ENHANCING LONG-VIDEO VLM EMBEDDINGS WITH QUERY-AWARE STREAMING LATENT REASONING

Anonymous authors
Paper under double-blind review

ABSTRACT

Long-video embedding requires capturing sparse query-relevant evidence under a limited visual-token budget. Uniform sampling can miss brief events in videos spanning minutes or hours, whereas encoding more frames in a single context increases memory and computation. We introduce **Query-Aware Streaming Latent Reasoning** (QASLR), a post-training framework that accumulates evidence across clips while keeping the embedding size fixed. QASLR selects a bounded set of frames, restores their temporal order, and processes them clip by clip with a vision-language backbone. A compact set of persistent think tokens cross-attends to each clip's features, while an embed token reads out a normalized representation after every update. This design integrates evidence across multiple backbone calls without requiring all selected frames to share a single context. Training combines final contrastive learning, step-wise contrastive supervision, and final-embedding self-distillation. Intermediate supervision trains partial-video readouts for retrieval, while self-distillation regularizes them toward the final representation. Query-aware selection produces query-conditioned representations for candidate-set scoring and reranking, whereas query-independent selection enables reusable corpus indexing. Under the full training recipe, HourVideo retrieval Hit@1 increases from 54.2 to 70.7 and from 57.4 to 72.8 for 2B and 8B Qwen3-VL-Embedding backbones, respectively. Gains extend to the evaluated moment-retrieval and video-QA tasks, and the streaming head transfers to a second Qwen-family embedding backbone. These results support streaming latent aggregation as an effective approach to integrating long-video evidence into fixed-dimensional representations.

1 INTRODUCTION

Multimodal embedding models provide a common representation interface for retrieving, organizing, and comparing text, images, and videos (Radford et al., 2021; Jia et al., 2021; Girdhar et al., 2023; Meng et al., 2026). Their fixed-dimensional outputs are attractive for large collections because candidate representations can be indexed once and searched efficiently (Johnson et al., 2021; Karpukhin et al., 2020). Extending this interface to long videos is difficult, however, because the visual input grows with duration whereas the model's visual-token budget remains bounded (Zhang et al., 2025a; Shen et al., 2025). A video may span tens of minutes, while the evidence that determines its relevance to a query may occupy only a few seconds (Chandrasegaran et al., 2024; Wu et al., 2024). Consequently, the quality of a long-video embedding depends not only on semantic alignment, but also on whether the model observes the relevant evidence and retains it while processing the rest of the sequence.

Existing pipelines usually address the input constraint by sampling a fixed number of frames. Uniform sampling provides broad temporal coverage at low cost, but can miss brief relevant events when a fixed frame budget is spread across a longer video (Yu et al., 2023; Tang et al., 2025). Query-aware selection concentrates a limited budget on semantically relevant frames (Wang et al., 2025b; Tang et al., 2025), but it does not by itself solve the representation problem. Selected frames may be temporally fragmented, and encoding all of them in one context couples the frame budget to the backbone's memory cost. Moreover, query-aware selection changes the retrieval setting: because the selected frames depend on the query, the resulting video representation cannot generally

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.

be reused as a single query-independent corpus embedding. A practical system must therefore distinguish query-independent indexing from query-conditioned candidate scoring, as in embedding-and-reranking pipelines (Li et al., 2026).

A second issue concerns how selected evidence is aggregated. Video-retrieval encoders have explored joint spatiotemporal encoding and pooling of frame-level representations (Bain et al., 2021; Xu et al., 2021; Luo et al., 2022), while memory-augmented video models explicitly retain information across observations (Song et al., 2024; He et al., 2024). These developments motivate studying how a compact streaming state can be trained to produce useful retrieval embeddings throughout a video. This issue is particularly important when relevant observations are distributed across time or when the number of selected frames still exceeds the context that can be processed economically in one forward pass. We therefore study long-video evidence accumulation through a bounded latent state, building on latent bottlenecks and recurrent memory architectures (Jaegle et al., 2021; Bulatov et al., 2022; Hutchins et al., 2022).

We propose **Query-Aware Streaming Latent Reasoning** (QASLR), a post-training framework for long-video embeddings. As shown in Figure 1, QASLR processes selected frames as chronologically ordered clips, using learnable think tokens to accumulate clip-level evidence in a persistent latent memory. An embed token reads out a fixed-dimensional representation after each step. Query-independent frame selection enables reusable video indexing, whereas query-aware selection produces query-conditioned representations for a bounded candidate set, such as candidates returned by a first-stage retriever.

Training combines a final contrastive loss with step-wise supervision (Lee et al., 2015) and self-distillation toward the final embedding (Zhang et al., 2019), aligning both intermediate and complete representations with their targets. Following MMEB and MMEB-V2 (Jiang et al., 2025; Meng et al., 2026), a unified query–positive interface supports retrieval, moment retrieval, classification, and multiple-choice video question answering (QA). Interleaved sub-batching groups compatible input formats during mixed-task training and is evaluated separately from streaming aggregation.

The full QASLR training recipe improves both 2B and 8B Qwen3-VL-Embedding backbones (Li et al., 2026), increasing HourVideo retrieval Hit@1 from 54.2 to 70.7 and from 57.4 to 72.8, respectively, relative to the original checkpoints. Gains extend to the evaluated moment-retrieval and video-QA tasks, while experiments on RzenEmbed (Jian et al., 2025) demonstrate applicability to a second Qwen-family embedding backbone.

Our contributions are threefold:

- We introduce QASLR, a streaming framework that decouples temporal coverage from representation size for long-video embedding. A persistent latent memory incrementally integrates temporally ordered clips and emits a fixed-dimensional embedding without placing all selected frames in one backbone context.
- We design a streaming post-training objective that combines final contrastive learning, step-wise contrastive supervision, and final-embedding self-distillation. This objective directly supervises intermediate representations through the same query–positive interface used across retrieval, classification, and video QA.
- We evaluate QASLR on nine long-video tasks using 2B and 8B Qwen3-VL-Embedding models and demonstrate within-family transfer to RzenEmbed 7B. The full QASLR recipe improves every reported task over the original checkpoints.

2 Related Work

2.1 Multimodal Embeddings and Long-Video Processing

Contrastive learning supports shared image–text and multimodal representations (Radford et al., 2021; Jia et al., 2021; Zhai et al., 2023; Girdhar et al., 2023), with temporal encoders and aggregation extending this approach to video (Bain et al., 2021; Xu et al., 2021; Luo et al., 2022). Instruction-guided retrieval and VLM-based embeddings unify diverse retrieval tasks (Wei et al., 2024; Jiang et al., 2024; 2025; Zhang et al., 2025b), with recent models supporting video and document inputs (Meng et al., 2026; Jian et al., 2025; Li et al., 2026). Long-video benchmarks highlight

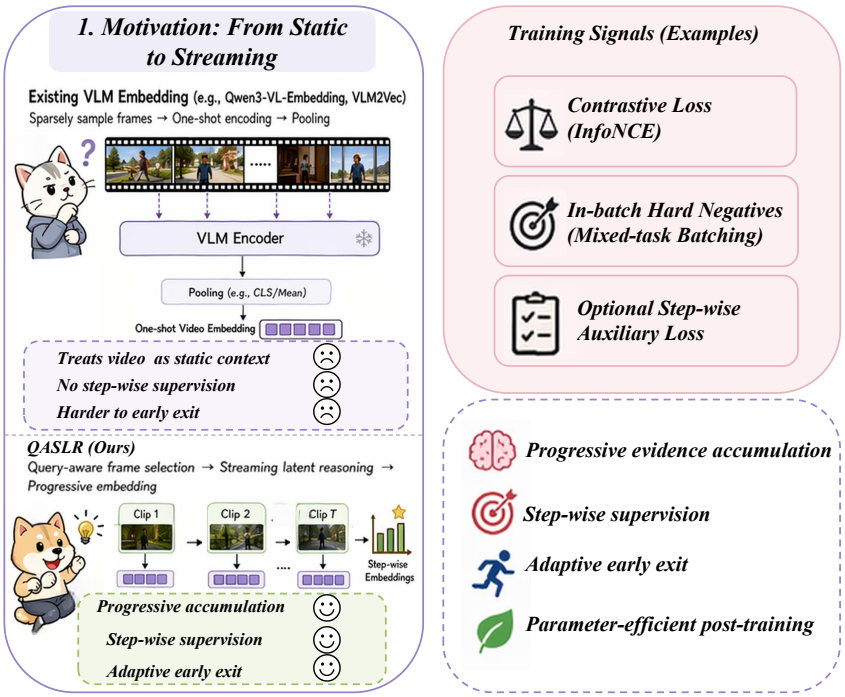


Figure 1: **From one-shot video encoding to streaming evidence accumulation.** The upper-left pathway illustrates a one-shot baseline that pools representations from sparsely sampled frames into a single video embedding. The lower-left pathway shows QASLR processing selected frames as ordered clips and producing embeddings throughout the sequence. The right-hand panels summarize example training signals and the intended capabilities of persistent memory, intermediate supervision, optional early exit, and parameter-efficient post-training.

challenges in temporal coverage and reasoning (Fu et al., 2025; Zhou et al., 2025; Wang et al., 2025a). Existing approaches extend context (Zhang et al., 2025a), compress visual tokens (Shen et al., 2025), aggregate clip descriptions (Zhang et al., 2024a), select query-relevant frames (Yu et al., 2023; Wang et al., 2025b; Tang et al., 2025), or allocate frame-wise resolution (Jin et al., 2026). Temporally grounded datasets provide localized language–video supervision (Krishna et al., 2017; Gao et al., 2017; Lei et al., 2021). Complementing input selection and compression, QASLR accumulates selected clip evidence into a fixed-dimensional embedding.

