REMEMBERING WHAT MATTERS: BALANCED MEMORY FOR ULTRA-LONG STREAMING VIDEOS

Anonymous authors
Paper under double-blind review

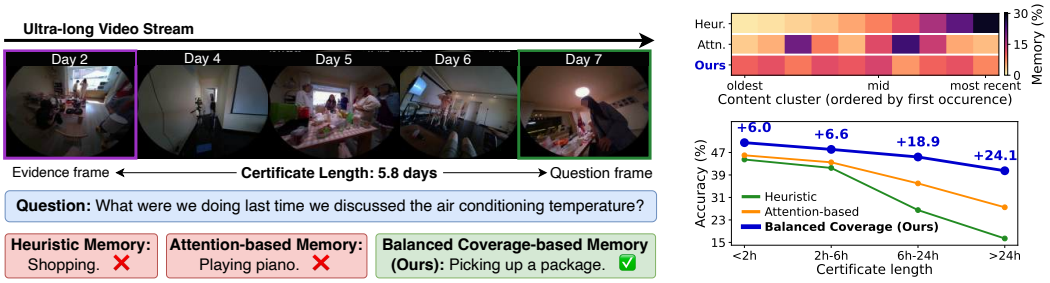


Figure 1: **Balanced temporal coverage enables stable performance on ultra-long streaming videos,** with days old evidence *(left)*, yet a fixed-size memory holds only a fraction of the stream. Existing memories decide what to keep by handcrafted heuristics or attention weighting, both degrading as the stream grows: heuristics evict by position, forgetting earlier content, while attention concentrates on dominant events and under-represents the rest *(right, top: memory distribution over the left stream)*, so both decline sharply on long-range questions *(right, bottom)*. Our memory instead retains content evenly across the timeline, sustaining stable performance as video length grows.

ABSTRACT

Multimodal Large Language Models (MLLMs) for streaming videos compress an unbounded stream of past frames into a fixed-size memory to stay within a limited LLM context budget. However, existing methods often produce unbalanced memory representations: heuristic writes induce a recency bias that coarsens older content, while attention-based writes concentrate on a few dominant events; both losing distant evidence as streams grow longer. Thus, we reformulate streaming memory compression as a **balanced temporal coverage** problem and propose **StreamBalance**, whose fixed-size memory is fully learnable and trained end-to-end through two mechanisms. A *state-aware, entropy-regularized optimal transport* formulation that routes tokens to their best-fit memory slots under a capacity constraint that keeps utilization even on write, and *adaptive decay-rate-aware retention mechanism* in which each slot learn its own persistence: fast slots merging recurring redundancy, and slow slots preserving rare events; so answer-bearing evidence is neither crowded out on write nor decayed away over time. Together, these mechanisms let StreamBalance set a new state of the art on two streaming benchmarks, such as 12.4% overall gain on OVOBench. On two ultra-long benchmarks spanning up to 44 hours, it maintains nearly stable accuracy where prior methods collapse, outperforming prior streaming model by 8% on EgoLife-QA at $1.2\times$ lesser compute and $8\times$ lower latency. Despite being trained for streaming, it further generalizes competitively to three offline long-video benchmarks.

1 INTRODUCTION

Multimodal large language models (MLLMs) have shown remarkable capabilities at video understanding, answering questions and describing events in clips spanning seconds to minutes (Li et al., 2023b; Zhang et al., 2023; Li et al., 2025a; Abdin et al., 2024). As videos grow longer, modeling such extensive temporal context strains their limited context length, so the content the LLM reasons over must be reduced. Recent works address this offline through adaptive frame sampling (Zhu et al., 2026; Tang et al., 2025; Wang et al., 2024c; 2025d), KV-cache compression (Zhang et al., 2024a; Shu et al., 2025; Liu et al., 2025b), and fixed-size external memory (Song et al., 2024a; He et al., 2024; Azad et al., 2025). Yet all share one assumption: the entire video and the query is available in advance. This breaks in the *streaming* setting, where frames arrive sequentially, queries may be

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.

posed at any moment, and the model can neither look ahead nor revisit the raw past. It must commit, online and irreversibly, to what it retains before knowing what will be asked; a constraint that makes streaming fundamentally harder than its offline counterpart, yet far closer to how interactive assistants, embodied agents, and live monitoring systems operate in practice.

Because incoming frames are unbounded, streaming models cannot sample from a complete video; they instead watch every frame but retain only a small, fixed-size representation, via KV-cache compression (Di et al., 2025; Xiao et al., 2026) or an external memory bank (Wang et al., 2025a; Zeng et al., 2025; Xiong et al., 2025). These are *heuristic* in both write and update: content is admitted at a fixed position (queue-front or tree-leaves) and compressed by heuristic similarity-based merging rules, with no downstream supervision over what is kept. They are further organized as a predefined temporal hierarchy that holds recent frames at full detail while coarsening older content (Azad et al., 2026; Xie et al., 2026); drawn by time rather than content, this induces strong *recency bias* that progressively compresses older content until distant-but-important events are lost. Learnable write modules (Ryoo et al., 2023; Piergiovanni et al., 2024), although not designed for streaming, could be adapted to it, write by attention weights: a winner-takes-all paradigm that concentrates memory on a few dominant events and under-represents the rest. Across both heuristic and attention-based memories, the same failure recurs: neither distributes memory in a balanced way, so as the required evidence recedes into the past, these memories progressively collapse (Figure 1).

Avoiding this collapse requires a memory that keeps coverage *balanced* across the timeline, admitting evidence from any point in the stream with comparable fidelity, rather than concentrating on recent or dominant content as above. We formalize this as a **balanced temporal coverage** problem, addressing the two failure points identified above in turn. First, to prevent content from being admitted at a fixed position or by winner-takes-all attention, each frame's tokens must instead be assigned to the memory slots they best *fit* while remaining aware of each slot's current content, under an explicit constraint that no slot is overloaded and no region of the timeline is starved: this makes memory allocation *even* and content-driven rather than tied to position or a few dominant slots. Second, assignment alone governs only *where* content lands on arrival; to prevent the recency bias induced by a fixed temporal partition, keeping coverage balanced over time also requires deciding *how long* each slot holds what it was given, rather than coarsening by age alone. Balanced temporal coverage thus demands both a balanced write and a principled adaptive retention policy, neither of which existing memories, built on fixed-position writes and heuristic eviction, can support.

Thus, we propose **StreamBalance**, a lightweight streaming video understanding framework that instantiates both halves of this formulation. Inspired by the Optimal Transport problem (Villani et al., 2009; Cuturi, 2013), StreamBalance realizes the **balanced write** with a state-aware, fully differentiable online optimal transport iteration that routes each frame's tokens into the fixed slot-set while keeping utilization even. It realizes **adaptive retention** by updating each slot on its own *adaptive decay-rate*: fast slots merge redundant, frequently recurring content into a single refreshed cell, while slow slots resist merging and instead allocate distinct cells to preserve rare events, letting persistence be allocated by *task relevance* rather than a hand-set temporal partition. This directly serves balanced coverage: removing redundancy where it occurs, rather than uniformly, frees capacity for genuinely distinct content, keeping memory from being consumed by repetition in one region while starving others. Both operations are jointly necessary, exactly as the formulation requires: the balanced write spreads content evenly across slots on arrival, and adaptive retention holds that spread steady as the stream lengthens, so evidence from any point in the timeline, however far back, remains recoverable at constant memory cost rather than eroding with distance. Both mechanisms are trained end-to-end only from the downstream loss, without any additional auxiliary loss.

We evaluate StreamBalance on two general streaming and two ultra-long streaming benchmarks, establishing new state of the art on the former, such as a 12.4% overall improvement on OVOBench. As the ultra-long benchmarks were orignally proposed for offline agentic use, we adapt them to a strictly causal streaming protocol and establish the *first* streaming baselines on them, re-evaluating prior streaming methods identically. Here, where prior methods collapse as video length increases, StreamBalance maintains nearly stable accuracy, outperforming the previous best streaming model by 8% on EgoLife-QA, using $1.2\times$ less compute and $8\times$ lower latency. It further generalizes to three offline long-video benchmarks, achieving competitive performance despite being trained for the streaming setting. To summarize, our contributions are as follows:

- We identify a shared failure of streaming memories, *unbalanced coverage* that over-weights recent or a few dominant events and worsens as the stream grows, and reinterpret streaming memory compression as a **balanced temporal coverage problem**: an end-to-end learned mechanism, in place of heuristics, that distributes an unbounded stream across a fixed-size memory while keeping coverage balanced as the video lengthens.
- We introduce **StreamBalance**, a lightweight streaming framework whose memory is made fully learnable through two mechanisms: (1) *Balanced-assignment write*, a state-conditioned optimal-transport formulation distributing content evenly across memory, and (2) *Adaptive decay rate-aware retention*, letting each slot learn its own persistence, merging redundancy while protecting distant evidence, keeping coverage balanced across time.
- StreamBalance sets a new state of the art on standard streaming benchmarks, generalizes competitively to offline long-video benchmarks, and sustains stable performance on ultra-long streaming video at $\mathbf{1.2}\times$ **lower compute and** $\mathbf{8}\times$ **lower latency**.

2 Related Works

Offline Long Video Understanding with MLLMs. Understanding long videos requires modeling temporal dependencies across minutes to hours, addressed through adaptive frame sampling (Zhu et al., 2026; Tang et al., 2025; Liu et al., 2025a; Pan et al., 2026), memory-augmented (He et al., 2024; Song et al., 2024b; Cheng et al., 2024; Azad et al., 2025), or agentic architectures (Wang et al., 2024b; Zhi et al., 2025; Montes & Lozano, 2025; Kugo et al., 2025). All assume the full video is available at query time and are inherently query-conditioned, recomputing memory or sampled frames per question. This is precisely what the streaming setting forbids: frames and queries arrive sequentially, and the model must commit to what it retains before knowing what will be asked.

Streaming Video Understanding with MLLMs. To meet this constraint, streaming methods process frames incrementally and reuse past context through a bounded memory (Qian et al., 2025; Zhang et al., 2025; Zeng et al., 2025; Xiong et al., 2025; Chen et al., 2026; Zhang et al., 2026b; Xie et al., 2026) or KV-cache retrieval (Di et al., 2025; Yang et al., 2025b; Kim et al., 2025; Zhang et al., 2026a; Wang et al., 2026a; Ning et al., 2026). Nearly all decide what to retain or discard by heuristic rules, with no gradient signal from the downstream task; the only learnable streaming memory (Wang et al., 2025c) is unbounded, growing its store and compute linearly with video length. While such memories perform reasonably on short streams, they degrade sharply as streams lengthen. We identify the cause as *unbalanced temporal coverage*, and build our framework around ensuring it, sustaining stable performance across ultra-long streams at constant cost and latency.

Token Compression in Videos. Orthogonal to the setting, a large body of work reduces tokens by spatial pruning (Bolya et al., 2023; Shang et al., 2025; Xing et al., 2025) or spatio-temporal merging (Shen et al., 2025; Tao et al., 2025; Liu et al., 2025c; Shao et al., 2025; Shen et al., 2024), but relies on bidirectional attention or global analysis over the full video, violating streaming causality. Among streaming-compatible methods, memory write and update mechanism determines coverage. Some write at a fixed position and merge by heuristic similarity (Song et al., 2024a; Cheng et al., 2024; Song et al., 2024b), inducing a *recency bias* that coarsens older content. Others learn what to write via attention (Ryoo et al., 2023; He et al., 2024; Wu et al., 2022); though not yet applied to streaming, such learnable writers can be adapted to the setting, but they still merge by heuristics, and being driven by attention weights, concentrate on a few *dominant events*, so learnability at write time does not extend to consolidation. Either way, coverage is left unbalanced. We instead propose a balanced-coverage memory with both a learnable write *and* a learnable update, trained end-to-end from the downstream loss; to our knowledge, the *first* such memory for streaming compression.

3 Method

Overview. To ensure stable performance over ultra-long streams, we reformulate streaming memory compression as a *balanced temporal coverage* problem: content from across the timeline must be retained evenly, rather than skewed toward recent or a few dominant events. We propose **StreamBalance**, an MLLM-based streaming framework built to support this formulation with a fully learnable memory coupling a balanced write with adaptive retention. StreamBalance encodes each incoming frame with a frozen vision encoder into frame features $f_t \in \mathbb{R}^{N\times D}$ (a set of N tokens of dimension D) and compresses it into a fixed-size memory bank $M \in \mathbb{R}^{N_k\times L\times D}$ of N_k slots, each a collection of L cells. Compression operates at two granularities: writing assigns tokens at the *slot* level under

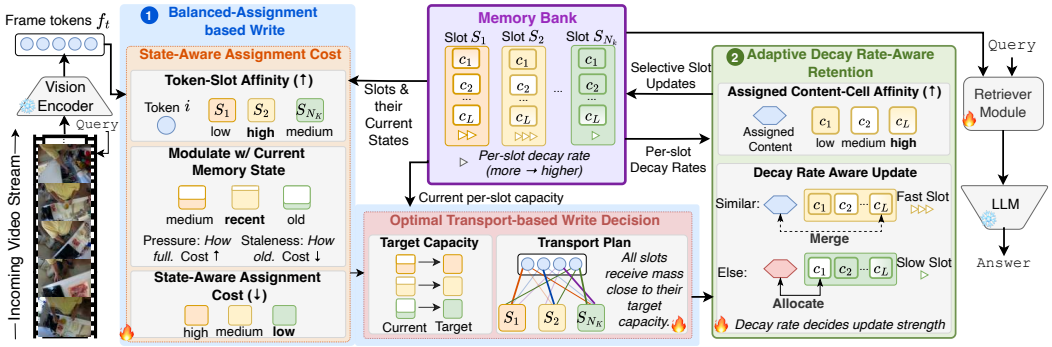


Figure 2: **Framework Overview.** StreamBalance incrementally encodes and compresses streaming observations into a balanced memory bank through state-aware balanced assignment and adaptive decay-rate-aware retention, preserving broad temporal coverage. At inference, the retriever selects query-relevant memory contents for the LLM to perform downstream reasoning.

a balanced-assignment constraint (§3.1), retention integrates content at the *cell* level via adaptive per-slot decay rate (§3.2): together keeping coverage balanced as the stream grows. When a query arrives, a lightweight retrieval module reads the query-relevant content at the cell level of M and passes it, to the frozen LLM to generate the response (§3.3). The framework is trained end-to-end under the downstream vision-language loss. An overview of the framework is shown in Figure 2.

