CONCURTREE: A NOVEL MEMORY STRUCTURE FOR LONG-VIDEO STREAMING

Anonymous authors
Paper under double-blind review

ABSTRACT

Vision-Language Models (VLMs) have advanced rapidly in visual perception and reasoning, yet long-video understanding in streaming applications remains challenging due to strict memory constraints and incomplete reasoning over extended temporal horizons. Existing streaming VLMs face a key trade-off: structured query-agnostic systems track objects or locations with unbounded token growth, whereas fixed-budget architectures sequentially merge adjacent events, losing interleaved activity threads when tasks are interrupted and resumed. To address these limitations, we introduce **ConcurTree**, a novel query-agnostic memory framework that maintains distinct trees under a fixed token budget. ConcurTree's memory structure organizes events into activity threads causally with an exception-based routing mechanism, while handling memory constraints via similarity-weighted token merging that compresses historical evidence in-place rather than evicting or flattening it. ConcurTree achieves state-of-the-art long-horizon performance among peer streaming systems across three held-out benchmarks under controlled evaluation. On our primary long-horizon endpoint (n=1,054), pooled across three benchmarks, ConcurTree reaches 50.4% accuracy, compared with 43.1% for the strongest matched fixed-budget streaming baseline. On EgoLifeQA and Ego-R1, ConcurTree reaches 48.6% and 39.5% when evidence is more than 24 hours old, respectively, and achieves 52.3% on TeleEgo's long-horizon categories. These results highlight the importance of preserving activity structure for long-horizon reasoning in bounded streaming video.

1 INTRODUCTION

Vision-Language Models (VLMs) have made rapid progress in video understanding (Zeng et al., 2025; Yang et al., 2026; Li et al., 2026b), but extending them to continuous, long-duration video remains challenging because no model can indefinitely retain an ever-growing sequence of frames as it processes a continuous stream of video (Xiao et al., 2024). This challenge manifests itself most clearly in streaming-capable (online) VLMs. Once a stream outgrows a fixed token budget, the binding question is how the model organizes its memory under constraint. We define streaming by its strictest definition: causal frame processing, bounded memory, and query-agnostic. Causal frame processing (no frame lookahead) and bounded memory are native requirements for streaming VLMs. Some VLMs are also query-agnostic (Zeng et al., 2025), which means that the system must commit to a memory organization before any question is asked of it.

VLMs often organize memory in a flat, chronologically ordered way, mimicking the real-world temporal flow of activities. However, one key attribute of human behavior in the real world that is underexplored in VLM research is "inherently concurrent", or overlapping, activity: individuals frequently suspend an activity, address an interruption, and resume it later (for example, alternating between preparing food and feeding a pet). We refer to these interrupted and resumed activities as activity threads. A purely chronological memory separates these related events whenever another activity intervenes. Long-horizon reasoning may instead require associating temporally distant events that belong to the same ongoing activity.

This motivates three requirements for online long-video streaming memory:

1. Long video: The model has the ability to process continuous video spanning many hours.

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.

2. Online/Streaming-capable, as specified:
   (a) *Causal frame processing*: The model processes frames without lookahead.
   (b) *Bounded memory*: Memory is bounded by a fixed token budget.
   (c) *Query-agnostic*: No query is available at encoding time.
3. Concurrent activity representation: the model can answer questions about activities that overlap, interleave, suspend, and resume.

Existing approaches satisfy subsets of these requirements (Section 2). Fixed-budget streaming memories provide causal, bounded processing, but typically compress events along the chronological stream; structured egocentric memories can represent simultaneously occurring objects or locations, but are not designed as bounded activity memories. Thus, preserving resumable activity structure while maintaining a strictly bounded, query-agnostic streaming memory remains underexplored.

To address this gap, we introduce **ConcurTree**, a bounded, query-agnostic memory structure that organizes video events into persistent activity trees. At any time, we maintain one active tree while others remain dormant and can later be resumed. ConcurTree uses an *exception-based routing* policy: temporal continuation is the default, visual change triggers a possible activity switch, and semantic reasoning is invoked only when deciding whether a new event should return to a dormant tree or begin a new one. Under memory pressure, ConcurTree compresses events within trees using similarity-weighted token merging rather than flattening activity structure into a single chronological memory. Throughout the paper, ConcurTree refers to the full system: the novel memory structure together with its fine-tuned reader unless otherwise specified. The structure's advantage emerges when the reader is trained on the memory it reads.

Our main contributions are as follows:

- **ConcurTree memory structure:** We introduce ConcurTree, a novel online, query-agnostic memory of persistent activity trees designed for streaming long-video understanding (Section 3).
- **Exception-based routing and token merging:** To manage multi-threaded memory causally, we propose an exception-driven routing system that uses visual-change gating and content judging to dynamically attach, switch, or resume activity threads. We employ in-place token merging to retain context despite a bounded memory.
- **State-of-the-art performance:** Evaluated across three held-out benchmarks in controlled settings, ConcurTree achieves its clearest gains when evidence must survive long periods of bounded-memory compression. On our pooled full long-horizon endpoint ($n = 1{,}054$), ConcurTree reaches 50.4% accuracy versus 43.1% for the strongest matched fixed-budget streaming comparator. Controlled analyses further show that the benefit is concentrated in questions requiring reasoning across temporally separated events.

The architecture of ConcurTree itself has no egocentric-specific components; it is a memory structure for streaming video under a bounded token budget, trained on a mix of egocentric and general video data. However, we evaluate on egocentric streams since that is where the stated constraints are most evident. Egocentric video often includes continuous, unedited, all-day capture, with no cuts to mark event boundaries, so it exposes the overlapping nature of human activity.

2 Related Works

Egocentric video and related VLMs. While earlier egocentric models targeted specialized tasks such as spatial awareness, object understanding, or action recognition (Cartas et al., 2017; Nagarajan et al., 2020; Di & Xie, 2024; Rodin et al., 2024; Plizzari et al., 2024), recent MLLMs aim for holistic, general-purpose egocentric video understanding (Ye et al., 2024).

Streaming-capable and query-agnostic VLMs. Streaming-capable (online) VLMs process live video streams causally, offering operational flexibility compared to offline architectures. Offline VLMs (Wang et al., 2024a; Zhang et al., 2025a) simultaneously access past and future frames relative to a query, retaining a performance advantage on long-video benchmarks (Fu et al., 2025). However, holding uncompressed video frames in context incurs prohibitive VRAM overhead (Shen

et al., 2025). Conversely, online VLMs enforce causal processing by blocking future frames and requiring real-time query responses (Lin et al., 2024; Niu et al., 2025). Models also differ in their memory commitment timing: query-conditioned approaches build memory or resample frames post-query (Wang et al., 2024b; Ye et al., 2025), whereas query-agnostic streaming models (Zeng et al., 2025) must commit to a structured memory during video ingestion before any question is posed.

Memory structures and compression. Online VLMs use compression or sliding windows to operate under memory constraints. VideoChat3 (Li et al., 2026b) and other sliding-window models maintain no persistent long-term memory, so content outside the context window is lost. Token merging (Bolya et al., 2023) compresses memory by merging tokens by similarity rather than directly discarding out-of-memory information. StreamForest (Zeng et al., 2025) organizes event-boundary segmented frames into a forest under a fixed token quota, merging nodes through penalty functions over temporal distance, frame similarity, and merge frequency. However, merging operates only over adjacent event nodes. Whareformer (Chalk et al., 2026) maintains a memory where each new observation belongs to an existing track or begins a new one, but its memory tracks objects and their locations, rather than activities. These mechanisms decide what to keep, but do not address how the model's memory relates to human activity structure.

Concurrent event understanding. Prior work has studied overlapping tasks or entities primarily as a recognition problem. Benchmarks such as Charades (Sigurdsson et al., 2016) supervise same-frame simultaneous occurrences, assuming multiple actions can overlap at each timestep. These approaches detect overlapping actions but do not target persistent activity representations that can later be queried. In egocentric video, UnweaveNet (Price et al., 2022) introduces a thread bank and a controller that decides whether each clip continues the ongoing thread, resumes a past thread, or begins a new one. However, UnweaveNet outputs video annotations rather than a queryable memory, and predates video MLLMs. AMEGO (Goletto et al., 2024) constructs an online, query-agnostic structured memory for objects and locations, explicitly addressing video "concurrency". However, its unbounded memory focuses on co-occurring objects or locations rather than activities, and its concurrent memory remains its weakest reasoning category. Meanwhile, event-structured memory in video MLLMs is sequential by construction, compressing only the immediately preceding event into global memory (Cheng et al., 2024). No existing system targets interleaving activities through routing between distinct trees as an LLM-parsable memory structure under a fixed token budget.

3 METHODOLOGY

Our system processes an egocentric stream in three stages (Figure 1), satisfying the three streaming conditions of Section 1:

1. **Encoding**: The model first processes frames causally, which enter a SigLIP detection stream, then into event segmentation, and is finally parsed as an "event" encoding.
2. **Memory**: The encoded event goes through two temporary stores, "now" and "short", and then enters the long ConcurTree memory, where we address encoded events according to the routing policies outlined in Section 3.2.
3. **Answer**: At query time, a fine-tuned Qwen-VL backbone ingests the query, receives the pre-built memory, and answers questions in the form of visual question-answering (VQA).

3.1 ENCODING AND EVENT SEGMENTATION

Event segmentation. After processing the initial frames causally, we compute cosine distances between consecutive SigLIP encoder (Zhai et al., 2023) features. Using a z-score test with no lookahead, we place an event boundary wherever the running z-score of the distances exceeds a tuned threshold. Each event span is bounded between 4 to 16 frames, and then encoded by the backbone from 8 uniformly sampled frames within the span (more details are included in Appendix A.2.1).

Event identification. The Qwen-VL backbone produces a concise one-line caption per event (verb, object, distinguishing detail) that is later used by the routing mechanism. We found in preliminary experiments that caption injection at the answering stage does not improve accuracy when evaluated across question types and times.

