GROVE: Growing and Reasoning over Temporally Stratified Memory from Streaming Video Experience

Sitong Gong[1,2], Caixin Kang[2,3]*, Tianyu Yan[1,2]*, Guo Chen[4], Bo Zheng[2], Kaipeng Zhang[2], Yunzhi Zhuge[1], Xiang Ruan[1], Huchuan Lu[1], Yifei Huang[2,3]†

[1]Dalian University of Technology [2]Alaya Lab [3]The University of Tokyo [4]NVIDIA
stgong@mail.dlut.edu.cn, hyf015@gmail.com

Abstract

A wearable assistant should both answer questions about its visual history and recognize when that history is useful to the present situation. Existing video-memory systems primarily support question-conditioned recall, whereas proactive assistants typically use separate memory and control mechanisms. We introduce GROVE, a training-free framework that supports both behaviors with one memory grown causally from a continuous video stream. GROVE retains fine-grained perceptual evidence and incrementally consolidates it into time-stamped moments, coherent episodes, and recurring cross-day patterns. Each stratum is paired with a scale-native retrieval skill for locating an observation, replaying an activity, or traversing long-range regularities. Reactive QA and proactive assistance share this memory and access interface, differing in whether retrieval is initiated by a user query or the current situation. Across multiple benchmarks including the challenging MM-lifelong and EgoServe, GROVE achieves the best results among the compared methods. Controlled ablations show that the temporal strata and their access skills are complementary, with patterns providing the largest benefit when evidence spans multiple days. Code will be available at https://github.com/SitongGong/GROVE.

Introduction

A wearable assistant observes the world from its user's perspective for hours, days, or even weeks. To remain helpful over this growing history, it must support two complementary forms of assistance. When a user asks, "Where did I leave my keys yesterday?", the assistant must *pull* relevant evidence from the past. When the user is about to repeat a mistake, it should instead *push* timely assistance without being asked. Reactive question answering and proactive assistance therefore depend on the same underlying capability: turning an open-ended video stream into a memory that remains faithful, searchable, and actionable over time.

Recent video-memory systems already address many difficulties of long and streaming video. They construct online event memories, organize experience hierarchically, and use iterative or agentic retrieval to locate sparse evidence (Xie et al. 2026; Long et al. 2025; Li, Numan, and Steed 2026; Chen et al. 2026a; Li et al. 2026; Liang et al. 2026). These advances change how efficiently the past is stored and searched, but most retain the same *query-first* contract: an explicit request determines what should be retrieved. Proactive assistants instead use the current situation to decide whether intervention is warranted (Zhang et al. 2025b; Gong et al. 2026), but their memory and control mechanisms are designed primarily for service triggering rather than as a common interface for open-ended recall. The missing capability is therefore not merely another memory level or a stronger search procedure. It is a single streaming memory that can be accessed from either of two starting signals: a question seeking past evidence or the unfolding situation indicating that past experience may now be useful.

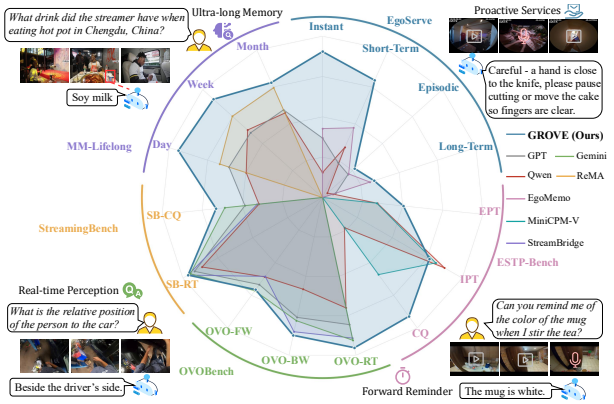


Figure 1: Performance of GROVE across five video understanding benchmarks. With a temporally structured memory, GROVE answers questions spanning ultra-long horizons, stays grounded in real-time perception, and decides when to deliver proactive services or forward reminders.

Our key idea is to make temporal scale the interface between what memory stores and how an agent reads it. Evidence required by a video assistant varies qualitatively with scale: an object count or state change may last seconds, an activity may unfold over minutes, and a routine may emerge only after related activities recur across days. Treating these observations as homogeneous records forces one general search operation to recover fundamentally different kinds of evidence. In GROVE, each scale instead represents a distinct kind of information and exposes an access operation suited to that information. The hierarchy is therefore not used only to compress the video; it determines how the stored experience can be reasoned over. Long-range regularities are explicitly constructed because they are absent from any individual

*These authors contributed equally.
†Corresponding author.

frame or event, and all memory updates are causal so that the memory remains usable while the stream is still arriving.

We realize this idea in **GROVE** (**G**rowing and **R**easoning **O**ver Streaming **V**ideo **E**xperience), a training-free framework that grows a temporally stratified memory online. Each arriving video window is processed through two complementary perception channels: a narrative caption and a structured perceptual registry containing entities, attributes, explicit counts, actions, and on-screen text. These observations are consolidated into time-stamped moments, coherent episodes, and cross-episode patterns capturing recurring behavior. GROVE exposes four scale-native retrieval skills (*Perception Lookup*, *Moment Recall*, *Episode Replay*, and *Pattern Traversal*) that enter the memory at increasing temporal scope. A multi-round agent composes these skills and accumulates non-redundant evidence.