3.1 Balanced Assignment-based Memory Write

Prior streaming memories primarily admit content at a *fixed position*, the front of a queue or the leaves of a tree, and consolidate by heuristic similarity (Wang et al., 2025a; Xiong et al., 2025). Such fixed-position writes push older content down and progressively merged; tying compression to *arrival time* rather than content, inducing a recency bias; attention-driven writes instead admit by dominance, concentrating on a few salient events. As a result, in both cases timeline coverage becomes uneven. We instead view the write as a *balanced-assignment* problem: an unbounded stream of tokens must be distributed across a fixed set of N_k slots so that assignment follows content-fit yet remains balanced, with no slot starved or overloaded. This makes each slot a *learned anchor* specializing in a kind of content rather than a positional bucket, letting any token be written to any slot by fit, removing the fixed, recency-driven entry point of prior memories.

Optimal Transport for Balanced Writing. Distributing token mass across slots under an optimal cost while keeping each slot near a target capacity fits naturally into the framework of an Entropy-Regularized oOptimal Transport problem (Villani et al., 2009; Cuturi, 2013): given a cost matrix, it yields the coupling of minimum cost between two distributions whose marginals are pulled toward matching them, regularized to be unique and differentiable rather than hard and combinatorial. Concretely, each token distributes its mass across slots by cost, while every slot's aggregate load is pulled toward its target share, yielding a balanced, differentiable assignment learnable through the downstream loss. Classical transport (Villani et al., 2009), however, is *stateless and stationary*: the cost and both marginals are fixed in advance, independent of any assignment history. Streaming memory violates both assumptions: since the memory already holds content, the right assignment depends on the memory's current occupancy, and the correct per-slot target shifts as the stream evolves.

We therefore make both inputs of the transport problem *state-conditioned*: a state-aware cost and a per-slot capacity, both predicted from the current memory, so that each frame instantiates a new transport problem shaped by the memory it faces rather than a fixed one solved in isolation.

State-Aware Assignment Cost. We measure token-slot affinity, where higher affinity lowers cost:

$$C = 1 - \cos(f_t, S) = \left[c_{ij}\right] \in \mathbb{R}^{N \times N_k}, \qquad c_{ij} = \text{cost of assigning token } i \text{ to slot } j. \tag{1}$$

$S \in \mathbb{R}^{N_k \times D}$ is a set of learnable slot anchors; one per slot, updated only via the task loss and independent of the memory bank's stored content; so tokens route by learned semantics rather than transient appearance. This affinity alone is blind to occupancy: it would keep routing to a well-matched slot even after it saturates, and never revisit one that has gone stale from disuse. We correct

for this with each slot's pressure, p_j and staleness, s_j, modulating C into a state-aware cost:

$$\hat{C}ij = Cij + \lambda_p p_j - \lambda_s s_j, \qquad \text{where, } p_j = \frac{m_j}{\max\left([m_1, \ldots, m_{N_k}]\right)}, \quad s_j = \frac{t - \xi_j}{t}, \tag{2}$$

m_j is the running sum of assigned mass (defined below) in slot j across all past frames, tracking how heavily a slot has been favored by assignment over time, and ξ_j is its last-written frame index. λ_p, λ_s are non-negative learnable coefficients. Pressure raises the cost of writing to full slots while staleness lowers it for dormant ones. Both trackers update after every write, so for every frame, the assignment adapts to current memory usage and slot occupancy self-regulates. Crucially, these modulate only slot-level *routing*, and do not by themselves decide what a slot retains; whether stored content survives is a separate cell-level decision (§3.2) governed by content redundancy.

Transport Plan. The per-slot capacity sets how much each slot should absorb given what it already holds, which we predict from the current memory with a lightweight MLP: the mean-pooled representation of all currently stored cells across slots of M, yielding a distribution b. Since both $\hat{C}$ and b are recomputed from the memory's current state at every frame, each frame poses a distinct transport problem rather than reusing a fixed one; which we solve via the Sinkhorn algorithm (Cuturi, 2013), yielding a unique transport plan P:

$$P = \mathrm{OT}(\hat{C}, b) \in \mathbb{R}^{N \times N_k}, \qquad \text{where, } P_{ij} = \text{fraction of token } i\text{'s mass routed to slot } j. \tag{3}$$

Each token's mass is thus distributed across slots by fit and state, while each slot's total received mass is balanced toward its learned capacity b_j, so no slot is overloaded or starved. From P, each slot j receives content Δ_j and write strength w_j:

$$\Delta_j = \frac{1}{w_j} \sum_i P_{ij} f_i \in \mathbb{R}^D, \qquad \text{where, } w_j = \sum_i P_{ij}, \tag{4}$$

P thus turns a frame's N tokens into at most N_k slot-level updates. How each slot integrates and retains Δ_j over time is governed by the adaptive retention mechanism. All operations leading to this transport plan are differentiable, so the entire write trains end-to-end from the downstream loss.

3.2 Adaptive Decay Rate Aware Retention

Although the transport plan assigns Δ_j to slot j, we only write to it when $w_j > c/N_k$, where c is a fixed threshold coefficient, since writing on every marginal or noisy assignment would waste capacity on content the transport plan itself considered a weak fit. For each written slot, the pressure/staleness trackers are updated, $m_j \leftarrow m_j + w_j, s_j = t$. Each slot is a learnable anchor implemented as a fixed-size collection of L cells that already store prior content. So once this threshold is cleared, the slot must still decide *how* to absorb Δ_j: overwrite an existing cell, or keep it as a distinct event. This is where content is actually retained or forgotten, and how a slot resolves it determines whether early evidence survives as the stream grows.

Learning Per-slot Decay Rates. Different content demands different persistence: fast-changing, redundant detail can be refreshed in place, while rare, salient events must be held long enough for distant queries to recover them. Prior memories impose this split by a fixed temporal partition, recent content at full detail, older content progressively coarsened, misjudging persistence by age alone: an early salient event is coarsened simply for being old, while recent redundancy is kept at full detail.

We instead let each slot *learn* its own decay rate α_j: fast slots (high α_j) overwrite readily into one refreshed cell, while slow slots (low α_j) preserve and accumulate distinct events, so persistence is allocated by *task relevance* rather than by recency. Concretely, α_j governs both *when* to merge incoming content into the closest existing cell and *how strongly*, based on their similarity:

$$c_{k^*} \leftarrow \begin{cases} (1-\lambda_j)c_{k^*} + \lambda_j \Delta_j, & \text{sim} \geq \tau_j, \\ \text{allocate new cell}, & \text{otherwise}, \end{cases} \quad \text{where, } \lambda_j = \alpha_j \frac{1 - \text{sim}}{2}, \text{and } \tau_j = \tau_0 + \eta\alpha_j; \tag{5}$$

where, $sim = \cos(\Delta_j, c_{k^*})$. Redundant content (sim $\to$ 1) drives $\lambda_j \to 0$, leaving cells almost unchanged. When a slot is full, and Δ_j is novel, rather than evicting by age, we merge the two most similar existing cells and allocate the freed cell to Δ_j, so consolidation removes the most redundant content regardless of when it arrived. Crucially, both this step and the merge decision above are governed by the slot's *adaptive* decay rate: unlike fixed similarity or age-based rules, what counts

as redundant and how aggressively it is consolidated is task-driven and differs per slot. Each cell additionally stores the interval $[t_{\text{first}}, t_{\text{last}}]$ of when it was first and last written; distinct from the per-slot staleness index ξ_j, providing an additional temporal signal for downstream reasoning.

Learning to Merge. Left as is, the merge-vs-allocate decision in Eqn. 5 would not be learnable: it is a hard threshold $d = \mathbb{1}[\text{sim} \geq \tau_j]$, a step function with zero gradient almost everywhere, so no learning signal can flow through it and and the decision would fall back to a fixed, heuristic threshold. We, therefore, apply a straight-through estimator (Bengio et al., 2013) to approximate the gradient of this discrete decision, defining a surrogate that preserves the hard decision in the forward pass while letting the gradient flow through it during backpropagation:

$$\hat{d} = d + \text{sim} - \text{sg}[\text{sim}], \tag{6}$$

where $\text{sg}[\cdot]$ is the stop-gradient operator: $\hat{d} = d$ in the forward pass, while $\partial\hat{d}/\partial\text{sim} = 1$ in the backward pass. The merge criterion is thus also learnable (Full gradient derivation in §E).

3.3 Query-Aware Retrieval and Reasoning

The write and retention mechanisms produce a memory whose coverage is balanced across the timeline but agnostic to any particular query; retrieval is where this balance is put to use, since evidence for any question, regardless of when it occurred, is equally accessible rather than crowded out by recency or a few dominant events.

Query-Aware Retrieval. When a query arrives at timestamp t, it is encoded into a text embedding q_t and passed to the *retriever*, a single transformer-decoder layer that returns the memory cells most relevant to the query. The top-K attended cells, each carrying its own timestamp interval, are projected and passed with q_t to the LLM, which decodes the answer.

Optimization. The entire pipeline: the balanced assignment write, the adaptive decay-rate-aware retention, and the retriever, is trained end-to-end under the standard cross-entropy-based vision-language loss, so what to store, keep, and retrieve is shaped directly by answer quality.

4 Experiments

Implementation Details. We build StreamBalance on top of frozen MLLM backbones and show generalization across 3 backbones: Qwen2-VL-7B (Wang et al., 2024a), Qwen2.5-VL-7B (Bai et al., 2025c), and Qwen3-VL-7B (Bai et al., 2025a), training only the balanced write, adaptive retention parameters, and the retriever. The memory bank holds $N_k = 128$ slots, each with $L = 8$ cells; the retriever returns the top-$K = 64$ cells. Additional details in §F.

Benchmarks and Metrics. We evaluate general streaming understanding on OVO-Bench (Li et al., 2025b) and StreamingBench (Lin et al., 2024), and ultra-long streaming on EgoLife-QA (Yang et al., 2025a) and Ego-R1 Bench (Tian et al., 2026a); the former average 1-30 minutes per video, the latter roughly 44.3 hours, testing StreamBalance over far longer horizons. We further test offline generalization on VideoMME (Fu et al., 2025a), MLVU (Zhou et al., 2024), and LongVideoBench (Wu et al., 2024). Videos are sampled at 1 FPS, with accuracy as reported evaluation metric.

4.1 Results

General Streaming Understanding. Table 1a reports performance on StreamingBench and OVO-Bench. StreamBalance consistently outperforms the previous best streaming method across backbones, with the largest single gain with Qwen2.5-VL on OVOBench's Backward split ($+22.5$), where evidence lies furthest in the past, indicating that balanced coverage matters most exactly where recency-biased memories have the most to lose.

Ultra-Long Streaming Understanding. Table 1b reports results on EgoLife-QA and Ego-R1 Bench, whose videos span 44.3 hours on average. These benchmarks were originally addressed only by agentic MLLMs, which operate offline with unconstrained access to the full video and query it with external tools and auxiliary models over multiple stages, unlike our *single* causal streaming pass. We therefore report them as a *performance ceiling*, not a direct comparison, and instead adapt the benchmarks to a strictly causal streaming protocol, establishing the *first* streaming baselines on them for matched comparison. Against these streaming comparators, StreamBalance is markedly stronger, e.g. $+8.3$ boost on EgoLife-QA with Qwen2.5-VL over the prior best streaming models,

Table 1: **Evaluation of general and ultra-long streaming video understanding.** Accuracy is the reported metric and models use 7B backbone, unless noted. Open-source **best** scores are highlighted.

(a) **General:** StreamingBench (StrmB) and OVOBench.

Method	StrmB	OVOBench			
	Real	Real	Back	Fwd	All
Proprietary Offline Video MLLMs					
Gemini 1.5 Pro (Team et al., 2024)	75.7	69.3	62.5	57.2	63.0
GPT 4o (OpenAI, 2024)	73.3	64.5	60.8	53.4	59.5
Open-Source Offline Video MLLMs					
InternVL3 8B (Chen et al., 2025)	68.7	61.8	45.2	58.0	57.0
LLaVA-NeXT-V 32B (Zhang et al., 2024b)	69.8	63.3	41.7	54.2	53.1
LLaVA-OneVision (Li et al., 2024a)	71.1	62.8	45.0	50.9	52.7
LongVU (Shen et al., 2024)	-	57.4	39.5	48.5	48.5
VITA 1.5 (Fu et al., 2025b)	68.2	63.5	41.5	53.5	55.5
Open-Source Online Video MLLMs					
VidLLM-online 8B (Chen et al., 2024a)	36.0	20.8	17.7	-	-
Flash-VStream (Zhang et al., 2025)	23.2	29.9	25.4	44.2	33.6
Dispider (Qian et al., 2025)	67.6	54.6	36.1	34.7	41.8
ReKV (Di et al., 2025)	37.4	64.4	46.7	47.8	52.6
STC (Wang et al., 2026b)	65.2	62.5	45.3	48.0	52.0
Qwen 2 VL (Wang et al., 2024a)	69.0	60.7	48.6	48.9	50.4
+ StreamForest (Zeng et al., 2025)	77.3	61.2	52.0	53.5	55.6
+ StreamBridge (Wang et al., 2025a)	77.0	71.3	68.1	48.4	62.6
+ ViSpeak (Fu et al., 2025c)	70.4	66.3	57.5	54.3	61.1
+ InfiniPot-V (Kim et al., 2025)	76.4	65.9	47.6	47.9	53.6
+ StreamReady (Azad et al., 2026)	78.3	73.6	72.2	58.8	68.2
+ StreamBalance (Ours)	**81.2**	**83.4**	**79.5**	**60.8**	**74.6**
Qwen 2.5 VL	73.7	59.9	44.7	38.5	46.5
+ TimeChat-Online (Yao et al., 2025a)	75.4	58.6	42.0	36.4	47.7
+ HERMES (Zhang et al., 2026a)	79.4	70.0	49.4	-	-
+ Streamo (Xia et al., 2026)	-	66.0	46.1	54.8	55.6
+ QueryStream (Zhang et al., 2026b)	75.3	63.1	44.9	40.2	49.4
+ Thinking-Qwen (Zhang et al., 2026c)	71.6	64.7	44.3	36.1	52.5
+ StreamBalance (Ours)	**82.1**	**75.1**	**71.9**	**57.1**	**68.0**
Qwen 3 VL 8B (Bai et al., 2025a)	77.3	78.7	51.8	43.4	58.0
+ StreamBalance (Ours)	**82.7**	**85.1**	**79.2**	**59.3**	**74.5**

(b) **Ultra-long:** EgoLife-QA (EgoLife) and Ego-R1 Bench (Ego-R1B). Agentic models are shown as a performance ceiling, not a direct comparison.