2.2 Streaming Latent Reasoning and Adaptive Computation

Latent-query architectures compress inputs through cross-attention (Jaegle et al., 2021; 2022; Alayrac et al., 2022; Li et al., 2023), while recurrence and memory tokens carry information across sequence blocks (Dai et al., 2019; Bulatov et al., 2022; Hutchins et al., 2022). Video models similarly retain historical information (Song et al., 2024; He et al., 2024) and support streaming interaction (Chen et al., 2024; Zhang et al., 2024b). Related approaches introduce computation before prediction through pause tokens (Goyal et al., 2024), continuous hidden states (Hao et al., 2025), or explicit reasoning traces for embeddings (Cui et al., 2026). Building on these ideas, QASLR updates persistent latent memory across clips and contrastively supervises intermediate embedding readouts. Early-exit methods adapt computation using intermediate predictions (Xin et al., 2020; Elbayad et al., 2020; Schuster et al., 2022). QASLR's optional exit instead controls the number of observed clips.

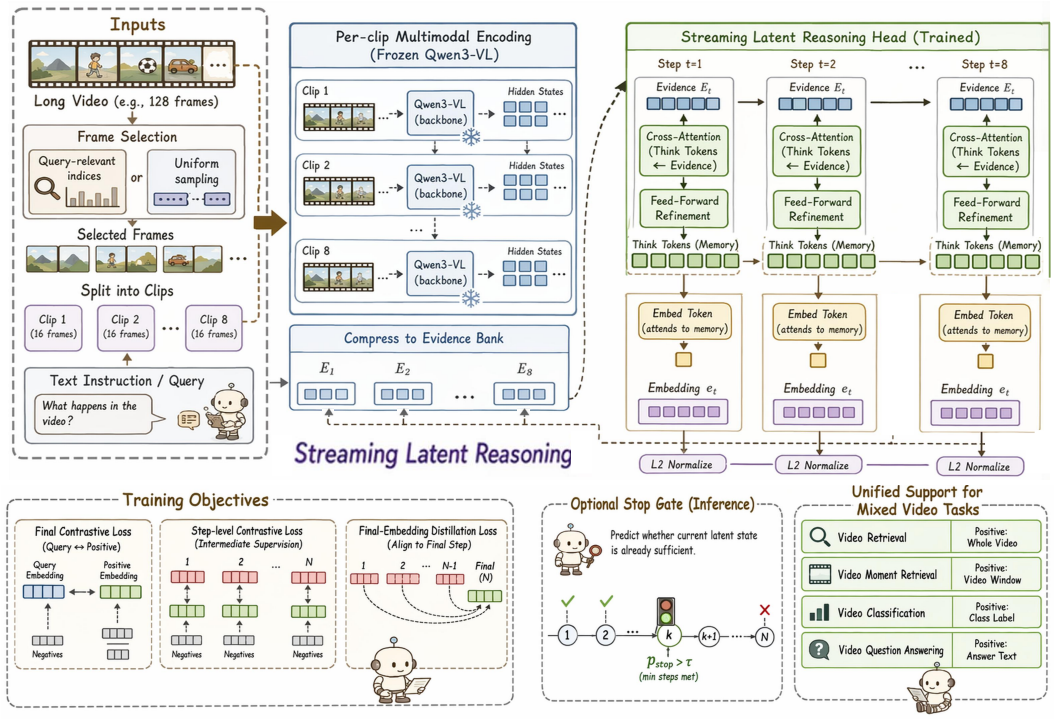


Figure 2: **QASLR architecture, training objectives, and optional early-exit inference.** Selected frames are ordered and partitioned into clips. The illustrated 128-frame configuration contains eight clips of 16 frames each. Per-clip backbone features are compressed into evidence banks that update persistent think-token memory through cross-attention and feed-forward refinement. An embed token reads the memory to produce an L2-normalized embedding after each update. Training combines final contrastive learning, step-wise supervision, and final-embedding self-distillation across the supported video tasks. The optional stop gate terminates processing when its stopping criterion is met.

3 METHOD

3.1 OVERVIEW AND RETRIEVAL SETTINGS

Figure 2 summarizes QASLR. A fixed set of think tokens accumulates clip evidence, and an embed token reads the latent state after each update. Streaming refers to aggregation over preselected clips, not necessarily online frame selection from an unseen video.

Following unified multimodal embedding frameworks (Wei et al., 2024; Jiang et al., 2025; Meng et al., 2026), we formulate all tasks as query–positive pairs (A_i, B_i). The query A_i may contain text, an image–text input, or a video–question input, while B_i may be a video, temporal window, class label, or answer. Non-video and short-video inputs retain the original embedding pathway. Long videos use the proposed streaming pathway.

QASLR supports two retrieval settings, distinguishing reusable dense representations (Karpukhin et al., 2020) from pair-conditioned scoring (Li et al., 2026). In the *indexable setting*, frame selection is query-independent, and each candidate video produces one reusable embedding $\mathbf{v}_j$. In the *query-conditioned setting*, frame selection and video encoding depend on query A_i, producing $\mathbf{v}_{ij}$. The similarity score is

$$s_{ij} = \begin{cases} \mathbf{q}_i^\top \mathbf{v}_j / \tau, & \text{indexable}, \\ \mathbf{q}_i^\top \mathbf{v}_{ij} / \tau, & \text{query-conditioned}, \end{cases} \tag{1}$$

where $\mathbf{q}_i$ is the normalized query embedding and τ is the contrastive temperature. Query-conditioned scoring is restricted to a bounded candidate set, such as the output of a first-stage

retriever. Every candidate for a given query is selected and encoded using the same conditioning procedure.

3.2 Streaming Video Encoding

Query-guided frame selection has been explored for video QA (Yu et al., 2023; Tang et al., 2025). QASLR treats selection as an input stage: for a long video V and optional conditioning input c, the selector returns an ordered sequence of at most $K_{\max}$ indices:

$$\mathcal{I}(V, c) = (k_1, \ldots, k_{K_V}), \qquad k_1 < \cdots < k_{K_V}, \qquad K_V \le K_{\max}. \tag{2}$$

The selected frames are restored to chronological order and divided into clips:

$$X(V, c) = [C_1, \ldots, C_T], \qquad T = \left\lceil \frac{K_V}{L} \right\rceil, \tag{3}$$

where L is the number of frames per clip. We use $K_{\max} = 128$, with $L = 16$ at full budget and fewer frames per clip for shorter inputs. Padding in the final clip is excluded by the attention mask. Ground-truth labels and temporal boundaries are not provided to the frame selector during evaluation.

At step t, the vision-language backbone F_θ, instantiated from Qwen3-VL-Embedding (Li et al., 2026) built on Qwen3-VL (Bai et al., 2025), encodes the current clip together with the task instruction and optional conditioning input:

$$U_t = F_\theta(C_t, c) \in \mathbb{R}^{B \times S_t \times D}, \tag{4}$$

where B, S_t, and D denote the batch size, valid sequence length, and backbone hidden dimension.

To bound the input size of the streaming head, the valid hidden states are compressed into at most M evidence tokens:

$$E_t = \mathrm{Compress}(U_t) \in \mathbb{R}^{B \times M \times D}. \tag{5}$$

The compression operator uses cosine similarity to compress valid hidden states after masking. We use $M = 256$ and $N = 32$ think tokens by default. Evidence banks with fewer than M states are padded and masked in cross-attention. Clips may share a backbone forward pass, but their evidence banks and latent states remain independent.

3.3 Streaming Latent Memory and Embedding Readout

Motivated by compact latent bottlenecks (Jaegle et al., 2021) and recurrent memory tokens (Bulatov et al., 2022), QASLR maintains N learnable think tokens as a persistent latent memory:

$$H_0 \in \mathbb{R}^{B \times N \times D}. \tag{6}$$

The memory is initialized once for each video and is not reset between clips. At streaming step t, the previous memory attends to the current clip evidence using multi-head attention (Vaswani et al., 2017) and layer normalization (Ba et al., 2016):

$$\widetilde{H}_t = \mathrm{LN}\left[H_{t-1} + \mathrm{MHA}\left(\mathrm{LN}(H_{t-1}), \mathrm{LN}(E_t), \mathrm{LN}(E_t)\right)\right], \tag{7}$$

$$H_t = \mathrm{LN}\left[\widetilde{H}_t + \mathrm{FFN}(\widetilde{H}_t)\right]. \tag{8}$$

The recurrent update allows the model to integrate new visual evidence while retaining information from earlier clips. Because the number of think tokens remains fixed, the size of the persistent state does not grow with video duration.

After each update, a learnable embed token e reads the current memory through query-based attention pooling, as in Perceiver IO (Jaegle et al., 2022):

$$r_t = \mathrm{MHA}(e, H_t, H_t), \qquad \mathbf{z}_t = \frac{W r_t}{\|W r_t\|_2}, \tag{9}$$

where W projects the readout into the shared d-dimensional embedding space. The streaming pathway returns

$$Z(V, c) = (\mathbf{z}_1, \ldots, \mathbf{z}_T), \tag{10}$$

and uses $\mathbf{z}_T$ as the final video representation. Intermediate embeddings expose the representation after each observed clip and are used by the auxiliary training objectives below.

