SHALLOWSTREAM: INDEX SHALLOW THEN ANSWER DEEP FOR STREAMING VIDEO UNDERSTANDING

Anonymous authors
Paper under double-blind review

ABSTRACT

Streaming video understanding is essential for real-time embodied and assistive systems, but processing open-ended streams with multimodal large language models (MLLMs) is computationally expensive. Existing methods reduce visual tokens or stored context, yet still apply full-depth prefill to admitted frames, causing repeated computation and depth-proportional KV-cache growth. We observe that shallow MLLM layers already provide effective signals for retrieving question-relevant evidence. Building on this observation, we propose **ShallowStream**, which decouples lightweight stream processing from selective full-depth answering. During streaming, ShallowStream processes frames only through shallow layers, using their KVs as a historical index with optional cluster compression. At query time, a text-only gate activates cross-layer token voting and diversity-aware retrieval when needed, after which only the selected history and recent context undergo full-depth prefill. ShallowStream reduces per-frame prefill and 20-second end-to-end latency by up to **52.1**× and **15.3**×, respectively, while improving average scores by approximately **1.7 points** over strong existing baselines. Our code is available at https://anonymous.4open.science/r/ShallowStream/.

1 INTRODUCTION

Multimodal Large Language Models (MLLMs) have demonstrated strong capabilities across diverse visual tasks (Bai et al., 2025b; Chen et al., 2023; Bai et al., 2025a). These advances have spurred growing interest in streaming video understanding, which supports applications such as embodied intelligence, autonomous driving, surveillance and early warning, AR glasses, sports commentary, and fitness coaching (Chen et al., 2024; Li et al., 2025; Lin et al., 2024; Di et al., 2025). Unlike offline long-video understanding, where the complete video and question are available before inference, streaming video understanding must causally process an open-ended stream without access to future frames or advance knowledge of future questions.

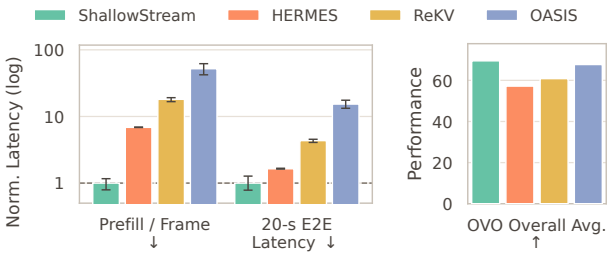

Figure 1: **Continuous-stream efficiency.** Per-frame prefill and end-to-end latency with one query every 20s. Latencies and their 95% confidence intervals are scaled by the corresponding ShallowStream mean and shown on a logarithmic scale. Performance is the OVO-Bench average.

Streaming workloads are inherently *asymmetric*: frames arrive continuously, whereas queries occur only intermittently (Qian et al., 2024; Wang et al., 2024b; Chen et al., 2026b; Yang et al., 2025a; Chen et al., 2025b; Lu et al., 2026; Ning et al., 2025; Kim et al., 2025a; Zhang et al., 2026b). During *query-agnostic stream processing*, a model encodes incoming frames and maintains historical context before future questions are known (Di et al., 2025; Zhang et al., 2026a). During *query-time answering*, it determines whether earlier context is needed, retrieves relevant evidence, and generates an answer (Shen et al., 2026; Liang et al., 2026). Because stream processing is continuous while question answering is sparse, per-frame prefill becomes a **first-order system cost**. As shown in Figure 1, the substantial per-frame overhead of representative methods accumulates into high end-to-end latency.

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.

This setting exposes three limitations in existing streaming systems:

- **Expensive Stream Processing (*Stream Processing*).** Many KV-centric methods prefill every incoming frame through the full Transformer stack before knowing whether its deep representations will ever be used (Di et al., 2025; Kim et al., 2025b). Subsequent cache reduction lowers memory consumption but cannot recover the computation already spent constructing deep KVs.
- **Lossy Evidence Reduction (*Stream Processing*).** Query-agnostic compression, merging, or eviction limits KV-cache growth but may discard evidence needed by future retrospective queries (Chen et al., 2026b; Yang et al., 2025a; Li et al., 2024b); retaining complete states preserves evidence but incurs prohibitive memory costs and bandwidth-limited offloading (Kim et al., 2025a).
- **Inefficient History Use (*Query-Time Answering*).** Supplying history to every query increases latency and can interfere with current-scene perception (Shen et al., 2026). Conversely, on-demand methods may first perform costly answer-level inference to determine whether historical evidence is needed (Liang et al., 2026; Zhang et al., 2026b).

Our key observation is that historical evidence can be retrieved effectively without processing every frame through the full model depth. As shown in Figure 2, strong retrieval capability already emerges at layer 4 of 28 in Qwen3-VL-8B (Bai et al., 2025a) and layer 3 of 32 in LLaVA-OneVision-7B (Li et al., 2024a). Deeper stream-time processing therefore incurs substantial additional cost while providing limited benefit for evidence selection.

Motivated by this observation, we propose **ShallowStream**, which addresses these limitations by decoupling lightweight, query-agnostic stream processing from selective full-depth answering:

- **Shallow Encoding:** During continuous stream processing, ShallowStream propagates each incoming video unit through only the shallow layers of the language Transformer, avoiding unnecessary deep-layer prefill and substantially reducing steady-state computation.
- **Full-History Visual Indexing:** The resulting shallow-layer KVs form a lightweight index over the observed stream, preserving access to historical evidence without retaining full-depth states for every frame. Optional long-cluster compression further bounds memory over long streams.
- **Selective Query-Time Answering:** A lightweight query-logit gate activates retrieval when needed, after which shallow-layer token voting and diversity-aware selection identify complementary evidence for full-depth answering with recent context.

Extensive experiments show that, without task-specific training, ShallowStream reduces per-frame prefill and 20-second end-to-end latency on long videos by up to **52.1**$\times$ and **15.3**$\times$, respectively, while improving average scores by approximately **1.7 points** over strong existing baselines. These results show that shallow indexing with selective deep answering provides an effective and efficient operating point for continuous video understanding.

2 Related Work

Unlike offline long-video understanding, streaming video understanding causally processes sequential frames without access to future content or advance knowledge of questions, which may arrive at any time (Chen et al., 2024; Qian et al., 2024; Lin et al., 2024; Yang et al., 2025b; Wang et al., 2024b; Li et al., 2025; Di et al., 2025; Yang et al., 2025a; Ning et al., 2025; Shen et al., 2026). Its asymmetric workload has two stages: query-agnostic stream processing and query-time answering.

Query-Agnostic Stream Processing. Before questions arrive, existing methods represent the stream using recurrent memory, KV-caches, or compact visual representations. VideoStreaming (Qian et al., 2024) propagates memory across clips, while VideoLLaMB (Wang et al., 2024b) links them through recurrent memory bridges. KV-centric methods maintain layer-wise visual states: ReKV (Di et al., 2025) offloads complete video KVs to host memory or disk; StreamKV (Chen et al., 2026b) applies segment-wise adaptive compression; and StreamMem (Yang et al., 2025a), InfiniPot-V (Kim et al., 2025b), and HERMES (Zhang et al., 2026a) bound cache growth using proxy-query attention, temporal redundancy with Value Norm, and hierarchical recency-attention cues, respectively. Other methods reduce visual input before MLLM prefill: TimeChat-Online (Yao et al., 2025) and StreamingTOM (Chen et al., 2025b) compress visual tokens, whereas Vista (Lu et al., 2026) indexes compact scene representations and offloads high-resolution frames to CPU memory.

Despite these differences, every frame or token admitted to the MLLM still traverses its *full* Transformer depth. Thus, these methods reduce the content processed or retained but leave model depth,

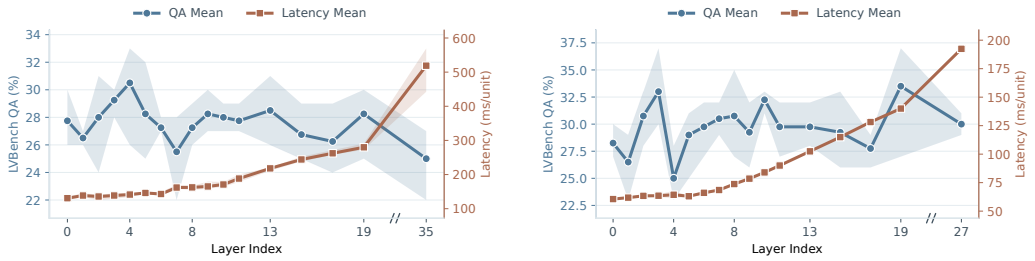

Figure 2: **Layer-wise retrieval quality and stream-time prefill cost on LVBench.** Results for Qwen3-VL-8B (*left*) and LLaVA-OneVision-7B (*right*) as the retrieval layer varies, with all other settings fixed. QA measures full-depth answer accuracy using units selected at that layer, while latency measures stream-time prefill per video unit up to that layer.

and its associated stream-time cost, unchanged. ShallowStream instead reduces computation at its source: each frame traverses only the shallow layers, whose KVs also form a lightweight full-history index, while optional cluster compression bounds memory over long streams.