Method	EgoLife	Ego-R1B
Proprietary Offline Video MLLMs		
Gemini 1.5 Pro (Team et al., 2024)	36.9	38.3
GPT-4o (OpenAI, 2024)	36.2	-
Open-Source Agentic Video MLLMs		
VideoAgent (Wang et al., 2024b)	29.2	32.6
EgoRAG (Yang et al., 2025a)	52.0	49.0
Ego-R1 3B (Tian et al., 2026a)	36.0	46.0
M3-Agent (Long et al., 2026)	53.5	52.0
WorldMM 8B (Yeo et al., 2026)	56.4	52.0
EgoRetriever (Tang et al., 2026)	41.5	-
Open-Source Offline Video MLLMs		
InternVideo 2.5 (Wang et al., 2025b)	33.0	34.0
LLaVA-Video (Zhang et al., 2024c)	36.4	29.0
LLaVA-OneVision (Li et al., 2024a)	30.8	31.6
VideoChat-Flash (Li et al., 2024b)	34.2	42.7
LongVA (Zhang et al., 2024a)	33.0	23.0
Open-Source Online Video MLLMs		
Qwen 2 VL (Wang et al., 2024a)	31.1	28.4
+ StreamReady (Azad et al., 2026)	31.5	29.8
+ StreamBalance (Ours)	**38.5**	**32.6**
Qwen 2.5 VL (Bai et al., 2025c)	35.4	34.7
+ HERMES (Zhang et al., 2026a)	30.9	28.3
+ TimeChat-Online (Yao et al., 2025a)	30.8	25.3
+ StreamBalance (Ours)	**43.7**	**38.3**
Qwen 3 VL 8B (Bai et al., 2025a)	38.6	35.7
+ StreamBalance (Ours)	**42.3**	**39.7**

and even closes much of the gap to the agentic ceiling using only a fixed, bounded memory and no external retrieval, highlighting that balanced coverage scales even on ultra-long streamings.

Offline Video Understanding. While specifically designed for streaming, we test StreamBalance's generalization to offline benchmarks, where the full video is available at once rather than arriving as a stream. Table 2 shows consistent, competitive performance across all benchmarks, with the largest gain on VideoMME's Long split (+7.9 on Qwen3-VL), similarly confirming that balanced coverage helps most where evidence must be integrated across the longest durations.

4.2 Ablation

Impact of Each Mechanism and their Components. Table 3*(a)* shows removing either balanced write or adaptive retention (falling back on heuristics) degrades performance, and removing both is far more damaging than either alone, confirming the two are complementary. Degradation concentrates on temporal-reasoning splits, sharpest for evidence furthest in the past, while performance relying on recent frames (OVB-Real) stays comparatively robust; this pattern holds on both OVOBench and EgoLife-QA, but is stronger on EgoLife-QA, confirming balanced coverage matters most as the timeline a query must recover from grows longer.

Within balanced write (Table 3*(b)*), removing state-aware modulation hurts most, degrading temporal-reasoning performance across both benchmarks; removing pressure alone hurts long-range recall most, as saturated slots keep being overwritten, while removing staleness has a milder, broader effect. Removing the learned capacity constraint also drops performance, though less: coverage stays balanced either way, but loses whether balancing is state-aware and adaptively targeted rather than content-blind. Within adaptive retention (Table 3*(c)*), replacing per-slot decay rates with a single global rate again costs most on long-range recall, since per-slot rates let slow slots retain long-range information while fast slots track recent content; removing the learned merge likewise degrades performance, isolating the value of learning what counts as redundant.

Table 2: **Evaluation of offline video understanding** on VideoMME (VidMME), MLVU, LongVideoBench (LVB). Accuracy is reported.

Method	VidMME All	VidMME Long	MLVU M-avg	LVB Avg
Proprietary Offline Video MLLMs				
Gemini 1.5 Pro (Team et al., 2024)	75.0	67.4	-	64.0
GPT 4o (OpenAI, 2024)	71.9	65.3	64.6	66.7
Open-source Offline Video MLLMs				
InternVL3.5 (Chen et al., 2024b)	66.0	-	70.2	62.1
LLaVA-Video (Zhang et al., 2024c)	62.6	50.6	70.8	58.2
LLaVA-OneVision (Li et al., 2024a)	58.2	46.7	64.7	56.4
Open-source Online Video MLLMs				
Flash-VStream (Zhang et al., 2025)	61.2	50.3	66.3	-
Focus (Zhu et al., 2026)	65.4	56.1	-	63.5
StreamingTom (Chen et al., 2026)	59.9	50.6	67.9	-
Qwen 2 VL (Wang et al., 2024a)	63.3	54.4	65.8	-
+ InfiniPot-V (Kim et al., 2025)	62.8	-	65.8	56.5
+ StreamBridge (Wang et al., 2025a)	64.4	-	69.6	59.1
+ StreamReady (Azad et al., 2026)	65.8	54.5	71.3	62.5
+ Streambalance (Ours)	**68.3**	**62.1**	**72.6**	**63.8**
Qwen 2.5 VL (Bai et al., 2025c)	65.1	51.6	70.2	54.7
+ TimeChat-Online (Yao et al., 2025a)	62.5	48.4	62.6	55.4
+ Streamo (Xia et al., 2026)	67.9	-	-	59.2
+ HERMES (Zhang et al., 2026a)	60.6	53.4	-	-
+ TimeSearch-R (Pan et al., 2026)	64.1	53.3	**71.3**	61.6
+ Streambalance (Ours)	**68.1**	**61.4**	71.1	**62.6**
Qwen 3 VL 8B (Bai et al., 2025a)	**71.8**	56.3	**75.1**	61.5
+ Streambalance (Ours)	71.4	**64.2**	74.8	**64.2**

Table 3: **Ablation studies for each component.** Here, ELife denotes EgoLife-QA.

Method	OVOBench Real	OVOBench Back	OVOBench Fwd	OVOBench Overall	ELife
(a) Impact of each mechanism					
Full model	**83.4**	**79.5**	**60.8**	**74.6**	**43.7**
– Balanced write	79.2	68.5	53.4	67.0	30.5
– Adaptive retention	75.7	70.3	52.5	66.2	32.4
– Both	68.2	51.5	45.1	54.9	25.0
(b) Impact of each balanced write component					
– State-aware cost	80.1	70.7	54.1	68.3	32.1
– Pressure	81.6	73.1	56.4	70.4	34.4
– Staleness	80.8	76.9	57.5	71.7	35.3
– Learned slot capacity	82.5	74.6	56.2	71.1	37.7
(c) Impact of each adaptive retention component					
– Adaptive decay rate	78.4	75.1	55.1	69.5	34.7
– Learned merge	80.2	76.3	58.4	71.6	37.1
(d) Impact of different learnable write mechanisms					
Balanced write	**83.4**	**79.5**	**60.8**	**74.6**	**43.7**
Expert choice (Zhou et al., 2022)	81.2	78.8	59.1	73.0	40.1
TTM (Ryoo et al., 2023)	72.2	59.4	51.4	61.0	34.6
QFormer (Li et al., 2023a)	76.2	65.4	54.4	65.3	37.1
(e) Impact of different retention mechanisms					
Adaptive retention	**83.4**	**79.5**	**60.8**	**74.6**	**43.7**
Fixed hierarchy (Azad et al., 2026)	82.4	62.7	51.1	65.4	30.1

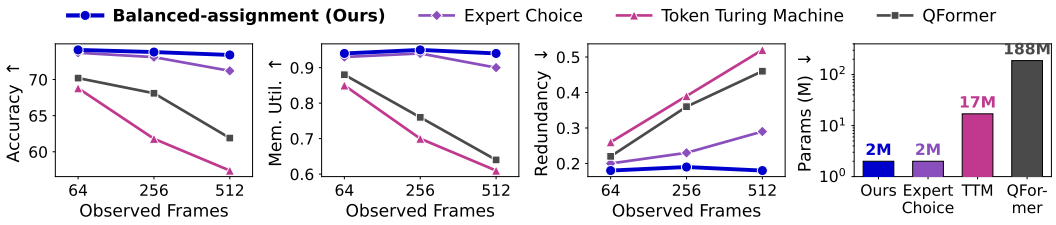


Figure 3: **Different learnable write mechanisms across stream lengths** (OVOBench).

Alternative Choices for Each Component. Table 3(*d*) shows that swapping balanced write for other learnable write schemes degrades accuracy throughout: Expert-choice routing (Zhou et al., 2022) comes closest, while attention-based writers (Ryoo et al., 2023; Li et al., 2023a) fall furthest, with the gap widest on long-range tasks. Swapping adaptive retention for a fixed hierarchy (Azad et al., 2026) (Table 3(*e*)) that coarsens content by age also degrades long-range performance, confirming that time-drawn hierarchies discard distant evidence, while learned, task-driven decay rates retain it. Notably, here a single component is swapped for an alternative while holding the rest of the architecture fixed, so any difference is attributable to that component alone.

4.3 Analysis

Effect of different learnable memory write mechanisms on stream length. Figure 3 shows the behavior of different learnable memory write mechanisms across stream length, reporting accuracy alongside *memory utilization* (effective fraction of slots in use) and *content redundancy* (mean pairwise cosine among stored cells) at matched memory budget. All four writers are comparable at short streams, but diverge as length grows and content competes for a fixed budget. Attention-based writers (Token Turing Machine, Q-Former) allocate by softmax weights that concentrate on a few dominant events: utilization falls, redundancy climbs, and accuracy degrades, despite having far more trainable parameters. Enforcing any per-slot capacity constraint (Ours, Expert Choice) keeps utilization high and lifts accuracy substantially over unconstrained attention, showing that bounding how much each slot absorbs is what first separates balanced from collapsing memories. Utilization alone, however, is not sufficient: Expert Choice keeps utilization balanced yet lets *redundancy* rise with length, because each slot selects its tokens independently and several slots can store the same content, so balanced occupancy is spent storing duplicates, and accuracy trails ours. Our optimal-transport formulation instead solves a single joint assignment over all tokens and slots at once, which discourages different slots from claiming the same content and keeps redundancy low and *flat* across

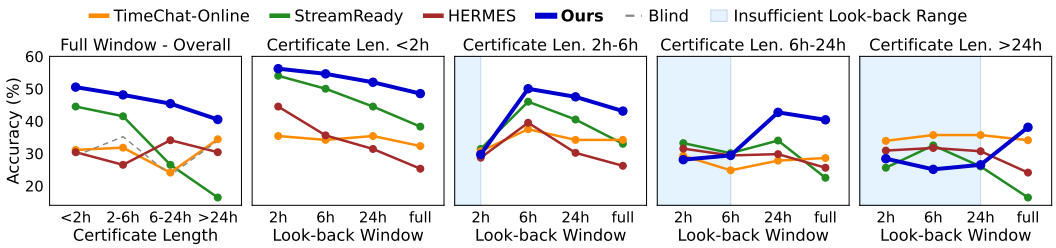


Figure 4: **Impact of certificate length and look-back window on ultra-long streaming (EgoLife-QA).** *Leftmost*: accuracy vs. certificate length (minimum span needed to answer); *Others*: accuracy vs. look-back window within each certificate group, where shading marks windows too short to cover the certificate. Blind denotes a no-video model (chance floor).

lengths while utilization stays near-full. Balanced coverage thus requires *both* even utilization and low redundancy; our formulation is the only writer that sustains both as the stream grows, and does so at a fraction of the parameters of the attention-based writers.

Robustness Across Varying Certificate Lengths. Figure 4 (*leftmost*) plots performance against certificate length, the minimum video span needed to answer, alongside a *blind* (no-video) model that answers at chance, marking a no-evidence floor. A model that truly reasons over evidence should decay gently as certificates lengthen, since the answer sits further back and more compressed. Prior methods instead trace erratic curves near this floor: at long certificates their memory no longer holds the evidence, so they answer at near-chance, indistinguishable from having no video. StreamReady holds a meaningful trend but collapses at the longest certificates. Ours stays well above the floor and declines gently, the graceful decay expected of a genuine visual reasoning model.

To trace *what drives* this graceful-vs-collapsing split, we sweep the *look-back window* (how much past video the model sees) within each certificate group (Figure 4, *remaining subplots*); shaded regions mark windows too short to cover the certificate, where all methods sit at the floor. Once the window covers it, accuracy rises then falls as added history competes for the fixed budget, far more sharply for prior methods. The longest-horizon group (Figure 4, *rightmost*) is decisive: at the widest window ours keeps rising as evidence arrives, while prior methods decline, one even below the blind floor, suggesting its long-range memory has degraded to actively misleading content. Our balanced memory instead stays *stable across horizon*, the core requirement for ultra-long streaming.

Inference Latency and Compute Analysis. Figure 5 reports latency as Time To First Token (TTFT) and compute as peak VRAM as the observed stream grows, on a single A100 40GB GPU. Prior methods either scale directly with length or carry a heavy component: an unbounded token bank (TimeChat-Online), a bounded but heavy KV cache (HERMES), or a heavy retriever (StreamReady), so latency and memory rise or sit high, as length grows. StreamBalance instead compresses the stream into a fixed-size memory read by a lightweight retriever, so the LLM always attends over a small, constant token set, keeping both latency and memory flat and low throughout. At 16k frames, we use roughly $8\times$ lower latency and $1.2\times$ less memory than the strongest prior method, remaining the fastest and most memory-efficient at every length.

