ReToken: One Token to Improve Vision–Language Models for Visual Retrieval

Yao Xiao[1] **Reuben Tan**[2] **Zhen Zhu**[1,3]* **Yuqun Wu**[1]

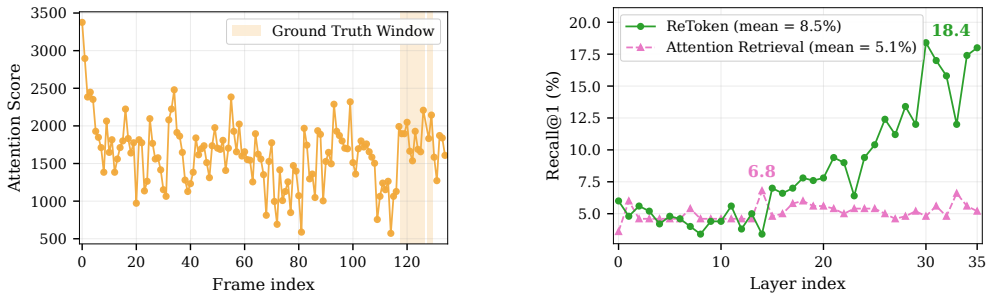
Jianfeng Gao[2] **Derek Hoiem**[1]

[1]University of Illinois at Urbana-Champaign, [2]Microsoft Research, [3]Google DeepMind

[1]{yaox11, dhoiem}@illinois.edu

Abstract

Long visual context poses a challenge for vision-language models: performance degrades as the number of distractors grows, and processing all tokens at once is computationally infeasible under GPU memory constraints. We present RETOKEN, a single learnable embedding trained as an explicit retrieval target that selects a sparse set of query-relevant visual tokens from a pre-filled visual KV cache. Trained on only a small image-QA dataset, RETOKEN yields consistent gains across image and video benchmarks: on Visual Haystacks it improves Qwen3VL-8B by 13.4 points and InternVL3.5 by 12.4 points (>20% relative), and on LVBench it transfers zero-shot to long video for an 8.0-point gain with Qwen3VL-8B. Thanks to its lightweight design, both training and long-video inference fit on a single H100. Code is available at: https://github.com/avaxiao/ReToken.



(a) Attention score fails to distinguish relevant frames from distractors.

(b) ReToken achieves more than 3x recall in the last layer. Layer-wise VLM Recall@1 on $\text{QAEgo4D}_{\text{Test-MC}}$

Figure 1: **Attention-based retrieval mechanisms are not effective for retrieving relevant visual information in VLMs**. (a) shows attention scores from question tokens (as queries) to frame tokens (as keys), across frames in one video; (b) shows whether the top-1 retrieved frame falls within the ground-truth time window (240 candidate frames per video on average, sampled at 0.5 FPS).

1 Introduction

Long visual contexts, e.g. from image collections or hour-long videos, are now within the input range of vision-language models (VLMs) [26, 38, 4, 9]. However, VLMs have difficulty answering

*This work was done while the author was at UIUC (currently at Google DeepMind).

questions from long visual contexts when only a small subset of images or frames is relevant to the prompt [45], and sometimes processing the full context at once is computationally infeasible under GPU memory constraints. So, handling long visual input reduces to a retrieval problem: selecting a small subset of frames or tokens from which the model can produce a correct answer. When processing text, the attention signals produced by large language models (LLMs) can be selective with respect to the input context, thanks to extensive long-context training [47, 16]. For VLMs, we find that the analogous signal is unreliable: attention between text and visual features is weakly correlated with relevance, as shown in Fig. 1a, and the attention-based retriever achieves only 5.1% average recall@1 across layers on QAEgo4D$_{\text{Test-MC}}$ for Qwen3VL-8B (Fig. 1b).

Looking at what does work, we find a striking asymmetry: matching a precise target phrase against the average visual *value* projections, rather than against the keys, increases recall@1 from 65.7 to 78.0 on Qwen3VL and from 78.8 to 83.8 on InternVL3.5 in a controlled two-image setting (Tab. 1). Values carry the content that is actually propagated through attention, and they appear to provide a better space for text-based visual retrieval. However, value–value pooling does not universally outperform query–key scoring for arbitrary retrieval text. Simply averaging over all question tokens introduces noise, reversing the advantage of the value space.

Building on this finding, we propose RETOKEN, a single learnable embedding that is appended to the question and trained explicitly as a retrieval target. RETOKEN scores each frame by the cosine similarity between its projected embedding and the frame's mean value vector at the final layer, and is supervised with a class-balanced binary cross-entropy loss against ground-truth relevance labels. The token and a single projection matrix are the only added parameters; the VLM is frozen by default.

Despite its minimal footprint, RETOKEN yields consistent gains across image and video benchmarks. On Visual Haystacks [45], it improves Qwen3VL-8B [4] by 13.4 points and InternVL3.5 [42] by 12.4 points, corresponding to over 20% relative gain. More notably, although trained only on multi-image QA, **RETOKEN generalizes to improve performance on long-video understanding**: it transfers zero-shot to long video, yielding an 8.0-point improvement on LVBench [41] with Qwen3VL-8B, where the average video length exceeds an hour.

Our contributions are:

- We identify that retrieval scores computed in the *value* space, rather than the conventional query-key space, provide a substantially stronger signal for retrieving visual information.
- We introduce RETOKEN, a lightweight learnable retrieval token that enables pretrained VLMs to identify relevant visual information more effectively.
- We show that RETOKEN trained only on multi-image QA transfers zero-shot to long-video benchmarks, suggesting a practical path toward scalable long-context multimodal reasoning.

2 Related Work

Visual Retrieval. The dominant approach to visual retrieval is to train image-text embedding models that align visual and language feature spaces, either with dual encoders such as CLIP [31], Perception Encoder [5], and SigLIP2 [39], or by adapting a VLM into a universal embedder, as in E5-V [18] and LamRA [27]. While effective, all of these operate as external retrievers separate from the VLM: relevant images must first be selected by the retriever and then re-encoded by the VLM for answer generation. A related family in long-video QA chains a separate localizer with an answerer model: SeViLA [49] repurposes BLIP-2 [22] as both a keyframe localizer and an answerer, while VideoAgent [43] and VideoTree [44] build agentic pipelines that iteratively retrieve and caption keyframes for an LLM. A second line of work, originating in long-context language modeling, instead treats the model's own attention scores as a retrieval signal, unifying retrieval and generation within a single forward pass. InfLLM [46] and EM-LLM [14] demonstrate this effectively in NLP by selecting key-value blocks based on query-to-key attention, and ReKV [13] transfers the mechanism to the visual domain without modification. We diagnose the limitations of attention-based retrieval in VLMs and propose RETOKEN, which performs retrieval in the value space via a learnable token.

Long Context Understanding. When handling long visual contexts, prior work follows three main directions. Memory-based methods such as MovieChat [35] and MA-LMM [17] maintain a fixed-size memory bank over streaming input, merging or evicting older content to bound state. Token compression methods such as Chat-UniVi [19], LLaMA-VID [23], LongVU [32], and Video-XL [33]

prune, merge, or summarize visual tokens before the LLM to shorten the prefix and reduce attention and KV-cache cost. Both families bound the visual context independently of the query. Retrieval-based methods instead defer context selection to inference time, picking a query-relevant subset to preserve fine-grained evidence: Goldfish [3] chunks the video into clips and retrieves the top-K by caption-query similarity; Video-RAG [28] augments retrieval with visually-aligned auxiliary text from ASR, OCR, and object detectors; and DrVideo [29] converts the video into a long document and retrieves question-relevant passages for the LLM. End-task accuracy in this family, however, is bottlenecked by the retriever rather than the LLM, and our work targets this bottleneck directly.