Query-Time Answering. Existing methods differ in *how* and *when* they access history. StreamMem (Yang et al., 2025a), InfiniPot-V (Kim et al., 2025b), and HERMES (Zhang et al., 2026a) answer from compressed KVs, whereas ReKV (Di et al., 2025), StreamKV (Chen et al., 2026b), LiveVLM (Ning et al., 2025), and V-Rex (Kim et al., 2025a) retrieve historical states on demand. StreamingTOM (Chen et al., 2025b) retrieves token groups from quantized memory, while Vista (Lu et al., 2026) recalls compact scene representations with high-resolution content. Regarding *when*, SimpleStream (Shen et al., 2026) shows that unnecessary history increases latency and can impair current-scene perception. OASIS (Liang et al., 2026) and WeaveTime (Zhang et al., 2026b) therefore retrieve history only when an initial answer lacks support or has high uncertainty, but this requires answer-level multimodal inference followed by another inference when retrieval is activated.

ShallowStream separates routing from answering: a single text-only query-logit evaluation determines whether history is needed without generating a candidate answer. If activated, cross-layer token voting and diversity-aware selection retrieve relevant, nonredundant units, and only the selected history and recent context undergo full-depth visual prefill. More related work is provided in Appendix A.

3 Observations

Background and Notation. A streaming video arrives causally as a sequence of video units $\mathcal{X} = \{x_t\}_{t\in\{1,2,\cdots\}}$, where x_t is the t-th unit. Each unit contains two sampled frames for Qwen3-VL and one for LLaVA-OneVision. At query time T, the system has access only to the observed prefix $\mathcal{X}_{\leq T} = \{x_t\}_{t\in\{1,\cdots,T\}}$; neither the future stream $\mathcal{X}_{>T}$ nor the incoming question q_T is known beforehand. When unambiguous, we write $q = q_T$ and denote the generated answer by $\boldsymbol{y}$.

We build on a pretrained MLLM comprising a vision encoder E_{v} and an L-layer language Transformer $\mathcal{F} = \{F^i\}_{i=0}^{L-1}$. The vision encoder maps each unit x_t to visual tokens indexed by $\mathcal{P}_t$. Each Transformer layer F^i maps hidden states $\boldsymbol{H}_t^i$ to $\boldsymbol{H}_t^{i+1}$ while producing key and value states $\boldsymbol{K}_t^i$ and $\boldsymbol{V}_t^i$. We define the pruning boundary $P \ll L$ as the first deep layer: layers $[0, P)$ constitute the **shallow MLLM**, layers $[P, L)$ constitute the **deep MLLM**, and together they form the full MLLM.

Observation: Shallow Layers Suffice for Retrieval. We find that shallow MLLM representations effectively retrieve question-relevant units from long histories, consistent with evidence that shallow and middle layers preserve rich semantic information (Xin et al., 2020; Meng et al., 2022; Hao et al., 2025c). To isolate retrieval quality from answer generation, we conduct a layer-wise LVBench diagnostic (Wang et al., 2024a) under ReKV's attention-sink, bounded-window prefill setting (Di et al., 2025): each candidate layer ranks historical units, and the full-depth model answers using only the highest-ranked units. Appendix B provides a dense full-history control.

Figure 2 shows that effective retrieval signals emerge in shallow layers, while stream-time prefill cost continues to increase with depth. This quality-cost separation motivates ShallowStream: it uses only shallow layers for continuous video encoding and evidence indexing, deferring full-depth computation until a question arrives and applying it only to the retrieved evidence.

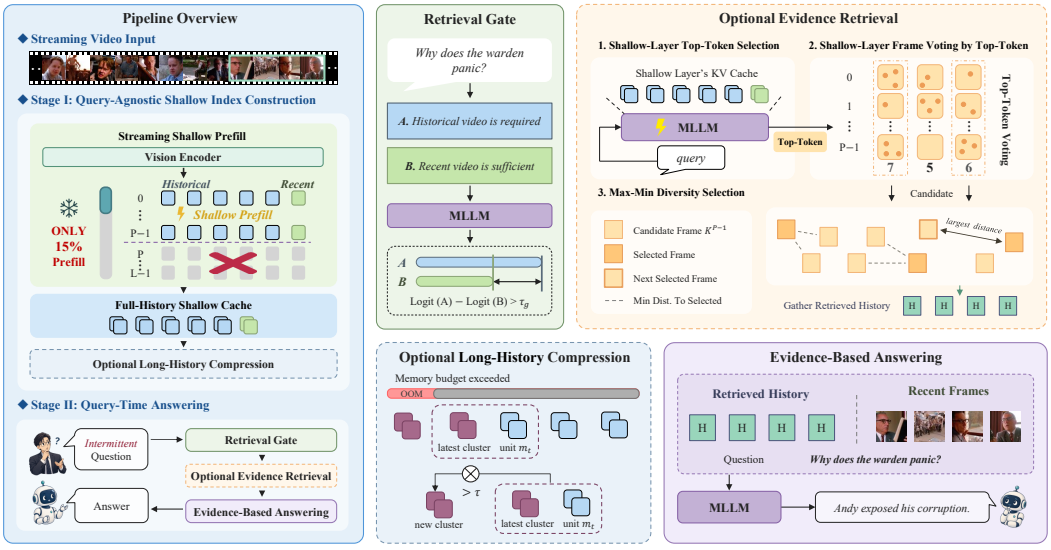

Figure 3: **Overview of ShallowStream. Stage I: Query-Agnostic Shallow Index Construction** (Section 4.1). Incoming video units traverse only the shallow MLLM layers, whose KVs form a lightweight index that can optionally be compressed into fixed-size historical clusters. **Stage II: Query-Time Answering** (Section 4.2). A text-only query-logit gate determines whether to retrieve history; if activated, shallow-layer attention, cross-layer unit voting, and max-min diversity selection identify complementary evidence for full-depth answering alongside recent context.

4 SHALLOWSTREAM

We present ShallowStream, which assigns different computation budgets to the two asymmetric stages of streaming video understanding. Figure 3 illustrates the pipeline.

4.1 QUERY-AGNOSTIC SHALLOW INDEX CONSTRUCTION

Streaming Shallow Prefill. Each incoming video unit x_t is first encoded as $\boldsymbol{H}_t^0 = E_{\mathrm{v}}(x_t)$ and then propagated only through the shallow layers $[0, P)$:

$$(\boldsymbol{H}_t^{i+1}, \boldsymbol{K}_t^i, \boldsymbol{V}_t^i) = F^i(\boldsymbol{H}_t^i; \mathcal{C}_{t-1}^i), \qquad i \in \{0, \cdots, P-1\}, \tag{1}$$

where $\mathcal{C}_{t-1}^i$ is the causal streaming KV cache available to layer i. The detailed memory retains $\boldsymbol{H}_t^0$ and the resulting shallow KVs, allowing a selected unit to be re-prefilled later from layer 0 without storing any deep-layer KVs. Under optional long-cluster compression, older per-unit states are replaced by the fixed-size representatives defined below. By skipping the deep layers $\{F^P, \cdots, F^{L-1}\}$ during streaming, ShallowStream reduces per-unit Transformer computation from L layers to P.

Full-History Shallow Cache. To help queries access distant history, ShallowStream retains the KVs of all observed units at each shallow layer:

$$\mathcal{C}_T^{\text{shallow}} = \{(\boldsymbol{K}_{1:T}^i, \boldsymbol{V}_{1:T}^i)\}_{i=0}^{P-1}. \tag{2}$$

Here, $\boldsymbol{K}_{1:T}^i$ and $\boldsymbol{V}_{1:T}^i$ concatenate the KVs of $\{x_1, \cdots, x_T\}$ at layer i. For each detailed unit x_t, we store its visual encoding, shallow-layer KV cache, and timestamp as $m_t = \{\boldsymbol{H}_t^0, (\boldsymbol{K}_t^i, \boldsymbol{V}_t^i)_{i=0}^{P-1}, t\}$. The cached KVs allow an incoming query to attend to the observed history throughout the shallow layers $[0, P)$, while the visual-token keys support fine-grained evidence retrieval.

For memory management and diversity-aware selection, we summarize each unit with a fixed-length descriptor. Specifically, we average its raw visual keys from the final shallow layer over the visual-token indices $\mathcal{P}_t$ and apply ℓ_2 normalization:

$$\boldsymbol{k}_t = \mathrm{Norm}\left(\tfrac{1}{|\mathcal{P}_t|} \textstyle\sum_{p \in \mathcal{P}_t} \mathrm{vec}(\boldsymbol{K}_{t,p}^{P-1})\right), \tag{3}$$

where $\mathrm{vec}(\cdot)$ flattens all key heads and $\mathrm{Norm}(\boldsymbol{z}) = \boldsymbol{z}/\|\boldsymbol{z}\|_2$. The descriptor $\boldsymbol{k}_t$ captures unit-level structure, while the unpooled shallow keys retain token-level retrieval signals.

At query time T, we divide the streaming memory $\mathcal{M}_T$ into recent and historical contexts. The latest N_r units form $\mathcal{R}_T$, and all earlier units form $\mathcal{H}_T$:

$$\mathcal{M}_T = \mathcal{H}_T \cup \mathcal{R}_T, \qquad \mathcal{R}_T = \{m_t \mid T - N_r < t \le T\}, \quad \mathcal{H}_T = \mathcal{M}_T \setminus \mathcal{R}_T. \tag{4}$$

Both sets retain KVs at every shallow layer. The recent context is always available for current-scene perception, whereas historical evidence is retrieved only when required.