3.4 TRAINING OBJECTIVE

For a batch of B query–positive pairs, let $\mathbf{a}_i$ denote the final query-side embedding and $\mathbf{b}_{ij}$ the representation of candidate B_j when scored against A_i. In the indexable setting, $\mathbf{b}_{ij} = \mathbf{b}_j$. In the query-conditioned setting, each candidate is encoded under A_i. The final loss uses an InfoNCE-style contrastive objective (van den Oord et al., 2018; Radford et al., 2021):

$$\mathcal{L}_{\text{final}} = \text{CE}(\widetilde{S}, y), \qquad s_{ij} = \frac{\mathbf{a}_i^\top \mathbf{b}_{ij}}{\tau}, \qquad y_i = i. \tag{11}$$

Query-independent embeddings can be gathered across workers to enlarge the negative pool (Chen et al., 2020). Query-conditioned scores instead require each candidate to be encoded under the current query. Embeddings conditioned on another query are not interchangeable. Valid co-positives from the same video are masked from the denominator, while the paired target remains the sole positive (Khosla et al., 2020; Jian et al., 2025).

Here, $\widetilde{S}$ equals S with excluded co-positive logits set to $-\infty$. Following deep supervision (Lee et al., 2015), we also train intermediate embeddings. Let $\mathcal{B}_t$ contain pairs with valid step-t readouts on all streaming sides. Non-streaming embeddings are reused. The step-wise loss is

$$\mathcal{L}_{\text{step}} = \frac{\sum_t |\mathcal{B}_t| \, \text{CE}(\widetilde{S}^{(t)}, y^{(t)})}{\sum_t |\mathcal{B}_t|}, \qquad s_{ij}^{(t)} = \frac{\left(\mathbf{a}_i^{(t)}\right)^\top \mathbf{b}_{ij}^{(t)}}{\tau}. \tag{12}$$

At each step, invalid pairs and padded readouts are excluded, targets are reindexed within $\mathcal{B}_t$, and the same co-positive mask is applied. Text-to-video retrieval reuses text embeddings. Video QA reuses answer embeddings.

To regularize the streaming trajectory, we adapt knowledge distillation (Hinton et al., 2015) and self-distillation (Zhang et al., 2019) by using the final embedding as a stop-gradient teacher:

$$\mathcal{L}_{\text{distill}} = \frac{1}{Z} \sum_{i \in \mathcal{S}} \sum_{t=1}^{T_i - 1} \left[1 - \left(\mathbf{z}_i^{(t)}\right)^\top \text{sg}\left(\mathbf{z}_i^{(T_i)}\right) \right], \tag{13}$$

where $\mathcal{S}$ contains the streamed inputs, T_i is the number of valid steps, and $Z = \sum_{i \in \mathcal{S}} (T_i - 1)$. When $Z = 0$, we define $\mathcal{L}_{\text{distill}} = 0$. Unlike the moving-average target network in BYOL (Grill et al., 2020), the teacher here is the detached final-step embedding from the same forward trajectory.

The complete objective is

$$\mathcal{L} = \mathcal{L}_{\text{final}} + \lambda_{\text{step}} \mathcal{L}_{\text{step}} + \lambda_{\text{distill}} \mathcal{L}_{\text{distill}}. \tag{14}$$

Mixed-task training combines retrieval, moment retrieval, classification, and video QA through the same query–positive interface. Interleaved sub-batching groups examples with compatible task and input formats. Their losses are weighted by the corresponding sub-batch sizes, and gradients are accumulated before one optimizer update.

Only the streaming head is trainable in the QASLR experiments. The backbone is frozen. The appendix describes data processing, computational complexity, and the optional early-exit mechanism.

4 EXPERIMENTS

4.1 EXPERIMENTAL SETUP

Tasks and metrics. We evaluated nine long-video embedding tasks spanning classification, video retrieval, moment retrieval, and multiple-choice QA. These comprised HourVideo (Chandrasegaran et al., 2024) classification, retrieval, and moment retrieval, LVBench (Wang et al., 2025a) retrieval and QA, LongVideoBench (Wu et al., 2024) moment retrieval, MovieChat QA (Song et al., 2024), EgoSchema (Mangalam et al., 2023), and TimeScope (Zohar et al., 2025). We report Hit@1/Hit@5 for classification and retrieval, except LongVideoBench moment retrieval (Hit@1 only), and accuracy for QA. All reported gains are in percentage points.

Table 1: **Long-video embedding performance on Qwen3-VL-Embedding backbones.** The original checkpoints, single-round post-training (PT), and full QASLR recipe are compared at the 2B and 8B scales. CLS, Ret., and MRet. denote classification, video retrieval, and video moment retrieval, respectively. Retrieval-style entries report Hit@1/Hit@5, except LongVideoBench MRet., which reports Hit@1 only. Question-answering (QA) entries report accuracy. Avg. is the unweighted mean across the four QA tasks. All scores are percentages, with higher values indicating better performance. Bold numeric entries identify the best result for each metric within each backbone scale.

Retrieval-style tasks						
Backbone	Method	HourVideo CLS	HourVideo Ret.	HourVideo MRet.	LVBench Ret.	LongVideoBench MRet.
Qwen3-VL 2B	Original	51.1 / 85.1	54.2 / 77.1	24.3 / 62.2	65.4 / 98.1	34.2
	Single-round PT	52.4 / 86.3	55.0 / 76.7	25.2 / 63.0	66.6 / 98.0	35.1
	QASLR	**54.2 / 88.1**	**70.7 / 92.5**	**30.1 / 65.8**	**75.7 / 98.3**	**37.0**
Qwen3-VL 8B	Original	52.2 / 83.9	57.4 / 87.5	24.9 / 63.1	68.1 / **98.4**	33.2
	Single-round PT	53.0 / 84.3	57.8 / 88.4	25.6 / 63.8	69.2 / 98.1	34.5
	QASLR	**57.3 / 89.6**	**72.8 / 95.8**	**27.7 / 66.6**	**77.3** / 98.2	**38.8**
QA-style tasks						
Backbone	Method	MovieChat QA	LVBench QA	EgoSchema	TimeScope	Avg.
Qwen3-VL 2B	Original	54.1	41.1	59.8	38.0	48.3
	Single-round PT	54.5	41.4	60.4	39.2	48.9
	QASLR	**55.2**	**42.3**	**63.5**	**41.3**	**50.6**
Qwen3-VL 8B	Original	55.7	38.9	70.2	80.0	61.2
	Single-round PT	55.2	39.6	71.7	81.3	62.0
	QASLR	**60.3**	**44.7**	**74.5**	**85.6**	**66.3**

Models and training. We compared original Qwen3-VL-Embedding (Li et al., 2026) checkpoints, single-round post-training, and QASLR at 2B and 8B scales. Both post-training methods used the same training data. QASLR selected at most 128 frames, using 16-frame clips at full budget and shorter clips for fewer frames. Training combined final contrastive learning, step-wise contrastive supervision, and final-embedding self-distillation. Only the streaming head was trainable in QASLR. The backbone remained frozen.

Mixed-task data processing. All examples used a common query–positive format. Retrieval paired a query with a video or candidate moment, whereas QA paired a video–question input with its answer text. Selected frames were stored as indices and decoded from source videos on demand. Interleaved sub-batching grouped compatible task and input formats, with its effects examined in Section 4.4.

Comparison protocol. The original checkpoints establish performance before long-video post-training. Single-round post-training uses the mixed-task data without recurrent latent aggregation. Table 1 evaluates the complete QASLR recipe rather than isolated architectural changes. Table 2 additionally compares both post-training methods with sub-batching, while Appendix A.5 clarifies their interpretation. Training and evaluation details are provided in Appendices A.2–A.4.

4.2 Main Results on Qwen3-VL-Embedding

In Table 1, QASLR achieved the highest Hit@1 on all five retrieval-style tasks at both scales. Mean Hit@1 increased from 45.8 to 53.5 at 2B and from 47.2 to 54.8 at 8B, compared with single-round averages of 46.9 and 48.0. The similar gains over the original checkpoints show that improvement was not limited to one model scale.

Video retrieval and classification. The largest gains occurred on full-video retrieval. QASLR achieved 70.7/92.5 HourVideo Hit@1/Hit@5 at 2B and 72.8/95.8 at 8B, exceeding the original checkpoints by 16.5 and 15.4 Hit@1 points, respectively. Relative to single-round post-training, the corresponding Hit@1 gains were 15.7 and 15.0 points. HourVideo Hit@5 also improved by 15.4 and 8.3 points over the original checkpoints, showing gains in both top-one and top-five retrieval. On LVBench retrieval, Hit@1 improved by 10.3 and 9.2 points over the original checkpoints, while Hit@5 remained near 98. Thus, the LVBench gains primarily reflect better top-ranked ordering

Table 2: **Effects of streaming aggregation and interleaved sub-batching.** All configurations use query-aware selection. PT and QASLR are evaluated with or without sub-batching. Ret. and MRet. denote video retrieval and video moment retrieval. Retrieval entries report Hit@1/Hit@5, except LongVideoBench MRet., which reports Hit@1 only. Avg. H@1 averages Hit@1 across the three retrieval tasks. QA entries report accuracy. All scores are percentages, and bold numeric entries mark the best result within each backbone and metric. QASLR with sub-batching achieves the highest retrieval Hit@1, whereas the strongest QA configuration varies across tasks.