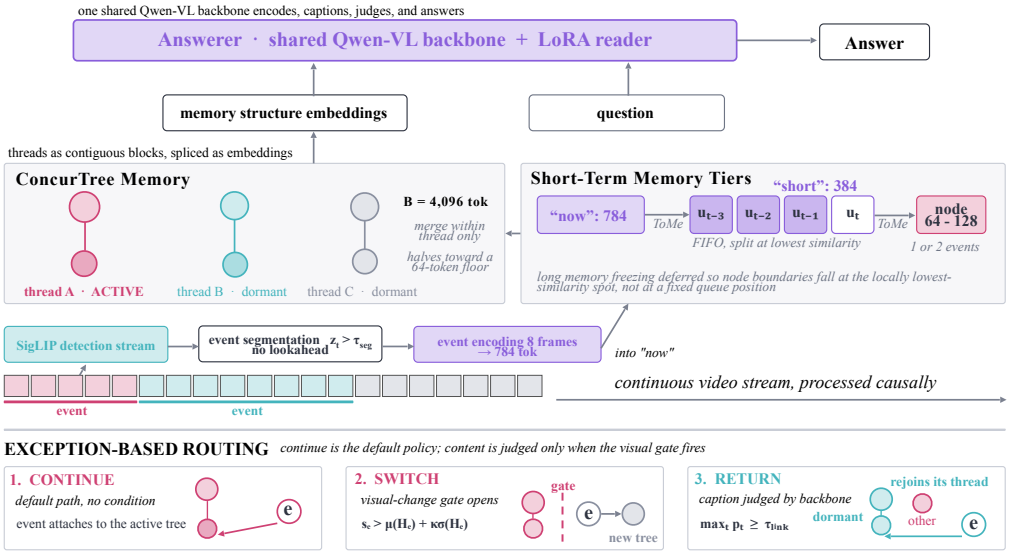

Figure 1: **ConcurTree overview.** Top: Streaming pipeline; one shared VLM backbone serves encoding, captioning, judging, and answering. Bottom: Exception-based routing for interleaved activities: continue by default, open a visual-change gate on potential switches, and invoke content-based judging only for return linking.

Backbone model. A single vision-language backbone, Qwen-VL (Bai et al., 2025), serves four roles: it encodes event frames into visual tokens, generates a short caption per event, acts as the routing judge (Section 3.2), and answers questions (Section 3.3). Sharing one backbone model keeps the system to a single approximately 8B-parameter backbone. The generation at event routing time (event captions and LLM judge) runs on frozen base weights, so answerer finetuning does not alter the memory being constructed. Therefore, later finetuning ablations compare answering ability for the same memory rather than mixing reader and memory-construction ability.

3.2 CONCURTREE MEMORY

"Now" and "Short" windows. After an event of 4–16 frames is encoded into 784 tokens, it passes through two short-term memory tiers before becoming an event node inside the ConcurTree memory structure. The "now" window holds the most recent event, at a budget of 784 tokens. When displaced, the event then gets ToMe (Bolya et al., 2023) compressed into 128 tokens and goes inside the "short" window, which has a budget of 384 tokens, so it can hold up to 3 distinct events. The "short" tier is a FIFO of 128-token events. When the tier is about to hit 4 events (1 over the budget, and we have event units $\{u_{t-3}, u_{t-2}, u_{t-1}, u_t\}$), we examine the similarity scores and decide what to freeze as a node (64–128 tokens). If the similarity is the lowest between $\{u_{t-3}, u_{t-2}\}$, we freeze $\{u_{t-3}\}$ as one node e (64 tokens, $m_e = 1$) and push it into the ConcurTree long memory store. If the similarity is lowest between $\{u_{t-2}, u_{t-1}\}$, we freeze $\{u_{t-3}, u_{t-2}\}$ as one node e (128 tokens, $m_e = 2$). Deferring freezing events into the ConcurTree memory lets node boundaries fall at the locally lowest similarity boundary rather than at a fixed queue position. No frames are revisited here, and this is also done query-agnostically.

Event node definition. An event node e inside the ConcurTree memory structure stores:

- **visual tokens** $V_e \in \mathbb{R}^{f_e \times 64 \times d_{\text{model}}}$, where f_e is the node's effective frame count: f_e counts constituent encoded events; nodes enter with $f_e \in \{1, 2\}$ (64–128 tokens), and merging two nodes yields $f_{a \oplus b} = \lceil (f_a + f_b)/2 \rceil$, so repeated merging keeps nodes between f_e=1 and f_e=2.
- a **timestamp** t_e, the frame-weighted mean of its constituent events' times; node timestamps order nodes for adjacency and merge selection and define each tree's time extent, but are never shown to the answerer.

- a **caption** c_e (at most 12 words: verb, object, one distinguishing detail, generated on the frozen base weights); a dormant thread is represented to the LLM judge for return linking by its most recent node's caption.
- a **merge count** m_e, the number of raw events u the node aggregates, used by compression to spread dilution evenly across the forest (below).

The memory is a forest of *event trees*, one per inferred activity thread; at any moment exactly one tree is *active* and the rest are *dormant*.

Exception-based routing. Each incoming event triggers one of three decisions: continue, switch, or return. Each of these decisions is made using a metric or signal we chose to use, quantified by tuning experiments:

- *Continue by default:* The event attaches to the active tree. In real egocentric footage, most consecutive events belong to the same activity, so temporal continuity is the default.
- *Switch to new activity thread:* When the pooled visual change between consecutive events exceeds an adaptive threshold (Equation 1), open the switch gate — the possibility to either switch activities or re-link to a previous activity thread. We found that visual scene change is a more reliable coarse switch detector than caption change. Let $\bar{p}_e$ be an event e's pooled SigLIP feature (which encodes multiple frames of video):

$$s_e = 1 - \cos\big(\bar{p}_e,\, \bar{p}_{e-1}\big), \qquad \text{gate opens at event } e \iff s_e > \mu(\mathcal{H}_e) + \kappa\,\sigma(\mathcal{H}_e), \quad (1)$$

  where $\mathcal{H}_e = \{s_1, \ldots, s_{e-1}\}$ is the stream's own switch score history, and $\kappa = 0.5$; the gate never opens during the first 16 events, when the history is too short to estimate. The switch gate (mean+$\kappa\sigma$, $\kappa = 0.5$) is permissive by design; precision is supplied downstream by the return judge.
- *Return linking by content, when switch gate activated.* When the switch gate opens, we allow dormant trees to possibly be resumed. We employ our Qwen-VL backbone as a judge that takes each candidate thread's most recent caption and the current event's caption to decide whether the new event continues the suspended activity. The best candidate above an acceptance threshold (based on its Yes/No probability) absorbs the new event, otherwise, we create a new tree. We never scan over the full memory, but only use this content comparison when the switch gate fires based on the statistical signal. See Appendix A.2.2 for more details.

Memory compression. We establish a budget of $B = 4{,}096$ tokens for our memory structure. When the budget is exceeded, we merge one pair of temporally adjacent nodes within a tree through similarity-weighted token averaging (ToMe, (Bolya et al., 2023)). The merged node keeps $f_{a\oplus b} = \lceil (f_a + f_b)/2 \rceil$ effective frames ($64f$ tokens). The merged node also inherits the later pre-merge node's caption (its most recent activity evidence) and the summed merge count, so no captions are regenerated. The pair is chosen by a weighted penalty over all adjacent within-tree pairs. High token similarity, older content, and low accumulated merge count are preferred, the last term spreading dilution evenly rather than repeatedly collapsing one region, with the three weighted $0.4/0.2/0.4$. Because two nodes already at $f{=}1$ merge into another node with $f{=}1$, the model undergoes graceful degradation over long horizons because compression dilutes old information instead of deleting it. Our policy only merges across trees when every tree has degenerated into a singleton state, which was not reached in any of our runs.

3.3 Answerer and LoRA Reader

At query time, we serialize the memory per tree. Trees appear as contiguous blocks ordered by start time and chronologically ordered within each tree. The answerer (Qwen-VL) receives the memory as continuous embeddings (created by the same backbone Qwen-VL) that are spliced into the answerer's input-embedding sequence at a single position inside a text preamble. We found that serializing per tree rather than chronologically does not change the accuracy by itself (Section 4.3), and we retain the per-tree serialization because it is a natural ordering of the structure ConcurTree maintains. The accuracy gains we observe relate to how the memory is built and read (Section 4.3), rather than input order.

The answerer, a shared Qwen-VL backbone, reads the serialized memory along with the question. For multiple-choice questions, the option list and answer-format instruction are appended to the prompt. The frozen backbone answers competitively in this interface, partly due to the memory being encoded by the same model. However, it was not explicitly trained to read merged, activity-grouped memory, so we adapt the reader to the memory states ConcurTree produces (Table 1). We finetune the answerer with a LoRA adapter to read the memory, and the training data and policy are detailed in Appendix A.3.

4 Experiments and Results

4.1 Experimental Setup

Training and validation data. We train ConcurTree's answerer reader, a LoRA adapter on Qwen3-VL-8B-Instruct, on 168,226 QA examples drawn from four sources, with evaluation footage excluded from training. Of these, 159,834 (95%) form an even 50:50 mixture of egocentric and general-domain instruction data. The egocentric portion uses Ego4D QA (Bärmann & Waibel, 2022) and a 10k subset of EgoIT-99K (Yang et al., 2026), excluding the EgoLife evaluation slice, while the general-domain portion comes from LLaVA-Video-178K (Zhang et al., 2024). The remaining 8,392 examples (5%) form a deep-evidence supplement derived from EgoIT-99K to improve recall of temporally distant events. Separately from training, we use EgoLongQA (Tran et al., 2026) (700 questions) and an HD-EPIC VQA slice (Perrett et al., 2025) (1,330 questions) for development and calibration, with MVBench (Li et al., 2024b) used only as an internal regression check. Further fine-tuning data-mixture ablations and hyperparameter details are provided in Appendix A.3.

Evaluation benchmarks. We evaluate ConcurTree on three complementary egocentric video QA benchmarks whose footage is excluded from training: **EgoLifeQA** (Yang et al., 2026) (500 four-way MCQs), **Ego-R1-Bench** (Tian et al., 2025) (300 four-way MCQs), and **TeleEgo** (Yan et al., 2025) (2,850 mixed-format MCQs). EgoLifeQA and Ego-R1 contain week-long streams with supporting evidence extending up to approximately six days before the query and are evaluated through causal replay. TeleEgo provides a native online evaluation setup with per-second ingestion. Detailed dataset descriptions, chance levels, ingestion settings, and evaluation setups are provided in Appendix A.4. We also evaluate on the EgoMemReason dataset (Wang et al., 2026), but for comparability, we do not report these results in the main table. Extended results are provided in Appendix A.1.

