qiu, Kai-Wei Chang, Song-Chun Zhu, Oyvind Tafjord, Peter Clark, and Ashwin Kalyan. Learn to explain: Multimodal reasoning via thought chains for science question answering, 2022. URL https://arxiv.org/abs/2209.09513.

Michael McCloskey and Neal J. Cohen. Catastrophic interference in connectionist networks: The sequential learning problem. volume 24 of *Psychology of Learning and Motivation*, pp. 109–165. Academic Press, 1989. doi: https://doi.org/10.1016/S0079-7421(08)60536-8. URL https://www.sciencedirect.com/science/article/pii/S0079742108605368.

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, 2019. doi: 10.1109/ICDAR.2019.00156.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision, 2021. URL https://arxiv.org/abs/2103.00020.

Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H. Lampert. icarl: Incremental classifier and representation learning, 2017. URL https://arxiv.org/abs/1611.07725.

Tejas Srinivasan, Ting-Yun Chang, Leticia Leonor Pinto Alva, Georgios Chochlakis, Mohammad Rostami, and Jesse Thomason. Climb: A continual learning benchmark for vision-and-language tasks, 2022. URL https://arxiv.org/abs/2206.09059.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. Llama: Open and efficient foundation language models, 2023. URL https://arxiv.org/abs/2302.13971.

Van-Tuan Tran, Shruthi Gowda, Merim Dzaferagic, and Marco Ruffini. Is our benchmark enough? an analysis of continual learning for mllms, 2026. URL https://arxiv.org/abs/2606.20961.

Jiazuo Yu, Yunzhi Zhuge, Lu Zhang, Ping Hu, Dong Wang, Huchuan Lu, and You He. Boosting continual learning of vision-language models via mixture-of-experts adapters, 2024. URL https://arxiv.org/abs/2403.11549.

Xiang Yue, Yuansheng Ni, Kai Zhang, Tianyu Zheng, Ruoqi Liu, Ge Zhang, Samuel Stevens, Dongfu Jiang, Weiming Ren, Yuxuan Sun, Cong Wei, Botao Yu, Ruibin Yuan, Renliang Sun, Ming Yin, Boyuan Zheng, Zhenzhu Yang, Yibo Liu, Wenhao Huang, Huan Sun, Yu Su, and Wenhu Chen. Mmmu: A massive multi-discipline multimodal understanding and reasoning benchmark for expert agi, 2024. URL https://arxiv.org/abs/2311.16502.

Hongbo Zhao, Fei Zhu, Haiyang Guo, Meng Wang, Rundong Wang, Gaofeng Meng, and Zhaoxiang Zhang. Mllm-cl: Continual learning for multimodal large language models, 2025. URL https://arxiv.org/abs/2506.05453.

A Appendix

You may include other additional sections here.