Optional Long-History Compression. When the memory footprint of $\mathcal{M}_T$ exceeds a budget B, ShallowStream compresses older history into long-term clusters $\mathcal{L}_T$, while preserving the recent context $\mathcal{R}_T$ and near-term history $\mathcal{S}_T \subset \mathcal{H}_T$ in detail:

$$\mathcal{M}_T^{(B)} = \begin{cases} \mathcal{H}_T \cup \mathcal{R}_T, & \mathrm{Mem}(\mathcal{M}_T) \le B, \\ \mathcal{L}_T \cup \mathcal{S}_T \cup \mathcal{R}_T, & \mathrm{Mem}(\mathcal{M}_T) > B. \end{cases} \tag{5}$$

Clusters are constructed online over temporally adjacent units. When a unit m_t leaves the detailed window, it joins the latest cluster C_j if its descriptor $\boldsymbol{k}_t$ is sufficiently similar to the cluster centroid $\boldsymbol{c}_j$; otherwise, it starts a new cluster:

$$C(m_t) = \begin{cases} C_j, & \boldsymbol{k}_t^\top \boldsymbol{c}_j \ge \tau, \\ C_{j+1}, & \text{otherwise}, \end{cases} \qquad \boldsymbol{c}_j \leftarrow \mathrm{Norm}\left(\frac{n_j \boldsymbol{c}_j + \boldsymbol{k}_t}{n_j + 1}\right), \tag{6}$$

where n_j is the current size of C_j. Its representative shallow KVs are updated by a running average:

$$(\overline{\boldsymbol{K}}_j^i, \overline{\boldsymbol{V}}_j^i) \leftarrow \frac{n_j(\overline{\boldsymbol{K}}_j^i, \overline{\boldsymbol{V}}_j^i) + (\boldsymbol{K}_t^i, \boldsymbol{V}_t^i)}{n_j + 1}. \tag{7}$$

Each cluster replaces its members' archived states with one input-level visual representative $\overline{\boldsymbol{H}}_j^0$, obtained from the source unit nearest $\boldsymbol{c}_j$ or by averaging visual embeddings. If selected, only this representative is transferred, yielding fixed per-cluster storage and cluster-size-independent host-to-device transfer. Compression is disabled when the uncompressed $\mathcal{H}_T \cup \mathcal{R}_T$ fits in memory.

4.2 Query-Time Answering

Retrieval Gate. Retrospective questions require historical evidence, whereas current-scene questions can often be answered from $\mathcal{R}_T$ alone. ShallowStream distinguishes them without generating a candidate answer by placing q in a fixed few-shot routing prompt $r(q)$ with two single-token choices: A for historical retrieval and B for recent context only. A single text-only forward pass produces:

$$\ell_{\mathrm{ret}}(q) = \mathrm{Logit}_A(r(q)), \qquad \ell_{\mathrm{recent}}(q) = \mathrm{Logit}_B(r(q)). \tag{8}$$

Retrieval is activated when their difference exceeds a backbone-specific threshold τ_g:

$$g(q) = \mathbf{1}[\ell_{\mathrm{ret}}(q) - \ell_{\mathrm{recent}}(q) \ge \tau_g]. \tag{9}$$

For $g(q) = 0$, the model uses only $\mathcal{R}_T$; for $g(q) = 1$, it retrieves from detailed history and optional long-term clusters. We calibrate τ_g once on a benchmark-independent set and freeze it for evaluation, requiring neither a separately trained router nor benchmark labels (Appendix C).

Optional Evidence Retrieval. When retrieval is activated, we append q to the retained shallow history and propagate it through layers $[0, P)$:

$$(\boldsymbol{H}_q^{i+1}, \boldsymbol{Q}_q^i) = F^i(\boldsymbol{H}_q^i; \mathcal{C}_T^i), \qquad i \in \{0, \cdots, P-1\}, \tag{10}$$

where $\mathcal{C}_T^i$ is the retained layer-i KV cache associated with $\mathcal{M}_T$, or $\mathcal{M}_T^{(B)}$ under Equation 5, and $\boldsymbol{Q}_q^i$ denotes the query states. The query attends to exact full-history shallow KVs without compression, or to detailed recent and representative historical KVs with compression.

For detailed history, ShallowStream retains token-level retrieval signals. Let $\mathcal{P}(\mathcal{S}_T) = \bigcup_{m_t \in \mathcal{S}_T} \mathcal{P}_t$ be the candidate visual tokens, with $\mathcal{S}_T = \mathcal{H}_T$ when uncompressed. At each shallow layer i, we compute scaled, RoPE-aware Q-K attention from the final prompt token to all candidates and average over attention heads. If $\alpha_{i,p}$ denotes this score, we retain the top M tokens at each layer:

$$\mathcal{V}_i^M = \mathrm{TopM}_{p \in \mathcal{P}(\mathcal{S}_T)} \alpha_{i,p}, \qquad i \in \{0, \cdots, P-1\}. \tag{11}$$

Each unit $m_t \in \mathcal{S}_T$ receives one vote per selected token: $v_t = \sum_{i=0}^{P-1} \sum_{p \in \mathcal{P}_t} \mathbf{1}[p \in \mathcal{V}_i^M]$.

We retain the top $4K$ units ranked by v_t, breaking ties by attention mass, then use max-min diversity selection (Alvar et al., 2025) in the descriptor space of Equation 3 to choose K nonredundant units. Each selected unit is then expanded with its temporal neighbors to restore local context:

$$\mathcal{E}_{\mathrm{det}}(q) = \begin{cases} \mathcal{R}_T, & g(q) = 0, \\ \mathcal{R}_T \cup \mathrm{Nbr}(\mathrm{DivTopK}_K(\mathcal{S}_T; v, \boldsymbol{k})), & g(q) = 1, \end{cases} \tag{12}$$

Here, DivTopK denotes vote-ranked, descriptor-based diversity selection, and $\mathrm{Nbr}(\cdot)$ adds the temporal neighbors of each selected unit. With long-cluster compression, we additionally select

$$\mathcal{E}_{\mathrm{cl}}(q) = \begin{cases} \varnothing, & g(q) = 0, \\ \mathrm{TopK}_{K_\ell}(\mathcal{L}_T; \boldsymbol{Q}_q^{P-1}, \boldsymbol{c}), & g(q) = 1, \end{cases} \tag{13}$$

where cluster centroids are scored against the final shallow query state. After deduplication and temporal ordering, the gate decides *whether* to retrieve, while token voting, diversity selection, and centroid retrieval decide *what* to retrieve.

Evidence-Based Answering. Shallow KVs support query contextualization and retrieval, but not generation. We gather $\boldsymbol{H}_t^0$ for each selected detailed unit and $\overline{\boldsymbol{H}}_j^0$ for each selected cluster without recovering its members, compose them with the question, and re-prefill the result through all L language layers:

$$\begin{aligned} &\widetilde{\boldsymbol{H}}^0 = \mathrm{Compose}(\{\boldsymbol{H}_t^0\}_{t\in\mathcal{E}_{\mathrm{det}}(q)}, \{\overline{\boldsymbol{H}}_j^0\}_{j\in\mathcal{E}_{\mathrm{cl}}(q)}, q), \quad \widetilde{\boldsymbol{H}}^{i+1} = F^i(\widetilde{\boldsymbol{H}}^i), \\ &\hat{\boldsymbol{y}} = \arg\max_{\boldsymbol{y}} p_\Theta(\boldsymbol{y} \mid \widetilde{\boldsymbol{H}}^L). \end{aligned} \tag{14}$$

This re-prefill ensures that generation uses consistent full-depth representations. Only the selected detailed units and cluster representatives receive full-depth computation, so host-to-device transfer is bounded by the evidence budget. Unselected history remains accessible through the shallow index without requiring full-depth prefill of the entire stream. All transfer and context-assembly costs are included in the reported end-to-end query latency.

5 Experiments

5.1 Experimental Setup

Datasets and Metrics. Following prior streaming-video studies (Di et al., 2025; Zhang et al., 2026a), we evaluate ShallowStream on **OVO-Bench** (Li et al., 2025) and **StreamingBench** (Lin et al., 2024) using their official protocols and reporting category-wise and aggregate scores. OVO-Bench separates Real-Time Visual Perception from Backward Tracing, evaluating current-scene understanding and historical retrieval, while StreamingBench covers broader continuous-video scenarios. We additionally use **LVBench** (Wang et al., 2024a) to analyze layer-wise retrieval quality and stream-time prefill cost.

Baselines. We compare with two baseline groups. Specialized **open-sourced online MLLMs** include VideoLLM-online (Chen et al., 2024), Flash-VStream (Zhang et al., 2025), Dispider (Qian et al., 2025), TimeChat-Online (Yao et al., 2025), StreamForest (Zeng et al., 2025), and Streamo (Xia et al., 2025), all designed specifically for causal video processing. More directly comparable are training-free **offline-to-online methods**. ReKV (Di et al., 2025), StreamKV (Chen et al., 2026b), and LiveVLM (Ning et al., 2025) retrieve historical full-depth or compressed states; HERMES (Zhang et al., 2026a) organizes retained KVs hierarchically; CausalMem (Song et al., 2026) maintains a dynamic fixed-budget memory; and SimpleStream (Shen et al., 2026) uses a strong recent-window baseline. OASIS (Liang et al., 2026) and WeaveTime (Zhang et al., 2026b) conditionally activate historical retrieval. Together, these baselines cover recent-only context, KV retention and retrieval, memory compression, fixed-budget memory, and on-demand retrieval. We also report the original **LLaVA-OneVision-7B** (Li et al., 2024a) and **Qwen3-VL-8B** (Bai et al., 2025a) backbones to contextualize gains within each model family.