Learnable Tokens for Visual Aggregation. RETOKEN is most directly related to methods that introduce learnable tokens into a VLM to aggregate or retrieve visual content. The Q-Former in BLIP-2 [22] and the Perceiver Resampler in Flamingo [1] use a small set of learnable queries to compress variable-length visual features into a fixed-size representation for a frozen LLM, and InstructBLIP [11] extends Q-Former to be text-conditioned. These modules are trained jointly with vision-language alignment as part of the bridge between encoder and LLM, and produce many tokens (typically 32 or 64) intended to carry the visual content forward into generation. SPRING [51] prepends pluggable soft-prompt tokens to externally retrieved passages to help a frozen text-only LLM consume them. The visual summarization token in Video-XL [33] similarly compresses a chunk of visual KVs into a single token. RETOKEN differs along three axes: (i) *operational mechanism*. Q-Formers and prompt tuning use learnable tokens as static, query-agnostic input conditions. In contrast, the ReToken is generated dynamically. A first-pass placeholder produces an output token that retrieves visual KV contexts relevant to the query. (ii) *supervision*. Q-Formers are trained as visual compressors under alignment or generation losses, while RETOKEN is supervised by an explicit retrieval loss scored directly against final-layer value vectors. (iii) *capacity*. Our method needs only a single token rather than a set of 32–64. The second half of the contribution is diagnostic: attention (query–key) scores are unreliable for visual retrieval in pretrained VLMs, while the value space carries a much stronger signal.

3 Approach

In this section, we begin by analyzing the limitations of attention-based retrieval methods for identifying relevant visual tokens in Sec. 3.1. Motivated by these insights, we then discuss our proposed RETOKEN approach in Sec. 3.2, which uses a learnable token embedding to help compute more informative retrieval scores over relevant visual tokens.

3.1 Attention-Based Retrieval

In our empirical setting, we consider a VLM based on a decoder-only LM comprising N transformer layers $\{L_1, \ldots, L_N\}$. Given input question tokens $\mathbf{T}_t$ and visual tokens $\mathbf{I}_v$, the VLM autoregressively generates the response $\mathbf{y}$ based on the conditional probability $p(\mathbf{y} \mid \mathbf{I}_v, \mathbf{T}_t)$. For multi-image and video inputs, the visual tokens are partitioned into F frames, $\mathbf{I}_v = \{\mathbf{I}_v^{(f)}\}_{f=1}^{F}$, with each frame contributing M/F tokens (where M the total number of visual tokens). For long videos, M can be prohibitively large, rendering full-context inference infeasible. Thus, our goal is to retrieve a subset of the K most relevant frames, where $K \ll F$, whose tokens suffice to answer the question.

Attention Scores as a Retrieval Signal. At each layer $l \in \{1, 2, \ldots, N\}$, the VLM computes attention scores $\mathbf{A}^{(l)} = \mathbf{Q}^{(l)}\mathbf{K}^{(l)\top}/\sqrt{d}$, which indicate how much each token contributes to the final response. Tokens with higher attention contributions can be interpreted as being more relevant to the question. For our analysis, we use the state-of-the-art ReKV [13] approach that directly uses attention score as the retrieval score and aggregates it at the frame level. At the $l-$th layer, we compute the mean key vector of each frame's visual tokens and the mean query vector over the question tokens as:

$$\bar{\mathbf{k}}_f^{(l)} = \frac{1}{|\mathbf{I}_v^{(f)}|} \sum_{i \in \mathbf{I}_v^{(f)}} \mathbf{k}_i^{(l)}, \qquad \bar{\mathbf{q}}^{(l)} = \frac{1}{E} \sum_{t=1}^{E} \mathbf{q}_t^{(l)} \tag{1}$$

where E denotes the number of total question tokens. Then, the frame-level retrieval score $\mathbf{s}_f^{(l)}$ and top-K frame selection are computed as:

$$\mathbf{s}_f^{(l)} = \bar{\mathbf{q}}^{(l)\top}\bar{\mathbf{k}}_f^{(l)}, \qquad \mathcal{S}_K^{(l)} = \mathrm{TopK}(\mathbf{s}_f^{(l)}) \tag{2}$$

Figure 2: **Retrieval Example**. Question/Sentence: "For the image with a cow, is there a truck?". Target phrase: "a cow".

where $\mathcal{S}_K^{(l)}$ denotes the K frames most relevant to the question at layer l. Given the per-layer retrieval sets $\{\mathcal{S}_K^{(l)}\}_{l=1}^N$, the model re-runs generation through the frozen VLM with *layer-wise* sparse attention: at each layer l, the question and answer tokens attend only to the visual tokens belonging to $\mathcal{S}_K^{(l)}$. The reduced visual context at layer l is therefore $[\mathbf{I}_v[\mathcal{S}_K^{(l)}]]$, and the answer is decoded autoregressively over this layer-dependent context $[\mathbf{I}_v[\mathcal{S}_K^{(l)}], \mathbf{T}_t]$.

Limitations of Attention-Based Retrieval. While attention-based retrieval is intuitive, two structural issues motivate our approach. First, attention is trained for next-token prediction rather than retrieval, so high-attention tokens are not guaranteed to correspond to query-relevant content. This mismatch is exacerbated by the composition of typical VLM training data: the input images or videos are almost always fully relevant to the question, so the model is never required to *select* among visual inputs. The consequences are visible in both image and video settings: on Visual Haystacks [45] the layer-wise $\bar{\mathbf{q}}^\top \bar{\mathbf{k}}_f$ retriever lands at 63.3% recall@1 even in the simplest two-image case (Tab. 1), and on long-video QAEgo4D$_{\text{Test-MC}}$ it averages only 5.1% recall@1 across layers (Fig. 1b). Second, averaging the question tokens to form a single query is itself a heuristic, and the resulting rankings shift with the choice of retrieval text. As shown in Tab. 1, replacing the full question sentence with a target phrase that directly names the entity of interest improves Recall@1. Fig. 2 illustrates a question sentence and its target phrase.

Why Values Carry Retrieval Signal. A more reliable signal sits in the value projections. Within a transformer attention layer, the value of a token carries the content propagated to any token that attends to it, while the query-key inner product only determines how that content is aggregated. Pooling the value projections within each frame therefore yields a representation of the content the frame contributes to attending tokens, making value features more sensitive to the retrieval text than key features. Tab. 1 confirms this consistently across both backbones: with a precise target phrase that names the entity of interest, value-space pooling identifies the ground-truth image at 78.0% recall@1 vs. 65.7% for the corresponding query-key score on Qwen3VL, and at 83.8% vs. 78.8% on InternVL3.5. This mirrors observations in visual segmentation, where value features are reported to be more informative while query-key features can be replaced by alternative aggregation signals [50, 40, 21]; TextRegion [48] in particular shows that, in image-text models, value features in the final attention block are rich in visual-language semantics, whereas attention weights primarily serve as an aggregation mechanism. This sensitivity cuts both ways, however: with the full question sentence, averaging mixes in many uninformative tokens, and the resulting noisy query erases or even reverses the value-space advantage. The value space thus rewards a precise retrieval target and penalizes a noisy one.

Table 1: **Value × Value is more informative, but it is sensitive to the input**. Recall@1 with 2 input images, retrieving 1 image based on the retrieval score.

Retrieval Text	$\bar{\mathbf{q}}^\top \bar{\mathbf{k}}_f$	$\bar{\mathbf{v}}^\top \bar{\mathbf{v}}_f$	Qwen3VL	InternVL3.5
Sentence	✓		63.3	78.5
		✓	62.6	75.6
Target Phrase	✓		65.7	78.8
		✓	**78.0**	**83.8**

3.2 ReToken: One Token for Visual Retrieval

To improve retrieval, we introduce ReToken, a single learnable embedding $\mathbf{X}_r \in \mathbb{R}^d$ that is appended to the question and trained explicitly as a retrieval target. ReToken addresses the

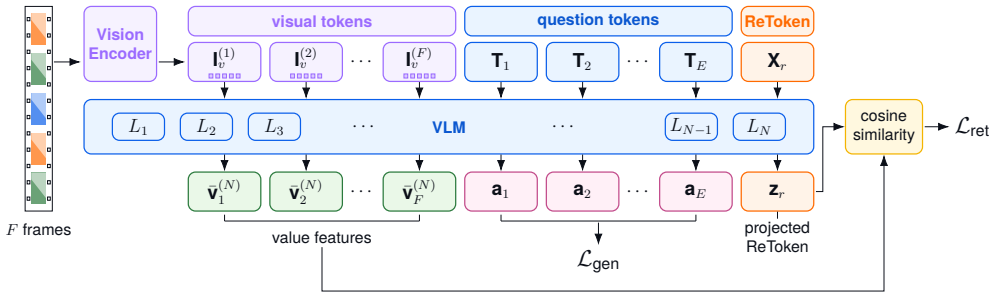


Figure 3: **Training Pipeline.** The visual tokens, question tokens, and a learnable embedding (RETOKEN) are fed into the LLM decoder. The RETOKEN output from the final layer serves as a retrieval query, scoring each frame based on its cosine similarity to the frame's pooled value features. The resulting scores are supervised using ground-truth frame-relevance labels and a class-balanced binary cross-entropy loss. By default, we train RETOKEN while keeping the VLM frozen. The generation loss is used only during VLM partial fine-tuning setting.

rephrasing instability by replacing the question-averaged query with a token *learned* from data, and strengthens the retrieval score by computing in the value space rather than the query–key space.