Retrieval-style tasks

Backbone	Method	HourVideo Ret.	LVBench Ret.	LongVideoBench MRet.	Avg. H@1
Qwen3-VL 2B	Original	54.2 / 77.1	65.4 / 98.1	34.2	51.3
	Single-round PT	55.0 / 76.7	66.6 / 98.0	35.1	52.2
	Single-round PT w/ sub-batching	66.1 / 84.2	67.5 / 98.2	36.7	56.8
	QASLR w/o sub-batching	55.4 / 78.8	73.3 / 98.0	34.8	54.5
	QASLR w/ sub-batching	**70.7 / 92.5**	**75.7 / 98.3**	**37.0**	**61.1**
Qwen3-VL 8B	Original	57.4 / 87.5	68.1 / **98.4**	33.2	52.9
	Single-round PT	57.8 / 88.4	69.2 / 98.1	34.5	53.8
	Single-round PT w/ sub-batching	67.3 / 93.5	75.8 / 98.0	36.7	59.9
	QASLR w/o sub-batching	59.2 / 89.9	72.4 / 98.3	35.6	55.7
	QASLR w/ sub-batching	**72.8 / 95.8**	**77.3** / 98.2	**38.8**	**63.0**

QA-style tasks

Backbone	Method	MovieChat	LVBench	EgoSchema	TimeScope
Qwen3-VL 2B	Original	54.1	41.1	59.8	38.0
	Single-round PT	54.5	41.4	60.4	39.2
	Single-round PT w/ sub-batching	**58.8**	**47.2**	62.9	**42.4**
	QASLR w/o sub-batching	47.0	36.8	60.7	38.8
	QASLR w/ sub-batching	55.2	42.3	**63.5**	41.3
Qwen3-VL 8B	Original	55.7	38.9	70.2	80.0
	Single-round PT	55.2	39.6	71.7	81.3
	Single-round PT w/ sub-batching	**61.5**	43.3	73.8	84.1
	QASLR w/o sub-batching	53.2	41.7	72.2	80.5
	QASLR w/ sub-batching	60.3	**44.7**	**74.5**	**85.6**

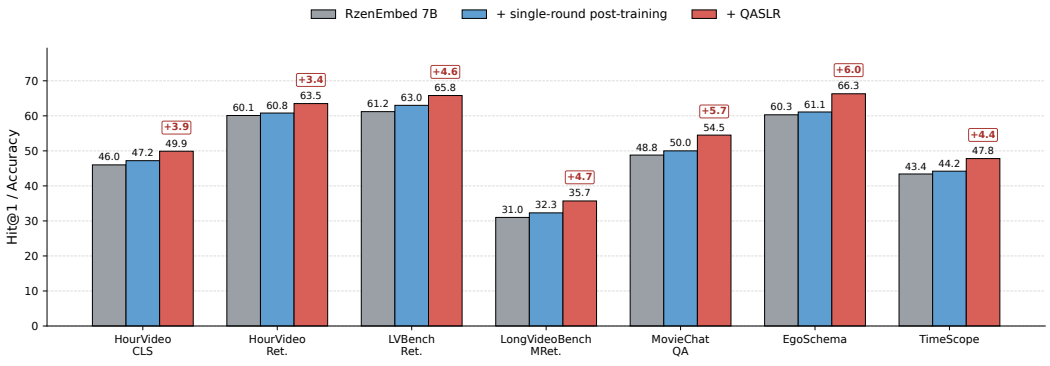


Figure 3: **Within-family transfer of QASLR to RzenEmbed 7B.** Grouped bars compare the original checkpoint, single-round post-training, and QASLR across seven tasks using the same RzenEmbed backbone. Classification (CLS), video retrieval (Ret.), and moment retrieval (MRet.) report Hit@1. Question-answering tasks report accuracy. All scores are percentages, and annotations above the QASLR bars give absolute improvements over the original checkpoint in percentage points. QASLR exceeds both baselines on every displayed task, with gains of 3.4–6.0 points over the original checkpoint.

rather than expanded top-five coverage. HourVideo classification improved more moderately, from 51.1 to 54.2 at 2B and from 52.2 to 57.3 at 8B.

Moment retrieval. HourVideo moment-retrieval Hit@1 increased from 24.3 to 30.1 at 2B and from 24.9 to 27.7 at 8B, with Hit@5 also improving at both scales. LongVideoBench Hit@1 rose from 34.2 to 37.0 and from 33.2 to 38.8, respectively. The larger HourVideo gain at 2B and larger LongVideoBench gain at 8B indicate task-dependent effects of model scale.

Video question answering. QASLR improved all four QA tasks over both baselines in Table 1. Mean accuracy increased from 48.3 to 50.6 at 2B and from 61.2 to 66.3 at 8B relative to the original checkpoints. The 8B gains were 4.6 points on MovieChat, 5.8 on LVBench, 4.3 on EgoSchema, and 5.6 on TimeScope. Improvements at 2B ranged from 1.1 to 3.7 points. Single-round post-training achieved mean QA accuracies of 48.9 and 62.0, so the additional gains with QASLR were larger at 8B. These results concern the complete recipe. Comparison with sub-batched single-round training reveals a less uniform QA advantage, as discussed below.

4.3 Generalization to Another VLM Embedding Backbone

Figure 3 compares the original RzenEmbed 7B checkpoint, single-round post-training, and QASLR. This Qwen2-VL-based backbone provides a test beyond Qwen3-VL-Embedding using the same training formulation. QASLR exceeded both baselines on all seven displayed tasks, with gains of 3.4–6.0 points over the original checkpoint. Improvements span classification, retrieval, moment retrieval, and QA, rather than a single task family. Relative to single-round post-training, QASLR raised EgoSchema accuracy from 61.1 to 66.3 and LongVideoBench moment-retrieval Hit@1 from 32.3 to 35.7. Because both backbones belong to the Qwen family, this experiment supports within-family transfer, not generalization across unrelated architectures.

4.4 Effect of Interleaved Sub-Batching

Table 2 shows that the sub-batching configuration, including a larger effective batch, improved QASLR's mean retrieval Hit@1 from 54.5 to 61.1 at 2B and from 55.7 to 63.0 at 8B. HourVideo retrieval gains were 15.3 and 13.6 points, respectively. LVBench retrieval improved by 2.4 and 4.9 Hit@1 points, while LongVideoBench moment retrieval improved by 2.2 and 3.2 points. At 2B, MovieChat and LVBench QA fell below the original checkpoint without sub-batching, reaching 47.0 and 36.8. Sub-batching recovered them to 55.2 and 42.3. At 8B, QA gains ranged from 2.3 points on EgoSchema to 7.1 on MovieChat. Sub-batching therefore benefited both retrieval and QA within the QASLR configuration.

However, sub-batching also improved single-round post-training: mean retrieval Hit@1 increased from 52.2 to 56.8 at 2B and from 53.8 to 59.9 at 8B. With sub-batching, QASLR retained higher Hit@1 on all three retrieval tasks at both scales. Mean scores were 61.1 versus 56.8 at 2B and 63.0 versus 59.9 at 8B. At the task level, HourVideo Hit@1 gains over this baseline were 4.6 points at 2B and 5.5 at 8B. The corresponding margins were 8.2 and 1.5 points on LVBench, and 0.3 and 2.1 on LongVideoBench, showing task- and scale-dependent differences. QA results were mixed: single-round post-training scored higher on three of four tasks at 2B and on MovieChat at 8B. These comparisons show substantial gains from the batching configuration, while QASLR's advantage over the sub-batched baseline is consistent for retrieval but task-dependent for QA.

5 Conclusion

We introduced QASLR, a post-training framework that represents long videos through a fixed-size latent memory updated over temporally ordered clips. The full training recipe improved retrieval, moment retrieval, classification, and multiple-choice QA relative to the original checkpoints at both the 2B and 8B scales. The streaming head also transferred to a second Qwen-family embedding model. QASLR stores frame indices rather than duplicated images and preserves the downstream embedding interface. Its query-aware path produces query-conditioned representations and is therefore intended for bounded candidate sets or second-stage reranking. Query-independent selection remains necessary when each video must be indexed once and reused across queries. Sub-batching benefits both single-round and streaming models, although improvements in QA remain task-dependent. Future work could assess generalization beyond Qwen-family backbones and evaluate whether adaptive stopping can reduce inference cost while preserving task performance.

AI USE STATEMENT

Generative AI tools were used solely to improve the clarity and readability of the manuscript. They were not used to formulate scientific claims, design experiments, generate results, or interpret findings. All AI-assisted edits were reviewed by the authors, who take full responsibility for the manuscript.

ETHICS STATEMENT

This work does not involve human participants, personally identifiable information, or private data. We recognize that the proposed method may inherit biases and limitations from its training data and underlying models. Potential misuse and unintended deployment risks should therefore be carefully considered in real-world applications. All authors have reviewed the ethical implications of this work and take responsibility for its content.

REPRODUCIBILITY STATEMENT

We provide detailed descriptions of the proposed method, model configuration, training procedure, datasets, evaluation protocols, and hyperparameters in the main paper and appendices. We report the computational resources and random seeds used in our experiments where applicable. The implementation details necessary to reproduce the main results are provided in the paper and appendices.

REFERENCES

Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, et al. Flamingo: A visual language model for few-shot learning. In *Advances in Neural Information Processing Systems*, volume 35, pp. 23716–23736, 2022.

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. Layer normalization. arXiv preprint arXiv:1607.06450, 2016.

Shuai Bai, Yuxuan Cai, Ruizhe Chen, et al. Qwen3-VL technical report. arXiv preprint arXiv:2511.21631, 2025.

Max Bain, Arsha Nagrani, Gül Varol, and Andrew Zisserman. Frozen in Time: A joint video and image encoder for end-to-end retrieval. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 1728–1738, 2021.

Aydar Bulatov, Yuri Kuratov, and Mikhail S. Burtsev. Recurrent memory transformer. In *Advances in Neural Information Processing Systems*, volume 35, pp. 11079–11091, 2022.