Implementation Details. We implement ShallowStream in PyTorch using Hugging Face Transformers and FlashAttention-2, with Qwen3-VL-8B-Instruct (Bai et al., 2025a) and LLaVA-OneVision-7B (Li et al., 2024a) as backbones. Videos are sampled at 1 FPS for both backbones on both benchmarks. All gate and retrieval settings are fixed before evaluation. Appendix C details gate calibration, and Appendix D provides the complete configuration.

Table 1: **Main results on OVO-Bench.** Baseline results marked with ‡ are from our reruns. Results marked with † enable long-cluster compression. Dashes indicate unreported entries.

Model / Method	# Frames	*Real-Time Visual Perception*							*Backward Tracing*				Avg. ↑
		OCR	ACR	ATR	STU	FPD	OJR	Avg. ↑	EPM	ASI	HLD	Avg. ↑	
Open-source Online MLLMs													
VideoLLM-online-8B (Chen et al., 2024)	2 fps	8.1	23.9	12.1	14.0	45.5	21.2	20.8	22.2	18.8	12.2	17.7	19.3
Flash-VStream-7B (Zhang et al., 2025)	1 fps	25.5	32.1	29.3	33.7	29.7	28.8	29.9	36.4	33.8	5.9	25.4	27.6
Dispider-7B (Qian et al., 2025)	1 fps	57.7	49.5	62.1	44.9	61.4	51.6	54.6	48.5	55.4	4.3	36.1	45.3
TimeChat-Online-7B (Yao et al., 2025)	1 fps	75.2	46.8	70.7	47.8	69.3	61.4	61.9	55.9	59.5	9.7	41.7	51.8
StreamForest-7B (Zeng et al., 2025)	1 fps	68.5	53.2	71.6	47.8	65.4	60.9	61.2	58.9	64.9	32.3	52.0	56.6
Streamo-7B (Xia et al., 2025)	1 fps	79.2	57.8	75.0	49.4	64.4	70.1	66.0	54.6	52.0	31.7	46.1	56.1
Offline-to-Online Methods													
LLaVA-OneVision-7B (Li et al., 2024a)	32	66.4	53.2	71.6	51.1	72.3	59.2	62.3	53.9	52.7	18.8	41.8	52.1
+ ReKV (Di et al., 2025)	0.5 fps	52.4	54.1	69.8	43.3	67.3	57.1	57.3	57.6	56.1	18.8	44.2	50.8
+ HERMES‡ (Zhang et al., 2026a)	0.5 fps	71.1	61.5	74.1	52.3	72.3	66.3	66.3	62.0	60.8	26.9	**49.9**	58.1
+ WeaveTime (Zhang et al., 2026b)	1 fps	72.5	69.7	74.1	53.4	75.2	67.9	68.8	–	–	–	–	–
+ CausalMem (Song et al., 2026)	0.5 fps	71.8	61.5	76.7	48.9	76.2	66.8	65.7	–	–	–	–	–
+ SimpleStream‡ (Shen et al., 2026)	4	77.2	71.6	75.9	52.3	77.2	71.7	71.0	53.2	52.0	43.6	49.6	60.3
+ **ShallowStream (Ours)**	1 fps	87.3	73.4	77.6	62.9	74.3	77.2	75.4	50.2	50.7	46.2	49.0	62.2
+ **ShallowStream (Ours)**†	1 fps	87.3	73.4	77.6	63.5	74.3	77.2	**75.5**	50.5	50.7	46.2	49.1	**62.3**
Qwen3-VL-8B (Bai et al., 2025a)	64	76.5	58.7	75.0	59.0	68.3	59.2	66.1	53.2	66.9	9.7	43.3	54.7
+ ReKV‡ (Di et al., 2025)	1 fps	81.9	68.8	71.6	59.0	74.3	69.0	70.8	58.6	71.6	22.6	50.9	60.8
+ HERMES‡ (Zhang et al., 2026a)	2 fps	85.2	64.2	73.3	55.6	70.3	65.8	69.1	50.5	65.5	19.9	45.3	57.2
+ OASIS (Liang et al., 2026)	–	92.0	80.7	81.0	67.4	67.3	79.9	78.1	62.0	60.1	47.3	57.2	67.7
+ SimpleStream‡ (Shen et al., 2026)	4	92.0	81.7	81.9	69.7	75.3	81.0	80.2	53.2	60.1	44.1	52.5	66.4
+ **ShallowStream (Ours)**	1 fps	92.0	83.5	82.8	71.9	75.3	79.9	80.9	52.5	72.3	49.5	**58.1**	**69.5**
+ **ShallowStream (Ours)**†	1 fps	92.0	83.5	82.8	71.9	76.2	79.9	**81.0**	51.2	71.0	50.0	57.4	69.2

5.2 Overall Performance and System Efficiency

Overall Streaming Performance. Tables 1 and 2 evaluate two complementary capabilities: retrieving evidence from the observed history and understanding an evolving visual scene. Without task-specific training, ShallowStream achieves the best overall results for both backbones: **69.5/78.2** on OVO-Bench/StreamingBench with Qwen3-VL-8B and **62.2/75.5** with LLaVA-OneVision-7B. Across the four dataset-backbone settings, it outperforms the second-best same-backbone method by **1.7 points** on average, showing that shallow stream-time indexing preserves evidence for both retrospective and real-time questions across different MLLM backbones.

Continuous-Stream Compute Efficiency. Efficiency is measured on one NVIDIA RTX 5090 with Qwen3-VL-8B over five long OVO-Bench Backward videos at 1 FPS. Timing excludes frame decoding and fixes generation to 16 visible tokens. We separately report per-frame prefill and per-query computation, including ShallowStream's archive updates, selected-state transfer, context assembly, and generation. Memory results report peak GPU allocation, excluding CPU capacity.

ShallowStream achieves performance on par with current state-of-the-art streaming methods at substantially lower continuous-stream cost. As shown in Figure 1, it reduces per-frame prefill latency by up to **52.1**× and 20-s end-to-end latency by up to **15.3**×. These gains reflect the two central design choices in Section 4: processing each incoming unit only through the shallow layers and reserving full-depth computation for the evidence selected at query time. Appendix E analyzes the trade-off between retrieval quality and stream-processing cost as the pruning boundary varies.

Memory Scaling under Long-History Compression. Long-history compression (Section 4.1) bounds GPU-memory growth by replacing older shallow states with fixed-size cluster representatives. The left panel of Figure 4 compares ShallowStream with compression enabled (**LC-on**) and disabled (**LC-off**); LC-off retains the complete layer-0–4 shallow K/V history. From 64 to 1,024 frames, LC-on keeps peak GPU allocation nearly constant at approximately 18 GiB, whereas LC-off grows to 21.76 GiB. At long prefixes, LC-on also uses less GPU memory than HERMES and OASIS.

Real-Time Processing Headroom. The right panel of Figure 4 evaluates total compute demand over different query intervals on five videos containing 588–1,198 sampled frames, without imposing

Table 2: **Main results on the real-time visual understanding subset of StreamingBench.** Baseline results marked with ‡ are from our controlled reruns. Results marked with † enable long-cluster compression. Dashes indicate unreported results.

Model / Method	# Frames	*Real-Time Visual Understanding*										Avg. ↑
		OP	CR	CS	ATP	EU	TR	PR	SU	ACP	CT	
Open-source Online MLLMs												
Flash-VStream-7B (Zhang et al., 2025)	–	25.9	43.6	24.9	23.9	27.3	13.1	18.5	25.2	23.9	48.7	23.2
VideoLLM-online-8B (Chen et al., 2024)	2 fps	39.1	40.1	34.5	31.1	46.0	32.4	31.5	34.2	42.5	27.9	36.0
Dispider-7B (Qian et al., 2025)	1 fps	74.9	75.5	74.1	73.1	74.4	59.9	76.1	62.9	62.2	45.8	67.6
TimeChat-Online-7B (Yao et al., 2025)	1 fps	80.2	82.0	79.5	83.3	76.1	78.5	78.7	64.6	69.6	58.0	75.4
StreamForest-7B (Zeng et al., 2025)	1 fps	83.1	82.8	82.7	84.3	77.5	78.2	76.9	69.1	75.6	54.4	77.3
Offline-to-Online Methods												
LLaVA-OneVision-7B (Li et al., 2024a)	32	77.7	76.6	77.6	81.9	71.7	71.7	66.7	65.5	65.7	44.0	71.0
+ ReKV (Di et al., 2025; Zhang et al., 2026a)	0.5 fps	74.4	78.9	78.6	77.1	68.3	67.9	67.6	62.6	64.3	44.6	69.2
+ HERMES‡ (Zhang et al., 2026a)	0.5 fps	78.2	79.7	86.8	81.5	69.2	73.2	75.0	64.2	68.6	45.6	73.2
+ StreamKV (Chen et al., 2026b)	1 fps	74.7	78.1	87.7	79.4	70.8	67.6	70.4	64.6	64.0	45.1	71.0
+ WeaveTime (Zhang et al., 2026b)	1 fps	71.5	81.3	86.8	78.6	75.2	73.2	72.2	69.1	68.8	44.7	72.1
+ SimpleStream‡ (Shen et al., 2026)	4	80.4	71.9	85.2	85.5	76.1	73.8	69.4	65.5	71.1	37.3	73.5
+ LiveVLM (Ning et al., 2025)	0.5 fps	79.8	79.7	84.9	80.7	67.1	70.1	74.1	66.3	68.0	42.5	71.3
+ CausalMem (Song et al., 2026)	0.5 fps	82.6	79.7	83.2	83.2	69.6	78.4	75.0	67.3	70.9	39.4	74.3
+ **ShallowStream (Ours)**	1 fps	82.8	67.2	83.9	90.8	71.7	81.9	64.8	70.7	74.8	35.2	75.5
+ **ShallowStream (Ours)**†	1 fps	82.8	67.2	83.9	90.8	72.3	81.9	64.8	70.7	74.8	35.2	**75.6**
Qwen3-VL-8B (Bai et al., 2025a)	64	80.4	75.0	83.0	83.5	74.2	84.4	77.8	63.8	69.7	57.0	75.9
+ SimpleStream‡ (Shen et al., 2026)	4	79.3	72.7	89.9	84.8	73.6	79.1	83.3	71.1	76.8	39.4	76.5
+ HERMES‡ (Zhang et al., 2026a)	2 fps	79.0	77.3	81.4	84.2	73.6	77.9	87.0	70.7	72.5	61.1	76.6
+ **ShallowStream (Ours)**	1 fps	79.3	71.9	89.9	85.8	77.4	79.1	88.9	71.5	77.3	52.9	**78.2**
+ **ShallowStream (Ours)**†	1 fps	79.3	71.9	90.2	85.5	77.4	79.1	88.9	71.5	77.3	49.7	78.0