Retrieval Score. Given an input sequence with visual tokens $\mathbf{I}_v$, question tokens $\mathbf{T}_t$, and the appended retrieval token $\mathbf{X}_r$, we compute retrieval scores at the final layer L_N via a lightweight projection $\mathbf{W}_r \in \mathbb{R}^{d\times d}$, which together with $\mathbf{X}_r$ constitutes the only added parameters. The retrieval score for the f-th frame is the cosine similarity between the projected embedding and the frame's mean value vector $\bar{\mathbf{v}}_f^{(N)}$,

$$\bar{\mathbf{v}}_f^{(N)} = \frac{1}{|\mathbf{I}_v^{(f)}|}\sum_{i\in\mathbf{I}_v^{(f)}} \mathbf{v}_i^{(N)}, \quad \mathbf{Z}_r = \mathbf{W}_r\,\mathbf{X}_r^{(N)}, \quad s_f^{(N)} = \cos\Big(\mathbf{Z}_r,\ \bar{\mathbf{v}}_f^{(N)}\Big) \tag{3}$$

For inference, we compute $\mathcal{S}_K^{(N)} = \mathrm{TopK}(s_f^{(N)})$ once at the final layer L_N and broadcast it to all layers, rather than maintaining per-layer subsets $\{\mathcal{S}_K^{(l)}\}$. We do not apply this at training time, since our training data is short-context and retrieval is unnecessary.

Training. We train RETOKEN with the VLM kept frozen by default; only the retrieval token $\mathbf{X}_r$ and a single final-layer projection $\mathbf{W}_r$ are updated. We supervise these parameters with a *retrieval loss* $\mathcal{L}_{\text{ret}}$ that compares the final-layer retrieval scores against ground-truth relevance labels $y_f \in \{0,1\}$.

Let $\mathcal{F}^+ = \{f : y_f = 1\}$ and $\mathcal{F}^- = \{f : y_f = 0\}$ denote the sets of relevant and irrelevant images, respectively. To prevent the loss from being dominated by the irrelevant images, we apply a class-balanced binary cross-entropy in which the positive and negative terms are averaged separately,

$$\mathcal{L}_{\text{ret}} = -\frac{1}{|\mathcal{F}^+|}\sum_{f\in\mathcal{F}^+}\log\sigma(\tau s_f^{(N)}) \;-\; \frac{1}{|\mathcal{F}^-|}\sum_{f\in\mathcal{F}^-}\log\sigma(-\tau s_f^{(N)}) \tag{4}$$

where $s_f^{(N)}$ is the retrieval score for image f at the final layer L_N, $\sigma(\cdot)$ is the sigmoid, and τ is a learnable logit scale parameter.

We additionally explore a *partial-tuning* variant in which the early layers of the VLM are tuned to give the visual representation more flexibility. In this setting, training is driven by two complementary losses, illustrated in Fig. 3. In addition to the retrieval loss above, we use a *generation loss* $\mathcal{L}_{\text{gen}}$, the standard next-token prediction loss on the answer, which preserves the model's question-answering ability and prevents the unfrozen layers from drifting under $\mathcal{L}_{\text{ret}}$ alone. The total loss is

$$\mathcal{L} = \mathcal{L}_{\text{ret}} + \lambda\,\mathcal{L}_{\text{gen}} \tag{5}$$

Inference. At test time, we use a two-pass *retrieve-then-answer* pipeline (Fig. 4). The video is first encoded once, and each question then retrieves a subset of visual KV cache for answer generation.

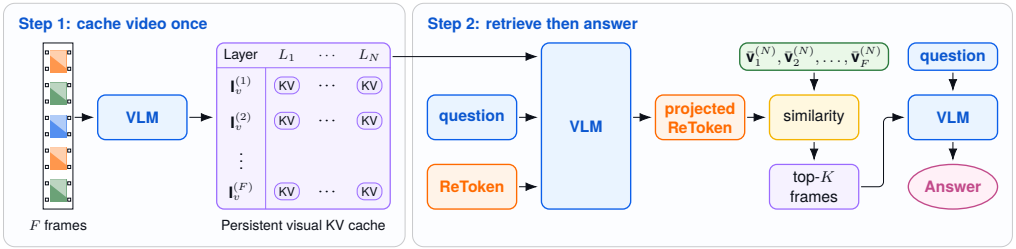


Figure 4: **Inference Pipeline**. Our method first caches features for all input frames. Given a text query, ReToken retrieves the relevant frames, and the answer is generated from their cached features.

Cache video once: At video ingestion time, the VLM processes the visual tokens and stores the resulting per-layer KV cache in a persistent cache. For short videos, this is a single full-context forward pass. For long videos, we follow ReKV [13] and encode the video chunk by chunk with a sliding-window attention mask: each chunk attends only to the most recent l_m visual tokens, while its newly produced KV states are appended to the cache. Older KV states can be offloaded to CPU memory when GPU memory is limited.

Retrieve then answer: Given a question, we run two forward passes through the frozen VLM over the cached visual context. The first pass computes the retrieval set $\mathcal{S}_K^{(N)}$ from the final layer; the second pass generates the answer conditioned on the selected frames. We use two passes because ReToken is supervised only at L_N (Eq. 4). The first pass enables the retrieval token to fully integrate textual and visual content, while the second pass enables attention to a consistent set of visual tokens, matching how it was trained.

First pass – retrieval. We append the retrieval token $\mathbf{X}_r$ to the question and run the first pass over the cached visual KV. For short videos $\mathbf{X}_r$ attends to all cached visual tokens at layer $l < N$, and the index $\mathcal{S}_K^{(N)}$ is computed at L_N via Eq. 3. For long videos the early-layer attention budget cannot accommodate the full cache, so at each layer $l < N$ we restrict $\mathbf{X}_r^{(l)}$'s attention as follows: (1) project the contextualized $\mathbf{X}_r^{(l)}$ through that layer's frozen value projection; (2) score every frame by the cosine similarity between this projection and the frame's mean value vector $\bar{\mathbf{v}}_f^{(l)}$ at layer l; (3) restrict $\mathbf{X}_r^{(l)}$'s attention to the top-K' frames under this score before computing layer l's output. We use $K' = 256$ by default, which keeps each early-layer attention well within GPU memory while leaving headroom for the final-layer ranking to refine the selection. At L_N we then compute Eq. 3 over the per-frame value means to obtain $\mathcal{S}_K^{(N)}$, reordered by original timestamp.

We note an asymmetry between this retrieval-pass budget K' and the answer-stage budget K: ReToken's retrieval pass attends to up to $K' = 256$ frames per early layer, which is different from the K used at answer time. ReToken therefore has access to more visual context when *ranking* candidates, while it still uses only K-frame visual context when *generating* the answer.

Second pass – answer. For each layer l, we load only the visual KV cache belonging to frames in $\mathcal{S}_K^{(N)}$, and run a standard generation pass over this reduced visual context together with the question. The answer stage therefore attends to about $K \cdot M/F$ visual tokens instead of all M, while the expensive video encoding is performed only once per video. This is particularly efficient when multiple questions are asked about the same video. Full prompt templates and a worked example of how ReToken interacts with the contextual visual tokens and the question are provided in Supp. A.

4 Experiments

4.1 Settings

Implementation Details. We use the greedy decoding configuration for ReToken and every baseline. We train ReToken using `Qwen3VL-8B` and `InternVL3.5-8B` on an image question-answering (QA) dataset. For the default frozen-VLM setting, we use an effective batch size of 64 and a learning

rate of 3×10^{-4} to train the learnable token for 3 epochs on a single H100 GPU with `Qwen3VL-8B`. Since `InternVL3.5-8B` requires more tokens to represent an image, resulting in substantially higher computational and memory costs, we train `InternVL3.5-8B` for only one epoch. We adopt a linear warmup schedule followed by cosine learning rate decay. For the partial fine-tuning setting, we use a learning rate of 2×10^{-5} and an effective batch size of 64. As this setting converges rapidly and begins to overfit after one epoch, we apply early stopping at the end of the first epoch. Training `Qwen3VL-8B` under this setting takes approximately 4 hours on a single H100 GPU.