Long-horizon memory. ConcurTree specifically targets evidence that must survive sustained compression under a bounded memory budget. ConcurTree's 4,096-token long-term memory corresponds to only 64 events. We therefore separately evaluate questions whose supporting evidence is older than 12 and 24 hours on EgoLifeQA ($n = 174/109$) and Ego-R1 ($n = 173/129$), together with TeleEgo's Ultra-long and Long-term Memory categories. Our primary pooled long-horizon endpoint combines the $>$24h subsets of EgoLifeQA and Ego-R1 with the corresponding TeleEgo long-horizon questions. We report this endpoint using both TeleEgo P1 ($n = 420$ total) for direct comparison with published baselines and the complete P1–P5 evaluation ($n = 1{,}054$) for our expanded evaluation.

Evaluation protocol. All evaluations use a single causal pass over each video: frames are processed in temporal order without lookahead, revisiting, or query-dependent resampling, and memory is constructed fully query-agnostically. Questions are issued only from memory available at their annotated query time. We adapt each benchmark to our streaming, query-agnostic evaluation setting using causal replay (Lin et al., 2024; Niu et al., 2025). Benchmark-specific ingestion rates and evaluation details are provided in Appendix A.2 and Appendix A.4.

Baselines and comparison regimes. We compare ConcurTree against representative streaming video-memory architectures. StreamForest maintains persistent event memory by hierarchically merging temporally adjacent events; we evaluate both its persistent query-agnostic and its query-conditioned configuration (Zeng et al., 2025). HERMES treats the KV cache as hierarchical streaming memory, while InfiniPot-V maintains a fixed-size KV cache by removing temporally redundant tokens and retaining semantically salient ones (Zhang et al., 2026; Kim et al., 2026). Flash-VStream maintains complementary context and detail-oriented memories for real-time long-video understanding (Zhang et al., 2025b), while VideoChat-Online provides a bounded online memory

baseline (Huang et al., 2025). We additionally include VideoChat3 (Li et al., 2026b) as a query-conditioned streaming reference. For offline reference points, we evaluate LLaVA-OneVision (Li et al., 2024a), LLaVA-Video (Zhang et al., 2024), EgoGPT (Yang et al., 2026), and Qwen-VL (Bai et al., 2025) configurations that resample frames from the available video prefix at query time.

We group these systems into three evaluation regimes: (i) streaming, fixed-budget, query-agnostic systems that match ConcurTree's constraints; (ii) streaming systems with query-conditioned or unbounded memory; and (iii) offline systems that may resample frames at query time. Only regime (i) provides a directly matched comparison. Where possible, we evaluate these baselines under ConcurTree's 4,096-token long-term memory budget and additionally report their released native configurations. We preserve each comparator's native memory organization and mark modified token-budget or resolution configurations as "matched." Full baseline configurations are provided in Appendix A.4.2.

4.2 Main Results

Long-horizon performance. Long-horizon memory is the primary target of ConcurTree. We therefore focus our main comparison on questions whose evidence must survive sustained compression: EgoLifeQA and Ego-R1 questions with evidence more than 24 hours old, together with TeleEgo's Ultra-long and Long-term Memory categories.

On the pooled full long-horizon endpoint ($n = 1{,}054$), ConcurTree achieves 50.4% accuracy, outperforming all evaluated streaming baselines with complete results. Among fixed-budget, query-agnostic systems operating under the same streaming constraints, the strongest comparator is matched StreamForest-persistent at 43.1%, giving ConcurTree a $+7.3$-point improvement. The margin increases over the other matched streaming baselines: ConcurTree exceeds matched InfiniPot-V (41.0%) by 9.4 points, resolution-matched HERMES (40.6%) by 9.8 points, matched Flash-VStream (37.7%) by 12.7 points, and VideoChat-Online (29.8%) by 20.6 points. ConcurTree also exceeds the evaluated query-conditioned streaming systems, StreamForest (42.2%) and VideoChat3-4B (41.5%), despite their less restrictive memory setting. These results make the full long-horizon endpoint our most comprehensive comparison: across more than one thousand questions, ConcurTree consistently outperforms alternative streaming memory systems.

The improvement is also consistent across the constituent long-horizon benchmarks. ConcurTree reaches 48.6% on EgoLifeQA at $>$ 24h and 39.5% on Ego-R1 at $>$ 24h, compared with 45.0% and 38.0%, respectively, for matched StreamForest-persistent, and achieves 52.3% on the full TeleEgo long-horizon set. For systems without complete TeleEgo results, we additionally report the common P1-based pooled endpoint ($n = 420$), where ConcurTree achieves 47.6% versus 41.2% for matched StreamForest-persistent. Together, these results show that ConcurTree's strongest advantage appears in the regime it is designed for: retaining useful evidence over long horizons under a fixed streaming memory budget.

Performance across evidence horizons. Figure 2 reports EgoLifeQA performance across evidence-age subsets. Because these bins contain different questions and unequal numbers of examples, absolute accuracy should not be compared across bins as if evidence age alone caused performance differences; question composition and difficulty can also shift the absolute scores. We instead focus on the between-model gap within each bin. On recent-evidence subsets, the streaming systems remain relatively close, whereas ConcurTree shows its clearest separation on the $>$ 24h subset. There, ConcurTree reaches 48.6%, compared with 45.0% for matched StreamForest-persistent, 44.0% for matched Flash-VStream, 37.6% for matched InfiniPot-V, and 33.0% for resolution-matched HERMES. Thus, the relevant observation is not that accuracy itself increases with evidence age, but that ConcurTree's relative advantage is largest in the long-horizon regime, where evidence must survive sustained routing and compression. Additional evidence-age and question-type analyses are provided in Ap-

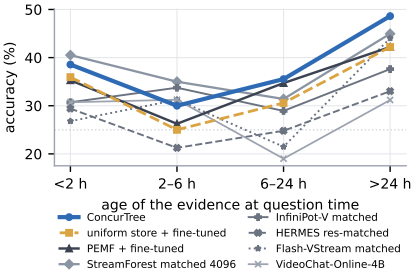

Figure 2: EgoLifeQA accuracy by evidence age.

Table 1: **Main results: full-test and long-horizon evaluation sets, grouped by evaluation regime.**

		EgoLifeQA			Ego-R1			TeleEgo				pooled, TeleEgo P1		pooled, TeleEgo P1–P5	
System	Budget / exposure	all	>12h	>24h	all	>12h	>24h	P1	P1 long-hor.	P1–P5	P1–P5 long-hor.	all	long-hor.	all	long-hor.
(iii) offline, frame budget over the prefix, unbounded storage															
LLaVA-OneVision-7B, 32 f	≈6.4k	32.6*	36.2	39.4	35.7	35.8	33.3	46.1*	45.1	44.0*	42.6*	38.6*	40.0*	41.8*	41.2*
LLaVA-Video-7B, 32 f	≈6.4k	37.0	45.4	**50.5**	34.3	33.5	32.6	49.5	48.9	47.4*	47.7*	41.3	44.3	44.9*	46.1*
EgoGPT-7B, 32 f, video only	≈6.4k	33.8	35.6*	36.7	34.0	35.3	35.7	48.4	41.8*	46.7*	45.2*	39.6*	38.6*	43.9*	43.2*
Qwen2.5-VL-7B, TeleEgo official script	n/a	n/p	n/p	n/p	n/p	n/p	n/p	47.6*	44.5	43.8*	41.4*	n/p	n/p	n/p	n/p
Qwen3-VL-8B offline, 64 uniform frames	none / 3,200	30.0*	34.5*	31.2*	29.3*	28.3*	28.7*	52.2	48.9	50.0	49.4	38.6*	38.1*	45.6*	45.0*
Qwen3-VL-8B offline, 768 uniform frames	none / 38,400	32.4*	37.4	36.7*	32.0	31.8	31.0	55.4	51.6	51.6	53.8	41.4	41.4*	47.3	49.2
Qwen2.5-VL-7B offline, 64 uniform frames	none / 4,608	31.4*	40.2	40.4	30.3	33.5	34.9	45.5*	46.7	41.9*	42.5*	36.7*	41.4*	39.5*	41.4*
(ii) streaming, query-conditioned or unbounded store															
StreamForest 128 f	query-cond.	37.8	38.5	42.2	**38.3**	38.7	37.2	44.7*	39.6*	44.2*	43.0*	40.7*	39.5*	42.8*	42.2*
VideoChat3-4B	query-cond.	37.6	42.5	39.4	32.0	29.5*	29.5	47.8	47.8	46.0*	43.6*	40.4*	40.0*	43.7*	41.5*
(i) streaming, fixed budget, query-agnostic															
VideoChat-Online-4B, native bank	832 / 832	28.0*	28.2*	31.2*	27.7*	28.9*	30.2	33.3	33.0	32.3*	29.5*	30.0*	31.7*	31.3*	29.8*
HERMES, kv 4096, native resolution	4,096 / 4,110	27.4*	37.4	38.5	28.7*	28.3*	27.1*	38.4	34.1	37.6*	33.3*	32.0*	33.1*	35.4*	33.1*
HERMES, kv 4096, resolution-matched	4,096 / 4,110	27.8*	32.2*	33.0*	28.0*	30.6	33.3	46.5	42.3	42.8*	42.8*	35.2*	37.1*	39.5*	40.6*
HERMES, kv 6000, native	6,000 / 6,014	27.8*	34.5*	33.9*	28.7*	30.1	29.5	45.5*	41.2*	41.3*	40.6*	35.0*	35.7*	38.4*	38.5*
InfiniPot-V, matched	3,872 / 2.9–3.9k	31.6*	38.5	37.6	31.7	35.3	31.8	48.6	48.9	43.7*	42.9*	38.3*	40.7*	41.0*	41.0*
InfiniPot-V, native 6K	5,808 / ≤5.8k	31.0*	36.2	36.7	29.0*	31.2	31.8	45.7	47.3	42.9*	42.8*	36.4*	39.8*	40.1*	40.8*
Flash-VStream-Qwen-7B, matched	4,096 / 4,096	30.2*	36.8	44.0	31.3	32.4	29.5	45.5*	38.5*	43.9*	38.1*	36.5*	37.1*	41.0*	37.7*
Flash-VStream-Qwen-7B, native	11,520 / 11,520	30.0*	37.4	41.3	34.0	32.4	31.0	44.7*	40.7*	42.9*	38.0*	36.7*	37.9*	40.3*	37.5*
StreamForest-persistent, native	5,120 / ≈8.2k	38.6	39.7	41.3	35.7	37.6	37.2	44.4*	38.5*	43.7*	42.5*	40.2*	38.8*	42.3*	41.7*
StreamForest-persistent, matched	4,096 / ≈7.1k	38.4	41.4	45.0	36.3	38.7	38.0	44.0*	41.2*	43.5*	43.6*	40.1*	41.2*	42.2*	43.1*
ConcurTree + frozen Qwen2.5-VL-7B reader	4,096 / 5,264	32.6	43.7	43.1	30.3	31.2	31.8	48.9	49.5	46.5*	45.5*	38.5*	42.4	43.3*	43.5*
ConcurTree + frozen reader	4,096 / 5,264	34.0	38.5	43.1	32.3	32.9	31.8	**56.8**	50.5	**52.8**	50.7	42.6	42.9	**48.5**	47.6
ConcurTree + fine-tuned reader (ours)	4,096 / 5,264	**38.8**	**45.4**	48.6	36.3	**39.3**	**39.5**	53.0	**52.7**	51.3	**52.3**	**43.8**	**47.6**	48.3	**50.4**