The same memory and skill library support two control policies. For *reactive* question answering, the user query initiates a decide–retrieve–answer loop in which the agent selects the type and scale of evidence to retrieve. For *proactive* assistance, either a forward-looking request or the current perceptual state initiates retrieval of related moments, episodes, and patterns before the agent decides whether and when to act. The two policies read the same constructed past through the same scale-native interface.

We evaluate GROVE on MM-Lifelong (Chen et al. 2026c), OVO-Bench (Niu et al. 2025), StreamingBench (Lin et al. 2026), ESTP-Bench (Zhang et al. 2025b), and EgoServe (Gong et al. 2026), covering lifelong question answering, online video understanding, forward assistance, and query-free proactive services. GROVE achieves the best overall results among the compared methods on all three benchmarks. Controlled ablations show that both the temporal organization of memory andits scale-aware retrieval interface contribute to these gains. The full design also improves retrieval efficiency, requiring fewer reasoning rounds and lower latency than a flat caption index.

Our contributions are threefold:

- We formulate long-video memory as a shared substrate for reactive question answering and proactive assistance.
- We introduce GROVE, a **causal, temporally stratified memory** paired with scale-native retrieval skills.
- Across 5 benchmarks, we demonstrate the effectiveness of this shared memory and retrieval interface, with ablations validating both its memory structure and retrieval design.

Related Works

Streaming Video Understanding. Offline long-video models extend the context window or compress an entire clip before answering (Chen et al. 2025; Song et al. 2024; Azad, Vineet, and Rawat 2025), an assumption that fails when the video is still arriving. Online video LLMs instead process each window as it comes and learn when to respond (Chen et al. 2024a; Wang et al. 2024b; Zhang et al. 2024); recent systems refine this with disentangled perception–decision loops (Qian et al. 2025), redundancy-aware token pruning (Yao et al. 2025), persistent event memory (Zeng et al. 2026; Wang et al. 2026), and visual instruction feedback (Fu et al. 2025). Progress in this setting is measured by OVO-Bench (Niu et al. 2025), which probes real-time perception, backward tracing and forward active responding, and Streaming-Bench (Lin et al. 2026), which adds contextual understanding; both are part of our evaluation. Such models maintain a compact state for the immediate present, whereas we retain a structured memory that stays queryable over days.

Memory-based Video Agents. A complementary, training-free line of work equips a frozen MLLM with an external memory and retrieves from it on demand. Retrieval-augmented pipelines index captions or clips of the whole video (Ren et al. 2026; Luo et al. 2026), while agentic systems add tool use and multi-round search over that index (Zhang et al. 2026; Liu et al. 2025b; Yang et al. 2026). Closest to us are memories with explicit structure: VAM commits deduplicated moments into an age-layered store (Li, Numan, and Steed 2026), M3-Agent maintains entity-centric episodic and semantic memory from audio-visual streams (Long et al. 2025), StreamRAG segments events online for query-adaptive retrieval (Xie et al. 2026), and MemDreamer (Chen et al. 2026a) and MAGIC-Video (Li et al. 2026) pair hierarchical or multimodal memory graphs with an agentic tool bank; ReMA shows that dynamic memory management becomes essential at the lifelong horizon (Chen et al. 2026c). Our stratification echoes the classic episodic–semantic distinction in human memory (Tulving 1972), but differs in three respects: the memory is grown causally online, it is stratified by temporal scale with one retrieval skill per stratum, and it serves proactive assistance in addition to question answering.

Proactive Assistants. Proactivity was first studied in text-only dialogue, where an agent must decide not only what to say but when to speak unprompted (Liu et al. 2025a). In egocentric video it becomes a streaming decision problem: ProAssist generates task-guidance dialogue from the stream (Zhang et al. 2025a), Eyes Wide Open times synchronized proactive answers and introduces ESTP-Bench (Zhang et al. 2025b), and Vinci2 defines the service-triggering protocol over EgoServe (Gong et al. 2026). These systems tie the trigger decision to current perception; none is built on a memory that also serves reactive reasoning, and none exposes the cross-day structure that long-term services require.

Methodology

Problem setting. We represent a streaming video as consecutive windows $V = \{w_1, w_2, \ldots, w_t\}$. At time t, the system can access only the observed prefix and its memory $\mathcal{M}_{\le t}$. GROVE supports two tasks. In *reactive question answering*, a query q requests evidence from the observed history. In *proactive assistance*, the system receives either no request or a forward-looking request q^+ and must decide whether the current situation warrants a response:

$$y = \pi(q,\, \mathcal{M}_{\le t})\,, \qquad s_t = \pi\big(q^+,\, \mathcal{M}_{\le t}\big)\,, \qquad (1)$$

where π is the reasoning agent, q^+ may be absent, and s_t is either empty or a response anchored at a second $\tau \le t$.