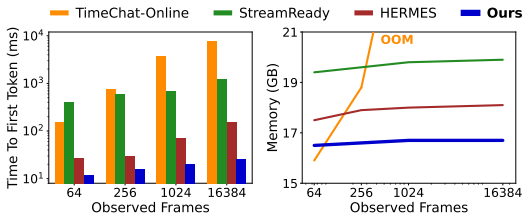


Figure 5: **Latency, compute analysis** (EgoLife-QA).

5 Conclusion

Ultra-long streaming demands a memory that covers the whole timeline evenly, yet prior memories write at fixed positions or by attention, inducing a recency or dominant-event bias that loses distant evidence under a fixed budget. We reformulate streaming memory compression as a **balanced temporal coverage** problem and build **StreamBalance**, whose fully learnable memory couples a balanced-assignment write with adaptive decay-rate-aware retention, trained end-to-end so what to store, keep, and retrieve is shaped by answer quality. It sets a new state of the art across general and ultra-long streaming, sustaining stable accuracy over hour- to day-scale streams where prior methods collapse, while remaining competitive offline at markedly lower latency and memory.

REFERENCES

Marah Abdin, Sam Ade Jacobs, Ammar Ahmad Awan, Jyoti Aneja, Ahmed Awadallah, Hany Awadalla, Nguyen Bach, Amit Bahree, Arash Bakhtiari, Harkirat Behl, et al. Phi-3 technical report: A highly capable language model locally on your phone. *arXiv preprint arXiv:2404.14219*, 2024.

Jason Altschuler, Jonathan Niles-Weed, and Philippe Rigollet. Near-linear time approximation algorithms for optimal transport via sinkhorn iteration. *Advances in neural information processing systems*, 30, 2017.

Quentin Anthony, Yury Tokpanov, Paolo Glorioso, and Beren Millidge. Blackmamba: Mixture of experts for state-space models. *arXiv preprint arXiv:2402.01771*, 2024.

Shehreen Azad, Vibhav Vineet, and Yogesh Singh Rawat. Hierarq: Task-aware hierarchical q-former for enhanced video understanding. In *CVPR*, pp. 8545–8556, 2025.

Shehreen Azad, Vibhav Vineet, and Yogesh S Rawat. Streamready: Learning what to answer and when in long streaming videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 40494–40504, 2026.

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

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Mingkun Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, Jiabo Ye, Xi Zhang, Tianbao Xie, Zesen Cheng, Hang Zhang, Zhibo Yang, Haiyang Xu, and Junyang Lin. Qwen2.5-vl technical report, 2025b. URL https://arxiv.org/abs/2502.13923.

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, et al. Qwen2. 5-vl technical report. *arXiv preprint arXiv:2502.13923*, 2025c.

Udbhav Bamba, Arnav Chavan, Aryamaan Thakur, Steven Teig, and Deepak Gupta. DOT-moe: Differentiable optimal transport for moefication. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=WrXyXq11Yc.

Yoshua Bengio, Nicholas Léonard, and Aaron Courville. Estimating or propagating gradients through stochastic neurons for conditional computation. *arXiv preprint arXiv:1308.3432*, 2013.

Daniel Bolya, Cheng-Yang Fu, Xiaoliang Dai, Peizhao Zhang, Christoph Feichtenhofer, and Judy Hoffman. Token merging: Your vit but faster. In *ICLR*, 2023.

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 *CVPR*, pp. 18407–18418, 2024a.

Xueyi Chen, Keda Tao, Kele Shao, and Huan Wang. Streamingtom: Streaming token compression for efficient video understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 24675–24685, 2026.

Zhe Chen, Weiyun Wang, Hao Tian, Shenglong Ye, Zhangwei Gao, Erfei Cui, Wenwen Tong, Kongzhi Hu, Jiapeng Luo, Zheng Ma, et al. How far are we to gpt-4v? closing the gap to commercial multimodal models with open-source suites. *Science China Information Sciences*, 67 (12):220101, 2024b.

Zhe Chen, Weiyun Wang, En Yu, Shenglong Luo, Jiannan Cao, Wenhai Wang, Shuai Bai, Xiaoyi Dong, Jianyuan Wang, Bin Li, Conghui He, Lewei Lu, Jifeng Dai, and Yu Qiao. Internvl3: Exploring advanced training and test-time recipes for open-source multimodal models, 2025.

Dingxin Cheng, Mingda Li, Jingyu Liu, Yongxin Guo, Bin Jiang, Qingbin Liu, Xi Chen, and Bo Zhao. Enhancing long video understanding via hierarchical event-based memory. *arXiv preprint arXiv:2409.06299*, 2024.

Aidan Clark, Diego de Las Casas, Aurelia Guy, Arthur Mensch, Michela Paganini, Jordan Hoffmann, Bogdan Damoc, Blake Hechtman, Trevor Cai, Sebastian Borgeaud, et al. Unified scaling laws for routed language models. In *International conference on machine learning*, pp. 4057–4086. PMLR, 2022.

Marco Cuturi. Sinkhorn distances: Lightspeed computation of optimal transport. *Advances in neural information processing systems*, 26, 2013.

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. In *ICLR*, 2025.

Chaoyou Fu, Yuhan Dai, Yongdong Luo, Lei Li, Shuhuai Ren, Renrui Zhang, Zihan Wang, Chenyu Zhou, Yunhang Shen, Mengdan Zhang, et al. Video-mme: The first-ever comprehensive evaluation benchmark of multi-modal llms in video analysis. In *CVPR*, 2025a.

Chaoyou Fu, Haojia Lin, Xiong Wang, Yi-Fan Zhang, Yunhang Shen, Xiaoyu Liu, Haoyu Cao, Zuwei Long, Heting Gao, Ke Li, et al. Vita-1.5: Towards gpt-4o level real-time vision and speech interaction. *arXiv preprint arXiv:2501.01957*, 2025b.

Shenghao Fu, Qize Yang, Yuan-Ming Li, Yi-Xing Peng, Kun-Yu Lin, Xihan Wei, Jian-Fang Hu, Xiaohua Xie, and Wei-Shi Zheng. Vispeak: Visual instruction feedback in streaming videos. In *ICCV*, 2025c.

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 *CVPR*, pp. 13504–13514, 2024.

Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, et al. Gpt-4o system card. *arXiv:2410.21276*, 2024.

Minsoo Kim, Kyuhong Shim, Jungwook Choi, and Simyung Chang. Infinipot-v: Memory-constrained kv cache compression for streaming video understanding. In *NeurIPS*, 2025.

Noriyuki Kugo, Xiang Li, Zixin Li, Ashish Gupta, Arpandeep Khatua, Nidhish Jain, Chaitanya Patel, Yuta Kyuragi, Yasunori Ishii, Masamoto Tanabiki, et al. Videomultiagents: A multi-agent framework for video question answering. *arXiv preprint arXiv:2504.20091*, 2025.

Bo Li, Yuanhan Zhang, Dong Guo, Renrui Zhang, Feng Li, Hao Zhang, Kaichen Zhang, Peiyuan Zhang, Yanwei Li, Ziwei Liu, et al. Llava-onevision: Easy visual task transfer. *arXiv preprint arXiv:2408.03326*, 2024a.

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

KunChang Li, Yinan He, Yi Wang, Yizhuo Li, Wenhai Wang, Ping Luo, Yali Wang, Limin Wang, and Yu Qiao. Videochat: Chat-centric video understanding. *arXiv preprint arXiv:2305.06355*, 2023b.

Kunchang Li, Xinhao Li, Yi Wang, Yinan He, Yali Wang, Limin Wang, and Yu Qiao. Videomamba: State space model for efficient video understanding. In *European Conference on Computer Vision*, pp. 237–255. Springer, 2025a.

Xinhao Li, Yi Wang, Jiashuo Yu, Xiangyu Zeng, Yuhan Zhu, Haian Huang, Jianfei Gao, Kunchang Li, Yinan He, Chenting Wang, et al. Videochat-flash: Hierarchical compression for long-context video modeling. *arXiv preprint arXiv:2501.00574*, 2024b.

Yifei Li, Junbo Niu, Ziyang Miao, Chunjiang Ge, Yuanhang Zhou, Qihao He, Xiaoyi Dong, Haodong Duan, Shuangrui Ding, Rui Qian, et al. Ovo-bench: How far is your video-llms from real-world online video understanding? *arXiv preprint arXiv:2501.05510*, 2025b.

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 preprint arXiv:2411.03628*, 2024.

Shuming Liu, Chen Zhao, Tianqi Xu, and Bernard Ghanem. Bolt: Boost large vision-language model without training for long-form video understanding. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 3318–3327, 2025a.

Xiangrui Liu, Yan Shu, Zheng Liu, Ao Li, Yang Tian, and Bo Zhao. Video-xl-pro: Reconstructive token compression for extremely long video understanding. *arXiv preprint arXiv:2503.18478*, 2025b.

Xuyang Liu, Yiyu Wang, Junpeng Ma, and Linfeng Zhang. Video compression commander: Plug-and-play inference acceleration for video large language models. In *EMNLP*, 2025c.

Lin Long, Yichen He, Wentao Ye, Yiyuan Pan, Yuan Lin, Hang Li, Junbo Zhao, and Wei Li. Seeing, listening, remembering, and reasoning: A multimodal agent with long-term memory. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=PMz29A7Muq.

Arthur Mensch and Gabriel Peyré. Online sinkhorn: Optimal transport distances from sample streams. *Advances in Neural Information Processing Systems*, 33:1657–1667, 2020.

Tony Montes and Fernando Lozano. Viqagent: Zero-shot video question answering via agent with open-vocabulary grounding validation. *arXiv preprint arXiv:2505.15928*, 2025.

Duc Anh Nguyen, Huu Binh Ta, Nhuan Le Duc, Tan Minh Nguyen, and Toan Tran. Selective sinkhorn routing for improved sparse mixture of experts. *arXiv preprint arXiv:2511.08972*, 2025.

Zhenyu Ning, Guangda Liu, Qihao Jin, Chengwei Li, Wenchao Ding, Minyi Guo, and Jieru Zhao. Livevlm: Efficient online video understanding via streaming-oriented kv cache and retrieval. In *63rd ACM/IEEE Design Automation Conference (DAC)*, 2026.

OpenAI. Hello gpt-4o. https://openai.com/index/hello-gpt-4o, 2024.

Junwen Pan, Qizhe Zhang, Rui Zhang, Ming Lu, Xin Wan, Yuan Zhang, Chang Liu, and Qi She. Timesearch-r: Adaptive temporal search for long-form video understanding via self-verification reinforcement learning. In *The Fourteenth International Conference on Learning Representations*, 2026.

AJ Piergiovanni, Isaac Noble, Dahun Kim, Michael S Ryoo, Victor Gomes, and Anelia Angelova. Mirasol3b: A multimodal autoregressive model for time-aligned and contextual modalities. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 26804–26814, 2024.

Rui Qian, Shuangrui Ding, Xiaoyi 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. In *CVPR*, 2025.

Michael S Ryoo, Keerthana Gopalakrishnan, Kumara Kahatapitiya, Ted Xiao, Kanishka Rao, Austin Stone, Yao Lu, Julian Ibarz, and Anurag Arnab. Token turing machines. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 19070–19081, 2023.

Yuzhang Shang, Mu Cai, Bingxin Xu, Yong Jae Lee, and Yan Yan. Llava-prumerge: Adaptive token reduction for efficient large multimodal models. In *ICCV*, 2025.

Kele Shao, Keda Tao, Can Qin, Haoxuan You, Yang Sui, and Huan Wang. Holitom: Holistic token merging for fast video large language models. In *NeurIPS*, 2025.

Leqi Shen, Guoqiang Gong, Tao He, Yifeng Zhang, Pengzhang Liu, Sicheng Zhao, and Guiguang Ding. Fastvid: Dynamic density pruning for fast video large language models. In *NeurIPS*, 2025.

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

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

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

Enxin Song, Wenhao Chai, Tian Ye, Jenq-Neng Hwang, Xi Li, and Gaoang Wang. Moviechat+: Question-aware sparse memory for long video question answering. *arXiv preprint arXiv:2404.17176*, 2024b.

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

Yuanmin Tang, Jue Zhang, Xiaoting Qin, Jing Yu, Meikang Qiu, Gaopeng Gou, Gang Xiong, Qingwei Lin, Saravan Rajmohan, Dongmei Zhang, et al. Egomemory: Memory-augmented personalized retrieval for long-context egocentric video. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 7317–7349, 2026.

Keda Tao, Can Qin, Haoxuan You, Yang Sui, and Huan Wang. Dycoke: Dynamic compression of tokens for fast video large language models. In *CVPR*, 2025.

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

Shulin Tian, Ruiqi Wang, Hongming Guo, Penghao Wu, Yuhao Dong, Xiuying Wang, Jingkang Yang, Hao Zhang, Hongyuan Zhu, and Ziwei Liu. Ego-r1: Agentic chain-of-tool-thought for ultra-long egocentric video reasoning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2026a.

Xin Tian, Jiuliu Lu, Ephraim Tsalik, Bart Wanders, Colleen Knoth, and Julian Knight. Region-graph optimal transport routing for mixture-of-experts whole-slide image classification. *arXiv preprint arXiv:2604.07298*, 2026b.

Cédric Villani et al. *Optimal transport: old and new*, volume 338. Springer, 2009.

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. In *NeurIPS*, 2025a.

Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, et al. Qwen2-vl: Enhancing vision-language model's perception of the world at any resolution. *arXiv preprint arXiv:2409.12191*, 2024a.

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

Xijun Wang, Junbang Liang, Chun-Kai Wang, Kenan Deng, Yu Lou, Ming C Lin, and Shan Yang. Vila: Efficient video-language alignment for video question answering. In *European Conference on Computer Vision*, pp. 186–204. Springer, 2024c.

Yi Wang, Xinhao Li, Ziang Yan, Yinan He, Jiashuo Yu, Xiangyu Zeng, Chenting Wang, Changlian Ma, Haian Huang, Jianfei Gao, et al. Internvideo2. 5: Empowering video mllms with long and rich context modeling. *arXiv preprint arXiv:2501.12386*, 2025b.

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. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 18523–18533, 2026a.

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. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2026b.

Yuxuan Wang, Cihang Xie, Yang Liu, and Zilong Zheng. Videollamb: Long-context video understanding with recurrent memory bridges. In *ICCV*, 2025c.