For long video inference, we use an encoding chunk size of 128 frames and a sliding-window length of l_m=30,000 tokens, which corresponds to roughly 153 frames at 196 tokens per frame.

Training Datasets. Our default multi-image training dataset comes from the MIRAGE [45] fine-tuning dataset, whose examples are each annotated with a relevant/irrelevant label per image. We use a 95%/5% train/validation split. It is a multi-image QA (MIQA) dataset that combines existing MIQA datasets (RetVQA [30], SlideVQA [36], and WebQA [7]) with synthetic MIQA data adapted from the LLaVA Visual Instruct 150K dataset [25] via clustering and distractor sampling. Please refer to Supp. B.1 for more details.

Evaluation Datasets. We evaluate ReToken on four benchmarks chosen to probe complementary aspects of retrieval and long-context understanding.

Visual Haystacks (VHs) [45] is a benchmark introduced alongside MIRAGE to evaluate the pure visual recognition ability of vision-language models. Constructed from the COCO dataset [24], it consists of 1,000 question-answer pairs and provides query-relevant image annotations. The answer is always either "Yes" or "No". For each QA example, a varying number of distractor images is added to test the model's ability to locate informative content within a large pool of inputs.

QAEgo4D$_{\text{Test-MC}}$ [12] is the multiple-choice subset of the QAEgo4D-test benchmark [6], focusing on question answering over long egocentric videos. Each example is annotated with the video segments relevant to the question. Video lengths range from 4 to 20 minutes.

LVBench [41] is a benchmark for extreme long-video understanding, comprising 103 publicly sourced YouTube videos totaling roughly 117 hours, with an average length of 68 minutes and individual videos extending up to 2 hours. This is a multiple-choice dataset, and the metric is accuracy.

Video-MME [15] consists of 900 videos and 2,700 question-answer pairs. Video durations range from 11 seconds to 1 hour, partitioned into short (<2 min), medium (4–15 min), and long (30–60 min) splits. Each question is multiple-choice, and we report accuracy as the evaluation metric.

4.2 Visual Haystacks Retrieval and Accuracy

We evaluate ReToken against the baselines on the single-needle task of the Visual Haystacks benchmark [45], in which exactly one image in the haystack is relevant to the query. The context size C controls the number of distractor images, with larger C posing a greater challenge to the model. K denotes the number of retrieved images.

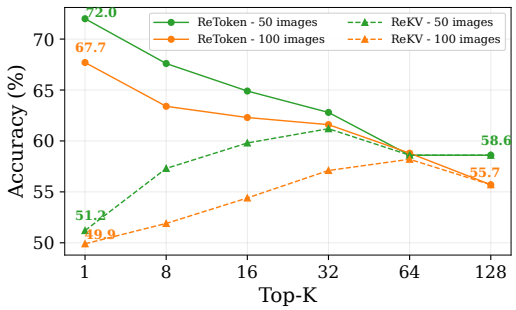


Figure 5: **Accuracy vs. Retrieve K Images**. Frozen `Qwen3VL-8B`.

Table 2: **Retrieval Strategy Comparison.** We compare `ReToken` against different retrievers on Visual Haystacks with retrieval budget K=1. "GT Cache" means using the KV cache of the ground truth image as input and serves as the oracle upper bound for any retriever. All results are based on `Qwen3VL-8B` with the VLM frozen.

Metric	Standard	GT Cache	SigLIP2	ReKV	CoT	**ReToken**
$C = 2, K = 1$						
Recall	N/A	100	76.4	63.3	65.3	**88.5**
Accuracy	82.0	86.5	82.8	73.0	77.8	**85.9**
$C = 50, K = 1$						
Recall	N/A	100	20.8	1.8	3.1	**64.7**
Accuracy	58.6	80.7	60.7	51.2	51.5	**72.0**

Retrieval Behavior of ReToken and Attention-Based Retrieval. Fig. 5 compares RETOKEN and attention-based retriever, ReKV [13], across retrieval budgets from K=1 to K=128 at C=50 and C=100. When $K >= C$, no retrieval is needed since all images are used as input. The two methods exhibit opposite trends in the meaningful retrieval regime ($K < C$): RETOKEN performs best at small K and degrades as more images (mostly distractors) are retrieved, whereas ReKV starts low and improves with larger K. The crossover reflects a precision–recall tradeoff: RETOKEN is precise per slot, while ReKV needs a wider budget to recall the relevant image.

Retrieval Strategy Comparison. We also compute recall to measure the retrieval ability of different strategies on Visual Haystacks. Here, recall measures whether the retriever successfully selects the ground-truth image when K=1. Tab. 2 compares RETOKEN against several retrieval strategies on Visual Haystacks. We compare against four baselines: *Standard* pre-fills all images (the default VLM inference pipeline); *GT Cache* uses the ground-truth image's KV cache as input, serving as an oracle upper bound; *SigLIP2* [39] first retrieves a single image with `siglip2-giant-opt-patch16-384` and feeds its KV cache to the VLM; and *CoT* prompts the VLM to first generate a target search phrase, then uses that phrase as the query for ReKV attention-based retrieval (full prompt in Supp. B.2).

RETOKEN substantially outperforms all baselines, retaining most of the GT Cache upper-bound accuracy at large context size ($C = 50$) while the others fail.

How the Stored KV Cache Differs From Re-encoding the Visual Input. Observing the accuracy for GT Cache in Tab. 2, we notice an interesting phenomenon: the accuracy degrades with higher C, even though only the cache for the same single ground truth relevant image is used in each case (86.5 for $C = 2$ vs. 80.7 for $C = 50$). This is because the image tokens have attended to previous images, which are irrelevant in this case, picking up distracting information.

To quantify this distraction, we compare the accuracy when using "GT Image" (re-encoding the ground-truth image alone, yielding a clean KV cache) with "GT Cache" (the fused KV cache for the ground-truth image). Their gap, $\Delta_{\text{GT Image}-\text{GT Cache}}$, measures the degree of distraction (Tab. 3); a smaller gap indicates a cleaner stored KV cache. Partial tuning on the retrieval dataset encourages the model to produce cleaner KV cache representations (KV cache is not affected when we only train the retrieval token). A cleaner KV cache benefits Visual Haystacks, but may pose a challenge for video understanding, which requires connecting information across adjacent frames.

Accuracy Comparison. Tab. 4 reports accuracy across context sizes from C=1 (only the query-relevant image, no distractors) to C=100. The retrieval budget K is set to 1. At C=1, RETOKEN matches the vanilla `Qwen3VL-8B` backbone, as we skip retrieval when $C <= K$.

As the context size grows, the gap between RETOKEN and the vanilla backbone widens substantially. The one exception is InternVL3.5-8B at C=100, which we attribute to its 1-epoch training budget (Sec. 4.1). RETOKEN also compares favorably against external baselines, surpassing all baselines across all context sizes, including the dedicated multi-image RAG framework MIRAGE.

4.3 Image-to-Video Transfer Evaluation

Having validated RETOKEN on image data, we now evaluate its transfer to the video setting. Notably, results in this section are obtained with RETOKEN trained *only* on the multi-image MIRAGE training data, demonstrating strong zero-shot transfer. By default, we evaluate the frozen `Qwen3VL-8B` on video benchmarks with videos sampled at 0.5 FPS. "Uniformly" means loading the KV cache of K uniformly sampled frames as input.

Table 3: **Partial tuning on the retrieval dataset yields a cleaner KV cache.** $\Delta_{\text{GT Image}-\text{GT Cache}}$ measures the degree of distraction. A smaller gap indicates a cleaner stored KV cache.

Model	Loss		GT Image	GT Cache		$\Delta_{\text{GT Image}-\text{GT Cache}}$	
	Generation	Retrieval		$C = 2$	$C = 50$	$C = 2$	$C = 50$
Qwen3VL-8B			87.8	86.5	80.7	1.3	7.1
Tuned Layer 1 - 3	✓		87.8	86.8	82.3	**1.0**	5.5
	✓	✓	**88.5**	**87.3**	**83.6**	1.2	**4.9**

Table 4: `ReToken`**'s advantage grows with context size, yielding over 20% relative gain at C=50, when freezing the VLM.** Performance on Visual Haystacks. C denotes the number of context images (i.e., images provided as input). E indicates a context overflow or CUDA out-of-memory error.