Overview. GROVE decouples the system into streaming memory construction and streaming memory access by agentic reasoning (Fig. 2). As video arrives, a perception model

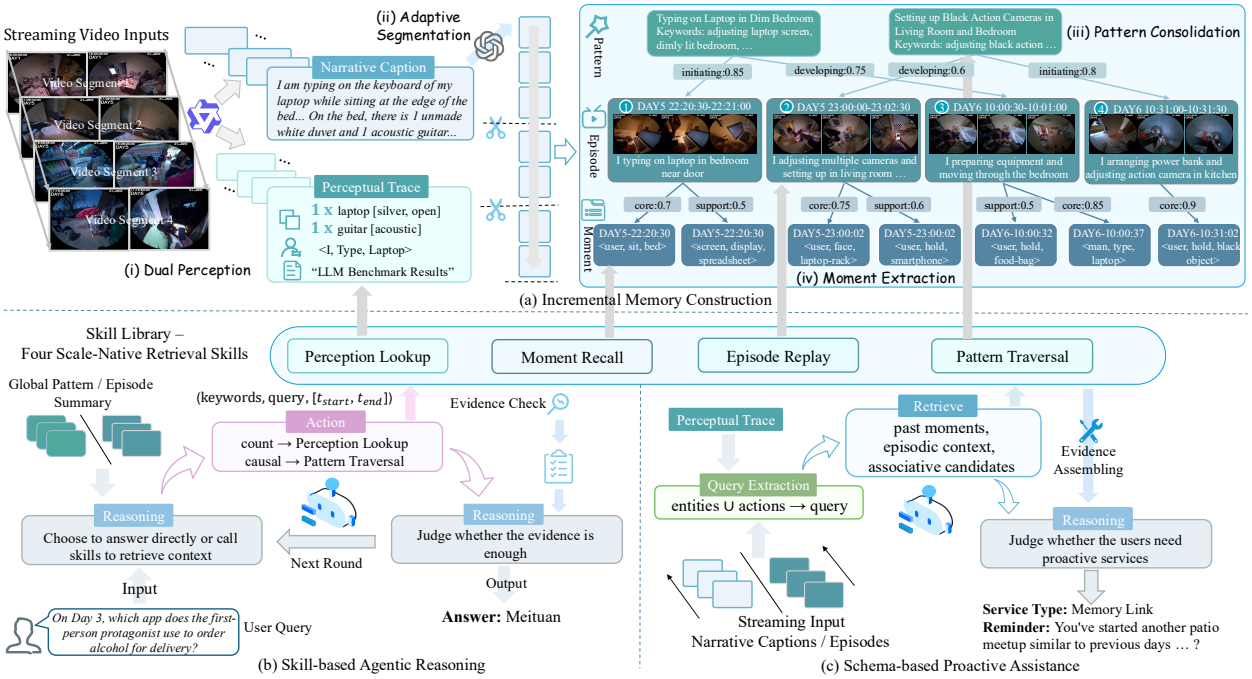


Figure 2: Overview of GROVE. (a) Memory construction proceeds in four incremental steps: dual perception, adaptive segmentation, pattern consolidation and moment extraction. (b) Four retrieval skills expose the strata to an agent that iteratively judges whether the retrieved evidence suffices to answer a user query. (c) Given no user query, the agent instead invokes the same skills to recall contextual clues from the memory, and decides on that evidence whether a proactive assistance is warranted.

and a lightweight consolidation model grow a textual memory stratified by temporal scale:

$$\mathcal{M}_{\leq t} = \{\mathcal{R},\ \mathcal{F},\ \mathcal{E},\ \mathcal{P}\}, \tag{2}$$

where $\mathcal{R}$ is the perceptual trace and $\mathcal{F}$, $\mathcal{E}$, and $\mathcal{P}$ contain MOMENTs, EPISODEs, and PATTERNs. Each stratum exposes one scale-native retrieval skill, forming a skill library $\mathcal{A} = \{a_1, \ldots, a_4\}$. Both tasks in Eq. 1 are then solved by the same agent over the same memory through the same skills: π invokes skills in $\mathcal{A}$ over multiple rounds until the retrieved evidence suffices, and never revisits raw frames.

Incremental Memory Construction

We build the stratified memory in four incremental steps. Each arriving window first undergoes dual perception, producing dense narrative captions and a perceptual trace entry. An adaptive segmenter groups the streaming captions into EPISODEs. When an EPISODE closes, it is atomized into time-stamped MOMENTs and consolidated with related, temporally non-contiguous EPISODEs into PATTERNs. Every operation updates the previous state $\mathcal{X}_{t^-}$ using only observations available by time t; no retrospective pass over future video is required. The resulting memory can therefore be queried while the stream is still arriving. The current window is immediately available through $\mathcal{R}$; MOMENTs, EPISODEs, and PATTERNs are updated only after the open EPISODE closes.

Dual perception. Each input window w_t is parsed by a VLM Φ into two complementary views $(c_t,\ r_t)$, where c_t contains timestamped captions of sampled frames and a short window-level summary. The perceptual trace entry r_t stores three types of timestamped evidence: object counts and attributes, subject–action–object events, and transcribed on-screen text. The perceptual trace supplements captions with explicit entities, counts, and text that narrative descriptions tend to blur. Both views are retained, $\mathcal{R}_t = \mathcal{R}_{t^-} \cup \{(c_t, r_t)\}$, yielding a time-ordered record of the observed stream.

Adaptive Episode Segmentation. The captions c_t are streamed into an LLM segmenter Seg that judges whether the ongoing activity continues:

$$b_t = \mathrm{Seg}\big(c_t,\ \tilde{e}_{t^-}\big) \in \{\textsc{extend},\ \textsc{cut}\}, \tag{3}$$

where $\tilde{e}_{t^-}$ denotes the EPISODE currently being accumulated. The segmenter matches its anchors against the arriving captions and remains conservative, so that ambiguous evidence keeps the activity open. On EXTEND the window is absorbed into the open unit and the activity keeps growing, until either a semantic change is detected or a maximum-length cap is reached. On CUT the unit is closed and appended, $\mathcal{E}_t = \mathcal{E}_{t^-} \cup \{e_t\}$, where the closed EPISODE e stores its start and end times together with two LLM-written texts: a brief summary of the activity, and a longer description that retains its fine-grained spatio-temporal details; a cut is also forced at a large recording gap or a day change. Episodes are therefore variable-length units aligned with activity boundaries rather than fixed slices, which keeps each unit semantically coherent and gives retrieval a meaningful span to land on.