a frame cap. Both LC-on and LC-off remain well below the real-time boundary. At an 80-s query interval, each requires approximately 2.6 s of compute, compared with 6.2 s for HERMES and 50.4 s for OASIS. Thus, ShallowStream retains substantial processing headroom whether it compresses old history or preserves the complete shallow index. Appendix F further examines how LC-off scales with the retained history.

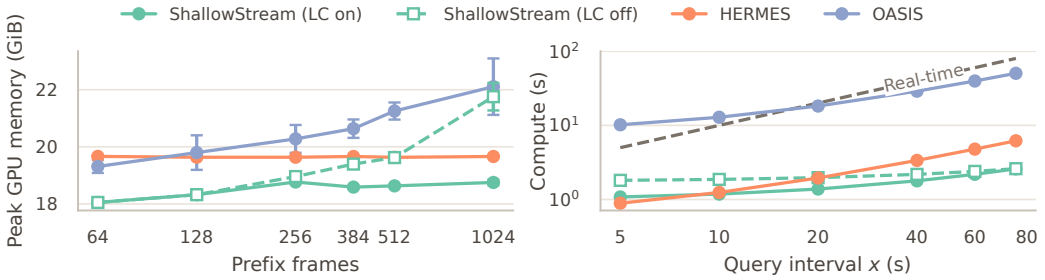

Figure 4: **Memory scaling and real-time compute demand.** *Left:* Peak GPU allocation from 64 to 1,024 frames, averaged over five videos; both ShallowStream variants use the calibrated Gate and token-vote retriever. *Right:* Total compute for a query interval x, obtained by combining mean query computation with stream-time prefill accumulated over x seconds at 1 FPS. The dashed boundary $y = x$ separates configurations that keep pace with the stream (below) from those whose compute exceeds the available interval (above).

5.3 Routing and Retrieval Ablations

The Retrieval Gate Identifies Historical Need. Always using recent context can omit necessary historical evidence, whereas indiscriminate retrieval can introduce distracting visual content. Figure 5 shows that the Query-Logit Gate from Section 4.2 balances these cases by activating retrieval for 48.7% of Backward questions but only 4.9% of Real-Time questions. Under controlled branch

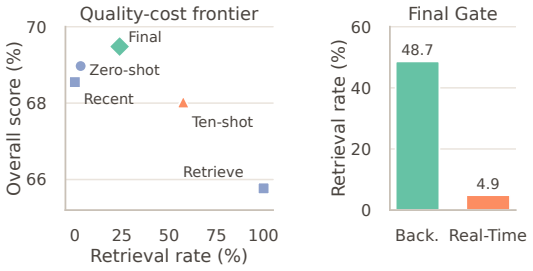

Figure 5: **Retrieval-gate ablation on OVO-Bench.** *Left*: Accuracy vs. retrieval rate with fixed branch outputs, isolating routing effects; *Right*: Calibrated Gate retrieval rates for Backward and Real-Time questions.

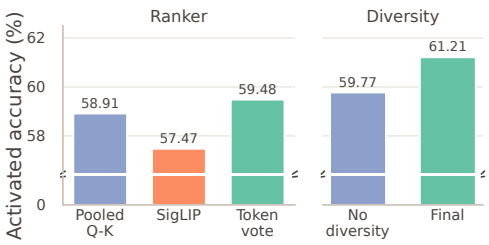

Figure 6: **Evidence-retrieval ablation on Gate-selected OVO-Bench queries.** *Left:* Rankers without diversity or temporal expansion; *Right:* Max-min diversity with token voting and preceding-unit expansion fixed.

outputs, this selective routing improves overall accuracy over both always-recent and always-retrieve policies. The result confirms that the Gate responds to the question's historical requirements rather than applying retrieval uniformly.

Token Voting and Diversity Improve Evidence Selection. Figure 6 isolates the evidence-retrieval components introduced in Section 4.2. Without diversity selection or temporal expansion, cross-layer token voting achieves 59.48%, outperforming pooled shallow Q-K (58.91%) and the independent SigLIP encoder (57.47%). This improvement demonstrates the value of query-contextualized, token-level attention for locating historical evidence. With token voting and preceding-unit expansion fixed, max–min diversity further raises accuracy from 59.77% to 61.21% by allocating the limited evidence budget across complementary moments.

5.4 Qualitative Evidence Retrieval

Figure 7 illustrates how the final retriever combines the two evidence-selection mechanisms in Section 4. Pooled shallow Q-K repeatedly selects similar kitchen views and predicts B. Max-min diversity broadens its temporal coverage, but the selections remain constrained by the pooled relevance signal and yield the same answer. Cross-layer token voting instead identifies query-specific evidence, while diversity selection distributes the retrieval budget across complementary views of the room, enabling the model to answer E correctly.



Figure 7: **Qualitative evidence retrieval on OVO-Bench EPM.** For the question "Where is the red and white checkered rug?" (ground-truth answer: E), the rows show the eight units selected by pooled shallow Q-K, pooled shallow Q-K with max-min diversity, and the final token-vote retriever with max-min diversity. Red outlines indicate frames containing the rug beneath the dining table. The final retriever selects complementary views and produces the correct answer.

6 Conclusions

We introduced **ShallowStream**, which treats model depth as a resource to allocate selectively in streaming video understanding. Rather than processing every incoming frame through the full MLLM, it builds a lightweight shallow index and applies full-depth computation only to evidence selected through query-logit routing and diversity-aware token voting. Across two MLLM backbones, ShallowStream matches state-of-the-art performance while substantially reducing computation, latency, and GPU-memory growth. These results establish shallow indexing with selective deep processing as a practical design principle for efficient, long-running streaming MLLMs.

AI Use Statement

Generative AI tools were used to generate the benchmark-independent synthetic questions and routing labels used for Gate calibration, and to polish the manuscript. The authors reviewed the generated calibration data for consistency with the retrospective-versus-recent routing definition and reviewed all AI-assisted text. The authors take full responsibility for the final content of this work.

Ethics Statement

This work develops a computational method for streaming video understanding and evaluates it using existing pretrained models and public benchmarks. It does not involve human-subject research, the collection of personal or sensitive data, or deployment in safety-critical settings. We identify no specific ethical concerns arising from the proposed method.

Reproducibility Statement

Implementation details, model and benchmark configurations, calibration procedures, and evaluation protocols are documented in the main paper and appendix. The source code and experiment configurations are available at https://anonymous.4open.science/r/ShallowStream/. Together, these materials specify the data processing, inference settings, and measurements needed to reproduce the reported results.

References

Saeed Ranjbar Alvar, Gursimran Singh, Mohammad Akbari, and Yong Zhang. Divprune: Diversity-based visual token pruning for large multimodal models. *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9392–9401, 2025. URL https://api.semanticscholar.org/CorpusID:276775957.

Shehreen Azad, Vibhav Vineet, and Yogesh Singh Rawat. Streamready: Learning what to answer and when in long streaming videos. *ArXiv*, abs/2603.08620, 2026. URL https://api.semanticscholar.org/CorpusID:286377314.

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Ke qin Chen, Xiong-Hui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Rongyao Fang, Chang Gao, et al. Qwen3-vl technical report. *ArXiv*, abs/2511.21631, 2025a. URL https://api.semanticscholar.org/CorpusID:283262018.

Shuai Bai, Ke qin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, et al. Qwen2.5-vl technical report. *ArXiv*, abs/2502.13923, 2025b. URL https://api.semanticscholar.org/CorpusID:276449796.

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. *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 18407–18418, 2024. URL https://api.semanticscholar.org/CorpusID:270560262.