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

Chao-Yuan Wu, Yanghao Li, Karttikeya Mangalam, Haoqi Fan, Bo Xiong, Jitendra Malik, and Christoph Feichtenhofer. Memvit: Memory-augmented multiscale vision transformer for efficient long-term video recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 13587–13597, 2022.

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

Jiaer Xia, Peixian Chen, Mengdan Zhang, Xing Sun, and Kaiyang Zhou. Streaming video instruction tuning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 31219–31229, 2026.

Junbin Xiao, Jiajun Chen, Tianxiang Sun, Xun Yang, and Angela Yao. Mukv: Multi-grained kv cache compression for long streaming video question-answering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2026.

Yiweng Xie, Bo He, Junke Wang, Xiangyu Zheng, Ziyi Ye, and Zuxuan Wu. Fluxmem: Adaptive hierarchical memory for streaming video understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 31272–31282, 2026.

Yujia Xie, Xiangfeng Wang, Ruijia Wang, and Hongyuan Zha. A fast proximal point method for computing exact wasserstein distance. In *Uncertainty in artificial intelligence*, pp. 433–453. PMLR, 2020.

Long Xing, Qidong Huang, Xiaoyi Dong, Jiajie Lu, Pan Zhang, Yuhang Zang, Yuhang Cao, Conghui He, Jiaqi Wang, Feng Wu, and Dahua Lin. Pyramiddrop: Accelerating your large vision-language models via pyramid visual redundancy reduction. In *CVPR*, 2025.

Haomiao Xiong, Zongxin Yang, Jiazuo Yu, Yunzhi Zhuge, Lu Zhang, Jiawen Zhu, and Huchuan Lu. Streaming video understanding and multi-round interaction with memory-enhanced knowledge. *arXiv preprint arXiv:2501.13468*, 2025.

Hongwei Xue, Tiankai Hang, Yanhong Zeng, Yuchong Sun, Bei Liu, Huan Yang, Jianlong Fu, and Baining Guo. Advancing high-resolution video-language representation with large-scale video transcriptions. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5026–5035. IEEE, 2022.

Jingkang Yang, Shuai Liu, Hongming Guo, Yuhao Dong, Xiamengwei Zhang, Sicheng Zhang, Pengyun Wang, Zitang Zhou, Binzhu Xie, Ziyue Wang, et al. Egolife: Towards egocentric life assistant. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 28885–28900, 2025a.

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 preprint arXiv:2508.15717*, 2025b.

Linli Yao, Yicheng Li, Yuancheng Wei, Lei Li, Shuhuai Ren, Yuanxin Liu, Kun Ouyang, Lean Wang, Shicheng Li, Sida Li, et al. Timechat-online: 80% visual tokens are naturally redundant in streaming videos. *arXiv preprint arXiv:2504.17343*, 2025a.

Linli Yao, Yicheng Li, Yuancheng Wei, Lei Li, Shuhuai Ren, Yuanxin Liu, Kun Ouyang, Lean Wang, Shicheng Li, Sida Li, et al. Timechat-online: 80% visual tokens are naturally redundant in streaming videos. In *Proceedings of the 33rd ACM international conference on Multimedia*, 2025b.

Woongyeong Yeo, Kangsan Kim, Jaehong Yoon, and Sung Ju Hwang. Worldmm: Dynamic multimodal memory agent for long video reasoning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 25599–25609, 2026.

Penghang Yin, Jiancheng Lyu, Shuai Zhang, Stanley Osher, Yingyong Qi, and Jack Xin. Understanding straight-through estimator in training activation quantized neural nets. *arXiv preprint arXiv:1903.05662*, 2019.

Abhay Zala, Jaemin Cho, Satwik Kottur, Xilun Chen, Barlas Oguz, Yashar Mehdad, and Mohit Bansal. Hierarchical video-moment retrieval and step-captioning. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 23056–23065. IEEE, 2023.

Xiangyu Zeng, Kefan Qiu, Qingyu Zhang, Xinhao Li, Jing Wang, Jiaxin Li, Ziang Yan, Kun Tian, Meng Tian, Xinhai Zhao, Yi Wang, and Limin Wang. Streamforest: Efficient online video understanding with persistent event memory, 2025.

Hang Zhang, Xin Li, and Lidong Bing. Video-llama: An instruction-tuned audio-visual language model for video understanding. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023 - System Demonstrations*, pp. 543–553, Singapore, December 6-10 2023. Association for Computational Linguistics.

Haoji Zhang, Yiqin Wang, Yansong Tang, Yong Liu, Jiashi Feng, Jifeng Dai, and Xiaojie Jin. Flash-vstream: Efficient real-time understanding for long video streams. In *ICCV*, 2025.

Haowei Zhang, Shudong Yang, Jinlan Fu, See Kiong Ng, and Xipeng Qiu. Hermes: Kv cache as hierarchical memory for efficient streaming video understanding. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 8411–8430, 2026a.

Kairui Zhang, Zhenyu Yang, Bing Wang, Shengsheng Qian, and Changsheng Xu. Querystream: Advancing streaming video understanding with query-aware pruning and proactive response. In *International Conference on Learning Representations*, 2026b.

Kecheng Zhang, Zongxin Yang, Mingfei Han, Haihong Hao, Yunzhi Zhuge, Changlin Li, Junhan Zhao, Zhihui Li, and Xiaojun Chang. Progressive online video understanding with evidence-aligned timing and transparent decisions. In *International Conference on Learning Representations*, 2026c.

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. *arXiv preprint arXiv:2406.16852*, 2024a.

Yuanhan Zhang, Bo Li, haotian Liu, Yong jae Lee, Liangke Gui, Di Fu, Jiashi Feng, Ziwei Liu, and Chunyuan Li. Llava-next: A strong zero-shot video understanding model, April 2024b. URL `https://llava-vl.github.io/blog/2024-04-30-llava-next-video/`.

Yuanhan Zhang, Jinming Wu, Wei Li, Bo Li, Zejun Ma, Ziwei Liu, and Chunyuan Li. Video instruction tuning with synthetic data. *arXiv preprint arXiv:2410.02713*, 2024c.

Zhuo Zhi, Qiangqiang Wu, Wenbo Li, Yinchuan Li, Kun Shao, Kaiwen Zhou, et al. Videoagent2: Enhancing the llm-based agent system for long-form video understanding by uncertainty-aware cot. *arXiv preprint arXiv:2504.04471*, 2025.

Junjie Zhou, Yan Shu, Bo Zhao, Boya Wu, Shitao Xiao, Xi Yang, Yongping Xiong, Bo Zhang, Tiejun Huang, and Zheng Liu. Mlvu: A comprehensive benchmark for multi-task long video understanding. *arXiv preprint arXiv:2406.04264*, 2024.

Yanqi Zhou, Tao Lei, Hanxiao Liu, Nan Du, Yanping Huang, Vincent Zhao, Andrew M Dai, Quoc V Le, James Laudon, et al. Mixture-of-experts with expert choice routing. *Advances in Neural Information Processing Systems*, 35:7103–7114, 2022.

Zirui Zhu, Hailun Xu, Yang Luo, Yong Liu, Kanchan Sarkar, Zhenheng Yang, and Yang You. FOCUS: Efficient keyframe selection for long video understanding. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=1OQKqLFcbB.

APPENDIX

The appendix is organized as follows:

- §A presents disclosures and statements (generative AI use, ethics, and reproducibility).
- §B presents additional analysis of the design choices underlying StreamBalance's memory compression.
- §C reports selected hyperparameters and their sensitivity analysis.
- §D discusses related optimal transport formulations in the literature.
- §E derives the gradient of the merge decision under the straight-through estimator.
- §F provides additional implementation details.
- §G presents full results on OVO-Bench and StreamingBench.
- §H presents the full training and inference algorithms for StreamBalance, including the per-frame balanced memory update shared by both.
- Finally, §I discusses limitations and future work.

A DISCLOSURES AND STATEMENTS

A.1 DISCLOSURE OF GENERATIVE AI USE

In this work, we used generative AI tools exclusively as writing assistants to improve grammar, refine sentence structure, and enhance the readability of the manuscript. We *did not use* generative AI tools for the following cases: "to assist in the writing of proofs, propose or refine hypotheses, design or provide feedback on research methodology or experiments, implement methods, support qualitative and thematic data analysis, and interpret results." Additionally, the following usage of generative AI tools are *not applicable* to this work: "generate synthetic data sets, help develop theoretical models or conceptual frameworks, formulate mathematical claims, provide critical ingredients for proving mathematical claims, assist with translation, clean and reformat dataset." We reviewed and verified all AI-assisted linguistic edits to ensure that they accurately reflected the intended meaning and scientific content. All scientific contributions, including the conception of ideas, methodology, experimental design, implementation, analysis, and conclusions, were developed solely by the authors. We take full responsibility for the final content of this work.

A.2 STATEMENT OF ETHICS

We have read and adhere to the ICLR Code of Ethics. Our research uses only publicly available video understanding benchmarks in accordance with their respective licenses. Our framework builds upon existing pre-trained models (Qwen Family), whose inherent limitations may reflect their training data, but our method does not introduce additional ethical risks. StreamBalance is intended to improve the performance of streaming video understanding systems while reducing computational cost through more efficient token processing. We declare no competing interests.

A.3 REPRODUCIBILITY STATEMENT

We have made every effort to ensure the reproducibility of our work. The source code for StreamBalance will be publicly released upon publication. Our framework is built on publicly available models, including Qwen2-VL-7B, Qwen2.5-VL-7B, Qwen3-VL-8B, with all relevant citations provided in the main text. We evaluate our method on standard public streaming and evaluation benchmarks, including StreamingBench, OVO-Bench, EgoLife-QA, EgoR1-Bench, VideoMME, MLVU and LongVideoBench. All critical implementation details, hyperparameters, and experimental settings are described in §4 and §F to facilitate reproducibility.

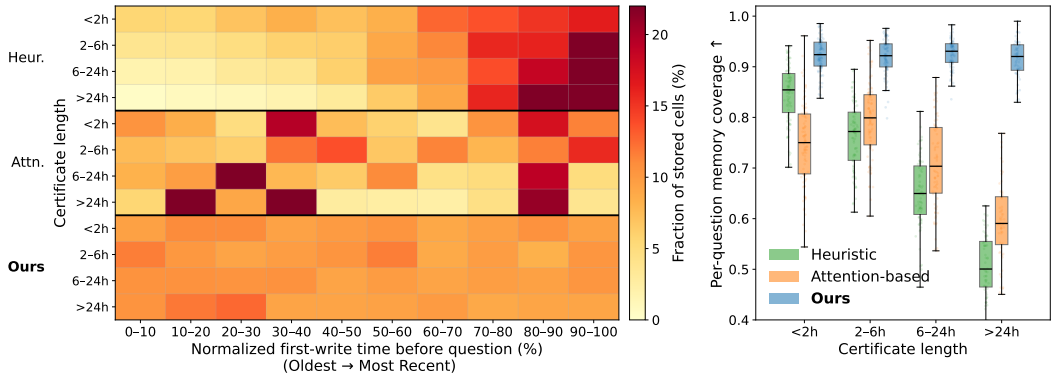


Figure 6: **Temporal distribution of memory-bank contents under different memory write strategies** on EgoLife-QA. *Left:* Distribution of stored memory cells according to their normalized first-write time in the available history, grouped by certificate length. *Right:* Distribution of per-question memory coverage across individual questions.

Table 4: **Robustness to the optimal transport solver.** Here, compression latency reports wall-clock time in ms/frame. ViT encoding latency (46.5 ms/frame) and query-decoding latency (TTFT = 16 ms/query) are solver-invariant. A single NVIDIA A100 40GB GPU was used for this analysis.

OT Solver	**OVOBench** (Overall)	**EgoLife-QA** (Overall)	**Compression Latency** ms/frame
Greenkhorn (Altschuler et al., 2017)	74.3	43.2	3.4
IPOT (Xie et al., 2020)	74.7	43.9	4.1
Sinkhorn (Ours)	**74.6**	**43.7**	**1.8**

B Additional Analysis

Impact of Balanced Coverage-based Memory Compression on Temporal Coverage. We compare the temporal coverage induced by different memory compression strategies by evaluating our balanced-coverage-based compression against two representative alternatives: heuristic compression (Azad et al., 2026) and a learnable transformer-based module (Q-Former) adapted as a learned memory write mechanism with heuristic updates (He et al., 2024). For each question, we identify the first-write time of every stored cell, normalize it relative to the history observed before that question, and aggregate within each certificate-length group; we further compute a per-question coverage score, the normalized entropy of this temporal distribution, to quantify how evenly the memory spans available history. Figure 6 (*left*) shows that as certificate length increases, heuristic updates grow increasingly recency-biased, concentrating almost exclusively on the most recent portion of the timeline, while the attention-based writer avoids this drift but instead allocates unevenly and inconsistently, concentrating on different, seemingly arbitrary regions rather than degrading in any fixed direction. Ours instead maintains a fairly balanced distribution regardless of certificate length. Per-question coverage scores (Figure 6 (*right*)) confirm this: ours consistently achieves the highest coverage with low variance across all certificate lengths, while both baselines degrade further, and grow more variable, as certificate length increases. These results show that balanced-coverage compression preserves information across diverse temporal regions, preventing collapse toward recent or narrowly dominant content as history grows.

Robustness to Different Optimal Transport Solvers. To verify that our gains stem from the balanced-coverage formulation itself rather than the specific numerical routine used to solve it, we replace our Sinkhorn solver with two alternatives: Greenkhorn (Altschuler et al., 2017), a greedy coordinate-wise variant with the same fixed point as Sinkhorn, and IPOT (Xie et al., 2020), which solves a sequence of proximal-point subproblems instead of a single entropic relaxation. Table 4 shows accuracy is essentially unchanged across all three solvers on both benchmarks, confirming that our balanced-assignment formulation, not the particular solver used to compute it, drives the

Table 5: **Impact of different decay rates for adaptive retention across certificate lengths** (EgoLife-QA).