Moment Extraction. Whenever an EPISODE closes, a lightweight consolidation model Ψ atomizes its dense perception into time-stamped MOMENTs $\mathcal{F}_t$, formulated as:

$$\{f_i\}_{i=1}^{n} = \Psi(e_t), \qquad \mathcal{F}_t = \mathcal{F}_{t^-} \cup \{f_i\}_{i=1}^{n}, \tag{4}$$

Table 1: The retrieval skill library of GROVE.

Skill	Reads	Best for
Perception Lookup	$\mathcal{R}$	exact counts, attributes, on-screen text at a moment
Moment Recall	$\mathcal{F}$	what/when facts by keyword and time range
Episode Replay	$\mathcal{E}$	locating and reading an activity segment
Pattern Traversal	$\mathcal{P}\to\mathcal{E}\to\mathcal{F}$	broad/causal/cross-period queries

where $\mathcal{F}_t$ contains all MOMENTs produced by time t. Each f_i pairs a timestamp within e_t with a minimal statement of an utterance, state change, or action, plus search keywords. The same pass creates a group edge from e_t to its MOMENTs and assigns each MOMENT a role (*core*, *context*, or *detail*) and a salience weight. The edge lets retrieval recover the surrounding EPISODE in one hop and favor core facts over peripheral detail.

Pattern Consolidation. Finally, an LLM matches the closed EPISODE against existing PATTERNs using its title, summary, and keywords:

$$p^* = \mathrm{Match}\big(e_t,\ \mathcal{P}_{t^-}\big), \tag{5}$$

where p^* is empty if no candidate accepts e_t. The stratum is then updated as

$$\mathcal{P}_t = \begin{cases} \big(\mathcal{P}_{t^-} \setminus \{p^*\}\big) \cup \{\mathrm{Upd}(p^*, e_t)\}, & p^* \text{ found}, \\ \mathcal{P}_{t^-} \cup \{\mathrm{New}(e_t)\}, & \text{otherwise}, \end{cases} \tag{6}$$

Here, Upd absorbs e_t and re-synthesizes the pattern title and summary from its episodic history, whereas New opens a candidate pattern. A PATTERN is stored as a group edge over temporally non-contiguous EPISODEs; after multiple EPISODEs have been assigned, it explicitly records recurrence, frequency, and typical time. This is the only stratum that represents regularities absent from any single EPISODE.

Scale-Native Retrieval Skills

GROVE exposes four *scale-native* retrieval skills, one per stratum (Table 1). Because the strata store different evidence at different granularities, each skill enters the memory at the requested scale instead of ranking all record types in one flat index. A skill may provide several calls, but all calls share a source stratum and return granularity. They accept keywords or a semantic query and, when available, a time range. Together, the four skills define the action space $\mathcal{A}$. A benchmark may expose only the subset required by its task.

Perception Lookup. This skill reads $\mathcal{R}$ at a requested time and returns nearby captions or structured entity counts, attributes, events, and on-screen text. It serves fine-grained questions and grounds real-time queries in the present scene.

Moment Recall. This skill searches $\mathcal{F}$ by keyword and time span, returning matching time-stamped moments or the ordered sequence within an interval. It addresses what-happened-when questions without replaying an entire EPISODE.

Episode Replay. This skill searches EPISODE summaries or reads recent units in $\mathcal{E}$, then returns the selected activity's detailed description and span $[t_0, t_1]$. The span can constrain subsequent Perception Lookup or Moment Recall calls, providing a coarse-to-fine path.

Pattern Traversal. This skill follows group edges along $\mathcal{P}\to\mathcal{E}\to\mathcal{F}$, with each stage constraining the candidates at the next. It is useful when a broad or cross-period query does not specify a time range. At each stage, reciprocal rank fusion combines BM25 and dense semantic rankings so that both named entities and paraphrased concepts remain reachable. Traversal may also begin at $\mathcal{E}$ or $\mathcal{F}$ when $\mathcal{P}$ is unnecessary.

Retrieval over the Memory

Both modes in Eq. 1 access memory through $\mathcal{A}$. Every call respects the causal cutoff and exposes only memory constructed by the current time t. The policies differ in how retrieval is initiated and controlled.

Skill-based Agentic Reasoning. Given a reactive query (Fig. 2b), π runs a bounded loop of at most $T_{\max}$ rounds. At round k, the agent either answers from the accumulated evidence $\mathcal{C}_{k-1}$ or invokes a skill $a_k \in \mathcal{A}$. The returned evidence is deduplicated before being appended:

$$o_k = a_k(\mathcal{M}_{\le t}), \qquad \mathcal{C}_k = \mathcal{C}_{k-1} \cup (o_k \setminus \mathcal{C}_{k-1}). \tag{7}$$

Query cues guide the initial scale: counts and literal text favor Perception Lookup, specific events favor Moment Recall, activities favor Episode Replay, and broad cross-period questions favor Pattern Traversal. The agent may move coarse-to-fine across rounds and stops when the evidence is sufficient. For real-time questions, the current perceptual trace is included before retrieval.