Keshigeyan Chandrasegaran, Agrim Gupta, Lea M. Hadzic, Taran Kota, Jimming He, Cristobal Eyzaguirre, Zane Durante, Manling Li, Jiajun Wu, and Li Fei-Fei. HourVideo: 1-hour video-language understanding. In *Advances in Neural Information Processing Systems*, volume 37, pp. 53168–53197, 2024.

Joya Chen, Zhaoyang Lv, Shiwei Wu, Kevin Qinghong Lin, Chenan Song, Difei Gao, Jia-Wei Liu, Ziteng Gao, Dongxing Mao, and Mike Zheng Shou. VideoLLM-online: Online video large language model for streaming video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 18407–18418, 2024.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119, pp. 1597–1607, 2020.

Xuanming Cui, Jianpeng Cheng, Hong-You Chen, et al. Think then embed: Generative context improves multimodal embedding. In *International Conference on Learning Representations*, pp. 2690–2709, 2026.

Zihang Dai, Zhilin Yang, Yiming Yang, Jaime Carbonell, Quoc V. Le, and Ruslan Salakhutdinov. Transformer-XL: Attentive language models beyond a fixed-length context. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 2978–2988, 2019.

Maha Elbayad, Jiatao Gu, Edouard Grave, and Michael Auli. Depth-adaptive transformer. In *8th International Conference on Learning Representations*. OpenReview.net, 2020. URL https://iclr.cc/virtual_2020/poster_SJg7KhVKPH.html.

Chaoyou Fu, Yuhan Dai, Yongdong Luo, 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 (CVPR)*, pp. 24108–24118, 2025.

Jiyang Gao, Chen Sun, Zhenheng Yang, and Ram Nevatia. TALL: Temporal activity localization via language query. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pp. 5267–5275, 2017.

Rohit Girdhar, Alaaeldin El-Nouby, Zhuang Liu, Mannat Singh, Kalyan Vasudev Alwala, Armand Joulin, and Ishan Misra. ImageBind: One embedding space to bind them all. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 15180–15190, 2023.

Sachin Goyal, Ziwei Ji, Ankit Singh Rawat, Aditya Krishna Menon, Sanjiv Kumar, and Vaishnavh Nagarajan. Think before you speak: Training language models with pause tokens. In *International Conference on Learning Representations*, pp. 27896–27923, 2024.

Jean-Bastien Grill, Florian Strub, Florent Altché, et al. Bootstrap your own latent - a new approach to self-supervised learning. In *Advances in Neural Information Processing Systems*, volume 33, pp. 21271–21284, 2020.

Shibo Hao, Sainbayar Sukhbaatar, DiJia Su, Xian Li, Zhiting Hu, Jason E. Weston, and Yuandong Tian. Training large language models to reason in a continuous latent space. In *Conference on Language Modeling*, 2025.

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 (CVPR)*, pp. 13504–13514, 2024.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. arXiv preprint arXiv:1503.02531, 2015.

DeLesley Hutchins, Imanol Schlag, Yuhuai Wu, Ethan Dyer, and Behnam Neyshabur. Block-recurrent transformers. In *Advances in Neural Information Processing Systems*, volume 35, pp. 33248–33261, 2022.

Andrew Jaegle, Felix Gimeno, Andy Brock, Oriol Vinyals, Andrew Zisserman, and Joao Carreira. Perceiver: General perception with iterative attention. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 4651–4664. PMLR, 2021. URL https://proceedings.mlr.press/v139/jaegle21a.html.

Andrew Jaegle, Sebastian Borgeaud, Jean-Baptiste Alayrac, Carl Doersch, Catalin Ionescu, David Ding, Skanda Koppula, Daniel Zoran, Andrew Brock, Evan Shelhamer, Olivier J. Hénaff, Matthew M. Botvinick, Andrew Zisserman, Oriol Vinyals, and João Carreira. Perceiver IO: A general architecture for structured inputs & outputs. In *International Conference on Learning Representations*. OpenReview.net, 2022. URL https://iclr.cc/virtual/2022/spotlight/6270.

Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc V. Le, Yunhsuan Sung, Zhen Li, and Tom Duerig. Scaling up visual and vision-language representation learning with noisy text supervision. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139, pp. 4904–4916, 2021.

Weijian Jian, Yajun Zhang, Dawei Liang, Chunyu Xie, Yixiao He, Dawei Leng, and Yuhui Yin. RzenEmbed: Towards comprehensive multimodal retrieval. arXiv preprint arXiv:2510.27350, 2025.

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.

Ziyan Jiang, Rui Meng, Xinyi Yang, Semih Yavuz, Yingbo Zhou, and Wenhu Chen. VLM2Vec: Training vision-language models for massive multimodal embedding tasks. In *International Conference on Learning Representations*, pp. 1255–1279, 2025.

Song Jin, Zhongtao Jiang, Chenglei Shen, Huanxuan Liao, Haozhe Chi, Zhiwei Wang, Kun Xu, and Yong Liu. Allocate before you embed: Adaptive visual input allocation for video embeddings. arXiv preprint arXiv:2609.01778, 2026.

Jeff Johnson, Matthijs Douze, and Hervé Jégou. Billion-scale similarity search with GPUs. *IEEE Transactions on Big Data*, 7(3):535–547, July 2021. doi: 10.1109/TBDATA.2019.2921572. First published online in 2019.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 6769–6781, 2020.

Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. Supervised contrastive learning. In *Advances in Neural Information Processing Systems*, volume 33, pp. 18661–18673, 2020.

Ranjay Krishna, Kenji Hata, Frederic Ren, Li Fei-Fei, and Juan Carlos Niebles. Dense-captioning events in videos. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pp. 706–715, 2017.

Chen-Yu Lee, Saining Xie, Patrick Gallagher, Zhengyou Zhang, and Zhuowen Tu. Deeply-supervised nets. In *Proceedings of the Eighteenth International Conference on Artificial Intelligence and Statistics*, volume 38, pp. 562–570, 2015.

Jie Lei, Tamara L Berg, and Mohit Bansal. Detecting moments and highlights in videos via natural language queries. In *Advances in Neural Information Processing Systems*, volume 34, pp. 11846–11858, 2021.

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 *Proceedings of the 40th International Conference on Machine Learning*, volume 202, pp. 19730–19742, 2023.

Mingxin Li, Yanzhao Zhang, Dingkun Long, et al. Qwen3-VL-Embedding and Qwen3-VL-Reranker: A unified framework for state-of-the-art multimodal retrieval and ranking. arXiv preprint arXiv:2601.04720, 2026.

Huaishao Luo, Lei Ji, Ming Zhong, Yang Chen, Wen Lei, Nan Duan, and Tianrui Li. CLIP4Clip: An empirical study of CLIP for end to end video clip retrieval and captioning. *Neurocomputing*, 508:293–304, 2022. doi: 10.1016/j.neucom.2022.07.028.

Karttikeya Mangalam, Raiymbek Akshulakov, and Jitendra Malik. EgoSchema: A diagnostic benchmark for very long-form video language understanding. In *Advances in Neural Information Processing Systems*, volume 36, pp. 46212–46244, 2023.

Rui Meng, Ziyan Jiang, Ye Liu, Mingyi Su, Xinyi Yang, Yuepeng Fu, Can Qin, Raghuveer Thirukovalluru, Xuan Zhang, Zeyuan Chen, Ran Xu, Caiming Xiong, Yingbo Zhou, Wenhu Chen, and Semih Yavuz. VLM2Vec-V2: Advancing multimodal embedding for videos, images, and visual documents. *Transactions on Machine Learning Research*, 2026. URL `https://mlanthology.org/tmlr/2026/meng2026tmlr-vlm2vecv2/`.

Alec Radford, Jong Wook Kim, Chris Hallacy, et al. Learning transferable visual models from natural language supervision. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139, pp. 8748–8763, 2021.

Tal Schuster, Adam Fisch, Jai Gupta, Mostafa Dehghani, Dara Bahri, Vinh Q. Tran, Yi Tay, and Donald Metzler. Confident adaptive language modeling. In *Advances in Neural Information Processing Systems*, volume 35, pp. 17456–17472, 2022.

Xiaoqian Shen, Yunyang Xiong, Changsheng Zhao, et al. LongVU: Spatiotemporal adaptive compression for long video-language understanding. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267, pp. 54582–54599, 2025.

Enxin Song, Wenhao Chai, Guanhong Wang, 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 (CVPR)*, pp. 18221–18232, 2024.

Xi Tang, Jihao Qiu, Lingxi Xie, Yunjie Tian, Jianbin Jiao, and Qixiang Ye. Adaptive keyframe sampling for long video understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 29118–29128, 2025.

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. arXiv preprint arXiv:1807.03748, 2018.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30, 2017. URL https://proceedings.neurips.cc/paper/7181-attention-is-all-you-need.

Weihan Wang, Zehai He, Wenyi Hong, et al. LVBench: An extreme long video understanding benchmark. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 22958–22967, 2025a.

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 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3272–3283, 2025b.

Cong Wei, Yang Chen, Haonan Chen, Hexiang Hu, Ge Zhang, Jie Fu, Alan Ritter, and Wenhu Chen. UniIR: Training and benchmarking universal multimodal information retrievers. In *Computer Vision – ECCV 2024*, volume 15145 of *Lecture Notes in Computer Science*, pp. 387–404. Springer, 2024. doi: 10.1007/978-3-031-73021-4_23. URL https://www.ecva.net/papers/eccv_2024/papers_ECCV/html/11927_ECCV_2024_paper.php.

Haoning Wu, Dongxu Li, Bei Chen, and Junnan Li. LongVideoBench: A benchmark for long-context interleaved video-language understanding. In *Advances in Neural Information Processing Systems*, volume 37, pp. 28828–28857, 2024.