Decay rates	**<2h**	**2h–6h**	**6h–24h**	**>24h**	**Overall**
Fixed $\alpha = 0.2$	44.8	42.6	38.9	34.7	40.3
Fixed $\alpha = 0.5$	47.1	44.3	38.0	30.8	40.1
Fixed $\alpha = 0.8$	48.0	43.5	34.6	25.2	37.8
Learned global α	47.4	44.1	38.8	31.9	40.8
Learned per-slot α_j (Ours)	**49.2**	**44.4**	**41.8**	**39.3**	**43.7**

Table 6: **Impact of different choice of per-cell temporal metadata.**

Per-cell Timestamp	**OVOBench**				**EgoLife-QA**		
	Real	Back	Fwd	Overall	<6h	≥6h	Overall
None	80.9	74.8	56.1	70.6	39.6	31.4	35.4
t_{first} only	81.5	77.1	57.2	71.9	42.4	36.7	39.4
t_{last} only	82.8	76.1	57.6	72.2	42.5	37.3	39.9
$[\mathbf{t_{first}}, \mathbf{t_{last}}]$ only	**83.4**	**79.5**	**60.8**	**74.6**	**45.2**	**42.1**	**43.7**

improvement over heuristic and attention-based memories. The solvers differ clearly in write latency, however, and this matters independently of query-time latency: write latency is incurred continuously as the stream is ingested, every frame, regardless of whether a query has arrived, whereas TTFT (Figure 5) is paid only once a query does, and reflects reading from an already-built memory rather than constructing it. In this per-frame pipeline, ViT encoding (Vision tower of Qwen-2.5-VL 7B) dominates cost at ~ 46.5ms/frame, and query-time decoding adds ~ 16ms/query; write latency is comparatively negligible, at most a few milliseconds regardless of solver, so solver choice barely moves the total per-frame or per-query cost. Still, a slower solver adds unnecessary overhead as the stream lengthens, and Sinkhorn's simultaneous row-and-column normalization is markedly cheaper here than Greenkhorn's sequential updates and IPOT's nested proximal-point loop, both of which require additional iterations per write with no accompanying accuracy gain. We therefore adopt Sinkhorn as our default solver for its favorable accuracy-latency tradeoff.

Effect of Different Decay Rates on Adaptive Retention Across Question Ages. Table 5 shows the effect of different retention mechanisms across certificate-length groups of EgoLife-QA. The decay rate α governs not only *whether* incoming content merges into an existing cell but also *how strongly*: a high α merges readily and absorbs new content aggressively, while a low α does both conservatively. A large fixed rate, $\alpha = 0.8$, discards redundant, fast-changing detail well, but applies the same aggressive overwrite to rare, salient events, so distant evidence is absorbed away before a later question can retrieve it, and accuracy collapses on long-range certificates. Conversely, a small rate, $\alpha = 0.2$, preserves distant evidence well, but its same reluctance lets stale, repetitive detail linger and consume capacity that recent content needs, underperforming on near-term questions. Both make the same error in opposite directions: a single global merge strength cannot simultaneously discard redundancy and preserve salience, since neither is determined by age alone. Learning a global α compromises between the two but still forces all slots to share one operating point. Our per-slot decay rate resolves this by letting each slot specialize: fast slots clear redundancy, slow slots preserve salient content close to its original form. This advantage grows with certificate length, since distant evidence increasingly depends on having been assigned to a slow slot in the first place, avoiding the short-term versus long-term trade-off any single fixed rate is forced into.

Effect of Per-cell Temporal Metadata. Table 6 evaluates different choices of temporal metadata stored with each memory cell. Incorporating temporal metadata consistently improves performance over storing no timestamps, with the full interval $[t_{first}, t_{last}]$ achieving the best results across both OVOBench and EgoLife-QA. Individually, t_{first} and t_{last} provide comparable gains, indicating that each captures complementary temporal information. While t_{first} records when a memory was first created and t_{last} records its most recent update, repeated memory consolidation causes the cell representation to evolve over time, making either timestamp alone an incomplete description of the

Table 7: **Impact of capacity of retriever vs. quality of memory.**

Memory	Retriever	Ret. Params (M) (↓)	OVOBench (Overall)	EgoLife-QA (Overall)
Heuristic (Azad et al., 2026)	Cosine Similarity	0	61.2	24.8
	Hierarchical Q-Former (Azad et al., 2026)	382	68.2	31.5
	Vanilla Q-Former (Li et al., 2023a)	188	67.5	30.1
	Single-layer Transformer Decoder (Ours)	27	66.7	29.3
Balanced Coverage (Ours)	Cosine Similarity	0	73.9	41.1
	Hierarchical Q-Former (Azad et al., 2026)	382	**74.9**	<u>43.2</u>
	Vanilla Q-Former (Li et al., 2023a)	188	74.2	43.0
	Single-layer Transformer Decoder (Ours)	**27**	<u>74.6</u>	**43.7**

current memory. Storing both timestamps enables the model to jointly reason about a memory's temporal origin and its subsequent evolution, resulting in the largest improvements. The benefit becomes particularly pronounced on longer EgoLife-QA certificates, where memory cells undergo substantially more consolidation and temporal reasoning spans much larger time horizons.

Impact of Retriever Capacity. Table 7 varies retriever capacity from a parameter-free cosine lookup to a 382M hierarchical Q-Former (Azad et al., 2026) under each memory. The decisive comparison is between blocks: our balanced memory with a *parameter-free* cosine retriever already outperforms the heuristic memory equipped with the 382M hierarchical Q-Former; i.e. balanced coverage with *no learned retrieval at all* beats a recency-biased memory with the heaviest retriever we test. On the balanced memory, scaling the retriever from 0 to 382M parameters moves performance only marginally, since evenly spread evidence makes retrieval a near-trivial lookup that even cosine similarity solves. The same scaling buys far more on the heuristic memory, $+7.0$ on OVOBench and $+6.7$ on EgoLife-QA, yet stays capped well below the balanced block, because *no retriever can recover what was never stored*: a recency-biased memory has already discarded the distant evidence. Retriever capacity thus matters most when the memory is weak; once it is balanced, a lightweight retriever suffices. This dissociation attributes StreamBalance's gains to balanced-coverage writing and retention rather than to a learned retriever, and justifies our lightweight decoder: at 27M parameters, 7–15$\times$ smaller than the Q-Former alternatives, it matches the best on OVOBench and exceeds it on EgoLife-QA.

C Hyperparameter Sweep

Selected Hyperparameters. Slot anchors S are randomly initialized and ℓ_2-normalized to match the cosine cost. Per-slot decay rate $\alpha_j = \sigma(z_j)$ is seeded from $z_j \sim \mathcal{U}(-2, 2)$ for a spread of initial retention rates. The pressure/staleness coefficients λ_p, λ_s are learnable scalars initialized to 0.1. The write threshold uses $c = 1.5$, while the adaptive merge threshold in Eq. 5 is defined by $\tau_j = \tau_0 + \eta\alpha_j$ with $\tau_0 = 0.5$ and $\eta = 0.1$. The following analyses evaluate the sensitivity of the model to the memory-bank capacity, decay-rate initialization, and memory-compression hyperparameters.

Sensitivity of Decay Rate Initialization on Temporal Specialization. Each memory slot's decay factor is parameterized as $\alpha_j = \sigma(z_j)$, with logits initialized as $z_j \sim \mathcal{U}(-2, 2)$, seeding a diverse spectrum of retention timescales at the start of training so different slots can specialize for short-, medium-, and long-term information, while remaining in a region where the sigmoid provides stable gradients. To evaluate the sensitivity of this choice, we compare three initialization ranges. Figure 7 shows that a narrower initialization, $z_j \sim \mathcal{U}(-1, 1)$, restricts the initial diversity of decay rates, resulting in weaker temporal specialization ($\rho = -0.77$) and lower downstream performance (42.9%), since slots start too similar to diverge into distinct roles. In contrast, a much wider initialization, $z_j \sim \mathcal{U}(-4, 4)$, instead pushes many slots into the sigmoid's saturated regions from the start, polarizing decay rates toward purely fast or purely slow and leaving little room to learn intermediate timescales; specialization remains strong ($\rho = -0.83$), but accuracy drops to 43.1%, as saturated gradients make these polarized assignments harder to refine. Our default initialization, $z_j \sim \mathrm{Uniform}(-2, 2)$, achieves the strongest specialization ($\rho = -0.89$) and the highest accuracy (43.7%), striking the balance between enough initial diversity to specialize and enough gradient signal to keep refining that specialization throughout training.

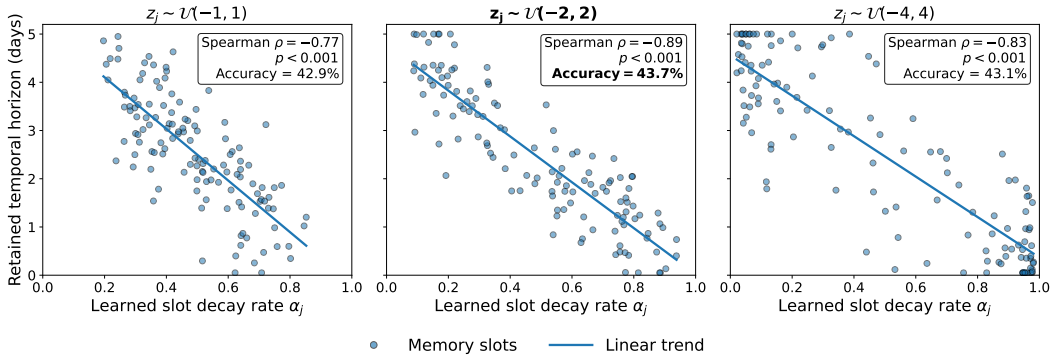


Figure 7: **Sensitivity analysis of decay rate initialization** on EgoLife-QA. Here, each point is one memory slot, plotting its learned decay rate α_j against the retained temporal horizon of its stored content. Spearman's correlation ρ measures the strength and direction of this monotonic relationship (more negative ρ indicates a stronger tendency for higher-decay slots to retain shorter horizons); p reports the statistical significance of this correlation, with $p < 0.001$ indicating the relationship is highly unlikely to have arisen by chance.

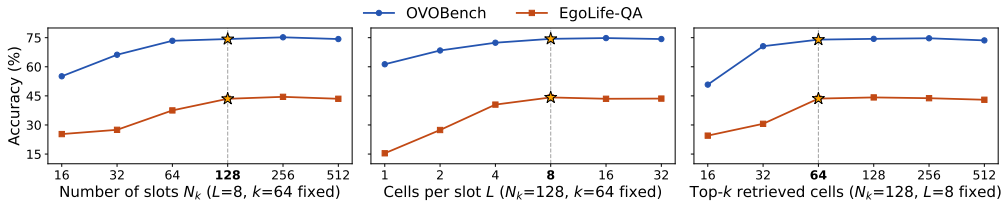


Figure 8: **Sensitivity analysis of memory bank size and retrieved cells' count.** The chosen configuration is highlighted.

Sensitivity of Memory Bank Size and Retrieved Cells Count. Figure 8 analyzes the effect of the memory bank configuration by varying the number of slots, the number of cells per slot, and the number of retrieved cells used for reasoning. Across both OVOBench and EgoLife-QA, performance consistently improves as memory capacity increases before reaching a plateau, indicating diminishing returns beyond a moderate memory size. Specifically, the best trade-off is achieved with 128 memory slots, 8 cells per slot, and top-$k = 64$ retrieved cells. We observe that OVOBench saturates slightly earlier than EgoLife-QA, whereas EgoLife-QA continues to benefit from larger memory configurations beyond OVOBench's saturation point. This is expected because EgoLife-QA consists of substantially longer videos with more diverse events, requiring greater memory capacity to preserve relevant context over extended temporal horizons. Nevertheless, the same memory configuration performs well on both datasets, demonstrating that our balanced-coverage based memory bank generalizes effectively across videos of varying durations, from relatively shorter sequences to ultra-long streaming videos.

Sensitivity of Write and Retention Thresholds. Figure 9 evaluates the sensitivity of the proposed memory compression mechanism to the write-threshold modulator c, base merge threshold τ_0, and decay-aware threshold slope η. Here, c controls the minimum transport mass required before writing to a slot, τ_0 determines the base similarity required for merging memory cells, and η adjusts this merge threshold according to each slot's learned decay rate. A single configuration ($c = 1.5, \tau_0 = 0.5, \eta = 0.1$), selected once on held-out data, disjoint from all evaluation benchmarks, is applied unchanged to every benchmark and performs well throughout, without any per-dataset tuning. Deviating from it degrades accuracy gracefully rather than collapsing, though the effect is more pronounced on EgoLife-QA than OVOBench, consistent with longer streams compounding compression errors over many more write steps; even so, with degradation staying within 1–2 points on EgoLife-QA and within 0.5 points on OVOBench across the full sweep, indicating the method is not fragile to this choice, only that the selected values are meaningfully better than poor ones.

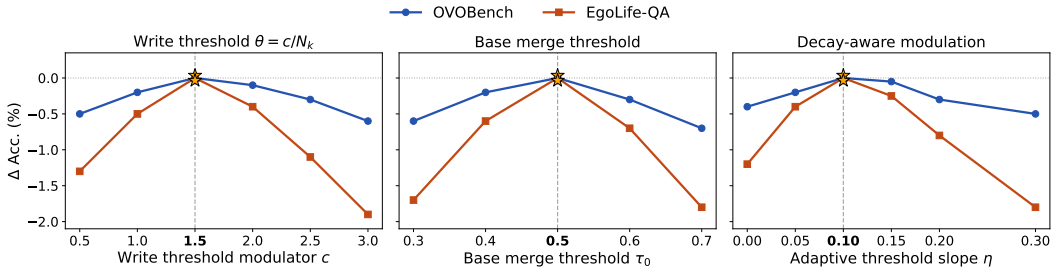


Figure 9: **Sensitivity analysis of write and retention thresholds.** Here, the chosen configuration is highlighted, and the y-axis reports the change in accuracy from the chosen values for each hyperparameter.

D Optimal Transport for Balanced Assignment: Stationary Vs. State-Aware

We use entropy-regularized optimal transport (OT) for our balanced write (§3.1), so a natural question is how this relates to existing OT formulations. Here we clarify that our use is *not* a streaming or online variant of classical OT, but a distinct *state-aware* formulation demanded by our setting. Classical and online OT alike solve a *stationary* problem: fixed cost and fixed marginals; whereas ours instantiates a new problem each frame, conditioned on the memory's current contents.