Joya Chen, Ziyun Zeng, Yiqi Lin, Yiqi Lin, Wei Li, Zejun Ma, and Mike Zheng Shou. Live: Learning video llm with streaming speech transcription at scale. *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 29083–29095, 2025a. URL https://api.semanticscholar.org/CorpusID:277993668.

Tao Chen, Kun Zhang, Qiong Wu, Xiao Chen, Chaokun Chang, Xiaoshuai Sun, Yiyi Zhou, and Rongrong Ji. Scaling the long video understanding of multimodal large language models via visual memory mechanism. *ArXiv*, abs/2603.29252, 2026a. URL https://api.semanticscholar.org/CorpusID:286977474.

Xueyi Chen, Keda Tao, Kele Shao, and Huan Wang. Streamingtom: Streaming token compression for efficient video understanding. *ArXiv*, abs/2510.18269, 2025b. URL https://api.semanticscholar.org/CorpusID:282246334.

Yilong Chen, Xiang Bai, Zhibin Wang, Chengyu Bai, Yuhan Dai, and Ming Lu. Streamkv: Streaming video question-answering with segment-based kv cache retrieval and compression. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 3120–3128, 2026b. URL https://api.semanticscholar.org/CorpusID:282913012.

Zhe Chen, Jiannan Wu, Wenhai Wang, Weijie Su, Guo Chen, Sen Xing, Zhong Muyan, Qinglong Zhang, Xizhou Zhu, Lewei Lu, Bin Li, Ping Luo, Tong Lu, Yu Qiao, and Jifeng Dai. Intern vl: Scaling up vision foundation models and aligning for generic visual-linguistic tasks. *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 24185–24198, 2023. URL https://api.semanticscholar.org/CorpusID:266521410.

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*, abs/2503.00540, 2025. URL https://api.semanticscholar.org/CorpusID:276741475.

Xin Ding, Hao Wu, Yifan Yang, Shiqi Jiang, Donglin Bai, Zhibo Chen, and Ting Cao. Streammind: Unlocking full frame rate streaming video dialogue through event-gated cognition. *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 13448–13459, 2025. URL https://api.semanticscholar.org/CorpusID:276903154.

Shenghao Fu, Qize Yang, Yuan-Ming Li, Yi-Xing Peng, Kun-Yu Lin, Xihan Wei, Jianfang Hu, Xiaohua Xie, and Wei-Shi Zheng. Vispeak: Visual instruction feedback in streaming videos. *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 21778–21788, 2025. URL https://api.semanticscholar.org/CorpusID:277066592.

Haonan Ge, Yiwei Wang, Hang Wu, and Yujun Cai. What should a streaming video model remember? *ArXiv*, abs/2606.16353, 2026. URL https://api.semanticscholar.org/CorpusID:289299850.

Jitai Hao, Qiang Huang, Hao Liu, Xinyan Xiao, Zhaochun Ren, and Jun Yu. A token is worth over 1,000 tokens: Efficient knowledge distillation through low-rank clone. *ArXiv*, abs/2505.12781, 2025a. URL https://api.semanticscholar.org/CorpusID:278740403.

Jitai Hao, Hao Liu, Xinyan Xiao, Qiang Huang, and Jun Yu. Uni-x: Mitigating modality conflict with a two-end-separated architecture for unified multimodal models. *ArXiv*, abs/2509.24365, 2025b. URL https://api.semanticscholar.org/CorpusID:281674376.

Jitai Hao, Yuke Zhu, Tianjian Wang, Jun Yu, Xin Xin, Bo Zheng, Zhaochun Ren, and Sheng Guo. Omnikv: Dynamic context selection for efficient long-context llms. In *International Conference on Learning Representations*, 2025c. URL https://api.semanticscholar.org/CorpusID:278601790.

Jitai Hao, Qiang Huang, Yaowei Wang, Min Zhang, and Jun Yu. Deltakv: Residual-based kv cache compression via long-range similarity. *ArXiv*, abs/2602.08005, 2026. URL https://api.semanticscholar.org/CorpusID:285454299.

Donghyuk Kim, Sejeong Yang, Wonjin Shin, and Joo-Young Kim. V-rex: Real-time streaming video llm acceleration via dynamic kv cache retrieval. *2026 IEEE International Symposium on High Performance Computer Architecture (HPCA)*, pp. 1–14, 2025a. URL https://api.semanticscholar.org/CorpusID:283895999.

Minsoo Kim, Kyuhong Shim, Jungwook Choi, and Simyung Chang. Infinipot-v: Memory-constrained kv cache compression for streaming video understanding. *ArXiv*, abs/2506.15745, 2025b. URL https://api.semanticscholar.org/CorpusID:279465408.

Bo Li, Yuanhan Zhang, Dong Guo, Renrui Zhang, Feng Li, Hao Zhang, Kaichen Zhang, Yanwei Li, Ziwei Liu, and Chunyuan Li. Llava-onevision: Easy visual task transfer. *ArXiv*, abs/2408.03326, 2024a. URL https://api.semanticscholar.org/CorpusID:271719914.

Yifei Li, Junbo Niu, Ziyang Miao, Chunjiang Ge, Yuanhang Zhou, Qi He, Xiao wen Dong, Haodong Duan, Shuangrui Ding, Rui Qian, Pan Zhang, Yuhang Zang, Yuhang Cao, Conghui He, and Jiaqi Wang. Ovo-bench: How far is your video-llms from real-world online video understanding? *2025*

IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 18902–18913, 2025. URL https://api.semanticscholar.org/CorpusID:275458756.

Yucheng Li, Huiqiang Jiang, Qianhui Wu, Xufang Luo, Surin Ahn, Chengruidong Zhang, Amir H. Abdi, Dongsheng Li, Jianfeng Gao, Yuqing Yang, and Lili Qiu. Scbench: A kv cache-centric analysis of long-context methods. *ArXiv*, abs/2412.10319, 2024b. URL https://api.semanticscholar.org/CorpusID:274763141.

Zhijia Liang, Jiaming Li, Weikai Chen, Yanhao Zhang, Haonan Lu, and Guanbin Li. Oasis: On-demand hierarchical event memory for streaming video reasoning. *ArXiv*, abs/2604.17052, 2026. URL https://api.semanticscholar.org/CorpusID:287635604.

Junming Lin, Zheng Fang, Chi Chen, Zihao Wan, Fuwen Luo, Peng Li, Yang Liu, and Maosong Sun. Streamingbench: Assessing the gap for mllms to achieve streaming video understanding. *ArXiv*, abs/2411.03628, 2024. URL https://api.semanticscholar.org/CorpusID:273850338.

Haocheng Lu, Nan Zhang, Wei Tao, Xiaoyang Qu, Guokuan Li, Jiguang Wan, and Jianzong Wang. Vista: Scene-aware optimization for streaming video question answering under post-hoc queries. *ArXiv*, abs/2602.08448, 2026. URL https://api.semanticscholar.org/CorpusID:285454172.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. Locating and editing factual associations in gpt. *Advances in Neural Information Processing Systems 35*, 2022. URL https://api.semanticscholar.org/CorpusID:255825985.

Zhenyu Ning, Guangda Liu, Qihao Jin, Wenchao Ding, Minyi Guo, and Jieru Zhao. Livevlm: Efficient online video understanding via streaming-oriented kv cache and retrieval. *ArXiv*, abs/2505.15269, 2025. URL https://api.semanticscholar.org/CorpusID:278783116.

Rui Qian, Xiao wen Dong, Pan Zhang, Yuhang Zang, Shuangrui Ding, Dahua Lin, and Jiaqi Wang. Streaming long video understanding with large language models. *ArXiv*, abs/2405.16009, 2024. URL https://api.semanticscholar.org/CorpusID:270063250.

Rui Qian, Shuangrui Ding, Xiao wen Dong, Pan Zhang, Yuhang Zang, Yuhang Cao, Dahua Lin, and Jiaqi Wang. Dispider: Enabling video llms with active real-time interaction via disentangled perception, decision, and reaction. *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 24045–24055, 2025. URL https://api.semanticscholar.org/CorpusID:275336693.

Jihao Qiu, Lingxi Xie, Xinyue Huo, Qi Tian, and Qixiang Ye. Longvideo-r1: Smart navigation for low-cost long video understanding. *ArXiv*, abs/2602.20913, 2026. URL https://api.semanticscholar.org/CorpusID:286000722.

Yujiao Shen, Shulin Tian, Jingkang Yang, and Ziwei Liu. A simple baseline for streaming video understanding. *ArXiv*, abs/2604.02317, 2026. URL https://api.semanticscholar.org/CorpusID:287073889.

Baiyang Song, Yu-Lin Lin, Qiong Wu, Tao Chen, Jun Peng, Xiao Chen, Yiyi Zhou, and Rongrong Ji. Towards a dynamic and fixed-budget memory bank for efficient streaming video understanding. *ArXiv*, abs/2606.25658, 2026. URL https://api.semanticscholar.org/CorpusID:289630861.

Haibo Wang, Bo Feng, Zhengfeng Lai, Mingze Xu, Shiyu Li, Weifeng Ge, Afshin Dehghan, Meng Cao, and Ping Huang. Streambridge: Turning your offline video large language model into a proactive streaming assistant. *ArXiv*, abs/2505.05467, 2025a. URL https://api.semanticscholar.org/CorpusID:278394112.