Schema-based Proactive Assistance. Given a forward-looking query or no query at all (Fig. 2c), there is no past-oriented user query to route. Instead, π converts the entities and actions in the current perceptual trace into a retrieval query. A fixed schema then fills three evidence slots: preceding MOMENTs, the surrounding EPISODE, and associated PATTERNs that represent routines or earlier commitments. The assembled evidence is used to decide whether a service is needed and, if so, what to provide. When every EPISODE must be screened, scale-matched experts evaluate service families in parallel: immediate safety and guidance at the moment scale, reminders at the episode scale, and longer-term coaching at the pattern scale. A fired service is aligned to the perception grid for second-level timing.

Experiments

Experimental Setup

Benchmarks and Evaluation. We evlauate our method on 5 representative benchmarks. **EgoServe** (Gong et al. 2026) contains over 3,000 proactive service instances across ten service sub-types, measured with Macro-F1 and an LLM-judged quality score. **MM-Lifelong** (Chen et al. 2026c) comprises 181.1 hours of video across day-, week-, and month-scale splits, evaluating lifelong question answering under the official GPT-5 judge protocol. **OVO-Bench** (Niu et al. 2025)

Table 2: Proactive service performance on EgoServe with best results per column in **bold**.

Model	Instant		Short-term			Episodic		Long-term			Overall
	SA	TU	NSG	ER	RR	MR	TR	HC	ML	RO	
Qwen3-VL-Plus (Bai et al. 2025)	5.2	1.5	8.6	10.4	3.4	0.0	1.8	4.4	0.0	0.0	3.5
GPT-5-mini (Singh et al. 2025)	12.5	3.6	9.5	1.0	5.2	0.0	**9.4**	5.7	0.0	0.0	4.7
EgoMemo (Gong et al. 2026)	11.4	7.5	**24.7**	1.7	4.7	3.8	5.7	3.7	4.9	**11.8**	8.0
GROVE (Ours)	**19.7**	**20.2**	21.8	**23.8**	**6.8**	**5.9**	5.9	**8.6**	**6.5**	7.1	**12.6**

Table 3: Performance comparison on the MM-Lifelong.

Method	Month	Week	Day
Human	80.4	95.6	99.2
GPT-5 (Singh et al. 2025)	14.87	15.00	15.25
Qwen3-VL-235B-A22B (Bai et al. 2025)	14.33	15.63	12.44
Video-XL-2-8B (Qin et al. 2025)	9.07	12.00	9.00
Eagle-2.5-8B (Chen et al. 2026b)	6.10	7.00	8.25
Nemotron-v2-12B (Deshmukh et al. 2025)	9.63	11.00	7.25
VideoMind-7B (Liu et al. 2025b)	8.35	11.75	7.50
LongVT-7B (Yang et al. 2026)	7.54	9.75	7.00
DeepVideoDiscovery (Zhang et al. 2026)	10.57	9.02	10.25
ReMA (Chen et al. 2026c)	18.62	18.82	16.75
GROVE (ours)	**19.98**	**22.75**	**23.50**

evaluates online video comprehension across backward tracing, real-time perception, and forward active responding. **StreamingBench** (Lin et al. 2026) assesses real-time and contextual video understanding under a streaming constraint. **ESTP-Bench** (Zhang et al. 2025b) evaluates ego-proactive video understanding through temporally grounded questions timed at opportune moments, scored with the official *valid-ScoreF1* metric averaged per task type.

Implementation details. Perception and caption generation use Qwen3.5-35B-A3B (Team 2026), while memory consolidation uses GPT-4.1-mini. The reasoning backbone is GPT-5.2 for MM-Lifelong and GPT-5-mini for the other benchmarks. Local video processing runs on NVIDIA H200 GPUs. For MM-Lifelong and EgoServe, we sample one frame every 5s and cap episodes at 10min; for the remaining benchmarks, we sample one frame every 2s and cap episodes at 2min. Dataset-specific configurations and all construction and inference prompts are provided in the supplementary material.

Main Results

Egoserve. GROVE achieves the best overall macro-F1 on EgoServe (12.6), exceeding EgoMemo by 4.6 points and ranking first on seven of ten service subtypes (Table 2). Its largest gains occur in safety (19.7 vs. 12.5), tool use (20.2 vs. 7.5), and error recovery (23.8 vs. 1.7), which depend on fine-grained current and recent evidence. GROVE also improves habit coaching (8.6 vs. 3.7) and memory-linked reminders (6.5 vs. 4.9), where cross-episode context is useful.

MM-Lifelong. On MM-Lifelong (Table 3), GROVE gets the best score on all three horizons: 23.50 (day), 22.75 (week), and 19.98 (month), improving over the strongest agentic baseline ReMA by 6.75, 3.93, and 1.36 points, respectively. The gain is largest on the day split, where the perceptual trace and time-stamped moments preserve details such as counts, attributes, text, and action timing. The smaller but consistent gains at longer horizons indicate that this fine-grained evidence remains accessible after consolidation.

OVO-Bench and StreamingBench. On OVO-Bench and StreamingBench (Table 4), GROVE obtains the best OVO-Bench overall score (67.5) and StreamingBench average (65.1) among the compared models. It leads OVO real-time perception (76.2), while obtaining 69.9 on backward tracing and 56.5 on forward active responding. On StreamingBench, it achieves 53.1 on contextual understanding, 9.2 points above the strongest listed baseline, while remaining competitive on real-time understanding (77.0 vs. 77.3). These results show that retrieval over the observed history improves temporally contextual questions without sacrificing grounding in the current window.