Stars denote paired McNemar significance versus ConcurTree + fine-tuned reader on the same questions, $p < 0.05$; see Appendix A.2.4.

pendix A.1.1. Across week-scale streams, ConcurTree structure routes events to different activity trees and performs compression. The long-horizon results are therefore obtained after repeated routing and compression rather than by retaining the original event stream (Appendix A.2.2).

Overall performance. The long-horizon gains do not come at the expense of full-test performance. ConcurTree achieves 38.8% on EgoLifeQA, compared with 38.4% for matched StreamForest-persistent, and 36.3% on Ego-R1, matching the strongest fixed-budget streaming comparator. On the full TeleEgo evaluation, ConcurTree achieves 51.3%. Thus, while ConcurTree's clearest advantage occurs at long evidence horizons, it remains competitive on the complete benchmark distributions. Efficiency analyses in Appendix A.1.3 further show that greater query-time visual-token exposure alone does not explain these performance differences.

Comparison across evaluation regimes. Table 1 also reports systems operating under less restrictive evaluation settings for context. Query-conditioned streaming methods (regime (ii)) may select or construct memory after receiving the query, while offline models (regime (iii)) resample frames from the available video prefix for each question; we therefore treat these results as reference points rather than directly matched comparisons. Nevertheless, ConcurTree exceeds the evaluated query-conditioned streaming systems on the pooled full long-horizon endpoint. Offline performance is less consistent across benchmarks: for example, LLaVA-Video reaches 50.5% on EgoLifeQA at > 24h but only 32.6% on Ego-R1 at > 24h.

Backbone differences alone also do not explain the long-horizon result. With a frozen Qwen2.5-VL-7B reader, matching the backbone generation used by several comparison systems, ConcurTree reaches 43.1% on EgoLifeQA at > 24h, compared with 37.6% for matched InfiniPot-V and 33.0% for resolution-matched HERMES. Section 4.3 further isolates the effects of memory structure and reader adaptation.

To further examine the gains of the long-horizon memory, we also split EgoLifeQA by the type of information required: episodic questions (EventRecall, TaskMaster, HabitInsight), which require recalling and relating events across an activity, and lookup questions (EntityLog, RelationMap), which retrieve a fact about an entity. ConcurTree's advantage is concentrated in episodic questions (Appendix A.1.1).

4.3 Ablation Study

Memory structure. We investigate the contribution of ConcurTree's memory structure by comparing it with three alternative long-term memory storage methods: (1) an unbounded uniform store,

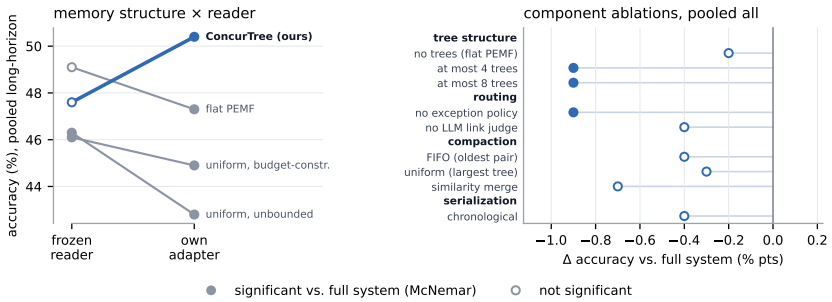

Figure 3: **Ablations on ConcurTree's memory structure and architectural components.**

which retains all events during ingestion and uniformly samples them to the reader's input budget at query time; (2) a budget-constrained uniform store, which maintains a fixed memory budget by iteratively removing the interior event whose two temporal neighbors are closest together; and (3) flat PEMF (Persistent Event Memory Forest), the memory structure used by StreamForest. We evaluate each memory policy under two reader conditions: a frozen backbone and a structure-specific reader trained separately on memory states produced by that policy (Figure 3). Under the frozen reader, ConcurTree's own memory structure does not significantly outperform alternative structures. However, with a policy-specific adapter trained on each memory's own states, ConcurTree reaches 50.4% on the pooled long-horizon endpoint, significantly outperforming all three alternative memory structures (Figure 3). In contrast, none of the alternatives benefits comparably from its policy-specific adapter. The memory structure's advantage therefore depends on a reader trained on its memory states. This result is consistent with ConcurTree's gain concentrating in long-horizon questions and episodic questions, rather than in more entity-centric ones (Appendix A.1.1).

Architectural components of ConcurTree. Ablating the mechanisms that construct and maintain the memory structure further identifies which design choices matter (Figure 3). The LLM return-link judge and the exception-based routing policy were added to let a dormant thread be resumed by content rather than recency alone. Replacing either of them lowers accuracy, suggesting that content-based returns contribute to what the memory retains. Capping the number of trees or using PEMF's single tree in the forest results in accuracy drops, suggesting that constraining how events are partitioned degrades what the memory retains. We also compare ConcurTree's weighted compaction strategy (Section 3.2) with three alternative strategies, none of which differs significantly from the full system. Changing our serialization strategy from per-tree to chronological serialization shows no significant difference; we keep per-tree serialization as a design choice that reflects ConcurTree's construction philosophy of distinguishing between trees. In Figure 3, we display component ablation numbers from all pooled questions (across the three benchmarks).

5 Conclusion

In this work, we presented ConcurTree, a novel memory structure for streaming, query-agnostic long-video understanding under a bounded token budget, motivated by the interleaved and resumable structure of human activities. Experiments on three held-out benchmarks under strict streaming model requirements show that ConcurTree improves long-horizon accuracy by 7.3 points over the strongest matched streaming baseline. In particular, ConcurTree memory structure performs better with an adapter trained to read its memory structure, with its clearest gains on long-horizon questions and those requiring reasoning across temporally separated events. Ablations further indicate that the benefit comes primarily from how events are organized and retained, rather than from their serialization order at read time. Passing in memory chronologically does not change accuracy significantly, while constraining how events are partitioned into trees lowers performance. Two limitations point to future work. ConcurTree shows no significant gain on entity lookups, where answers depend less directly on activity structure. Our evaluation benchmarks do not annotate interleaved activity threads, so the structure is evaluated only through QA accuracy. Benchmarks that explicitly address interleaving activity structure are a promising area of future work.

AI USE STATEMENT

In this work, we used generative AI tools to assist with research implementation, experimental refinement, data preparation, analysis, literature review, and manuscript preparation. Specifically, generative AI assisted with software development and experiment implementation; proposed ablation ideas and refinements to answerer-training experiments. AI-generated experimental proposals were reviewed by the authors before execution, and the authors independently evaluated the resulting experimental evidence and interpretations.

Generative AI also assisted with cleaning and reformatting training data, including identifying data-quality issues, constructing a supplementary training set, and performing automated sample audits against source video narrations. These procedures were initiated and supervised by the authors, and AI-suggested data modifications and audit decisions were reviewed against the intended data criteria.

For literature and manuscript preparation, we used generative AI to identify relevant literature, retrieve and aggregate supporting information, assist with paper organization and formatting, and edit or draft text for clarity and grammatical correctness. References identified with AI assistance were manually checked for validity and relevance.

Generative AI was not used to originate the core ConcurTree architecture, its exception-based routing policy, or the primary evaluation design, and it was not used to generate synthetic research datasets. AI assistance was used for subsequent refinements and ablations of the human-designed system and experimental procedure.

We reviewed and validated all AI-assisted research contributions. AI-generated code was tested for correctness, experimental implementations and outputs were checked against the intended experimental design and sanity checks, proposed ablations and methodological refinements were reviewed before execution, and data-processing changes were verified against our specified criteria. The authors independently verified the reported results and take responsibility for the final content of this work, including all text, claims, code, analyses, and other artifacts produced with the assistance of generative AI.

ETHICS STATEMENT

This work only uses existing, publiclly released egocentric video benchmarks and related datasets. We did not collect any data related to human subjects. There are no conflicts of interest or issues related to sponsorship sources between authors and funding sources.

REPRODUCIBILITY STATEMENT

For reproducibility, we anonymously release our source code. In our paper, we have also tried our best to include implementation specifics to ensure the approach can be replicated. The code can be found here: https://anonymous.4open.science/r/ConcurTree-4B45.