Method	$C=1$	$C=2$	$C=3$	$C=5$	$C=10$	$C=20$	$C=50$	$C=100$
`Generalist`								
Gemini-1.5 Pro [37]	88.4	82.0	78.3	76.0	71.9	68.6	62.8	57.4
InternVL2 [8]	88.1	80.5	72.3	63.9	58.8	55.2	E	E
`Specialist`								
SigLIP2 [39]	72.0	69.2	68.1	65.3	64.1	60.3	58.7	58.3
MIRAGE [45]	83.2	77.8	76.6	72.8	70.5	66.0	63.6	62.0
$\text{REN}_{\text{DINO-SigLIP2}}$ [20]	81.2	78.6	77.4	76.0	74.0	72.1	68.3	65.5
`Freezing the VLM`								
Qwen3VL-8B [4]	**87.8**	82.0	77.9	74.7	69.2	64.0	58.6	55.7
+ ReToken	**87.8** (+0.0)	**85.9** (+3.9)	**84.4** (+6.5)	**82.5** (+7.8)	**80.7** (+11.5)	**76.8** (+12.8)	**72.0** (+13.4)	**67.7** (+12.0)
InternVL3.5-8B [42]	**87.5**	81.6	79.3	73.1	69.0	65.0	57.3	55.3
+ ReToken	**87.5** (+0.0)	**84.1** (+2.5)	**83.7** (+4.4)	**80.7** (+7.6)	**76.9** (+7.9)	**73.0** (+8.0)	**69.7** (+12.4)	**59.7** (+4.4)
`Tuning the first 3 layers of the VLM`								
Qwen3VL-8B [4]	**88.5**	85.6	82.4	79.6	76.5	71.2	64.2	61.5
+ ReToken	**88.5** (+0.0)	**86.5** (+0.9)	**85.2** (+2.8)	**84.3** (+4.7)	**82.8** (+6.3)	**79.8** (+8.6)	**75.0** (+10.8)	**70.6** (+9.1)

Accuracy on $\text{QAEgo4D}_{\text{Test-MC}}$. Tab. 5 shows that ReToken's advantage is most pronounced under tight retrieval budgets. With $K=1$, ReToken achieves a 6.8-point improvement over uniform sampling, while ReKV actually performs slightly worse than uniform sampling. Unlike multi-image QA where images are independent, neighboring video frames provide complementary context, so accuracy continues to climb as K grows even where ReToken's lead over uniform sampling narrows.

Table 5: Zero-shot performance on $\text{QAEgo4D}_{\text{Test-MC}}$.

K	Uniformly	ReKV	**ReToken**
1	42.8	42.6	**49.6**
16	53.6	57.8	**60.0**
32	55.2	59.8	**60.6**

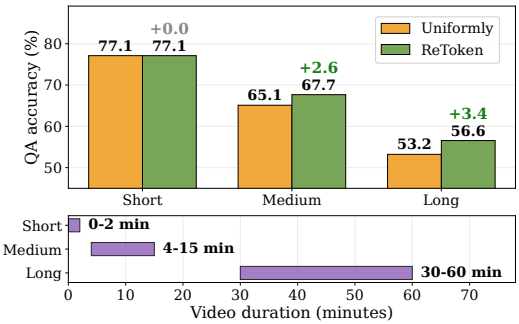
Long Video Understanding. The benefits of retrieval grow with video length (Tab. 6a). On the Short split of Video-MME, ReToken provides no gain, since these videos last only 2 minutes (60 frames at 0.5 FPS), which falls below our retrieval input budget of 100 frames and therefore does not trigger retrieval. As video length grows, however, the gain becomes substantial. ReToken achieves an 8.0-point improvement on LVBench (Tab. 6b).

4.4 Ablations

We ablate the design choices of ReToken with `Qwen3VL-8B` on Visual Haystacks by default, focusing on: (1) retrieval based on visual key or value; (2) training the token only versus partial tuning; (3) the influence of the inference setting; (4) efficiency analysis; and (5) error analysis.

Table 6: **ReToken helps more with long videos, even when trained only with images**. Freezing the VLM and only tuning ReToken.

(a) Performance gain for Qwen3VL on different split of VideoMME.



(b) All results in the `Process raw frames` section are sourced from the technical report [4].

Model Duration	# Frames	LVBench 30 – 140 min	Video-MME 0 – 60 min
`Process raw frames`			
Gemini 2.5 Pro [10]	512	69.0	80.6
OpenAI GPT-5 [34]	256	-	77.3
Claude Opus 4.1 [2]	100	-	73.3
Qwen3VL-8B [4]	2 FPS	58.0	71.4
`Retrieve from the stored KV cache at 0.5 FPS`			
Qwen3VL-8B [4]	100	40.6	65.1
+ ReToken	100	48.6 (+8.0)	67.1 (+2.0)

Table 8: **Tuning early layers can improve performance on the image task, but it degrades performance on the video task**. (a) Tuning the first few VLM layers (Layer 1–3) gives the best recall–accuracy trade-off at C=50, $K = 1$. (b) Training the VLM on images decreases accuracy on video; evaluated on QAEgo4D$_{\text{Test-MC}}$ with K=16.

(a) Train on images, evaluate on images.

Tuning Layers	Freezing	1	1-3	1-10	N
Recall	64.7	**68.3**	68.2	58.2	64.3
Acc	72.0	73.3	**75.0**	72.6	71.3

(b) Train on images, evaluate on video.

Strategy	Uniformly	ReKV [13]	**ReToken**
Freezing	53.6	57.8	**60.0**
Layer 1 - 3	50.0	55.2	**58.0**

We further ablate design choices on whether a single token is sufficient for retrieval in Supp. C.1, and whether RETOKEN can skip attending to visual tokens in Supp. C.2.

Table 7: Key vs. Value.

	× Key	× Value
Recall	59.4	**64.7**
Accuracy	70.6	**72.0**

Retrieval Score Based on Average Vision Key or Value. We ablate RETOKEN's scoring mechanism in Tab. 7 with $C = 50, K = 1$: "×Key" trains RETOKEN to retrieve via the average image keys, while "×Value" uses the average image values. "×Value" pulls clearly ahead in both recall and accuracy. We attribute this to values carrying the content actually propagated through attention, providing a more discriminative signal for distinguishing the relevant image among many distractors. Supp. B.3 provides further evidence for this.

Train Only Token or Partial Tuning VLM Layers. Tab. 8a ablates the partial-tuning depth. Tuning the first few layers (1–3) gives the best accuracy while preserving high recall, indicating that early layers are where visual features can be best shaped without compromising downstream performance.

Tab. 8b compares the transfer ability of the frozen and partial-tuning settings. RETOKEN shows a clear advantage in both. However, overall performance under partial tuning is lower than under the frozen setting. A possible reason is that tuning the LLM layers on image data can cause domain shift, especially given the distribution gap between image training data and video benchmarks.

Single or Two-pass Inference. Since RETOKEN is trained to retrieve at the last layer, we use two-pass inference to obtain retrieval results at the final layer and then broadcast them to all layers. We ablate its effect and report the accuracy in Tab. 9 under $C = 50, K = 1$. "Single" means each layer retrieves its own images and directly generates the response based on that. "Two" means early layers attend to all visual tokens, and the last-layer retrieval result is broadcast to generate the response. The conclusions are different for ReKV and RETOKEN. One possible explanation is that attention-based retrieval is suited for answering directly, so it does not rely on the final layer's results; in contrast, RETOKEN is only trained with the retrieval loss at the final layer. Therefore, the default setting for ReKV is single-pass, while for RETOKEN it is two-pass.

Table 9: Two-pass Inference.

	ReKV	**RETOKEN**
Single	**51.2**	50.4
Two	50.4	**72.0**

Table 10: **Runtime vs. quality on long-video QA** (QAEgo4D$_{\text{Test-MC}}$, ∼240 frames per video at 0.5 FPS, single H100). Latency is reported *per question* after one-time video encoding. RETOKEN achieves higher accuracy with only modest overhead in the retrieval pass, and identical cost for the encode and answer-stage.