ESTP-Bench. On ESTP-Bench (Table 5), GROVE achieves the best contextual-question score among the non-benchmark-specific comparison methods (41.0 vs. 26.6) and the best overall score (28.6 vs. 22.9). It also leads the explicit proactive track (22.7), but trails Qwen2-VL on implicit proactive tasks (34.4 vs. 39.3). The large contextual gain is consistent with GROVE's design: these questions benefit most from connecting the present to earlier events.

Ablations

We conduct ablations on MM-Lifelong day/week and EgoServe. MM-Lifelong tests reactive reasoning over long-horizon memory, whereas EgoServe tests proactive assistance. To control evaluation cost, MM-Lifelong ablations use GPT-5-mini rather than the GPT-5.2 backbone in Table 3; all variants within an ablation share the same backbone and retrieval budget.

Memory structure. Table 6 replaces each stratum in turn. Removing the EPISODE layer causes the largest day-scale drop (18.75→14.57), while flattening the hierarchy reduces performance to 14.75/15.50 on day/week despite preserving an equal volume of captions. The effect is scale-dependent: removing PATTERN slightly improves the day split (18.75→19.90), where cross-day recurrence is unnecessary, but sharply reduces the week split (19.75→13.00). On EgoServe, removing MOMENT causes the largest single-stratum decline (12.62→11.77).

Retrieval skills. Table 7 deletes one skill at a time while its content remains reachable. Disabling retrieval entirely collapses accuracy (18.75→7.25 on day), and removing *Moment Recall* or *Pattern Traversal* sharply lowers the week split (15.75 / 15.58) and EgoServe (11.05); on the day split the remaining skills largely compensate, as the episode skeleton already answers coarse questions. Read against the structural ablation, this separates *content value* from *entry-point value*: a stratum helps only when both its content and a native way to reach it are present.

Table 4: Online video understanding on OVO-Bench and StreamingBench. We report the overall score of each track (OVO real-time/backward/forward and their overall; StreamingBench real-time, contextual, and their average). The results of Online MLLMs are reported from (Azad, Vineet, and Rawat 2026).

Model	OVO-Bench				StreamingBench		
	Real-Time	Backward	Forward	Overall	Real-Time	Contextual	Avg
Human Agents	93.20	92.33	92.90	92.81	91.46	93.55	92.51
Proprietary MLLMs							
GPT-4o (Hurst et al. 2024)	64.5	60.8	53.4	59.5	73.3	38.7	56.0
Claude 3.5 Sonnet (Anthropic 2024)	–	–	–	–	72.4	37.7	55.1
Open-Source Offline MLLMs							
InternVL2 (Chen et al. 2024b)	60.7	44.0	45.4	50.0	63.7	32.4	48.1
LLaVA-OneVision (Li et al. 2024)	62.8	45.0	50.9	52.9	71.1	32.7	51.9
Qwen2-VL (Wang et al. 2024a)	56.0	46.7	48.7	52.7	69.0	31.7	50.4
LLaVA-NeXT-Video (Li et al. 2024)	63.3	41.7	54.2	53.1	69.8	34.3	52.1
VITA-1.5 (Fu et al. 2026)	63.5	41.5	53.5	52.8	52.3	27.4	39.9
HierarQ (Azad, Vineet, and Rawat 2025)	67.3	48.3	48.3	54.6	69.7	35.1	52.4
Open-Source Online MLLMs / Agents							
VideoLLM-online (Chen et al. 2024a)	20.8	17.7	–	19.3	36.0	26.6	31.3
Flash-VStream (Zhang et al. 2024)	29.9	25.4	44.2	33.2	23.2	24.1	23.7
Dispider (Qian et al. 2025)	54.6	36.1	34.7	41.8	67.6	33.6	50.6
StreamForest (Zeng et al. 2026)	61.2	52.0	53.5	55.6	**77.3**	–	–
StreamBridge (Wang et al. 2026)	71.3	68.1	48.4	62.6	77.0	32.6	54.8
ViSpeak (Fu et al. 2025)	66.3	57.5	54.3	59.4	70.4	43.9	57.2
TimeChat-Online (Yao et al. 2025)	58.6	42.0	36.4	45.7	75.4	35.3	55.4
StreamAgent (Yang et al. 2025)	61.3	41.7	45.4	49.5	74.3	34.6	54.5
GROVE (Ours)	**76.2**	**69.9**	**56.5**	**67.5**	77.0	**53.1**	**65.1**

Table 5: Experimental results on the ESTP-Bench. EyeWO* is the benchmark's own model (gray). Best non-gray per column in **bold**.

Model	EPT	IPT	CQ	Overall
EyeWO* (Zhang et al. 2025b)	23.6	52.5	43.6	34.7
LLaVA-OneVision (Li et al. 2024)	13.6	31.8	8.2	18.0
Qwen2-VL (Wang et al. 2024a)	15.4	**39.3**	10.4	21.3
MiniCPM-V (Yao et al. 2024)	15.2	36.5	26.6	22.9
LLaVA-NeXT-Video (Li et al. 2024)	16.5	34.6	13.8	21.3
InternVL-V2 (Chen et al. 2024b)	5.6	6.7	5.7	5.9
LIVE (Chen et al. 2024a)	9.5	25.6	18.9	15.5
MMDuet (Wang et al. 2024b)	9.1	34.2	20.3	17.8
GROVE (Ours)	**22.7**	34.4	**41.0**	**28.6**

Table 6: Ablation of memory structure. Each masked memory layer is replaced by an equal volume of raw captions, so that all rows receive the same information and differ only in structural organization.