REFERENCES

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, Wenbin Ge, Zhifang Guo, Qidong Huang, Jie Huang, Fei Huang, Binyuan Hui, Shutong Jiang, Zhaohai Li, Mingsheng Li, Mei Li, Kaixin Li, Zicheng Lin, Junyang Lin, Xuejing Liu, Jiawei Liu, Chenglong Liu, Yang Liu, Dayiheng Liu, Shixuan Liu, Dunjie Lu, Ruilin Luo, Chenxu Lv, Rui Men, Lingchen Meng, Xuancheng Ren, Xingzhang Ren, Sibo Song, Yuchong Sun, Jun Tang, Jianhong Tu, Jianqiang Wan, Peng Wang, Pengfei Wang, Qiuyue Wang, Yuxuan Wang, Tianbao Xie, Yiheng Xu, Haiyang Xu, Jin Xu, Zhibo Yang, Mingkun Yang, Jianxin Yang, An Yang, Bowen Yu, Fei Zhang, Hang Zhang, Xi Zhang, Bo Zheng, Humen Zhong, Jingren Zhou, Fan Zhou, Jing Zhou, Yuanzhi Zhu, and Ke Zhu. Qwen3-vl technical report, 2025. URL https://arxiv.org/abs/2511.21631.

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

Daniel Bolya, Cheng-Yang Fu, Xiaoliang Dai, Peizhao Zhang, Christoph Feichtenhofer, and Judy Hoffman. Token merging: Your ViT but faster. In *International Conference on Learning Representations (ICLR)*, 2023.

Alejandro Cartas, Juan Marín, Petia Radeva, and Mariella Dimiccoli. Recognizing activities of daily living from egocentric images. In *Iberian Conference on Pattern Recognition and Image Analysis (IbPRIA)*, 2017.

Jacob Chalk, Saptarshi Sinha, Dima Damen, Yannis Kalantidis, and Diane Larlus. Whareformer: Learning to track what is where in long egocentric videos. In *European Conference on Computer Vision (ECCV)*, 2026.

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.

Shangzhe Di and Weidi Xie. Grounded question-answering in long egocentric videos. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

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 *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 24108–24118, 2025.

Gabriele Goletto, Tushar Nagarajan, Giuseppe Averta, and Dima Damen. Amego: Active memory from long egocentric videos. In *European Conference on Computer Vision (ECCV)*, volume 15071 of *Lecture Notes in Computer Science*, pp. 92–110. Springer, 2024. doi: 10.1007/978-3-031-72624-8_6.

Zhenpeng Huang, Xinhao Li, Jiaqi Li, Jing Wang, Xiangyu Zeng, Cheng Liang, Tao Wu, Xi Chen, Liang Li, and Limin Wang. Online video understanding: Ovbench and videochat-online. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3328–3338. IEEE, 2025.

Minsoo Kim, Kyuhong Shim, Jungwook Choi, and Simyung Chang. Infinipot-v: Memory-constrained kv cache compression for streaming video understanding. *Advances in Neural Information Processing Systems*, 38:138983–139013, 2026.

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.

Jiazheng Li, Chi-Hao Wu, Yunze Liu, Kaize Ding, Jundong Li, and Chuxu Zhang. Bridging modalities, spanning time: Structured memory for ultra-long agentic video reasoning, 2026a. URL https://arxiv.org/abs/2605.08271.

Kunchang Li, Yali Wang, Yinan He, Yizhuo Li, Yi Wang, Yi Liu, Zun Wang, Jilan Xu, Guo Chen, Ping Luo, et al. Mvbench: A comprehensive multi-modal video understanding benchmark. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 22195–22206, 2024b.

Xinhao Li, Yuhan Zhu, Xiangyu Zeng, Yuhao Dong, Haoning Wu, Zhiqiu Zhang, Yuandong Yang, Changlian Ma, Qingyu Zhang, Yansong Shi, Xinyu Chen, Haoran Chen, Zizheng Huang, Jun Zhang, Kun Ouyang, Lin Sui, Ziang Yan, Yicheng Xu, Chenting Wang, Yinan He, Hongjie Zhang, Yi Wang, Yu Qiao, Yali Wang, Ziwei Liu, Kai Chen, and Limin Wang. Videochat3: Fully open video mllm for efficient and generalist video understanding, 2026b.

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.

Quinn McNemar. Note on the sampling error of the difference between correlated proportions or percentages, 1947. ISSN 1860-0980.

Tushar Nagarajan, Yanghao Li, Christoph Feichtenhofer, and Kristen Grauman. Ego-topo: Environment affordances from egocentric video. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

Junbo Niu, Yifei Li, Ziyang Miao, Chunjiang Ge, Yuanhang Zhou, Qihao He, Xiaoyi Dong, Haodong Duan, Shuangrui Ding, Rui Qian, Pan Zhang, Yuhang Zang, Yuhang Cao, Conghui He, and Jiaqi Wang. Ovo-bench: How far is your video-llms from real-world online video understanding? In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 18902–18913, 2025.

Toby Perrett, Ahmad Darkhalil, Saptarshi Sinha, Omar Emara, Sam Pollard, Kranti Parida, Kaiting Liu, Prajwal Gatti, Siddhant Bansal, Kevin Flanagan, Jacob Chalk, Zhifan Zhu, Rhodri Guerrier, Fahd Abdelazim, Bin Zhu, Davide Moltisanti, Michael Wray, Hazel Doughty, and Dima Damen. Hd-epic: A highly-detailed egocentric video dataset, 2025. URL https://arxiv.org/abs/2502.04144.

Chiara Plizzari, Gabriele Goletto, Antonino Furnari, Siddhant Bansal, Francesco Ragusa, Giovanni Maria Farinella, Dima Damen, and Tatiana Tommasi. An outlook into the future of egocentric vision. *International Journal of Computer Vision*, pp. 1–57, 2024.

Will Price, Carl Vondrick, and Dima Damen. Unweavenet: Unweaving activity stories. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 13770–13779, 2022.

Ivan Rodin, Antonino Furnari, Kyle Min, Subarna Tripathi, and Giovanni Maria Farinella. Action scene graphs for long-form understanding of egocentric videos. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

Xiaoqian Shen, Yunyang Xiong, Changsheng Zhao, Lemeng Wu, Jun Chen, Chenchen Zhu, Zechun Liu, Fanyi Xiao, Balakrishnan Varadarajan, Florian Bordes, Zhuang Liu, Hu Xu, Hyunwoo J. Kim, Bilge Soran, Raghuraman Krishnamoorthi, Mohamed Elhoseiny, and Vikas Chandra. LongVU: Spatiotemporal adaptive compression for long video-language understanding. In *International Conference on Machine Learning (ICML)*, volume 267 of *Proceedings of Machine Learning Research*, pp. 54582–54599, 2025.

Gunnar A. Sigurdsson, Gül Varol, Xiaolong Wang, Ali Farhadi, Ivan Laptev, and Abhinav Gupta. Hollywood in homes: Crowdsourcing data collection for activity understanding. In *European Conference on Computer Vision (ECCV)*, 2016.

Shitong Sun, Ke Han, Yukai Huang, Weitong Cai, and Jifei Song. Egograph: Temporal knowledge graph for egocentric video understanding, 2026. URL https://arxiv.org/abs/2602.23709.

Shulin Tian, Ruiqi Wang, Hongming Guo, Penghao Wu, Yuhao Dong, Xiuying Wang, Jingkang Yang, Hao Zhang, Hongyuan Zhu, and Ziwei Liu. Ego-r1: Chain-of-tool-thought for ultra-long egocentric video reasoning, 2025. URL https://arxiv.org/abs/2506.13654.

Tuyen (Harry) Tran, Maxim Arap, Seungwhan Moon, Raffay Hamid, Alessandro Suglia, Zsolt Kira, Pascale Fung, and Mubarak Shah. Wearable ai workshop at eccv 2026. https://wearable-ai-workshop.github.io/, 2026. Workshop at the European Conference on Computer Vision (ECCV) 2026.

Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Yang Fan, Kai Dang, Mengfei Du, Xuancheng Ren, Rui Men, Dayiheng Liu, Chang Zhou, Jingren Zhou, and Junyang Lin. 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 (ECCV)*, 2024b.

Ziyang Wang, Yue Zhang, Shoubin Yu, Ce Zhang, Zengqi Zhao, Jaehong Yoon, Hyunji Lee, Gedas Bertasius, and Mohit Bansal. Egomemreason: A memory-driven reasoning benchmark for long-horizon egocentric video understanding, 2026. URL https://arxiv.org/abs/2605.09874.

Guangxuan Xiao, Yuandong Tian, Beidi Chen, Song Han, and Mike Lewis. Efficient streaming language models with attention sinks, 2024. URL https://arxiv.org/abs/2309.17453.

Jiaqi Yan, Ruilong Ren, Jingren Liu, Shuning Xu, Ling Wang, Yiheng Wang, Xinlin Zhong, Yun Wang, Long Zhang, Xiangyu Chen, Changzhi Sun, Jixiang Luo, Dell Zhang, Hao Sun, Chi Zhang, and Xuelong Li. Teleego: Benchmarking egocentric ai assistants in the wild, 2025. URL https://arxiv.org/abs/2510.23981.

Jingkang Yang, Shuai Liu, Hongming Guo, Yuhao Dong, Xiamengwei Zhang, Sicheng Zhang, Pengyun Wang, Zitang Zhou, Binzhu Xie, Ziyue Wang, Bei Ouyang, Zhengyu Lin, Marco Cominelli, Zhongang Cai, Yuanhan Zhang, Peiyuan Zhang, Fangzhou Hong, Joerg Widmer, Francesco Gringoli, Lei Yang, Bo Li, and Ziwei Liu. Egolife: Towards egocentric life assistant, 2026. URL https://arxiv.org/abs/2503.03803.

Hanrong Ye, Haotian Zhang, Erik Daxberger, Lin Chen, Zongyu Lin, Yanghao Li, Bowen Zhang, Haoxuan You, Dan Xu, Zhe Gan, Jiasen Lu, and Yinfei Yang. Mm-ego: Towards building egocentric multimodal llms for video qa. *arXiv preprint arXiv:2410.07177*, 2024.

Jinhui Ye, Zihan Wang, Haosen Sun, Keshigeyan Chandrasegaran, Zane Durante, Cristobal Eyzaguirre, Yonatan Bisk, Juan Carlos Niebles, Ehsan Adeli, Li Fei-Fei, Jiajun Wu, and Manling Li. Rethinking temporal search for long-form video understanding. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8579–8591, 2025.