Method	K	Latency (second, per question)			Peak GPU (GB)		QA (%)	
		Encode†	Retrieve / Load	Answer	Retrieve	Answer	Acc.	Recall
Uniformly	16	14.68	0.081	0.169	65.1	65.4	53.6	38.0
RETOKEN	16	14.68	0.519	0.167	66.6	65.4	**60.0**	**70.6**
Uniformly	32	14.70	0.128	0.166	66.0	66.7	55.2	59.6
RETOKEN	32	14.70	0.538	0.166	67.2	66.8	**60.6**	**81.2**

†Encoding is performed once per video; answers are generated from the visual tokens of the K retrieved frames.

Table 11: **RETOKEN helps most when the evidence is localized and nameable, and hurts when it is dispersed across the video.** Per-type accuracy on LVBench. K=100. *#QA* denotes the number of questions per type; a question can carry multiple type labels.

Question type	*#QA*	Uniformly	**RETOKEN**	Δ
Key information retrieval	291	42.3	57.7	+15.4
Entity recognition	677	40.0	50.5	+10.5
Reasoning	201	40.8	46.3	+5.5
Temporal grounding	220	35.0	38.6	+3.6
Event understanding	647	41.4	43.3	+1.9
Summarization	58	36.2	31.0	−5.2

Runtime and Memory Usage. Tab. 10 breaks down per-question cost into retrieval and answer phases, and reports video encoding separately because encoding produces a persistent KV cache shared across all questions about the same video. Encoding dominates the budget at ≈14.7 s per video. When few questions are asked of a video, the encoding dominates the compute time, but can be performed as a pre-process. RETOKEN adds roughly 0.4 seconds to the per-question retrieval and answering time, but significantly improves recall and question accuracy.

Error Analysis. LVBench annotates each question with a task-type label, allowing us to characterize when RETOKEN works best and when it fails (Tab. 11). RETOKEN is strongest when the evidence is localized and nameable: key information retrieval (+15.4) and entity recognition (+10.5) are exactly the regimes where a precise retrieval target locks onto the relevant frames. Gains shrink where relevance depends on cross-frame or temporal structure rather than per-frame content, as in temporal grounding and event understanding. RETOKEN hurts on summarization, where the evidence is dispersed across the entire video; uniform coverage is the better prior for this question type.

5 Conclusion

We diagnose the limitations of attention-based retrieval in VLMs and introduce RETOKEN, a learnable token that improves visual retrieval. Our diagnosis points to a broader principle: in pretrained VLMs, the value space carries a stronger text-aligned signal. Despite being trained only on multi-image data, RETOKEN yields significant improvements on both image and long-video benchmarks, transferring zero-shot from images to videos. Both training and long-video inference fit on a single H100, making RETOKEN a practical step toward scalable long-context multimodal reasoning.

Limitations. RETOKEN requires a two-pass forward and attends to more visual context in early layers, adding slightly to the memory requirements and response time. Our training data is limited to multi-image QA. Training on video data could help RETOKEN better capture temporal structure and improve video understanding.

Future Work. RETOKEN scores each frame independently by matching the query content against per-frame value means, and relaxing this design opens several directions. First, retrieval could target *sets* of consecutive frames whose information emerges from their temporal combination rather than from any single frame. Second, temporally offset queries such as "what happened before I entered the room" would require a scoring mechanism aware of temporal displacement, since content matching alone tends to locate the described event rather than the frames preceding it. Third, computing the retrieval score at the token level instead of the frame level could recover evidence that occupies only a few tokens and is diluted by frame-level mean pooling.

Acknowledgments

We thank Xiaodong Liu, Sethuraman T V, and Bolin Lai for their insightful comments and helpful discussions.

This research project has benefited from the Microsoft Agentic AI Research and Innovation (AARI) grant program, and was partially supported by the Office of Naval Research under grant N00014-23-1-2383. This work also used NVIDIA GPUs at NCSA Delta through allocation CIS240059 and

CIS250059 from the Advanced Cyberinfrastructure Coordination Ecosystem: Services & Support (ACCESS) program, which is supported by NSF Grants #2138259, #2138286, #2138307, #2137603, and #2138296.

References

[1] Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, et al. Flamingo: a visual language model for few-shot learning. *Advances in neural information processing systems*, 35:23716–23736, 2022.

[2] Anthropic. System card addendum: Claude Opus 4.1. Technical report, Anthropic, August 2025.

[3] Kirolos Ataallah, Xiaoqian Shen, Eslam Abdelrahman, Essam Sleiman, Mingchen Zhuge, Jian Ding, Deyao Zhu, Jürgen Schmidhuber, and Mohamed Elhoseiny. Goldfish: Vision-language understanding of arbitrarily long videos. In *European Conference on Computer Vision*, pages 251–267. Springer, 2024.

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

[5] Daniel Bolya, Po-Yao Huang, Peize Sun, Jang Hyun Cho, Andrea Madotto, Chen Wei, Tengyu Ma, Jiale Zhi, Jathushan Rajasegaran, Hanoona Rasheed, et al. Perception encoder: The best visual embeddings are not at the output of the network. *arXiv preprint arXiv:2504.13181*, 2025.

[6] Leonard Bärmann and Alex Waibel. Where did i leave my keys? — episodic-memory-based question answering on egocentric videos. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 1559–1567, 2022.

[7] Yingshan Chang, Mridu Narang, Hisami Suzuki, Guihong Cao, Jianfeng Gao, and Yonatan Bisk. Webqa: Multihop and multimodal qa. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 16495–16504, 2022.

[8] Zhe Chen, Weiyun Wang, Yue Cao, Yangzhou Liu, Zhangwei Gao, Erfei Cui, Jinguo Zhu, Shenglong Ye, Hao Tian, Zhaoyang Liu, et al. Expanding performance boundaries of open-source multimodal models with model, data, and test-time scaling. *arXiv preprint arXiv:2412.05271*, 2024.

[9] Zhe Chen, Jiannan Wu, Wenhai Wang, Weijie Su, Guo Chen, Sen Xing, Muyan Zhong, Qinglong Zhang, Xizhou Zhu, Lewei Lu, et al. Internvl: Scaling up vision foundation models and aligning for generic visual-linguistic tasks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 24185–24198, 2024.

[10] Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, et al. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *arXiv preprint arXiv:2507.06261*, 2025.

[11] Wenliang Dai, Junnan Li, Dongxu Li, Anthony Tiong, Junqi Zhao, Weisheng Wang, Boyang Li, Pascale N Fung, and Steven Hoi. Instructblip: Towards general-purpose vision-language models with instruction tuning. *Advances in neural information processing systems*, 36:49250–49267, 2023.

[12] Shangzhe Di and Weidi Xie. Grounded question-answering in long egocentric videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12934–12943, 2024.

[13] Shangzhe Di, Zhelun Yu, Guanghao Zhang, Haoyuan Li, Tao Zhong, Hao Cheng, Bolin Li, Wanggui He, Fangxun Shu, and Hao Jiang. Streaming video question-answering with in-context video kv-cache retrieval. *arXiv preprint arXiv:2503.00540*, 2025.

[14] Zafeirios Fountas, Martin A Benfeghoul, Adnan Oomerjee, Fenia Christopoulou, Gerasimos Lampouras, Haitham Bou-Ammar, and Jun Wang. Human-inspired episodic memory for infinite context llms. In *13th International Conference on Learning Representations Iclr 2025*. ICLR, 2025.

[15] Chaoyou Fu, Yuhan Dai, Yongdong Luo, Lei Li, Shuhuai Ren, Renrui Zhang, Zihan Wang, Chenyu Zhou, Yunhang Shen, Mengdan Zhang, et al. Video-mme: The first-ever comprehensive evaluation benchmark of multi-modal llms in video analysis. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 24108–24118, 2025.

[16] Chi Han, Qifan Wang, Hao Peng, Wenhan Xiong, Yu Chen, Heng Ji, and Sinong Wang. Lm-infinite: Zero-shot extreme length generalization for large language models. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 3991–4008, 2024.

[17] Bo He, Hengduo Li, Young Kyun Jang, Menglin Jia, Xuefei Cao, Ashish Shah, Abhinav Shrivastava, and Ser-Nam Lim. Ma-lmm: Memory-augmented large multimodal model for long-term video understanding. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 13504–13514, 2024.