Entropy-regularized optimal transport has been used in existing literature for routing and assignment under balance constraints, but always to solve a *stationary* problem. Online Sinkhorn (Mensch & Peyré, 2020) estimates optimal transport between two fixed, continuous distributions from a stream of i.i.d. samples, converging toward one ground-truth coupling as more samples are observed, with a representation that grows freely with the sample stream rather than staying bounded. Sinkhorn-based routing in Mixture-of-Experts architectures (Clark et al., 2022; Bamba et al., 2026; Nguyen et al., 2025; Tian et al., 2026b; Anthony et al., 2024) instead solves a single transport problem per batch, using a fixed cost and a fixed, typically uniform capacity target, with no dependence on prior assignments. Neither setting combines a bounded memory with an unbounded stream, so in neither case must a fixed-capacity store be continually reassigned in light of what it already holds: this is precisely the combination our setting introduces, and what forces our transport problem to be *state-aware* rather than fixed.

Concretely, neither side of our transport problem is fixed: the source is a single frame's tokens, and the target is a per-slot capacity predicted anew at each timestep from the memory's current occupancy, while the cost is modulated by the memory's pressure and staleness rather than a fixed ground metric. Each frame's Sinkhorn iteration still converges to that frame's balanced coupling, exactly as in any entropic OT solve; what differs is that this is a genuinely new coupling every frame, rather than successive estimates of one shared ground truth or an independent solve of the same fixed problem. A memory that stays fixed a N_k slots by design, rather than growing with samples or resetting each batch, is what makes every frame's problem distinct in the first place. Our transport plan is also not an end in itself but a differentiable intermediate step, produced fresh from the memory state and consumed immediately to update it, with its governing parameters trained end-to-end through the downstream loss rather than toward any fixed objective. To our knowledge, we are the *first* to make entropy-regularized optimal transport suitable for streaming memory compression through our state-aware formulation, and show that it can sustain balanced temporal coverage across ultra-long streams, where prior streaming compression methods fail.

E Gradient Derivation for the Merge Decision

We expand here why Equation (5) blocks gradient without STE, and derive the resulting gradient to the shared projection (W) used to compute the similarity (sim) in Equation (5).

Without STE. The decision $d = \mathbb{1}[\text{sim} \geq \tau_j]$ is a step function, so $\partial d/\partial \text{sim} = 0$ almost everywhere (and undefined at the threshold). By the chain rule,

$$\frac{\partial \mathcal{L}}{\partial W} = \frac{\partial \mathcal{L}}{\partial d} \cdot \underbrace{\frac{\partial d}{\partial \text{sim}}}_{=0} \cdot \frac{\partial \text{sim}}{\partial W} = 0, \tag{7}$$

so W receives no learning signal from the merge decision.

With STE. Substituting the surrogate $\partial \hat{d}/\partial \text{sim} := 1$ (from $\hat{d} = d + \text{sim} - \text{sg}[\text{sim}]$) gives

$$\frac{\partial \mathcal{L}}{\partial W} = \frac{\partial \mathcal{L}}{\partial \hat{d}} \cdot 1 \cdot \frac{\partial \text{sim}}{\partial W} \neq 0. \tag{8}$$

The remaining factor is exact: with $u = \Delta_j W$, $v = c_{k^*} W$ and $\text{sim} = \frac{u^\top v}{(\|u\|\|v\|)}$,

$$\frac{\partial \text{sim}}{\partial u} = \frac{1}{\|u\|}\left(\frac{v}{\|v\|} - \text{sim}\,\frac{u}{\|u\|}\right), \quad \frac{\partial \text{sim}}{\partial v} = \frac{1}{\|v\|}\left(\frac{u}{\|u\|} - \text{sim}\,\frac{v}{\|v\|}\right), \tag{9}$$

$$\frac{\partial \text{sim}}{\partial W} = \Delta_j^\top \frac{\partial \text{sim}}{\partial u} + c_{k^*}^\top \frac{\partial \text{sim}}{\partial v}. \tag{10}$$

The STE substitution is the only approximated term in this chain; all other terms are exact.

We note that $\partial \hat{d}/\partial \text{sim} := 1$ is a surrogate rather than the true derivative of the step function d; but such approximation through STE has been empirically proven to be a stable and effective training signal for discrete decisions of this kind (Bengio et al., 2013; Yin et al., 2019); and we have also observed the same in our setting. Additioanlly, STE is only required for the discrete branch decision d; the blending strength $\lambda_j = \alpha_j(1 - \text{sim})/2$ used within the merge branch is already a smooth function of sim and requires no approximation.

F Additional Implementation Details

Method-specific hyperparameters are given in §C. We report remaining architecture and training details below.

Video preprocessing. Videos are sampled at 1 FPS, following standard streaming video understanding protocol. Frames are resized to 224×224 and features are downsampled $\times 4$ via average pooling, yielding 64 tokens per frame.

Architecture. We build our framework on Qwen-2-VL 7B, Qwen-2.5-VL 7B, and Qwen-3-VL 8B as the frozen backbone model, with each having its corresponding vision tower and LLM as the frozen vision encoder and language decoder. The balanced-assignment write module contains a lightweight 2-layer MLP that predicts the per-slot capacity distribution b from the current memory state, measured as the mean-pooled representation of all currently stored cells. State-aware cost modulation (pressure, staleness) requires no additional network parameter beyond the two learnable scalar coefficients (λ_p, λ_s). The adaptive retention mechanism consists of a per-slot decay parameter $(\alpha_j = \sigma(z_j))$ implemented as a single learnable scalar logit per slot (N_k total). The retriever is a single transformer-decoder layer with 8 attention heads, operating over the $N_k \times L$ memory cells with the query embedding q_t as the decoding target.

Training data. We use the training data of TimeChat-Online (Yao et al., 2025b), a large corpus of varying-length videos with QA pairs, following standard streaming video understanding practice. For compliance with internal data-use policies, we exclude videos sourced from the HiREST (Zala et al., 2023) and HD-VILA (Xue et al., 2022) datasets, which together constitute $< 1\%$ of the corpus; their removal has no material impact on performance.

Training schedule. Trainable parameters are optimized for 5 epochs using AdamW ($\beta = [0.9, 0.999]$) with a learning rate of 2e-5 and cosine annealing scheduling. Each Sinkhorn solve runs for 15 iterations with entropic regularization $\epsilon = 0.1$, sufficient for stable convergence on the 64×128 token-slot transport problem while keeping per-frame write latency low. All experiments are conducted on 8 NVIDIA A100 40GB GPUs.

Adapting EgoLife-QA and EgoR1-Bench to Streaming. While EgoLife-QA and EgoR1-Bench were both originally proposed to evaluate ultra-long context retrieval in the offline setting, with exclusively agentic models as their proposed solution, which have unconstrained access to the entire video and can query it with multiple tools and passes rather than processing it causally in a single pass. We instead adapt both benchmarks to the streaming setting, restricting each model to content observed causally, one frame at a time, up to each query's timestamp, and evaluate existing streaming models alongside ours under this protocol to establish the *first* streaming baselines on these benchmarks and enable direct comparison with our streaming model. For the EgoLife-QA, we evaluate on the published 500-question benchmark for the participant Jake, whereas for EgoR1-Bench, we evaluate on the full benchmark.

Supporting Forward Reasoning (Optional). The retriever can optionally support forward reasoning, a task present only in OVOBench (Forward Active Reasoning subset) where a query may arrive before its supporting evidence has streamed in. Genuine forward reasoning requires knowing when evidence has arrived, so the model can wait and answer once it does; without this, a correct answer risks being a lucky guess or language prior rather than evidence-guided reasoning, and forward performance no longer reflects the balanced-coverage memory mechanism we want to study. To prevent this confound, the retriever includes one additional optional component, used only for this task and disabled otherwise. This mechanism is not a contribution of this work; it exists solely to keep forward performance a fair reflection of our memory mechanism rather than answering from an incomplete stream. We therefore follow established, off-the-shelf proactive/forward reasoning protocols (Azad et al., 2026; Wang et al., 2025a), we equip the retriever with a single learnable evidence token, monitored by a lightweight regression head that outputs a per-frame evidence score $R_t \in [0, 1]$; the LLM only answers once R_t crosses a threshold. As the evidence proxy, we use the retriever's *attention shift*: the frame-to-frame change in the query's attention over slots, gated by its magnitude, so that a large, confident shift marks the arrival of new supporting evidence. The regression head is trained via the contrastive timing objective of (Azad et al., 2026), with its features detached from the retriever.

Analysis metrics. For a distribution p over M bins, we measure its evenness by the normalized effective support $E(p) = \exp\big(H(p)\big)/M$, with $H(p) = -\sum p \log p$; $E = 1$ for a uniform distribution and $\to 0$ when mass concentrates in one bin. For Figure 3, *utilization* is $Util = E(u)$, applied to the stored-mass distribution over the $C = N_k \times L$ cells (u_{jk} = fraction of mass in cell k of slot j), and *redundancy* is the mean cell-wise cosine $Red = \frac{2}{C(C-1)} \sum_{i<j} \cos(v_i, v_j)$. For the temporal coverage analysis (Figure 6), we take each cell's first-write time t_{first}, normalize it as $\hat{t} = t_{\text{first}}/t_q \in [0, 1]$ relative to the question timestamp t_q, and bin $\hat{t}$ into $B = 10$ intervals to form a distribution h (*left*); per-question *coverage* is $Cov = E(h)$ (*right*). Thus $Util$ and Cov share one construction but differ in what they bin: $Util$ by *cell identity* (is capacity used evenly now?), Cov by *write time* (does stored content span the timeline evenly?). A memory can be fully utilized yet temporally skewed, as heuristic memories are in Figure 6.

G Full Results on OVOBench and StreamingBench

Table 8 and Table 9 show detailed result for OVOBench and StreamingBench respectively.

H Training and Inference Algorithms

Algorithms 1 to 3 present, respectively, the per-frame balanced memory update followed by the full training and inference procedures for StreamBalance, with the memory update shared by both.

I Limitations and Future Work

StreamBalance makes several deliberate design choices that are effective in current setting but also open natural directions for future work. First, it maintains balanced coverage within a memory of *fixed* capacity $N_k \times L$ set in advance; while our results show stable accuracy up to multiple days-scale streams, letting capacity grow or reallocate adaptively with stream complexity is an open problem for arbitrarily long or unbounded streams. Second, our state-aware cost conditions on two

occupancy signals, pressure and staleness; richer state: longer-range occupancy trends or explicit content-diversity signals is a promising direction for more robust routing as streams grow longer and more heterogeneous. Third, per-slot decay rates are learned at training time and fixed at deployment; extending them to continual, within-deployment adaptation would let retention timescales track shifts in a stream's content distribution over time. Finally, since the balanced-assignment formulation is defined generally over token streams and fixed-capacity memory, extending balanced-coverage memory to other unbounded-stream modalities, such as audio, long-context text, where non-recency-biased retention is equally needed, is a compelling avenue.

Table 8: **Full results on OVO-Bench.** For compliance with internal data-use policies, HiREST (Zala et al., 2023) sourced videos are removed from the ASI category; † marks scores affected by this removal, which is almost negligible across all backbones.

Method	Real-Time Visual Perception							Backward Tracing				Forward Active Responding				Overall
	OCR	ACR	ATR	STU	FPD	OJR	AVG.	EPM	ASI	HLD	AVG.	REC	SSR	CRR	AVG.	Overall AVG.
Human																
Human	94.0	92.6	94.8	92.7	91.1	94.0	93.2	92.6	93.0	91.4	92.3	95.5	89.7	93.6	92.9	92.8
Proprietary Offline MLLMs																
Gemini 1.5 Pro (Team et al., 2024)	85.9	67.0	79.3	58.4	63.4	62.0	69.3	58.6	76.4	52.6	62.5	35.5	74.2	61.7	57.2	63.0
GPT-4o (Hurst et al., 2024)	69.8	64.2	71.6	51.1	70.3	59.8	64.5	57.9	75.7	48.7	60.8	27.6	73.2	59.4	53.4	59.5
Open-Source Offline MLLMs																
InternVL-V2-8B (Chen et al., 2024b)	67.1	60.6	63.8	46.1	68.3	56.5	60.4	48.2	57.4	24.7	43.4	26.5	59.1	54.1	46.6	50.2
LLaVA-NeXT-Video (Zhang et al., 2024b)	69.1	58.7	68.8	49.4	74.3	59.8	63.5	56.2	57.4	7.5	40.4	34.1	70.0	60.4	54.8	52.9
LLaVA-OV-7B (Li et al., 2024a)	66.4	57.8	73.3	53.4	71.3	62.0	64.0	54.2	55.4	21.5	43.7	25.6	67.1	58.8	50.5	52.7
LongVU (Shen et al., 2024)	53.7	53.2	62.9	47.8	68.3	59.8	57.6	40.7	59.5	4.8	35.0	12.2	69.5	60.8	47.5	46.7
VITA 1.5 (Fu et al., 2025b)	74.5	60.6	70.7	53.4	63.4	58.7	63.5	46.1	54.1	24.2	41.5	37.5	60.7	62.1	53.5	55.5
Open-Source Online MLLMs																
VideoLLM-Online-8B (Chen et al., 2024a)	8.1	23.9	12.1	14.0	45.5	21.2	20.8	22.2	18.8	12.2	17.7	-	-	-	-	-
Flash-VStream-7B (Zhang et al., 2025)	24.2	29.4	28.5	33.7	25.7	28.8	28.4	39.1	37.2	5.9	27.4	8.0	67.3	60.0	45.1	33.6
Dispider (Qian et al., 2025)	57.7	49.5	62.1	44.9	61.4	51.6	54.6	48.5	55.4	4.3	36.1	18.1	37.4	48.8	34.7	41.8
ReKV (Di et al., 2025)	73.8	56.0	74.1	51.7	70.3	60.3	64.4	54.2	57.4	28.5	46.7	25.4	64.6	53.3	47.8	52.6
STC (Wang et al., 2026b)	68.5	57.8	73.3	47.2	68.3	59.8	62.5	50.5	55.4	30.1	45.3	27.9	63.3	52.9	48.0	52.0
Qwen 2 VL (Wang et al., 2024a)	60.4	50.5	56.0	47.2	66.3	55.4	56.0	47.8	35.5	56.1	46.5	31.7	65.8	48.8	48.7	50.4
+ StreamForest (Zeng et al., 2025)	71.8	51.4	72.4	46.6	67.3	59.8	61.6	59.3	63.5	33.3	52.0	32.5	70.8	56.7	53.3	54.3
+ StreamBridge (Wang et al., 2025a)	84.6	71.6	74.1	49.4	75.3	72.8	71.3	67.7	57.4	79.0	68.1	19.2	64.3	61.7	48.4	62.6
+ ViSpeak (Fu et al., 2025c)	75.2	58.7	71.6	51.1	74.3	66.9	66.3	59.9	48.7	64.0	57.5	33.8	68.5	60.4	54.3	61.1
+ StreamReady (Azad et al., 2026)	86.3	74.5	73.7	52.2	80.1	74.1	73.5	70.3	64.7	81.5	72.2	34.4	70.7	71.4	58.8	68.2
+ Ours	94.9	85.7	81.8	71.8	80.5	85.4	83.4	77.4	73.4† (73.9)	87.8	79.5† (79.7)	34.8	74.1	73.5	60.8	74.6
Qwen 2.5 VL (Bai et al., 2025b)	67.8	55.1	67.2	42.1	66.3	60.9	59.9	51.5	58.8	23.7	44.7	28.4	48.1	38.7	38.5	47.7
+ TimeChat-Online (Yao et al., 2025b)	69.8	48.6	64.7	44.9	68.3	55.4	58.6	53.9	62.8	9.1	42.0	32.5	36.5	40.0	36.4	45.6
+ HERMES (Zhang et al., 2026a)	85.2	64.2	71.6	53.4	74.3	65.2	69.0	48.5	62.2	37.6	49.4	-	-	-	-	-
+ Streamo (Xia et al., 2026)	79.2	57.8	75.0	49.4	64.4	70.1	66.0	54.6	52.0	31.7	46.1	30.0	51.0	83.3	54.8	55.6
+ QueryStream (Zhang et al., 2026b)	75.2	49.5	69.8	50.0	71.3	62.5	63.1	56.9	65.5	12.4	44.9	35.5	43.3	41.7	40.2	49.4
+ Thinking-Qwen (Zhang et al., 2026c)	74.1	57.2	68.1	55.3	75.0	58.3	64.7	48.0	56.3	28.8	44.3	29.1	39.3	40.0	36.1	52.5
+ Ours	87.1	79.4	71.4	63.7	73.7	75.4	75.1	68.2	67.8† (67.9)	79.8	71.9	30.2	68.4	72.8	57.1	68.0
Qwen 3 VL 8B (Bai et al., 2025a)	92.0	78.9	79.3	68.0	73.3	81.0	78.7	48.8	61.5	48.9	51.8	21.8	64.9	50.4	43.5	58.0
+ Ours	93.6	88.4	84.2	76.4	81.5	87.1	85.1	76.6	75.6† (75.9)	85.3	79.2	35.1	70.8	72.3	59.3	74.5