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. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025. Spotlight.

Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid loss for language image pre-training, 2023.

Boqiang Zhang, Kehan Li, Zesen Cheng, Zhiqiang Hu, Yuqian Yuan, Guanzheng Chen, Sicong Leng, Yuming Jiang, Hang Zhang, Xin Li, Peng Jin, Wenqi Zhang, Fan Wang, Lidong Bing, and Deli Zhao. Videollama 3: Frontier multimodal foundation models for image and video understanding. *arXiv preprint arXiv:2501.13106*, 2025a.

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

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, 2026.

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

Table 2: **EgoLifeQA accuracy by question type.** Episodic combines EventRecall, TaskMaster, and HabitInsight; lookup combines EntityLog and RelationMap.

System	EventRecall	TaskMaster	HabitInsight	EntityLog	RelationMap	episodic	lookup
External streaming systems							
StreamForest-persistent, matched 4096	43.7	38.1	37.7	34.4	37.6	40.8	36.0
VideoChat3-4B	39.7	33.3	49.2	36.8	32.8	40.4	34.8
InfiniPot-V, matched	32.5	36.5	27.9	32.8	28.8	32.4	30.8
HERMES, resolution-matched	31.7	25.4	27.9	30.4	22.4	29.2	26.4
Flash-VStream-Qwen, matched	34.9	34.9	27.9	28.0	26.4	33.2	27.2
VideoChat-Online-4B	26.2	27.0	27.9	31.2	27.2	26.8	29.2
Controlled memory structures							
recency window + fine-tuned	34.9	33.3	36.1	36.8	38.4	34.8	37.6
uniform store + fine-tuned	32.5	38.1	36.1	34.4	33.6	34.8	34.0
PEMF + fine-tuned	34.1	34.9	36.1	35.2	36.8	34.8	36.0
CONCURTREE + fine-tuned reader (ours)	40.5	41.3	41.0	34.4	39.2	40.8	36.8

A APPENDIX

A.1 EXTENDED RESULTS

A.1.1 MEMORY STRUCTURE GAINS: WHERE DOES CONCURTREE HELP?

Evidence age. We examine in more detail whether the benefit of structured memory depends on how long evidence must be retained. Figure 2 bins questions by the age of their supporting evidence at query time. On EgoLifeQA, systems remain relatively close for recent evidence, but ConcurTree separates most clearly once evidence is more than 24 hours old, reaching 48.6%, compared with 44.0% for the fine-tuned recency window, 42.2% for both the uniform store and flat PEMF, and 45.0% for matched StreamForest. This is the evaluation setting in which relevant evidence must survive sustained compression under the fixed memory budget.

Question type. We split EgoLifeQA by the type of information required (Table 2). ConcurTree's advantage is concentrated in the three episodic categories (EventRecall, TaskMaster, and HabitInsight) which require retaining and relating events across an activity. On the pooled episodic subset, ConcurTree reaches 40.8%, compared with 34.8% for each of the fine-tuned recency, uniform, and flat-PEMF controls. Relative to flat PEMF, this is a +6.0-point improvement (p=.004). In contrast, on the EntityLog and RelationMap lookup questions, ConcurTree reaches 36.8%, compared with 36.0% for PEMF and 37.6% for the recency window; the ConcurTree–PEMF difference is not significant (+0.8, p=.80). Together with the evidence-age analysis, these results indicate that ConcurTree's structural benefit is concentrated where temporally distant evidence must remain associated with an ongoing activity, rather than in recent or entity-centric retrieval.

Extended ablations. To supplement Section 4.3, we report our ablations results in detail in Table 3. All systems in this table use Qwen3-VL-8B with a 4,096-token long-term memory budget for comparability, and all component ablations reuse the full system's fine-tuned reader without retraining. All significance tests use the paired McNemar test against the full ConcurTree system.

EgoLifeQA shows larger component effects than comparing across all questions pooled. In particular, every tree and routing ablation drops significantly on EgoLifeQA's episodic questions, which reinforces that ConcurTree's advantage is more concentrated across episodic questions, compared to entity lookup questions.

We also report a few negative results from the ablation study: some ablation cells show gains compared to our full system ConcurTree, although only one cell is statistically significant. On Ego-R1 (all questions), flat PEMF with its own fine-tuned reader wins against ConcurTree (non-significantly). On TeleEgo (all questions), flat PEMF with a frozen reader wins against full-system ConcurTree, as the only significant negative result. In general, chronological serialization is slightly positive on EgoLifeQA, but because it is not clearly better than per-tree serialization, we use per-tree serialization in our final model.

Table 3: **Ablations on ConcurTree's memory structure and architectural components.**

	EgoLifeQA				Ego-R1		TeleEgo P1–P5		pooled	
System (what changes)	all	episodic	lookup	>24h	all	>24h	all	long-hor.	all	long-hor.
memory under the frozen Qwen3-VL-8B reader (no adapter) — what each long-store policy buys before any fine-tuning										
ConcurTree (our memory)	34.0 (−4.8)	34.4 (−6.4)	33.6 (−3.2)	43.1 (−5.5)	32.3 (−4.0)	31.8 (−7.8)	52.8 (+1.5)	50.7 (−1.6)	48.5 (+0.2)	47.6 (−2.8)
uniform store, unbounded (budget applied at read time)	32.2 (−6.6)*	32.0 (−8.8)*	32.4 (−4.4)	37.6 (−11.0)	34.3 (−2.0)	32.6 (−7.0)	51.2 (−0.1)	49.6 (−2.7)	47.2 (−1.1)	46.3 (−4.1)*
uniform store, budget-constrained (streaming)	34.0 (−4.8)	35.2 (−5.6)	32.8 (−4.0)	38.5 (−10.1)	32.3 (−4.0)	31.0 (−8.5)	50.8 (−0.4)	49.5 (−2.8)	47.0 (−1.3)	46.1 (−4.3)*
flat PEMF (StreamForest's memory)	33.2 (−5.6)*	34.0 (−6.8)	32.4 (−4.4)	43.1 (−5.5)	33.7 (−2.7)	30.2 (−9.3)*	53.3 (+2.0)*	52.9 (+0.6)	48.9 (+0.6)	49.1 (−1.2)
long-store policy under its own adapter — each policy read by an adapter trained on its own memory states with the identical recipe										
uniform store, unbounded (budget applied at read time)	33.0 (−5.8)*	36.8 (−4.0)	29.2 (−7.6)*	33.0 (−15.6)*	34.0 (−2.3)	34.9 (−4.7)	46.9 (−4.4)*	45.3 (−7.0)*	43.9 (−4.4)*	42.8 (−7.6)*
uniform store, budget-constrained (streaming)	34.2 (−4.6)*	34.4 (−6.4)*	34.0 (−2.8)	51.4 (+2.8)	35.3 (−1.0)	37.2 (−2.3)	47.2 (−4.0)*	45.2 (−7.1)*	44.5 (−3.9)*	44.9 (−5.5)*
flat PEMF (StreamForest's memory)	32.8 (−6.0)*	36.8 (−4.0)	28.8 (−8.0)*	44.0 (−4.6)	37.7 (+1.3)	34.9 (−4.7)	48.0 (−3.3)*	49.8 (−2.6)	45.1 (−3.3)*	47.3 (−3.0)*
tree structure — the full system opens a new tree at every accepted switch (no cap; 32 trees per EgoLife week)										
single tree (flat PEMF memory, ConcurTree's reader)	35.4 (−3.4)*	34.8 (−6.0)*	36.0 (−0.8)	42.2 (−6.4)	36.3 (+0.0)	38.0 (−1.6)	51.6 (+0.3)	53.4 (+1.1)	48.1 (−0.2)	50.4 (+0.0)
at most 4 trees (further switches continue the current tree)	36.0 (−2.8)*	36.8 (−4.0)*	35.2 (−1.6)	43.1 (−5.5)	35.3 (−1.0)	38.8 (−0.8)	50.7 (−0.6)	52.0 (−0.4)	47.4 (−0.9)*	49.4 (−0.9)
at most 8 trees	36.8 (−2.0)	36.8 (−4.0)*	36.8 (+0.0)	46.8 (−1.8)	36.3 (+0.0)	38.0 (−1.6)	50.5 (−0.8)	51.8 (−0.5)	47.4 (−0.9)*	49.6 (−0.8)
routing — continue the current tree unless the visual switch gate fires; at a switch an LLM judge chooses return vs. new tree										
no exception policy (every event to its most similar tree)	36.4 (−2.4)	36.4 (−4.4)*	36.4 (−0.4)	44.0 (−4.6)	36.3 (+0.0)	38.0 (−1.6)	50.5 (−0.8)	51.5 (−0.9)	47.4 (−0.9)*	49.1 (−1.3)
no LLM link judge (caption-noun overlap instead)	36.0 (−2.8)*	37.2 (−3.6)*	34.8 (−2.0)	46.8 (−1.8)	36.0 (−0.3)	39.5 (+0.0)	51.3 (+0.1)	53.4 (+1.1)	48.0 (−0.4)	51.0 (+0.7)
compaction under the quota — merge the adjacent pair with the lowest penalty of dissimilarity, age and prior merges										
FIFO compaction (oldest pair)	37.4 (−1.4)	40.0 (−0.8)	34.8 (−2.0)	42.2 (−6.4)	36.3 (+0.0)	40.3 (+0.8)	51.1 (−0.2)	52.5 (+0.1)	48.0 (−0.4)	49.9 (−0.5)
uniform compaction (oldest pair of the largest tree)	38.0 (−0.8)	40.4 (−0.4)	35.6 (−1.2)	43.1 (−5.5)	36.3 (+0.0)	38.8 (−0.8)	51.0 (−0.2)	52.7 (+0.4)	48.0 (−0.3)	50.0 (−0.4)
similarity-merge compaction (most similar pair)	37.6 (−1.2)	38.8 (−2.0)	36.4 (−0.4)	45.9 (−2.8)	35.7 (−0.7)	38.8 (−0.8)	50.7 (−0.6)	51.6 (−0.7)	47.7 (−0.7)	49.4 (−0.9)
serialization — the reader sees the memory tree by tree, events chronological within a tree										
chronological serialization (tree membership hidden)	39.2 (+0.4)	40.4 (−0.4)	38.0 (+1.2)	49.5 (+0.9)	35.7 (−0.7)	39.5 (+0.0)	50.8 (−0.5)	52.6 (+0.2)	47.9 (−0.4)	50.7 (+0.3)
full system (ConcurTree + fine-tuned reader)	**38.8**	**40.8**	**36.8**	**48.6**	**36.3**	**39.5**	**51.3**	**52.3**	**48.3**	**50.4**

Parentheses show changes relative to the full system.
Stars denote paired McNemar significance versus full system, $p < 0.05$.