[18] Ting Jiang, Minghui Song, Zihan Zhang, Haizhen Huang, Weiwei Deng, Feng Sun, Qi Zhang, Deqing Wang, and Fuzhen Zhuang. E5-v: Universal embeddings with multimodal large language models. *arXiv preprint arXiv:2407.12580*, 2024.

[19] Peng Jin, Ryuichi Takanobu, Wancai Zhang, Xiaochun Cao, and Li Yuan. Chat-univi: Unified visual representation empowers large language models with image and video understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13700–13710, 2024.

[20] Savya Khosla, Sethuraman TV, Barnett Lee, Alexander Schwing, and Derek Hoiem. Ren: Fast and efficient region encodings from patch-based image encoders. *arXiv preprint arXiv:2505.18153*, 2025.

[21] Mengcheng Lan, Chaofeng Chen, Yiping Ke, Xinjiang Wang, Litong Feng, and Wayne Zhang. Proxyclip: Proxy attention improves clip for open-vocabulary segmentation. In *European Conference on Computer Vision*, pages 70–88. Springer, 2024.

[22] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *International conference on machine learning*, pages 19730–19742. PMLR, 2023.

[23] Yanwei Li, Chengyao Wang, and Jiaya Jia. Llama-vid: An image is worth 2 tokens in large language models. In *European Conference on Computer Vision*, pages 323–340. Springer, 2024.

[24] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014.

[25] Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction tuning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 26296–26306, 2024.

[26] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *Advances in neural information processing systems*, 36:34892–34916, 2023.

[27] Yikun Liu, Yajie Zhang, Jiayin Cai, Xiaolong Jiang, Yao Hu, Jiangchao Yao, Yanfeng Wang, and Weidi Xie. Lamra: Large multimodal model as your advanced retrieval assistant. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 4015–4025, 2025.

[28] Yongdong Luo, Xiawu Zheng, Guilin Li, Shukang Yin, Haojia Lin, Chaoyou Fu, Jinfa Huang, Jiayi Ji, Fei Chao, Jiebo Luo, et al. Video-rag: Visually-aligned retrieval-augmented long video comprehension. *arXiv preprint arXiv:2411.13093*, 2024.

[29] Ziyu Ma, Chenhui Gou, Hengcan Shi, Bin Sun, Shutao Li, Hamid Rezatofighi, and Jianfei Cai. Drvideo: Document retrieval based long video understanding. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 18936–18946, 2025.

[30] Abhirama Subramanyam Penamakuri, Manish Gupta, Mithun Das Gupta, and Anand Mishra. Answer mining from a pool of images: towards retrieval-based visual question answering. *arXiv preprint arXiv:2306.16713*, 2023.

[31] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PmLR, 2021.

[32] Xiaoqian Shen, Yunyang Xiong, Changsheng Zhao, Lemeng Wu, Jun Chen, Chenchen Zhu, Zechun Liu, Fanyi Xiao, Balakrishnan Varadarajan, Florian Bordes, et al. Longvu: Spatiotemporal adaptive compression for long video-language understanding. *arXiv preprint arXiv:2410.17434*, 2024.

[33] Yan Shu, Zheng Liu, Peitian Zhang, Minghao Qin, Junjie Zhou, Zhengyang Liang, Tiejun Huang, and Bo Zhao. Video-xl: Extra-long vision language model for hour-scale video understanding. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 26160–26169, 2025.

[34] Aaditya Singh, Adam Fry, Adam Perelman, Adam Tart, Adi Ganesh, Ahmed El-Kishky, Aidan McLaughlin, Aiden Low, AJ Ostrow, Akhila Ananthram, et al. Openai gpt-5 system card. *arXiv preprint arXiv:2601.03267*, 2025.

[35] Enxin Song, Wenhao Chai, Guanhong Wang, Yucheng Zhang, Haoyang Zhou, Feiyang Wu, Haozhe Chi, Xun Guo, Tian Ye, Yanting Zhang, et al. Moviechat: From dense token to sparse memory for long video understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18221–18232, 2024.

[36] Ryota Tanaka, Kyosuke Nishida, Kosuke Nishida, Taku Hasegawa, Itsumi Saito, and Kuniko Saito. Slidevqa: A dataset for document visual question answering on multiple images. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 13636–13645, 2023.

[37] Gemini Team, Petko Georgiev, Ving Ian Lei, Ryan Burnell, Libin Bai, Anmol Gulati, Garrett Tanzer, Damien Vincent, Zhufeng Pan, Shibo Wang, et al. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *arXiv preprint arXiv:2403.05530*, 2024.

[38] Gemma Team, Thomas Mesnard, Cassidy Hardin, Robert Dadashi, Surya Bhupatiraju, Shreya Pathak, Laurent Sifre, Morgane Rivière, Mihir Sanjay Kale, Juliette Love, et al. Gemma: Open models based on gemini research and technology. *arXiv preprint arXiv:2403.08295*, 2024.

[39] Michael Tschannen, Alexey Gritsenko, Xiao Wang, Muhammad Ferjad Naeem, Ibrahim Alabdulmohsin, Nikhil Parthasarathy, Talfan Evans, Lucas Beyer, Ye Xia, Basil Mustafa, et al. Siglip 2: Multilingual vision-language encoders with improved semantic understanding, localization, and dense features. *arXiv preprint arXiv:2502.14786*, 2025.

[40] Feng Wang, Jieru Mei, and Alan Yuille. Sclip: Rethinking self-attention for dense vision-language inference. In *European conference on computer vision*, pages 315–332. Springer, 2024.

[41] Weihan Wang, Zehai He, Wenyi Hong, Yean Cheng, Xiaohan Zhang, Ji Qi, Ming Ding, Xiaotao Gu, Shiyu Huang, Bin Xu, et al. Lvbench: An extreme long video understanding benchmark. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 22958–22967, 2025.

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

[43] Xiaohan Wang, Yuhui Zhang, Orr Zohar, and Serena Yeung-Levy. Videoagent: Long-form video understanding with large language model as agent. In *European Conference on Computer Vision*, pages 58–76. Springer, 2024.

[44] Ziyang Wang, Shoubin Yu, Elias Stengel-Eskin, Jaehong Yoon, Feng Cheng, Gedas Bertasius, and Mohit Bansal. Videotree: Adaptive tree-based video representation for llm reasoning on long videos. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 3272–3283, 2025.

[45] Tsung-Han Wu, Giscard Biamby, Jerome Quenum, Ritwik Gupta, Joseph E Gonzalez, Trevor Darrell, and David M Chan. Visual haystacks: A vision-centric needle-in-a-haystack benchmark. *arXiv preprint arXiv:2407.13766*, 2024.

[46] Chaojun Xiao, Pengle Zhang, Xu Han, Guangxuan Xiao, Yankai Lin, Zhengyan Zhang, Zhiyuan Liu, and Maosong Sun. Infllm: Training-free long-context extrapolation for llms with an efficient context memory. *Advances in neural information processing systems*, 37:119638–119661, 2024.

[47] Guangxuan Xiao, Yuandong Tian, Beidi Chen, Song Han, and Mike Lewis. Efficient streaming language models with attention sinks. *arXiv preprint arXiv:2309.17453*, 2023.

[48] Yao Xiao, Qiqian Fu, Heyi Tao, Yuqun Wu, Zhen Zhu, and Derek Hoiem. Textregion: Text-aligned region tokens from frozen image-text models. *arXiv preprint arXiv:2505.23769*, 2025.

[49] Shoubin Yu, Jaemin Cho, Prateek Yadav, and Mohit Bansal. Self-chained image-language model for video localization and question answering. *Advances in Neural Information Processing Systems*, 36:76749–76771, 2023.

[50] Chong Zhou, Chen Change Loy, and Bo Dai. Extract free dense labels from clip. In *European conference on computer vision*, pages 696–712. Springer, 2022.

[51] Yutao Zhu, Zhaoheng Huang, Zhicheng Dou, and Ji-Rong Wen. One token can help! learning scalable and pluggable virtual tokens for retrieval-augmented large language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 26166–26174, 2025.

A Prompt Templates

In this section, we detail the prompt templates used during training and inference with ReToken. To highlight the modifications introduced by ReToken, we color-code the content-bearing tokens as follows: All Vision Tokens, Retrieved Vision Tokens, Question Tokens and the <Retrieval> token, and the model output Answer.