Algorithm 1 Per-frame balanced memory update (shared by training and inference)

Require: frame features $f_t \in \mathbb{R}^{N\times D}$; memory bank $M \in \mathbb{R}^{N_k\times L\times D}$; frame index t
Require: parameters Θ: slot anchors $S \in \mathbb{R}^{N_k\times D}$, capacity MLP, coefficients λ_p, λ_s, decay logits $\{z_j\}$
Require: trackers $\{m_j, \xi_j\}_{j=1}^{N_k}$; hyperparameters $\theta = c/N_k$, τ_0, η

function MEMUPDATE(f_t, M, t)
  $C \leftarrow 1 - \cos(f_t, S)$ ▷ token–slot affinity cost, Eq. (1)
  $p_j \leftarrow m_j / \max_k m_k$; $s_j \leftarrow (t - \xi_j)/t$ ▷ pressure and staleness
  $\hat{C}_{ij} \leftarrow C_{ij} + \lambda_p p_j - \lambda_s s_j$ ▷ state-aware cost, Eq. (2)
  $b \leftarrow \mathrm{MLP}(M)$ ▷ per-slot target capacity from current memory
  $P \leftarrow \mathrm{Sinkhorn}(\hat{C}, b)$ ▷ entropy-regularized OT plan, Eq. (3)
  $w_j \leftarrow \sum_i P_{ij}$; $\Delta_j \leftarrow \frac{1}{w_j}\sum_i P_{ij} f_i$ ▷ write strength and content, Eq. (4)
  for each slot j with $w_j > \theta$ **do** ▷ write only sufficiently-assigned slots
    $\alpha_j \leftarrow \sigma(z_j)$; $k^* \leftarrow \arg\max_k \cos(\Delta_j, c_k)$; $\mathrm{sim} \leftarrow \cos(\Delta_j, c_{k^*})$
    $\tau_j \leftarrow \tau_0 + \eta\,\alpha_j$; $\lambda_j \leftarrow \alpha_j(1 - \mathrm{sim})/2$
    $d \leftarrow \mathbb{1}[\mathrm{sim} \geq \tau_j]$; $\hat{d} \leftarrow d + \mathrm{sim} - \mathrm{sg}[\mathrm{sim}]$ ▷ STE surrogate (training only), Eq. (6)
    if $d = 1$ **then** ▷ merge into closest cell, Eq. (5)
      $c_{k^*} \leftarrow (1 - \lambda_j)\, c_{k^*} + \lambda_j\, \Delta_j$; $t_{\mathrm{last}}(k^*) \leftarrow t$
    else ▷ allocate as a distinct event
      if slot j is full **then**
        merge the two most similar cells in slot j ▷ redundancy-based, not age-based
        union their intervals: $[\min t_{\mathrm{first}}, \max t_{\mathrm{last}}]$
      end if
      write Δ_j to the free cell with interval $[t_{\mathrm{first}}, t_{\mathrm{last}}] \leftarrow [t, t]$
    end if
    $m_j \leftarrow m_j + w_j$; $\xi_j \leftarrow t$ ▷ update occupancy trackers
  end for
  return M
end function

Table 9: **Full results on StreamingBench.**

Method	Real-Time Visual Understanding										
	OP	CR	CS	ATP	EU	TR	PR	SU	ACP	CT	AVG.
Human											
Human	89.5	92.0	93.6	91.5	95.7	92.5	88.0	88.8	89.7	91.3	91.5
Proprietary Offline MLLMs											
Gemini 1.5 Pro Team et al. (2024)	79.0	80.5	83.5	79.7	80.0	84.7	77.8	64.2	72.0	48.7	75.7
GPT-4o (Hurst et al., 2024)	77.1	80.5	83.9	76.5	70.2	83.8	66.7	62.2	69.1	49.2	73.3
Open-Source Offline MLLMs											
LLaVA-OV-7B (Li et al., 2024a)	80.4	74.2	76.0	80.7	72.7	71.7	67.6	65.5	65.7	45.1	71.1
LLaVA-NeXT-Video 32B Zhang et al. (2024b)	80.1	71.1	80.7	80.7	71.4	73.2	63.0	59.4	63.2	36.8	69.8
InternVL-V2-8B (Chen et al., 2024b)	68.1	60.9	69.4	77.1	67.7	62.9	59.3	53.3	55.0	56.5	63.7
VITA 1.5 (Fu et al., 2025b)	77.7	82.8	82.3	79.3	72.7	71.3	67.6	62.2	69.1	32.1	70.9
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., 2024a)	39.1	40.1	34.5	31.1	46.0	32.4	31.5	34.2	42.5	27.9	36.0
Dispider (Qian et al., 2025)	74.9	75.5	74.1	73.1	74.4	59.9	76.1	62.9	62.2	45.8	67.6
ReKV (Di et al., 2025)	77.5	79.5	79.2	75.6	72.3	69.7	66.0	60.3	62.2	43.6	69.1
STC (Wang et al., 2026b)	71.3	77.3	74.4	72.9	44.0	66.9	69.4	52.4	58.9	60.9	65.2
Qwen 2 VL (Wang et al., 2024a)	75.2	82.8	73.2	77.5	68.3	71.0	72.2	61.2	61.5	46.1	69.0
+ StreamForest (Zeng et al., 2025)	83.1	82.8	82.7	84.3	77.5	78.2	76.9	69.1	75.6	54.4	77.3
+ StreamBridge (Wang et al., 2025a)	80.4	78.7	83.2	79.9	74.2	69.5	77.8	63.4	70.0	43.0	72.0
+ ViSpeak (Fu et al., 2025c)	79.8	71.1	81.4	78.8	74.5	70.1	63.9	64.2	71.4	28.0	70.4
+ StreamReady (Azad et al., 2026)	85.3	87.9	82.3	82.4	74.6	78.8	78.3	75.7	74.4	60.6	78.3
+ Ours	88.6	89.1	87.1	85.8	78.9	85.1	81.8	77.1	73.4	64.1	81.2
Qwen 2.5 VL Bai et al. (2025b)	78.3	80.5	78.9	80.5	76.7	78.5	79.6	63.4	66.2	53.2	73.7
+ TimeChat-Online (Yao et al., 2025b)	80.2	82.0	79.5	83.3	76.1	78.5	78.7	64.6	69.6	58.0	75.4
+ HERMES (Zhang et al., 2026a)	83.7	81.3	88.0	87.5	76.7	86.6	82.4	76.0	73.9	46.6	79.4
+ QueryStream (Zhang et al., 2026b)	82.4	84.4	79.2	82.4	78.0	81.3	78.7	65.0	69.3	47.3	75.3
+ Thinking-Qwen (Zhang et al., 2026c)	70.3	66.7	80.0	78.0	79.3	68.7	78.3	68.2	72.3	52.4	71.6
+ Ours	90.6	87.2	88.7	89.4	78.3	89.3	80.5	74.1	77.3	65.1	82.1
Qwen 3 VL 8B Bai et al. (2025a)	89.3	81.7	74.8	75.0	88.8	72.4	68.6	40.4	85.7	71.9	77.3
+ Ours	92.6	89.5	87.5	86.2	90.1	82.4	77.1	57.6	88.4	75.3	82.7

Algorithm 2 StreamBalance: end-to-end training

Require: dataset $\mathcal{D}$ of triplets (video V, query T_P posed at t_q, answer T_O)
Require: frozen vision encoder E and frozen LLM
Require: trainable $\Theta = \{S,\ \text{MLP},\ \lambda_p, \lambda_s,\ \{z_j\}_{j=1}^{N_k},\ \text{retriever } \mathcal{R},\ \text{evidence head}\}$

for epoch $= 1, \ldots, E_{\max}$ **do**
  for $(V, T_P, t_q, T_O) \in \mathcal{D}$ **do**
    $M \leftarrow \emptyset;\quad m_j \leftarrow 0, \xi_j \leftarrow 0$ ▷ fresh memory per video
    for $t = 1, \ldots, t_q$ **do** ▷ single causal streaming pass at 1 FPS
      $f_t \leftarrow E(V_t)$
      $M \leftarrow \textsc{MemUpdate}(f_t, M, t)$ ▷ forward uses hard d; backward flows via $\hat{d}$
      if proactive (forward) task **then** ▷ optional evidence monitoring
        $R_t \leftarrow \text{EvidenceHead}(\text{sg}[\mathcal{R}\text{'s attention shift at } t])$ ▷ features detached from $\mathcal{R}$
      end if
    end for
    $q_{t_q} \leftarrow \text{TextEncode}(T_P)$
    $\{c_k\}_{k=1}^{K} \leftarrow \mathcal{R}(q_{t_q}, M)$ ▷ top-K cells of $N_k \times L$, with timestamps
    $[T_{\text{first}}, T_{\text{last}}]_{k=1}^{K} \leftarrow \text{TextEncode}([t_{\text{first}}, t_{\text{last}}]_{k=1}^{K})$ ▷ Intervals verbalized as text
    $\mathcal{L}_{\text{VL}} \leftarrow \text{CE}\big(\text{LLM}(\text{proj}(c_k)) \oplus [T_{\text{first}}, T_{\text{last}}] \oplus q_{t_q},\ T_O\big)$ ▷ VL loss shapes answer quality.
    $\Theta \leftarrow \text{AdamW}\big(\Theta,\ \nabla_\Theta \mathcal{L}_{\text{VL}}\big)$ ▷ grads flow through balanced assignment and STE
    if proactive (forward) task **then**
      $\mathcal{L}_{\text{ev}} \leftarrow \text{ContrastiveTiming}(\{R_t\})$ ▷ shapes answer timing, not memory/ retriever
      $\Theta_{\text{ev}} \leftarrow \text{AdamW}\big(\Theta_{\text{ev}},\ \nabla_{\Theta_{\text{ev}}} \mathcal{L}_{\text{ev}}\big)$
    end if
  end for
end for

Algorithm 3 StreamBalance: streaming inference

Require: unbounded frame stream $\{V_t\}_{t \geq 1}$; trained Θ; frozen E, LLM

$M \leftarrow \emptyset;\quad m_j \leftarrow 0, \xi_j \leftarrow 0$
for $t = 1, 2, \ldots$ **do** ▷ frames arrive causally at 1 FPS; no look-ahead or revisiting
  $f_t \leftarrow E(V_t)$
  $M \leftarrow \textsc{MemUpdate}(f_t, M, t)$ ▷ hard merge decision; no STE at inference
  if query T_P arrives at time t **then**
    $q_t \leftarrow \text{TextEncode}(T_P)$
    $\{c_k\}_{k=1}^{K},\ A_t \leftarrow \mathcal{R}(q_t, M)$ ▷ attend over all $N_k \times L$ cells; keep top-K
    $[T_{\text{first}}, T_{\text{last}}]_{k=1}^{K} \leftarrow \text{TextEncode}([t_{\text{first}}, t_{\text{last}}]_{k=1}^{K})$ ▷ Intervals verbalized as text
    if proactive (forward) task **then** ▷ optional evidence monitoring
      compute evidence score $R_t \in [0, 1]$ from retriever attention shift
      if $R_t < \rho$ **then continue** ▷ defer answering until evidence arrives
    end if
    $T_O \leftarrow \text{LLM}\big(\text{proj}(\{c_k\}) \oplus \{[T_{\text{first}}, T_{\text{last}}]_k\} \oplus q_t\big)$ ▷ decode answer over constant-size context
  end if
end for