Config	MMLifelong		EgoServe
	day	week	
GROVE	18.75	19.75	**12.62**
w/o Perceptual Trace	15.50	**20.50**	12.27
w/o Moment	17.09	18.25	11.77
w/o Episode	14.57	17.25	12.47
w/o Pattern	**19.90**	13.00	11.49
Flat (w/o Hier.)	14.75	15.50	11.53

Retrieval mechanism. Table 8 varies how the skills query the memory. Removing the temporal-range constraint degrades localization (18.75→15.08 on day, 19.75→13.50 on week), and building the proactive query from the current perceptual trace outperforms an LLM-written query on EgoServe (12.62 vs. 12.17): grounding retrieval in observed entities and actions is more reliable than free-form query generation.

Structure and efficiency. Starting from a flat caption index, we add one stratum back at a time with the retrieval budget fixed, and measure accuracy together with the retrieved tokens, realized rounds, and latency per question (Table 9). Accuracy increases monotonically as the hierarchy is restored (12.75→18.75), while the agent needs *fewer* rounds (6.25→4.95) and less time (59.3→47.6 s) to answer. Structure therefore buys accuracy and efficiency at once: a memory organized by temporal scale lets the agent reach the right evidence directly instead of searching its way toward it.

Inference-Time Analysis

We analyse costs on both sides of the system: the retrieval budget in the agentic reasoning stage and the inference time in the memory construction stage.

Retrieval budget. To explore how the retrieval budget affects accuracy and cost, we vary the maximum number of retrieval rounds with the memory fixed. Accuracy is measured on the full day split, while the prompt tokens and LLM time per question are profiled on a batch of 20 questions. As illustrated in Fig. 3 (a), accuracy peaks at eight rounds (18.75) and declines beyond it (15.33 at ten), whereas tokens and latency grow almost linearly with the budget. Additional retrieval thus stops paying off once the evidence is sufficient-further calls mostly add low-information context-so we set the budget to eight rounds throughout, where the agent self-terminates after 4.95 rounds on average.

Memory construction. To accelerate construction, we run

Table 7: Ablation of retrieval skills. On EgoServe, the schema injects evidence blocks instead of exposing tools, so removing a skill amounts to removing its stratum for perceptual trace and pattern layer.

Config	MMLifelong		EgoServe
	day	week	
GROVE	18.75	19.75	**12.62**
w/o Perception Lookup	16.25	**20.25**	12.27
w/o Moment Recall	**19.50**	15.75	11.05
w/o Episode Replay	18.34	19.60	11.36
w/o Pattern Traversal	18.50	15.58	11.49
No retrieval	7.25	12.75	10.39

Table 8: Ablation of retrieval details on different benchmarks.

Config	MMLifelong		EgoServe
	day	week	
GROVE	**18.75**	**19.75**	**12.62**
w/o time boundary	15.08	13.50	-
Question Retri.	12.56	18.00	-
LLM-generated query	-	-	12.17

captioning, episode segmentation, and moment/pattern consolidation as three overlapping asynchronous stages. Fig. 3 (b) reports the build time per minute of video across sampling rates and window counts: it drops from 78.2 to 19.5 seconds, exceeding real time from one frame every five seconds onward and reaching 3.08× at the sparsest setting. Construction can thus keep pace with a live stream, trading temporal resolution for throughput without altering the memory structure.

Qualitative Analysis

Fig. 4 shows three cases. In the first, GROVE locates the hotpot episode among hours of unrelated footage and reads the drink from its perceptual trace, matching the ground truth. In the second, a procedural query, it replays the cooking episode and follows its ordered moments to recover the next step. In the third, with no query at all, it traverses the pattern linking the Day-4 assembly to the Day-1 session and offers the earlier record. The cases show one memory read at the right scale for both reactive and proactive use.

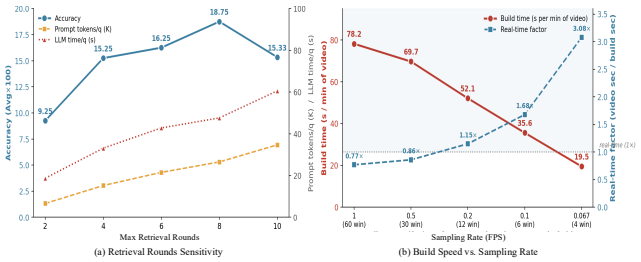


Figure 3: Inference and construction cost. (a) Accuracy, total prompt tokens, and LLM time per question as the maximum retrieval budget varies on MM-Lifelong day. (b) Build time per minute of video and the resulting real-time factor across sampling rates.