Weihan Wang, Zehai He, Wenyi Hong, Yean Cheng, Xiaohan Zhang, Ji Qi, Shiyu Huang, Bin Xu, Yuxiao Dong, Ming Ding, and Jie Tang. Lvbench: An extreme long video understanding benchmark. *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 22958–22967, 2024a. URL https://api.semanticscholar.org/CorpusID:270391637.

Yiyu Wang, Xuyang Liu, Xiyan Gui, Xinying Lin, Boxue Yang, Chenfei Liao, Tailai Chen, and Linfeng Zhang. Accelerating streaming video large language models via hierarchical token compression. *ArXiv*, abs/2512.00891, 2025b. URL https://api.semanticscholar.org/CorpusID:283450494.

Yuxuan Wang, Yiqi Song, Cihang Xie, Yang Liu, and Zilong Zheng. Videollamb: Long streaming video understanding with recurrent memory bridges. *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 24170–24181, 2024b. URL https://api.semanticscholar.org/CorpusID:280422760.

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. *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3272–3282, 2024c. URL https://api.semanticscholar.org/CorpusID:270094803.

Siwei Wen, Zhangcheng Wang, Xingjian Zhang, Lei Huang, and Wenjun Wu. Eventmemagent: Hierarchical event-centric memory for online video understanding with adaptive tool use. *ArXiv*, abs/2602.15329, 2026. URL https://api.semanticscholar.org/CorpusID:285659749.

Hang Wu, Sherin M. Mathews, Yujun Cai, Ming-Hsuan Yang, and Yiwei Wang. Semantic-aware adaptive visual memory for streaming video understanding. *ArXiv*, abs/2605.07897, 2026. URL https://api.semanticscholar.org/CorpusID:288147794.

Jiaer Xia, Peixian Chen, Mengdan Zhang, Xing Sun, and Kaiyang Zhou. Streaming video instruction tuning. *ArXiv*, abs/2512.21334, 2025. URL https://api.semanticscholar.org/CorpusID:284153645.

Jun Xiao, Jiajun Chen, Tianxiang Sun, Xun Yang, and Angela Yao. Mukv: Multi-grained kv cache compression for long streaming video question-answering. *ArXiv*, abs/2605.22269, 2026. URL https://api.semanticscholar.org/CorpusID:288650113.

Junlin Xie, Quanlong Zheng, Ruifei Zhang, Kuo Wang, Yanhao Zhang, Jinguo Luo, Haonan Lu, Xiang Wan, and Guanbin Li. Streamrag: Enhancing real-time video understanding with retrieval augmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 38870–38879, June 2026.

Ji Xin, Rodrigo Nogueira, Yaoliang Yu, and Jimmy J. Lin. Early exiting bert for efficient document ranking. In *SUSTAINLP*, 2020. URL https://api.semanticscholar.org/CorpusID:226283755.

Shuhang Xun, Sicheng Tao, Jungang Li, Yibo Shi, Zhixin Lin, Zhanhui Zhu, Yibo Yan, Hanqian Li, Linghao Zhang, Shikang Wang, Yixin Liu, Hanbo Zhang, Xuming Hu, and Yinghao Ma. Rtv-bench: Benchmarking mllm continuous perception, understanding and reasoning through real-time video. *ArXiv*, abs/2505.02064, 2025. URL https://api.semanticscholar.org/CorpusID:278326974.

Yibin Yan, Jilan Xu, Shangzhe Di, Yikun Liu, Yudi Shi, Qirui Chen, Zeqian Li, Yifei Huang, and Weidi Xie. Learning streaming video representation via multitask training. *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 9900–9912, 2025. URL https://api.semanticscholar.org/CorpusID:278165314.

Yanlai Yang, Zhuokai Zhao, Satya Narayan Shukla, Aashu Singh, Shlok Kumar Mishra, Lizhu Zhang, and Mengye Ren. Streammem: Query-agnostic kv cache memory for streaming video understanding. *ArXiv*, abs/2508.15717, 2025a. URL https://api.semanticscholar.org/CorpusID:280700196.

Zhenyu Yang, Yuhang Hu, Zemin Du, Dizhan Xue, Shengsheng Qian, Jiahong Wu, Fan Yang, Weiming Dong, and Changsheng Xu. Svbench: A benchmark with temporal multi-turn dialogues for streaming video understanding. *ArXiv*, abs/2502.10810, 2025b. URL https://api.semanticscholar.org/CorpusID:276408575.

Linli Yao, Yichen Li, Yuancheng Wei, Lei Li, Shuhuai Ren, Yuanxin Liu, Kun Ouyang, Lean Wang, Shicheng Li, Sida Li, Lingpeng Kong, Qi Liu, Yuanxing Zhang, and Xu Sun. Timechat-online: 80% visual tokens are naturally redundant in streaming videos. *Proceedings of the 33rd ACM International Conference on Multimedia*, 2025. URL https://api.semanticscholar.org/CorpusID:278033234.

Woongyeong Yeo, Kangsan Kim, Jaehong Yoon, and Sung Ju Hwang. Worldmm: Dynamic multimodal memory agent for long video reasoning. *ArXiv*, abs/2512.02425, 2025. URL https://api.semanticscholar.org/CorpusID:283458398.

Xiangyun Zeng, Kefan Qiu, Qingyu Zhang, Xinhao Li, Jing Wang, Jiaxi Li, Ziang Yan, Kun Tian, Meng Tian, Xinhai Zhao, Yi Wang, and Limin Wang. Streamforest: Efficient online video understanding with persistent event memory. *ArXiv*, abs/2509.24871, 2025. URL https://api.semanticscholar.org/CorpusID:281674907.

Haoji Zhang, Yiqin Wang, Yansong Tang, Yong Liu, Jiashi Feng, and Xiaojie Jin. Flash-vstream: Efficient real-time understanding for long video streams. *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 21059–21069, 2025. URL https://api.semanticscholar.org/CorpusID:280011608.

Haowei Zhang, Shudong Yang, Jinlan Fu, See-Kiong Ng, and Xipeng Qiu. Hermes: Kv cache as hierarchical memory for efficient streaming video understanding. *ArXiv*, abs/2601.14724, 2026a. URL https://api.semanticscholar.org/CorpusID:284917012.

Yulin Zhang, Cheng Shi, and Sibei Yang. Weavetime: Stream from earlier frames into emergent memory in videollms. *ArXiv*, abs/2602.22142, 2026b. URL https://api.semanticscholar.org/CorpusID:286011636.

A Other Related Work

Model Architecture and Adaptation. StreamFormer (Yan et al., 2025), LiveCC (Chen et al., 2025a), and Streamo (Xia et al., 2025) improve streaming representations through architectural adaptation and training, while Uni-X (Hao et al., 2025b) addresses modality conflict with a two-end-separated architecture. ShallowStream instead retains the pretrained MLLM and schedules computation across its depth, using shallow layers for continuous indexing and the full model for query-time answering.

Proactive Interaction and Routing. StreamMind (Ding et al., 2025), Dispider (Qian et al., 2025), and StreamBridge (Wang et al., 2025a) use event detection, decision modules, or activation models to determine when to invoke language-model inference or generate proactive responses; ViSpeak (Fu et al., 2025) and StreamReady (Azad et al., 2026) additionally consider visual interaction cues and evidence readiness. ShallowStream addresses a distinct routing problem: after a question arrives, its text-only Gate determines whether that question requires historical retrieval.

Memory Construction and Retrieval. StreamRAG (Xie et al., 2026) and EventMemAgent (Wen et al., 2026) organize event-level evidence; MuKV (Xiao et al., 2026), FlexMem (Chen et al., 2026a), and DeltaKV (Hao et al., 2026) explore multiscale KV retrieval, compressed visual states, and residual KV representations, respectively. Concurrent SelectStream and SAVEMem further investigate budgeted historical memory and query-adaptive recall (Ge et al., 2026; Wu et al., 2026). These methods vary how history is represented, compressed, or retrieved. ShallowStream instead focuses on the cost of constructing that history: it builds a shallow index, optionally compresses older states into clusters, and selects evidence before full-depth visual prefill.

Efficient Streaming Execution. Streaming Token Compression combines visual-feature reuse with token pruning (Wang et al., 2025b), while TimeChat-Online and StreamingTOM compress visual tokens (Yao et al., 2025; Chen et al., 2025b); Low-Rank Clone improves knowledge-transfer efficiency through distillation (Hao et al., 2025a). ShallowStream complements these approaches by reducing Transformer depth during continuous stream processing and invoking the full model only for selected query-time evidence, without task-specific training.

Query-Driven Long-Video Reasoning. VideoTree, WorldMM, and LongVideo-R1 use hierarchical or multi-memory search to acquire question-relevant evidence (Wang et al., 2024c; Yeo et al., 2025;

Table 3: **Query-logit Gate calibration on the benchmark-independent query-only set.**

Backbone	τ_g	Precision	95% lower bound	Recall
Qwen3-VL-8B	11.0	96.04%	91.47%	97.00%
LLaVA-OneVision-7B	0.875	97.78%	90.63%	44.00%

Qiu et al., 2026). ShallowStream similarly performs query-conditioned evidence selection, but builds its index causally before future questions and frames are available.