A.1 Original Prompt Template

The standard prompt format used by vision-language models feeds *all* visual tokens into the model alongside the textual query:

Original Prompt

<system><|vision_start|>All Vision Tokens<|vision_end|>Question Tokens<|im_end|><|im_start|>assistant: Answer

A.2 Training Prompt Template

During training, we employ two complementary prompt formats depending on the training mode. The first, which appends the <Retrieval> token after the question (teaching the model *when* to invoke retrieval), is used in both frozen and partially fine-tuned settings:

Training Prompt 1: learning to trigger retrieval

<system><|vision_start|>All Vision Tokens<|vision_end|>Question Tokens<Retrieval>

The second format, designed to preserve the model's original QA abilities, is incorporated alongside the first exclusively when the LLM is partially fine-tuned:

Training Prompt 2: keeping its original QA ability

<system><|vision_start|>All Vision Tokens<|vision_end|>Question Tokens<|im_end|><|im_start|>assistant: Answer

A.3 Inference Pipeline

At inference time, we adopt a two-stage *retrieve-then-answer* pipeline. In Stage 1, at most K' frames are attended during retrieval and the model uses the <Retrieval> token to aggregate query-relevant information, performing retrieval at the final layer. In Stage 2, only the retrieved visual tokens Retrieved Vision Tokens are supplied to the model to generate the final answer.

Inference Stage 1: retrieval triggering

<system><|vision_start|>All Vision Tokens<|vision_end|>Question Tokens<Retrieval>

Inference Stage 2: answering with retrieved tokens

<system><|vision_start|>Retrieved Vision Tokens<|vision_end|>Question Tokens<|im_end|><|im_start|>assistant: Answer

B Additional Details

B.1 Training Data Filtering

The original MIRAGE fine-tuning set aggregates multiple open-source VQA datasets, many of which were already seen during Qwen3VL pretraining. We observe that on a substantial fraction of these examples, Qwen3VL achieves *lower* generation loss when conditioned on the full set of distractor images than on the target ground-truth image alone, suggesting that the model has memorized the

Table 12: **The influence of training dataset.** "w/o filtered" means directly sampling 70,686 examples from the original MIRAGE fine-tuning set without filtering, and using them to train ReToken. "w/ filtered" is our default setting.

Method	$C = 1$	$C = 2$	$C = 3$	$C = 5$	$C = 10$	$C = 20$	$C = 50$	$C = 100$
Qwen3VL-8B [4]	87.8	82.0	77.9	74.7	69.2	64.0	58.6	55.7
w/o filtered	87.8	85.2	83.1	80.8	78.1	75.5	69.3	62.7
w/ filtered	87.8	85.9	84.4	82.5	80.7	76.8	72.0	67.7

underlying QA pairs and bypasses the intended retrieval task. To address this, we apply three filters to the MIRAGE-augmented dataset (in which images from different QAs are combined into a single multi-image input):

- **Memorization filter.** We discard examples where the generation loss conditioned on the target image alone is higher than that conditioned on the full image set, as such examples no longer require retrieval to be answered.
- **Difficulty filter.** We rank remaining examples by the average attention score between the question query tokens and the target image key tokens, and retain those with higher scores. In a controlled experiment, we find that easier retrieval examples (higher attention scores on the target image) lead to a stronger learned retriever than harder examples.
- **Multi-image filter.** We remove single-image QA examples (which lack distractor images for retrieval) and OCR-style queries (which test text recognition rather than visual recognition).

After filtering, there are 70,686 examples in our final training dataset. We note that our filters are applied *only* to the MIRAGE training split; the Visual Haystacks evaluation split and video datasets are left untouched. The filtering signals (generation loss and attention scores) are computed from the frozen Qwen3VL-8B on training examples only, with no access to evaluation labels or examples. Tab. 12 shows the influence of filtering.

B.2 Chain-of-Thought (CoT) Retrieval Baseline

CoT Prompt Template (query-rewriting baseline)

Given the following question, what phrase should I search for to find the visual evidence needed to answer it? Respond with the search phrase only, no more than 5 words.
Question: {question}

B.3 Key or Value? Training Comparison

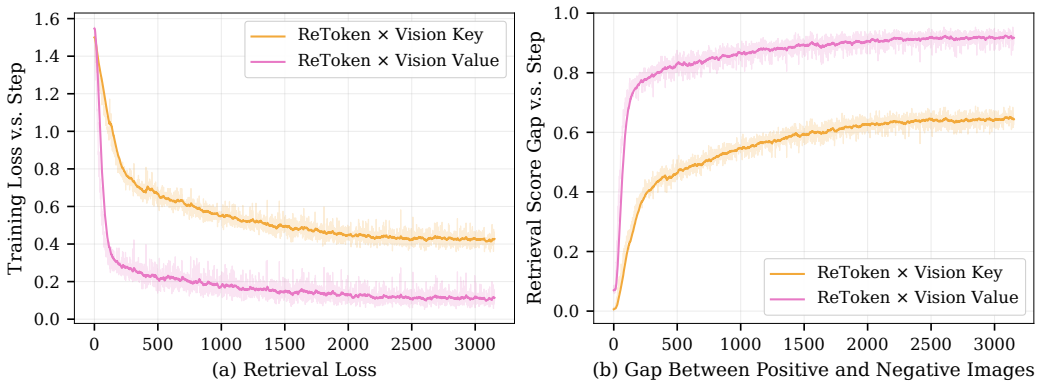


Figure 6: **Training Comparison.** The RETOKEN ×Value variant (a) converges much faster in retrieval loss, and (b) learns a substantially larger retrieval score gap between relevant and irrelevant images.

Fig. 6 shows the retrieval loss and retrieval score gap across training steps with the VLM frozen and only RETOKEN trained: ×Value yields a clear advantage.

C Design Choice

C.1 Multiple Tokens

Tab. 13 reports results for training 3 ReTokens on the retrieval task. We append the 3 ReTokens sequentially, and compute the retrieval score by computing the cosine similarity between each projected ReToken and the averaged image value feature. We then average the resulting logits to obtain the final score.

Table 13: **Multiple tokens perform on par with a single token.** Qwen3VL-8B.

Method	$C = 1$	$C = 2$	$C = 3$	$C = 5$	$C = 10$	$C = 20$	$C = 50$	$C = 100$
3 tokens	87.8	85.5	84.3	82.5	82.0	77.6	72.1	67.9
1 token	87.8	85.9	84.4	82.5	80.7	76.8	72.0	67.7

Performance is on par overall, with small fluctuations in both directions. One reason may lie in our training recipe: the multi-token design treats the embeddings equally, since all three are supervised through one averaged score, nothing encourages them to specialize, so the tokens likely converge on a similar solution. A promising direction could be to empower specialization of different tokens so that a token is more pronounced for its specialized domain. But this is beyond the scope of this work and we leave it for future exploration.

C.2 Query Composition or visual Summarization

Since we append the RETOKEN to the input, it can attend to both the images and the query, allowing it to serve as both a visual summarization token and a query summarization token. To better understand the role of the RETOKEN, we design an ablation in which the RETOKEN is only allowed to attend to the query tokens. In this setting, we perform two forward streams. In stream A, we encode the vision tokens and cache their KV. In stream B, we process only the query and the RETOKEN, so that neither the query token nor the RETOKEN can attend to the visual part. This way, the RETOKEN sees only the query tokens and learns to summarize what we want to retrieve based on the input query alone. We then compute the retrieval score by computing the cosine similarity between the projected RETOKEN from the last layer of stream B and the averaged visual value feature from the last layer of stream A.

Table 14: **Allowing the ReToken to attend to images yields better results.** Question tokens also skip the visual tokens in the "skip images" setting. The VLM is frozen.

Method	$C = 1$	$C = 2$	$C = 3$	$C = 5$	$C = 10$	$C = 20$	$C = 50$	$C = 100$
Qwen3VL-8B [4]	87.8	82.0	77.9	74.7	69.2	64.0	58.6	55.7
skip images	87.8	83.4	78.9	76.2	73.7	68.5	63.1	62.0
attend images	87.8	85.9	84.4	82.5	80.7	76.8	72.0	67.7

Tab. 14 reports the results with and without attention to the images. The results show that skipping the visual tokens entirely still yields decent performance, but underperforms the variant that attends to the images. This suggests that when the RETOKEN can understand what happens in the input video/images, it achieves better retrieval results.