Ji Xin, Raphael Tang, Jaejun Lee, Yaoliang Yu, and Jimmy Lin. DeeBERT: Dynamic early exiting for accelerating BERT inference. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 2246–2251, 2020.

Hu Xu, Gargi Ghosh, Po-Yao Huang, Dmytro Okhonko, Armen Aghajanyan, Florian Metze, Luke Zettlemoyer, and Christoph Feichtenhofer. VideoCLIP: Contrastive pre-training for zero-shot video-text understanding. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 6787–6800, 2021.

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

Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid loss for language image pre-training. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 11975–11986, 2023.

Ce Zhang, Taixi Lu, Md Mohaiminul Islam, Ziyang Wang, Shoubin Yu, Mohit Bansal, and Gedas Bertasius. A simple LLM framework for long-range video question-answering. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 21715–21737, 2024a.

Haoji Zhang, Yiqin Wang, Yansong Tang, Yong Liu, Jiashi Feng, Jifeng Dai, and Xiaojie Jin. Flash-VStream: Memory-based real-time understanding for long video streams. arXiv preprint arXiv:2406.08085, 2024b.

Linfeng Zhang, Jiebo Song, Anni Gao, Jingwei Chen, Chenglong Bao, and Kaisheng Ma. Be your own teacher: Improve the performance of convolutional neural networks via self distillation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 3713–3722, October 2019. doi: 10.1109/ICCV.2019.00381. URL https://openaccess.thecvf.com/content_ICCV_2019/html/Zhang_Be_Your_Own_Teacher_Improve_the_Performance_of_Convolutional_Neural_ICCV_2019_paper.html.

Peiyuan Zhang, Kaichen Zhang, Bo Li, Guangtao Zeng, Jingkang Yang, Yuanhan Zhang, Ziyue Wang, Haoran Tan, Chunyuan Li, and Ziwei Liu. Long context transfer from language to vision. *Transactions on Machine Learning Research*, 2025a. URL https://mlanthology.org/tmlr/2025/zhang2025tmlr-long/.

Xin Zhang, Yanzhao Zhang, Wen Xie, Mingxin Li, Ziqi Dai, Dingkun Long, Pengjun Xie, Meishan Zhang, Wenjie Li, and Min Zhang. Bridging modalities: Improving universal multimodal retrieval by multimodal large language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9274–9285, 2025b.

Junjie Zhou, Yan Shu, Bo Zhao, et al. MLVU: Benchmarking multi-task long video understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 13691–13701, 2025.

Orr Zohar, Xiaohan Wang, Rui Li, Andrés Marafioti, Miquel Farré, Merve Noyan, Leandro von Werra, Serena Yeung-Levy, and Thomas Wolf. Apollo2: Exploring the long-video frontier of large multimodal models. Official citation for the TimeScope benchmark, 2025. URL https://huggingface.co/datasets/Apollo-LMMs/TimeScope.

A APPENDIX

This appendix describes the training configuration, evaluation data, component comparisons, and theoretical properties of QASLR.

A.1 DETAILED ALGORITHM

Table 3 summarizes the QASLR training pathway. Each example is represented through a query–positive interface, and selected frame indices are decoded on demand. Streaming memory is initialized once for each input and retained across its clips.

Table 3: **QASLR post-training procedure for unified query–positive pairs.** Selected frame indices are decoded on demand and restored to chronological order before clip-wise backbone encoding. A bounded evidence bank updates the persistent think-token memory, which is initialized once per video and read out after every clip. The optimization combines final contrastive learning with the enabled auxiliary objectives. Interleaved sub-batching groups compatible task and input formats before parameter updates. The gate objective applies only when training the optional early-exit module; it is not required for fixed-length streaming inference.

Input: training pairs $\{(q_i, p_i)\}_{i=1}^{P}$, selected frame indices, VLM backbone F_θ, streaming head G_ϕ, temperature τ, loss weights $\lambda_{\text{step}}, \lambda_{\text{distill}}$.

1. Sample a mini-batch of query-positive pairs. If interleaved sub-batching is enabled, group examples with similar task and input format into homogeneous sub-batches.
2. For each streaming video item, decode selected frame indices from the original video and restore chronological order.
3. Split selected frames into clips $[C_1, \ldots, C_T]$.
4. For each clip C_t, encode $U_t = F_\theta(C_t, c)$, where c is the permitted conditioning input, omitted for query-independent video encoding.
5. Compress valid hidden states into a fixed-size evidence bank E_t.
6. Initialize think-token memory H_0 and update it sequentially: $H_t = \text{Update}_\phi(H_{t-1}, E_t)$.
7. Read out step embeddings $z_t = \text{Readout}_\phi(H_t)$.
8. Compute the masked final contrastive loss and enabled auxiliary embedding losses; train the optional gate separately after embedding training.
9. Update the streaming-head parameters while keeping the pretrained backbone frozen.

State isolation and padding. Each video trajectory has an independent memory state. Packing clips into a backbone batch does not permit attention or memory sharing between different inputs. Padded evidence tokens and unavailable readouts are masked. Non-streaming inputs retain their original embedding pathway.

Inference. Fixed-length inference returns the final valid readout. Optional early exit returns an intermediate readout when the stopping rule is satisfied. Early-exit use in local-clip QA is discussed separately in Section A.10.

A.2 IMPLEMENTATION DETAILS

Backbone input and output. The image processor uses a pixel-area range of 4,096–1,310,720, with height and width aligned to multiples of 32. The output embedding dimensions are 2,048 for Qwen3-VL-Embedding 2B and 4,096 for Qwen3-VL-Embedding 8B.

Streaming configuration. The main QASLR configuration selects at most 128 frames and uses 32 think tokens and at most 256 evidence tokens per clip. At the full frame budget, 16 frames per clip give eight streaming updates. For inputs with fewer selected frames, the number of frames per clip is reduced. Only valid frames and readouts contribute to attention and auxiliary losses. Compression uses cosine similarity between valid hidden-state representations to construct the evidence bank.

Trainable modules. Only the streaming latent reasoning head is updated in QASLR, while pretrained backbone parameters remain frozen. The trainable parameter counts are approximately 200 million for the 2B configuration and 600 million for the 8B configuration. These counts refer to the trainable head rather than the total model size.

Optimization. Post-training uses AdamW with a learning rate of 10^{-4} and approximately three to five epochs, depending on the configuration. Training runs on NVIDIA H100 GPUs, with a per-GPU batch size of 4 or 8 and a global effective batch size of 256 or 512 across configurations. The step-wise contrastive and self-distillation weights are $\lambda_{\text{step}} = 0.1$ and $\lambda_{\text{distill}} = 0.01$, respectively. The reported experimental results are averaged over three random seeds.

Interleaved sub-batching. Compatible examples are grouped by task and input format. Negative candidates are shared across sub-batches and GPUs, and the effective batch size increases in the sub-batching configuration. The sub-batching configuration therefore changes both the effective optimization batch size and the negative pool. For query-conditioned scoring, shared candidates retain the query-conditioning requirement defined in the main objective.

Frame storage. Selected frames are recorded as indices with optional timestamps and decoded from the original videos when needed. This avoids storing duplicated image copies for every query–video annotation. Frame indices and timestamps provide references to the original video without duplicating the selected images.

A.3 Additional Experimental Details

Post-training data. Post-training primarily uses LLaVA-Video-178K, supplemented with 10,000 ActivityNet videos.

Benchmark annotations and task construction. QA annotations are taken from the corresponding original benchmark datasets. The four QA results in the main comparison are MovieChat, LVBench, EgoSchema, and TimeScope. Classification, video retrieval, and moment retrieval use sampled and filtered subsets of the source datasets. Their sizes vary by dataset and task, from tens to thousands of examples. GPT-4o assists annotation by constructing pairs of visual frames from video segments and textual questions. We will publicly release the complete evaluation sets for all reported tasks to support reproducibility and future research on long-video understanding.

Candidate sets. These evaluations use bounded candidate sets. Video-retrieval questions generally have 4 or 8 candidates, and moment-retrieval questions generally have 8 or 16 candidate segments, depending on video length. Classification uses 4–8 categories, depending on the dataset. QA questions generally contain four answer options. Retrieval scores measure ranking within these task-specific candidate sets. The candidate-count distribution also affects Hit@5: with four candidates and a valid positive, top-five retrieval necessarily includes that positive.

QA scoring. The video–question input is embedded on the query side, and the original answer options are embedded on the candidate side. The selected answer is the option with the highest embedding similarity. No additional QA prompt is introduced beyond this input construction.

Metrics and averaging. Retrieval-style tasks report Hit@1/Hit@5 where available, and QA reports accuracy. LongVideoBench moment retrieval reports Hit@1. Results are averaged over three random seeds. Task averages include only the tasks specified in the relevant table and do not weight by the number of examples.

A.4 Evaluation Modes and Query Conditioning

Query-aware frame selection. We use Q-Frame for training-free frame selection conditioned on text. For a given retrieval query, the same query conditions selection and encoding for both positive and negative candidates. Ground-truth labels and annotated temporal boundaries are not provided to the selector during evaluation. The resulting video representation is query-conditioned and is intended for bounded candidate scoring.