Table 4: **EgoMemReason** (hidden test set, live leaderboard; chance = 16.8%).

System	Overall	Rank	Setting
ConcurTree + ft (champion)	30.4	#10[a]	streaming, causal, 4096-tok, 8B
Flat PEMF + ft (champion)	31.0	#10[a]	streaming, causal, 4096-tok, 8B
ConcurTree + frozen	31.2	#10[a]	streaming, causal, 4096-tok, 8B
Qwen-3-VL-8B (our backbone)	29.6	#12	offline, 1024f/question
Random	16.8	—	chance

[a] all three of our rows are statistically indistinguishable on hidden test set.

A.1.2 EgoMemReason Leaderboard Results and Scoring

We additionally report results on the EgoMemReason dataset (Wang et al., 2026). However, we do not report these thoroughly in the main paper because it is difficult to perform rigorous comparisons on a leaderboard hidden test set. We do not outperform competing models, nor is our model comparable to these other models. Regarding visual token budget, the EgoMemReason leaderboard's non-agentic MLLM rows re-sample 1,024 frames from the full video history at every question, with no shared or reused encoding across questions. Using our own pipeline's visual encoding rate (784 tokens per 8 sampled frames, Section 3.1, i.e. 98 tokens/frame) as a proxy for the same Qwen-VL backbone, 1,024 sampled frames correspond to approximately $1{,}024 \times 98 \approx 100{,}352$ visual tokens per question. Our system instead answers from a fixed budget built once and reused for every question over the entire week: 784 ("now") $+ 384$ ("short") $+ 4{,}096$ (ConcurTree long-term memory) $= 5{,}264$ tokens (Section 3.2). This gives a ratio of $100{,}352/5{,}264 \approx 19.1\times$ fewer visual tokens per question. We report results on EgoMemReason in Table 4.

A.1.3 Efficiency and Inference Cost

Efficiency. Figure 4 compares accuracy with query-time visual-token exposure. Increasing exposure does not consistently improve performance: the larger native configurations of HERMES, InfiniPot-V, and Flash-VStream attend to substantially more visual tokens than their matched configurations without corresponding accuracy gains. ConcurTree attends to at most 5,264 visual tokens per query while matching the accuracy reported above. Moreover, reducing its long-term memory budget from 4,096 to 1,024 tokens still yields 35.8% overall and 40.4% on $>$24h evidence. These results show that greater query-time exposure alone does not explain long-horizon performance; how evidence is retained under the memory budget also matters.

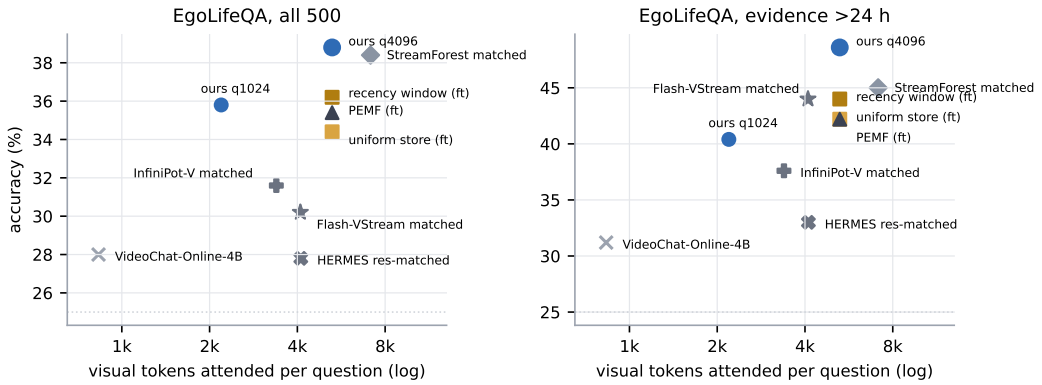

Figure 4: **Accuracy versus query-time visual-token exposure on EgoLifeQA.** Results are shown for the full test set (left) and questions with evidence older than 24 hours (right). Filled circles denote ConcurTree with 1,024- and 4,096-token long-term memory budgets.

Table 5: **Computational cost** (seconds per question).

Model	EgoLongQA	EgoLifeQA	Ego-R1 [a]
Ours — Qwen3 generation (uncalibrated)			
ConcurTree + ft answerer (champion) [b]	9.0	13.2	103.0
Flat PEMF + ft (ablation) [b]	—	10.5	91.3
ConcurTree + frozen answerer	13.7	12.5	104.9
Streaming tier, persistent memory			
StreamForest-persistent (native budget)	—	8.3	77.6
Offline-replay tier, per-question re-ingest			
StreamForest-Qwen2-7B (128 frames)	—	127.7	129.7

[a] Single-pass rows (ours, StreamForest-persistent): n=275 (the manual-A1 duplicate subset is excluded). Offline-replay baselines: full n=300.
[b] The fine-tuned answerer outputs a single answer letter, while the frozen base generates longer responses.

Computational cost. Overall, our online model is faster than offline models (Table 5). Cost per question is measured in seconds (single H100-class GPU), computed as total time (including ingest) divided by questions answered. Per-question baselines re-ingest frames for every question, so this is their true unit cost; online systems (ours and persistent StreamForest) ingest the stream once, so the same figure overstates their marginal answering cost.

A.1.4 External Results

Table 6 compares ConcurTree to externally published runs from other models. These other models are proprietary/API models or offline agentic models with tool calls. None of these are directly comparable to our own model, but are reported here for completeness. We do not outperform all of these models, because they operate under looser constraints — unbounded memory, frame retrieval, larger backbones, or audio data access.

A.2 Methodology Details

A.2.1 Event Segmentation

We use a SigLIP encoder (Zhai et al., 2023) to create embeddings for the causally processed frames. We then use these embeddings ϕ_t to calculate distances d_t, which are then used to calculate a z-score z_t to place event boundaries based on a z-score test (Equation 2). Based on preliminary experiments, a trained boundary model transferred poorly, while this untrained simple statistical detector matched

Table 6: **External runs compared with ConcurTree**.

System	Setting	EgoLifeQA	Ego-R1-Bench	TeleEgo P1
Ours — Qwen3 generation				
ConcurTree + ft answerer	online	38.8	36.3	53.0
Offline agentic / frontier tier (not directly comparable)				
MAGIC-Video (120B)	offline	67.6[a]	64.7[b]	—
EGAgent	offline	57.5	—	—
Ego-R1 agent	offline	—	46.0	—
Gemini-2.5-Pro	API	46.8	—	45.55
EgoGraph (GPT-4o)	offline	45.8	41.3	—

[a] 500 question subset. [b] A1-50 block score.
(Yang et al., 2026; Tian et al., 2025; Yan et al., 2025; Li et al., 2026a; Sun et al., 2026)

the boundary model's in-domain accuracy, so we keep our detection method training-free. Each identified event is encoded using the shared Qwen-VL backbone from 8 frames sampled uniformly within the span of the event, so each event is encoded at 784 visual tokens. The same SigLIP stream also yields a per-event switch score, the cosine distance between consecutive events' mean-pooled span features (Equation 1). The pooled features exist only to compute switch scores: the encoder retains just $\bar{p}_{e-1}$, and no pooled feature is stored in the memory.

$$d_t = 1 - \cos\big(\phi_t,\, \phi_{t-1}\big), \qquad z_t = \frac{d_t - \mu(\mathcal{Q}_t)}{\sigma(\mathcal{Q}_t)}, \qquad \text{boundary at } t \iff z_t > \tau_{\text{seg}} = 0.5, \quad (2)$$

where ϕ_t is the SigLIP feature of frame t, $\mathcal{Q}_t$ is a FIFO queue of the last $\Delta = 21$ distances (excluding d_t itself, while distances at flagged boundaries are included), $\mu(\cdot), \sigma(\cdot)$ are the queue's mean and standard deviation (no boundaries are set until the queue holds 5 distances). The boundary detector is a change detector, not an outlier test. $\tau_{\text{seg}} = 0.5$ was selected through development sets, with the span bounds of 4–16 frames regularizing event granularity.

A.2.2 Return Linking LLM Judge

We employ a LLM judge that compares the flagged event node to the previous active and dormant threads. Although the number of candidate threads that we compare to is not explicitly capped, it is structurally bounded since the number of trees can never exceed the node capacity $B/64 = 64$. In practice, the forest stays much smaller. The LLM judge is used in the following way:

$$p_T = \frac{P(\text{Yes})}{P(\text{Yes}) + P(\text{No})}, \qquad T^* = \arg\max_{T \in \mathcal{D}} p_T, \qquad \text{accept } T^* \iff p_{T^*} \geq \tau_{\text{link}}, \quad (3)$$

where $P(\text{Yes})$ and $P(\text{No})$ are from the backbone's first-token probabilities (softmax over full vocab) when asked whether the current event's caption and dormant thread T's most recent caption describe the same ongoing activity. This raw probability was discovered to be skewed toward zero, so we set $\tau_{\text{link}} = 0.01$, calibrated for precision-recall balance on held-out participants of the tuning domain. $\mathcal{D}$ is the set of dormant threads. If no candidate clears the threshold, a new tree is created.

A.2.3 Frame Ingestion Rates

Frame-ingestion setting per benchmark shown in Table 7. The segmenter's parameters are all defined in frame counts, not wall-clock time. Event spans are bounded at 4–16 frames, and the event detection window holds the last 21 distances (Equation 2). The same frame-count bounds therefore cover very different real-world durations depending on each benchmark's ingestion cadence: at TeleEgo's 1 fps, an event spans 4–16 seconds, while at the week-scale, the same 4–16 frames span roughly 2–8 minutes at $\sim$1-frame-per-30-seconds.