Streaming and Interactive Evaluation. RTV-Bench evaluates evolving answers, ProReady-QA examines evidence timing and proactive questions, and ViSpeak-Bench tests visually triggered interaction (Xun et al., 2025; Azad et al., 2026; Fu et al., 2025). Together with OVO-Bench, StreamingBench, and SVBench (Li et al., 2025; Lin et al., 2024; Yang et al., 2025b), these benchmarks cover complementary aspects of streaming understanding. Our main experiments use OVO-Bench and StreamingBench to evaluate retrospective evidence tracing and real-time visual understanding under causal access to the observed video prefix.

B Attention Context and Retrieval Depth

The layer-wise diagnostic in Figure 2 adopts ReKV's stream-time prefill configuration (Di et al., 2025), in which each incoming video unit attends to an attention sink and a bounded local window. This constraint limits stream-time attention but not retrieval: the full-history shallow index covers the entire observed stream, enabling effective shallow video–question matching without deeper prefill.

A dense complete-causal-prefix control further shows that the preferred retrieval depth depends on the attention regime. Full-history attention can improve retrieval representations at deeper layers, but its prefill cost grows with the accumulated history, preventing bounded per-unit computation on open-ended streams. ShallowStream therefore targets bounded-context streaming, where retrieval signals emerge early and full-depth computation is reserved for query-relevant evidence.

C Query-Logit Gate Calibration

We construct a benchmark-independent, query-only calibration set of synthetically generated questions labeled using the same retrospective-versus-recent definition as the ten-shot prompt; no OVO-Bench or StreamingBench queries are included. For each backbone, we apply the prompt and compute $s(q) = \ell_{\text{ret}}(q) - \ell_{\text{recent}}(q)$. We then select a backbone-specific threshold using a precision-first criterion requiring the one-sided 95% Wilson lower bound on retrieval precision to reach at least 90%, and freeze it before evaluating either benchmark (Table 3).

D Implementation Details

Table 4 summarizes the fixed configurations used in our main experiments. Both backbones use 1 FPS on both benchmarks and share the same evidence selection and decoding budgets, while the pruning boundary, Gate threshold, and video unit definition follow their respective architectures. Shallow KVs and input-level visual states are archived in host memory. In LC-off runs, this archive preserves every detailed unit; in LC-on runs, units outside the detailed window are replaced by fixed-size cluster KVs and one input representative per cluster. Query latency is measured from query entry to generation completion and therefore includes host-to-device transfer and selected-context assembly.

E Pruning-Boundary Accuracy and Cost

The first five layers already form an effective visual evidence index, allowing ShallowStream to retain its strongest accuracy without unnecessary continuous-stream depth. Figure 8 reports a full OVO-Bench Backward sweep that varies only the pruning boundary while holding routing, evidence selection, and recent-only outputs fixed. The $P = 5$ boundary, selected beforehand from LVBench,

Table 4: **Main implementation settings for the two evaluated backbones.**

Setting	Qwen3-VL-8B	LLaVA-OneVision-7B
Backbone and Stream Processing		
Pruning boundary P	5	4
Sampling rate (OVO / StreamingBench)	1 / 1 FPS	1 / 1 FPS
Video unit	2 frames	1 frame
Stream prefill window	64 units, 128 frames	128 units, 128 frames
Query Routing		
Gate prompt	ten shot	ten shot
Shared Gate calibration set	200 query-only examples	200 query-only examples
Gate threshold τ_g	11.0	0.875
Evidence Selection and Generation		
Query representation	last prompt token	last prompt token
Visual candidates per shallow layer	64 tokens	64 tokens
Vote-ranked candidate pool	32 historical units	32 historical units
Selected historical evidence	8 units	8 units
Diversity selection	max-min	max-min
Temporal expansion per selected unit	1 preceding unit	1 preceding unit
Recent context, retrieval / recent-only	6 / 2 units	6 / 2 units
Maximum new tokens	16	16

Table 5: **Cost summary under the final calibrated Gate and token-vote retriever.** Stream-time prefill is averaged over the same five long videos; query components use the matched full-history LC-off setting.

Measurement	Setting or Component	Cost
Stream-time prefill	$P = 1$	9.47 ms/frame
	$P = 5$	10.72 ms/frame
	$P = 19$	15.53 ms/frame
Query-time computation	Full query	1.759 s/query
	Gate	92.7 ms/query
	Evidence selection	106.0 ms/query

achieves the strongest EPM, HLD, and Backward average, with the latter reaching 58.27; increasing the boundary through $P = 19$ provides no further gain.

Shallow Processing Preserves Accuracy at Lower Cost. ShallowStream's efficiency stems from applying shallow computation to every incoming frame while reserving deeper processing for sparse queries. Because stream-time prefill is repeated continuously, even modest per-frame savings accumulate substantially over long videos. As shown in Table 5, the observation-guided boundary $P = 5$ reduces per-frame prefill by 31.0% relative to $P = 19$, while achieving the highest accuracy in Figure 8. This result indicates that deeper continuous prefill adds recurring cost without improving retrieval quality. At query time, Gate routing and evidence selection contribute only a small fraction of the total computation, so their overhead does not offset the savings from shallow encoding. Shallow-

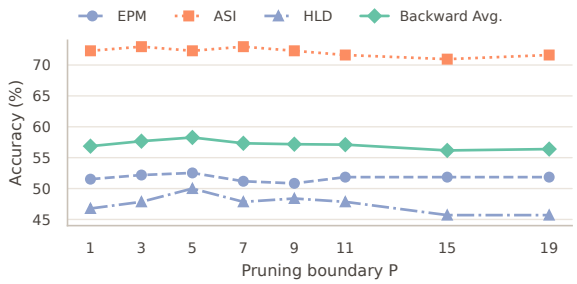

Figure 8: **Sensitivity to the pruning boundary P on OVO-Bench Backward under the final calibrated Gate and token-vote retriever.** Routing and evidence selection are held fixed across depths.

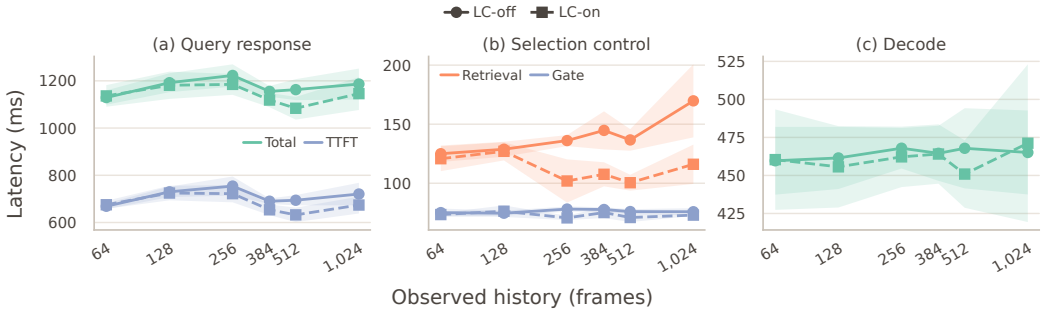

Figure 9: **Query-time scaling with retained history on an NVIDIA RTX 5090.** Results span six history lengths from 64 to 1,024 frames. *(a)* Total query latency and time to first token (TTFT). *(b)* Evidence-selection and Gate latency. *(c)* Decoding latency. Solid circles and dashed squares denote full-history LC-off and the matched final LC-on setting, respectively. Curves show means over five retrieval queries; shaded regions indicate 95% confidence intervals.

Stream therefore maintains a favorable accuracy-efficiency trade-off even with its stronger query-time evidence selection.

F Query-Time Scaling with Retained History

To assess whether query cost grows with retained history, we compare full-history LC-off with the matched final LC-on setting across a 16× range of history lengths (Figure 9). Wall-clock latency includes transfers from the host-resident archive and selected-context assembly. Under both settings, total query latency and time to first token remain nearly constant as history grows, while Gate and decoding costs are largely invariant. Long-cluster compression also keeps evidence-selection cost essentially constant, while Gate and decoding latency remain nearly invariant. Thus, even the complete shallow history remains responsive, while compression additionally bounds the history-dependent selection stage.

G Ten-Shot Router Prompt

For reproducibility, the complete prompt template used by our ten-shot query-logit router is given below. `[Input video question]` denotes the question inserted at inference time.

Ten-Shot Query-Logit Router Prompt

```
You are deciding whether answering a video question requires retrieving older video memory. The system already
      has the latest video segment immediately available. Do not answer the video question itself.

Routing options:
A. Answering reliably requires inspecting earlier video memory.
B. The latest video segment is sufficient; older video memory is unnecessary.

Here are labeled routing examples:

Example 1:
Video question: What object is the person holding in the latest visible moment?
Correct routing option: B

Example 2:
Video question: Where did the person leave the keys near the beginning of the video?
Correct routing option: A

Example 3:
Video question: What word is currently displayed on the screen?
Correct routing option: B

Example 4:
Video question: How many times did the bell ring over the course of the video?
Correct routing option: A
```

```
Example 5:
Video question: Which item did the cook pick up first?
Correct routing option: A

Example 6:
Video question: What color is the vehicle visible in the latest scene?
Correct routing option: B

Example 7:
Video question: What was inside the box before it was emptied?
Correct routing option: A

Example 8:
Video question: What action is the person performing right now?
Correct routing option: B

Example 9:
Video question: Which object is immediately to the left of the cup in the latest frame?
Correct routing option: B

Example 10:
Video question: How has the room changed compared with its appearance earlier in the video?
Correct routing option: A

Now route this video question:
<video_question>
[Input video question]
</video_question>

Output only A or B.
Routing answer:
```