Indexing and two-stage scoring. Query-independent selection and encoding produce one reusable representation per video. Query-aware selection instead requires candidate processing after the query is available. Table 4 distinguishes these deployment modes. In its two-stage comparison, the single-round and QASLR methods use the same first-stage candidate lists, and first-stage misses

Table 4: **Indexable and query-conditioned retrieval on HourVideo with Qwen3-VL-Embedding 2B.** Uniform-selection methods produce one query-independent representation per video that can be reused across queries. SB denotes interleaved sub-batching; the uniform single-round row does not use SB. Two-stage methods apply query-aware selection and candidate scoring only to the top-M videos returned by a query-independent first stage. First-stage R@M measures positive-video recall before second-stage scoring, and Hit@1 measures final retrieval performance; both are percentages. *First stage* indicates that only the initial retrieval stage is pre-indexable, and a dash marks an inapplicable shortlist-recall entry. Latency is measured in seconds per query on NVIDIA H100 hardware with a warm cache.

Mode	Query-aware	Pre-indexable	First-stage R@M	Hit@1	Latency (s)
Uniform, single-round	No	Yes	–	55.0	0.18
Uniform, QASLR + SB	No	Yes	–	64.1	0.61
Two-stage, single-round + SB	Yes	First stage	95.8	66.1	1.26
Two-stage, QASLR + SB	Yes	First stage	95.8	**70.7**	1.72

remain in the denominator of final Hit@1. Table 2 uses query-aware selection throughout, whereas Table 4 additionally evaluates query-independent indexing.

First-stage recall and final accuracy. Let R denote shortlist size, to distinguish it from the evidence-bank size M. Let $\mathcal{C}_R(q)$ be the shortlist and $\mathcal{P}(q)$ the valid positive set. First-stage recall measures whether $\mathcal{C}_R(q) \cap \mathcal{P}(q)$ is nonempty, while final Hit@1 measures whether the top-ranked candidate is positive. A first-stage miss contributes zero to the latter. Consequently, end-to-end Hit@1 cannot exceed first-stage recall. The reported values imply approximate top-one success rates conditional on a first-stage hit of $66.1/95.8 = 69.0\%$ and $70.7/95.8 = 73.8\%$. Both conditional rates are calculated from the rounded values in Table 4.

A.5 Attribution of the Main Gain

Training configuration. All configurations in Table 2, including the original checkpoints, use query-aware frame selection. At 2B, single-round HourVideo Hit@1 increases from 55.0 to 66.1 with the sub-batching configuration. This configuration combines task-compatible grouping, cross-sub-batch and cross-GPU negatives, and a larger effective batch size. The comparison measures their combined contribution to post-training.

Streaming contribution. Relative to the original checkpoints, the complete QASLR recipe improves HourVideo Hit@1 by 16.5 and 15.4 points at 2B and 8B. Against single-round training with sub-batching, the corresponding gains are 4.6 and 5.5 points. This comparison evaluates the streaming training configuration with sub-batching present in both methods.

Task-dependent QA gains. At 2B, QASLR minus sub-batched single-round differences are -3.6, -4.9, $+0.6$, and -1.1 points on MovieChat, LVBench, EgoSchema, and TimeScope. At 8B, the corresponding differences are -1.2, $+1.4$, $+0.7$, and $+1.5$ points. Across these comparisons, the retrieval advantage is more consistent than the QA advantage.

A.6 Does the Latent Memory Use Temporal Evidence?

The aggregation controls retain the selected frames, backbone features, training setup, and objectives except for the component being ablated. Mean pooling, a recurrent aggregator, and a Transformer aggregator are compared with QASLR. Clip shuffling, reversal, and memory reset are applied only at test time, without retraining. The last-clip-only control retains the final clip of the selected chronological frame sequence and also does not retrain the model.

Relative to full QASLR, HourVideo Hit@1 decreases by 6.6 points with shuffled clips, 5.7 with reversed clips, 10.4 with memory reset, and 13.8 with only the last clip. QASLR exceeds the reported Transformer aggregator by 4.5 points. These controls support the usefulness of cross-clip state and chronological inputs under the evaluated settings. Because the perturbations are applied only at inference, the effects also include sensitivity to a change in the input or state-update distribution.

Table 5: **Aggregation, clip-order, and memory controls with Qwen3-VL-Embedding 2B.** Variants use the same selected frames and backbone features, except that the last-clip-only control retains only the final clip. Alternative aggregators test pooling and additional modeling capacity, while QASLR controls perturb clip order or remove cross-clip memory. *Ordered* indicates chronological clip presentation, not whether an aggregator is sensitive to order; *Persistent memory* indicates retention of information across clips. Ret. Hit@1 is HourVideo retrieval Hit@1, and QA Acc. is EgoSchema accuracy. Scores are percentages, and bold numeric entries mark the highest value in each metric.

Aggregator	Ordered	Persistent memory	Ret. Hit@1	QA Acc.
Mean pooling	Yes	No	58.6	60.2
Recurrent aggregator	Yes	Yes	63.5	61.8
Transformer aggregator	Yes	Yes	66.2	62.7
QASLR, shuffled clips	No	Yes	64.1	61.6
QASLR, reversed clips	No	Yes	65.0	61.9
QASLR, reset memory	Yes	No	60.3	60.8
QASLR, last clip only	Yes	No	56.9	59.7
QASLR	Yes	Yes	**70.7**	**63.5**

Table 6: **Auxiliary-objective ablation and intermediate retrieval quality on HourVideo.** The four Qwen3-VL-Embedding 2B configurations independently enable step-wise contrastive supervision and final-embedding self-distillation, with the final contrastive loss active throughout. Entries report retrieval Hit@1 (%) after processing 25%, 50%, 75%, or all selected clips; *Final* denotes the complete sequence. *Mean H@1* is the unweighted mean of the four checkpoint scores. Bold numeric entries mark the highest value in each column, including intermediate and final performance.

Step loss	Distillation	25%	50%	75%	Final	Mean H@1
No	No	43.8	54.6	62.8	67.9	57.3
Yes	No	49.7	59.8	66.5	69.8	61.5
No	Yes	51.2	61.7	67.9	70.1	62.7
Yes	Yes	**55.6**	**64.8**	**69.1**	**70.7**	**65.1**

A.7 Frame Budget and Streaming Depth

The full-budget configuration uses $K = 128$ frames, $L = 16$ frames per clip, and eight memory updates. For inputs with fewer selected frames, QASLR reduces the number of frames per clip. Frame budget and clip size determine different aspects of processing: K controls the observations supplied to the backbone, while L controls per-call context length.

For fixed K, changing L also changes the number of updates $T = \lceil K/L \rceil$. At fixed L, increasing K changes both temporal coverage and total computation. These dependencies distinguish the storage occupied by the persistent memory from the cost of obtaining and encoding its evidence.

A.8 Objective Ablations and Intermediate Embedding Quality

We isolate the effects of step-wise contrastive supervision and final-embedding self-distillation using a 2×2 objective ablation. The final contrastive loss is active in every configuration, and all other settings are identical across the four loss combinations. For text-to-video retrieval, the text query embedding is shared across steps and contrasted with each intermediate video embedding. For examples with unequal numbers of steps, padded steps are excluded from both the contrastive and distillation losses.

Table 6 reports Hit@1 after 25%, 50%, 75%, and all selected clips. Mean checkpoint Hit@1 is the unweighted average of these four scores, summarizing intermediate and final retrieval quality without treating the checkpoints as an integrated curve.

Negative construction. Consistent with the main objective, each query retains its paired target as the positive. Other valid co-positives from the same source video are excluded from the contrastive denominator, rather than added to a multi-positive target. The mask removes valid co-positives from the denominator while preserving the paired target.

Final and intermediate effects. Adding step-wise supervision alone increases final HourVideo Hit@1 from 67.9 to 69.8; distillation alone increases it to 70.1; using both reaches 70.7. The corresponding changes at the 25% checkpoint are $+5.9$, $+7.4$, and $+11.8$ points. Relative to final-only supervision, the mean over the four checkpoints increases from 57.3 to 65.1 with both objectives. The combined final gain is 2.8 points, with larger improvements at the early checkpoint. Both objectives therefore improve partial-video readouts as well as the final embedding in this comparison.

Partial evidence and distillation. An early clip may not yet contain the evidence needed for the paired target. Step-wise supervision therefore encourages a useful partial-video representation, rather than certifying that the answer is already observable. The final teacher also sees evidence unavailable at earlier steps. Alignment to that teacher is a training regularizer, not access to unseen frames at inference. Neither a high checkpoint accuracy nor a low average distillation loss guarantees correctness on every query or every candidate.

A.9 Frame Selection and Evidence Coverage

Q-Frame selects text-relevant frames within the input budget, and QASLR restores their chronological order before aggregation. The selector determines the available observations; the persistent memory combines their evidence across clips. Events omitted by selection remain outside the information available to the streaming head.

A.10 Early Exit and Computational Cost

Optional early exit. For local-clip QA, the optional stopping rule returns an intermediate readout at the first eligible step $t \geq t_{\min}$ satisfying $p_t \geq \eta$. If no eligible step meets the threshold, processing continues to the final readout. The fixed-length retrieval results use the complete sequence. The theoretical analysis below characterizes agreement with full-step predictions under its stated calibration assumption.

Warm-cache latency. Table 4 reports per-query latency on NVIDIA H100 hardware with a warm cache. The two-stage single-round and QASLR configurations take 1.26 s and 1.72 s, respectively. QASLR improves HourVideo Hit@1 by 4.6 points in this comparison, with a latency increase of 0.46 s, approximately 36.5%. The uniform-selection rows compare configurations that additionally differ in sub-batching.