Table 7: **Frame ingestion settings.**

Benchmark	Ingest rule	Frames/stream
EgoLifeQA / Ego-R1 / EgoMemReason	One mid-clip frame per $\sim$30 s recorded clip; single causal pass	6,266
TeleEgo	Official per-second harness	One / recorded sec
EgoLongQA (dev)	1 fps, capped at 128 frames/video; uniform subsampling beyond 128 s	≤ 128
HD-EPIC (dev)	Same capped rule over recordings	128

A.2.4 McNemar Significance Tests

We use McNemar's test (McNemar, 1947) whenever we compare two models. This makes per-question outcomes paired rather than independent, so a question's inherent difficulty affects the conditions of both models identically. This is why we use a paired test rather than an unpaired test. McNemar's test conditions on the 2×2 table of paired outcomes (both correct, both incorrect, either one correct only), and tests the discordant cells.

We compare two systems. Let b be the number of questions the first system answers correctly and the second system answers incorrectly, and c the reverse (the number of questions the second system answers correct and the first does not). We report the exact two-sided binomial p-value on these discordant counts,

$$p = \min\left(1,\ 2\sum_{i=0}^{\min(b,c)} \binom{n}{i}\left(\frac{1}{2}\right)^{n}\right), \qquad n = b + c, \tag{4}$$

rather than the chi-square approximation, since several of our comparisons have small discordant counts where the asymptotic test is unreliable. We call a difference significant at $p < 0.05$ (under a two-sided test).

A.3 Training and Tuning Details

Training data composition and effects. Table 8 details the ablations on the training data composition, holding rank constant at $r = 16$ except where specified. No single mix in this table dominates. We also detail the different training data compositions:

- *v3 mix*: a 50:50 egocentric/general-domain recipe
- *EgoIT*: egocentric-only instruction data with no general-domain data
- *OnlineIT*: StreamForest's public streaming-format temporal instruction corpus
- *A-bias records*: free-form answers with a spurious leading "A" token, later rebalanced
- *saturated memory*: training on caches compressed to the deployed 4,096-token budget rather than fresh, never-saturated caches.

Both Table 8 and Table 9 also report the column "Content-Free" max-letter, which is the result of asking the model content-free questions and checking which of the multiple choice answers it gives. This is to isolate possible issues in the fine-tuning data, which was verified by a fine-tuning set corruption that was discovered to have caused a letter bias.

Answerer Hyperparameter Tuning. Table 9 details the hyperparameter tuning sweep over rank, learning rate, epoch count, and deep-evidence mix percentage of training data on the final training mix (Section 4.1). "Rank 128, length-debiased mix" trains on the same mix as the champion, but with an answer-length bias corrected (Section 4.1). Rank $r = 128$ at $lr = 1e-4$ is the sweet spot: $r = 64$ undershoots, $r = 256$ collapses, a second epoch overfits, and a lower learning rate regresses at both ranks tested.

We write q1024/q4096 as shorthand for memory budget B=1,024/4,096 (Table 9). The memory budget policy is detailed in Section 3.2.

Table 8: **Training-mix ablations** (ConcurTree memory fixed; rank 16 unless noted).

Training mix	HD-EPIC	EgoLongQA	Content-Free (max-letter)	EgoLifeQA (q1024)	EgoLifeQA (q4096)	>24h (q4096)
v3 mix, ego:general 50:50	34.10 / 35.86	64.57	.437 (A)	34.0	35.8	∼40.4
EgoIT only, fixed cache	35.39 / 36.99	66.14	.426 (A)	33.0	34.0	38.5
OnlineIT only	34.45 / 32.18	68.86	.410 (A)	29.4	—	—
v3 + OnlineIT blend	34.94 / 39.77	69.14	.303 (A)	30.6	30.8	—
EgoIT only, rank 64	—	56.14	.545 (A) ×	34.4	35.2	40.4
EgoIT only, rank 64, A-bias records removed	—	66.00	.306 (D), flat	34.0	32.8	42.2
EgoIT only, rank 64, continued-FT on saturated memory	—	—	—	—	33.6	33.9
no adapter (frozen reader)	36.81 / 35.34	65.14	—	30.0	34.0	43.1

× = fine-tuning set corruption discovered (letter bias).

Table 9: **Hyperparameter sweep for the answerer head** (ConcurTree memory fixed).

Configuration	HD-EPIC	EgoLongQA	Content-Free (max-letter)	EgoLifeQA (q1024)	EgoLifeQA (q4096)	>24h (q4096)
rank 64, lr 1e-4	—	70.53	.549 (A) ×	33.0	34.4	33.9
rank 128, lr 1e-4 (champion)	**34.61 / 39.17**	**65.57**	**.320 (B)**	**35.8**	**38.8**	**48.6**
rank 64, 10% deep-evidence	—	55.36 ×	.429 (A)	32.0	31.0	34.9
rank 64, lr 5e-5	—	68.00	.371 (A)	34.0	35.0	41.3
rank 128, lr 5e-5	—	62.29	.441 (A)	32.0	32.6	38.5
rank 256, lr 1e-4	—	50.57 ×	.363 (B)	35.2	37.0	40.4
rank 128, 2 epochs	—	43.14 ×	.391 (A)	30.0	29.8	39.4
rank 128, length-debiased mix	—	64.00	.262 (C), best	32.4	33.8	38.5
StreamForest bar	31.78 / 26.69	—	—	—	38.4–38.6	45.0

× = fine-tuning set corruption.

A.4 Dataset Details

A.4.1 Evaluation and Development Datasets

For our evaluation held-out benchmarks (Table 10), we feed questions at their annotated query times under the causal replay setup. For the development benchmarks (HD-EPIC and EgoLongQA), which carry no query timestamps, questions are posed after the final frame. Further details on each evaluation dataset follow:

- **EgoLifeQA** (Yang et al., 2026) spans a 44-hour, 7-day egocentric stream, and questions were published for one out of six participants. Question evidence may lie up to six days before the query, and 109 of 500 questions require evidence older than 24 hours, which is where memory is most heavily compressed. Activities recur across days as participants return to interrupted routines, but this is not labeled, so thread structure can only be assessed indirectly through QA accuracy.
- **Ego-R1-Bench** (Tian et al., 2025), draws on the same EgoLife footage, but covers all six participants (vs. EgoLifeQA's questions on a single participant). Half the questions carry machine-readable evidence timestamps; among these, the median query-to-evidence gap

Table 10: **Evaluation benchmarks.**

Benchmark	# Questions (format, chance)	Evidence horizon	Protocol
EgoLifeQA	500 (4-way, 25%)	∼6 d; 109/500 >24 h	causal replay
Ego-R1-Bench	300 (4-way, 25%)	142 h; 38% >24 h†	causal replay
TeleEgo	2,850 (mixed, ∼27%)	within-day	**native online**

is 14.3 hours and around 129 questions exceed 24 hours (max ∼6 days), and the reported >24 h proportion is computed over this timestamped half only.

- **EgoMemReason** (Wang et al., 2026) holds its test answers hidden, and also maintains a HuggingFace leaderboard. Unlike EgoLifeQA or Ego-R1-Bench, it mixes option counts from 4 to 10, giving a chance level of 16.8% (Appendix A.1.2).
- **TeleEgo** (Yan et al., 2025) has its questions fired at evidence timestamps with a 5-second decision window. Of its 3,291 total questions, we evaluate on only the video-only MCQ subsets of TeleEgo, which is 2,850 questions: 1,743 4-way single-choice (25%), 614 binary (50%), and 493 exact-set multi-select ($\sim$ 7%), for an aggregate chance level of $\sim$ 27%. Individual days span several distinct activities that interleave, though the dataset provides no thread annotation (Appendix A.4.3).

We also report results on our development/tuning benchmarks:

- **EgoLongQA** (Tran et al., 2026): 700 4-way MCQs from Meta's Wearable-AI Workshop, posed over a single video. Recordings are single-outing and linear, with no suspended activities to resume, so this surface cannot exercise return-linking and we expect parity with sequential-memory baselines rather than a gain.
- **HD-EPIC** (Perrett et al., 2025): 26,550 5-way MCQs over 41 hours of unscripted kitchen video across 9 kitchens, covering 69 recipes and 59K fine-grained actions. Sessions contain interleaved goals such as cooking, cleaning, and storing. However, we only use a 1,330 slice for development.

A.4.2 Baseline Setup

All baselines (primarily Table 1, and excluding Table 6) are run through our own evaluation pipeline under the causal-replay setup of Appendix A.2.3, but using each baseline's own released checkpoints and code. Regardless of baseline, we maintain two constraints: 1) we keep responses at a maximum of 16 new tokens generated (which corresponds to the desired output, one single MCQ answer letter), and 2) we use scaled dot-product attention instead of any attention variants. Everything else (memory mechanisms, individual memory budgets, hyperparameters and thresholds, prompt template, and decoding) are kept consistent with the original benchmark's released code, unless otherwise specified. We mark the frame resolution and token budget changes in our comparisons as "matched", to denote experiments that match ConcurTree's memory budget in order to isolate the effect of memory budget on the memory structure.

A.4.3 TeleEgo Benchmark Rules

TeleEgo (Yan et al., 2025) is the only benchmark that we use with a natively online evaluation method. It uses per-second ingest, and questions are asked during the stream at their own timestamps with a 5-second decision window. An additional metric, Memory Persistence Time (MPT), also re-queries the same question every 60 seconds up to ten times while the model continues answering correctly, which measures whether a correct answer survives further compression. We evaluate only the MCQ subsets of TeleEgo (2,850/3,291 questions): 1,743 4-way single-choice (25%), 614 binary (50%), and 493 exact-set multi-select (∼7%) give an aggregate chance level of ∼27%. Individual days span several interleaving activities, but the dataset does not provide annotation for distinct activities.

Our own evaluations report the mean across five legs of the dataset (P1-P5), shown in Table 1. Comparisons to other models (including externally published results) report P1 only, matching those

systems' published practice. The 5-second window is TeleEgo's stated real-time requirement for a deployable streaming system; the official TeleEgo system scores every answer regardless of timing, so meeting the window is just a property of a model, not something the benchmark explicitly enforces.