Cost decomposition. For a shortlist $\mathcal{C}_R(q)$ containing at most R candidates, online processing comprises query encoding, search, candidate processing, and ranking:

$$C_{\text{online}}(q) = C_{\text{query}}(q) + C_{\text{search}}(q) + C_{\text{rank}}(q) + \sum_{j \in \mathcal{C}_R(q)} \left[C_{\text{select}}(q,j) + C_{\text{decode}}(q,j) + C_{\text{encode}}(q,j) + C_{\text{head}}(q,j) \right]. \tag{15}$$

This decomposition defines the full online workload separately from the warm-cache measurements above. Query-independent corpus encoding and index construction contribute offline costs, amortized as C_{offline}/Q over Q queries. Cached media can reduce repeated decoding, whereas query-conditioned video encoding remains specific to each query–candidate pair.

A.11 Data Separation and Result Aggregation

Source-video separation. The post-training and evaluation sets have zero overlap in source videos. This separation concerns the constructed post-training corpus and evaluation sets. QA uses original benchmark annotations, while classification and retrieval use the task-specific subsets described in Section A.3.

Result aggregation. Experimental results are averaged over three random seeds. Differences between configurations are expressed in percentage points, and task-level averages use the metrics and task groups identified in each table. Retrieval metrics are interpreted within the candidate sets defined for each task.

A.12 Complexity Analysis

Let K be the selected-frame count, L the clip size, $T = \lceil K/L \rceil$ the number of steps, and D the backbone hidden dimension. The head uses N think tokens, at most M evidence tokens, and an output dimension d. Let $C_{\mathrm{vlm}}(L)$ denote one clip-encoding cost. For the cross-attention, fixed-width-ratio feed-forward network, and readout defined in Section 3, the per-step head cost is

$$C_{\mathrm{head}} = O\big(NMD + (N+M)D^2 + Dd\big)\,. \tag{16}$$

The terms account for attention mixing, token projections and feed-forward transformations, and the output projection, respectively. This expression does not assume an additional latent self-attention layer. For full-step inference,

$$C_{\mathrm{full}} = C_{\mathrm{fixed}} + T\,[C_{\mathrm{vlm}}(L) + C_{\mathrm{head}}]\,, \tag{17}$$

where C_{fixed} includes any up-front selection or preprocessing cost. In query-conditioned retrieval, clip encoding and aggregation are repeated for each query–candidate pair; first-stage retrieval and query encoding are additional costs.

The persistent memory occupies $O(ND)$ values per example. Excluding model parameters and backbone activations, a conventional single-step attention implementation requires working space of order

$$O((N+M)D + NM + d)\,. \tag{18}$$

These are inference-state bounds, not bounds on training peak memory. Backpropagation through the streaming head may retain activations across steps even when the backbone is frozen; checkpointing or recomputation changes that trade-off.

If the model exits after $\widehat{T} \le T$ steps, then, for fixed per-step cost and negligible gate overhead,

$$\mathbb{E}[C_{\mathrm{exit}}] = C_{\mathrm{fixed}} + \mathbb{E}[\widehat{T}]\,[C_{\mathrm{vlm}}(L) + C_{\mathrm{head}}]\,. \tag{19}$$

When fixed costs are negligible, the remaining cost ratio is approximately $\mathbb{E}[\widehat{T}]/T$, and the saved fraction is $1-\mathbb{E}[\widehat{T}]/T$. Up-front frame selection and decoding can reduce end-to-end savings relative to this approximation.

A.13 Theoretical Analysis

The following conditional results describe perturbation stability and agreement with full-step retrieval. They do not establish that optimization attains the stated assumptions or that QASLR is optimal.

Assumption 1: regular memory update and readout. Let $H_t = U_\phi(H_{t-1}, E_t)$. On the states under consideration, assume

$$\|U_\phi(H,E) - U_\phi(H',E')\|_F \le \rho\|H-H'\|_F + \beta\|E-E'\|_F, \tag{20}$$

with $\rho, \beta \ge 0$, and assume the normalized readout is L_R-Lipschitz. For normalization, this requires the pre-normalization readout norm to remain bounded away from zero, in addition to regularity of the preceding operations.

Proposition 1: propagation of evidence perturbations. Consider two executions with the same initial memory and $\|E_t - \widetilde{E}_t\|_F \le \epsilon$ at every step. Under Assumption 1,

$$\|H_T - \widetilde{H}_T\|_F \le \beta\epsilon \sum_{k=0}^{T-1} \rho^k, \qquad \|z_T - \widetilde{z}_T\|_2 \le L_R\beta\epsilon \sum_{k=0}^{T-1} \rho^k. \tag{21}$$

Proof. Writing $\delta_t = \|H_t - \widetilde{H}_t\|_F$ gives $\delta_t \le \rho\delta_{t-1} + \beta\epsilon$. Unrolling from $\delta_0 = 0$ yields the memory bound; applying readout Lipschitz continuity yields the embedding bound. □

If $\rho < 1$, the memory bound is at most $\beta\epsilon/(1-\rho)$, uniformly in T. If $\rho = 1$, it grows linearly, and if $\rho > 1$, it can grow geometrically. The architecture alone does not verify a contractive update.

Scope of step-wise supervision. The step-wise contrastive objective directly evaluates intermediate retrieval scores, whereas the final-only objective directly evaluates only the terminal readout. This distinction motivates intermediate supervision, but does not prove it is necessary for early exit. Nor does minimizing contrastive loss alone establish a prescribed distance to an ideal intermediate embedding. Both shared parameters and other auxiliary objectives can influence intermediate states without a step-wise contrastive term.

Proposition 2: ranking preservation under an embedding-distance bound. For a fixed normalized query q, let p be the positive candidate and n index negatives. Define

$$\gamma_t = q^\top z_t^p - \max_n q^\top z_t^n. \tag{22}$$

Suppose $\gamma_T \geq m > 0$ and every candidate satisfies $\|z_t - z_T\|_2 \leq \epsilon$. If $2\epsilon < m$, the step-t representation preserves the correct full-step top-1 decision.

Proof. Cauchy–Schwarz gives $\left|q^\top (z_t - z_T)\right| \leq \epsilon$. The positive score can decrease by at most ϵ, and the largest negative score can increase by at most ϵ. Hence

$$\gamma_t \geq \gamma_T - 2\epsilon \geq m - 2\epsilon > 0. \tag{23}$$

□

For normalized embeddings, $\|z_t - z_T\|_2^2 = 2(1 - z_t^\top z_T)$, linking the distillation objective to embedding distance. However, a small average distillation loss does not guarantee the uniform bound required for every candidate. The proposition assumes a fixed query and does not directly cover a simultaneously changing query-side embedding.

Assumption 2: calibration conditional on the stopping history. Let $\mathcal{F}_t$ contain the information available to the stopping rule through step t, including the query, observed states, and previous gate outputs. Let A_t denote agreement between the step-t and full-step retrieval decisions. Assume the gate output p_t is measurable with respect to $\mathcal{F}_t$ and satisfies

$$|p_t - \Pr(A_t \mid \mathcal{F}_t)| \leq \epsilon_g \tag{24}$$

almost surely at every eligible step, with $\epsilon_g \geq 0$ and $0 \leq \eta \leq 1$. This history-conditional assumption is stronger than a small average calibration error.

Proposition 3: disagreement risk at a threshold-based stopping time. Let $\widehat{T}$ be the first step $t \geq t_{\min}$ with $p_t \geq \eta$, or T if no earlier exit occurs. Under Assumption 2,

$$\Pr(A_{\widehat{T}}^c) \leq \min\{1,\, 1 - \eta + \epsilon_g\}. \tag{25}$$

Proof. For $t < T$, the event $\{\widehat{T} = t\}$ belongs to $\mathcal{F}_t$. On this event, calibration and the threshold imply $\Pr(A_t^c \mid \mathcal{F}_t) \leq 1 - \eta + \epsilon_g$. Since A_T holds by definition,

$$\Pr(A_{\widehat{T}}^c) = \sum_{t<T} \mathbb{E}\Big[\mathbf{1}_{\{\widehat{T}=t\}} \Pr(A_t^c \mid \mathcal{F}_t)\Big] \tag{26}$$

$$\leq (1 - \eta + \epsilon_g) \Pr(\widehat{T} < T) \leq 1 - \eta + \epsilon_g. \tag{27}$$

A probability is also at most one. □

This is an agreement bound, not a guarantee of correctness against ground truth. It is conditional on Assumption 2, which is not verified by the reported experiments. A larger threshold delays or prevents exit for a fixed gate-output trajectory, but does not by itself prove improved task accuracy.

Storage cost of index-only annotations. Suppose a video has R query annotations, each selecting K frames of size $H \times W$ with c channels. If each raw pixel-channel value occupies s bytes and each index occupies b bytes, uncompressed frame copies require $RKHWcs$ bytes. Index lists require RKb bytes, giving an additional-annotation storage ratio of

$$\frac{HWcs}{b}. \tag{28}$$

Both approaches retain the original video. Compressed image formats, metadata, and caching alter the practical ratio; this comparison concerns duplicated annotation storage, not total dataset size or decoding latency.

A.14 LIMITATIONS

QASLR depends on frame selection: its latent memory cannot recover evidence that is absent from the selected observations. The query-aware pathway produces query-conditioned representations for bounded candidate scoring. Reusable corpus indexing instead requires query-independent selection and encoding.

The comparisons with sub-batched single-round training show consistent retrieval gains but task-dependent QA gains. Test-time order and memory perturbations measure sensitivity to altered inputs and state updates, rather than identifying the internal reasoning process. Transfer is evaluated within the Qwen family.

Warm-cache latency depends on selection, decoding, candidate count, and hardware. Fixed memory size bounds the persistent state, but does not imply a proportional reduction in total computation. The seed-averaged scores describe empirical differences without establishing statistical significance. Early-exit bounds concern agreement with full-step predictions under the stated assumptions, rather than unconditional ground-truth accuracy.