Table 9: Ablation of the memory strata on MM-Lifelong (day). The inference budget is computed over the same 20 questions as in Fig. 3

Level	day	tok/q	rounds	lat.(s)
caption flat	12.75	15.1K	6.25	59.3
+ Episode	13.75	18.2K	6.40	59.1
+ Moment/fact	13.57	19.1K	5.05	48.1
+ Pattern	15.50	29.7K	5.60	50.9
Full	**18.75**	26.5K	**4.95**	**47.6**

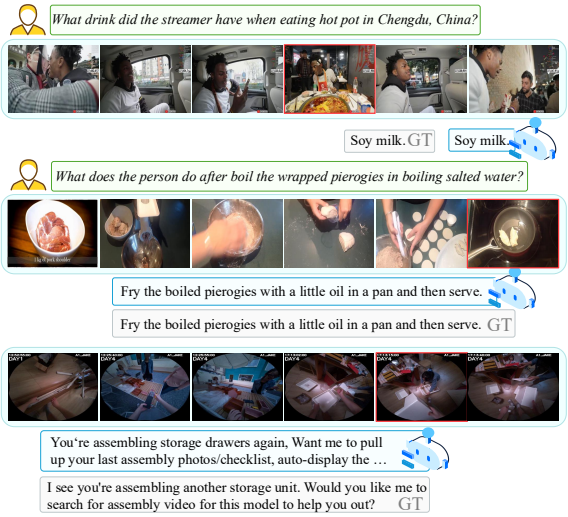


Figure 4: Visualization of GROVE's predictions. The first two cases answer user queries about a fine-grained detail and a procedural step from a long-past video, while the third fires a proactive service without any query.

Conclusion

We presented GROVE, a training-free framework that organizes streaming video memory and its access around temporal scale. GROVE causally transforms incoming observations into a perceptual trace, time-stamped moments, coherent episodes, and recurring patterns, with a retrieval skill specialized for each stratum. Reactive QA and proactive assistance share this memory and skill library. Results across five benchmarks, together with controlled ablations, show that pairing temporal structure with scale-native access improves both long-horizon recall and situation-aware assistance. GROVE therefore provides a practical memory substrate for assistants that must continuously observe, remember, and act.

Limitations

GROVE inherits errors from its frozen perception model, and evidence missed during ingestion cannot be recovered through later consolidation. Higher-level memory is updated only when an episode closes, so current-window queries rely on the perceptual trace. Memory construction and multi-round retrieval also become more costly as the observed history grows. Finally, because the main benchmarks use different reasoning backbones, component-level conclusions rely on the fixed-backbone ablations rather than absolute cross-benchmark results.

A. Methods

A.1 Details on Memory Construction

This section complements the construction pipeline described in the main paper with the implementation-level rules, the hyperparameters of every stage, and the dataset-specific instantiations that we use to build all memories reported in the experiments. Table 10 lists all hyperparameters for both memory construction and retrieval stages, while Algorithm 1 summarizes the online procedure.

Window Perception. A stream is consumed as fixed-length windows. Within a window, we sample frames at a constant interval and issue a *single* VLM call that returns one detailed caption per sampled frame, each with its own time span, plus one window-level description; the same call emits the perceptual trace entry of that window. The sampling interval is increased from 2s to 5s on MM-Lifelong and EgoServe, whose recordings run for tens of hours: the number of VLM calls grows linearly with duration, so a denser interval would make these two settings account for most of the construction cost. A window incorporating captions and perceptual trace is also the smallest unit the memory ever exposes—its perception becomes retrievable as soon as the VLM call returns, without waiting for the surrounding EPISODE to close.

Deterministic Rules around the Segmenter. The LLM segmenter receives the frame-level captions of the arriving window together with those of the three most recent windows of the open unit and makes a binary judgment on the window as a whole. It compares stable anchors across the two: whether the high-level activity continues, whether the location has changed, and whether the key objects and people involved are still present. A boundary is declared only when such a change holds for the entire window, and the open unit is then closed *before* it, which becomes the first window of the next unit. Three deterministic rules bound the judgment: a cut is forced without consulting the LLM at recording gaps above 60s, at day changes, and at the episode length cap; the LLM is not queried until the open unit holds a minimum number of frames, so a boundary is never declared from a single caption; and an unparsable output defaults to EXTEND, keeping the activity open rather than fragmenting it.

Adaptive Moment Density. The number of MOMENTs requested from a closed EPISODE is a target rather than a fixed rate: it is set to $\lceil \text{duration}/\Delta \rceil$, with $\Delta = 5$s for units up to 5min, 10s for units between 5 and 15min, and 20s beyond that, and the extraction may return up to $1.5\times$ that target, with a floor of 5 and a hard ceiling of 200 per EPISODE. The timestamps are not uniformly spaced—they fall wherever something actually happens—so the target only controls how densely coverage is requested: it keeps a ten-minute unit from collapsing into a handful of sentences while preventing a pathological unit from flooding the stratum. The role and weight of each MOMENT are assigned within the same extraction stage, so no additional pass over the memory is needed to build the group edge.

Bounded Cost of Pattern Consolidation. Three measures keep consolidation from growing with the number of PATTERNs already stored. Candidates are examined in parallel batches of 10 and the first affirmative judgment absorbs the EPISODE, so a match short-circuits the remaining candidates. When a PATTERN is re-synthesized, only its most recent 5 EPISODEs contribute full summaries, and older ones are compressed into a single short subject each, which bounds the prompt regardless of how long the PATTERN has been running. Roles and weights of the grouped EPISODEs are assigned default values on insertion and re-estimated by a dedicated LLM pass only once every 3 updates of that PATTERN, together with the coherence score.

Dataset-specific Instantiations. Construction is otherwise identical across benchmarks, with three exceptions that we state explicitly for fairness. (i) Caption and extraction prompts are designed specifically to scenario—generic third-person QA, first-person tool manipulation, and first-person daily life—so that the perception stage attends to the evidence each domain requires. (ii) On the two instructional subsets of EgoServe, the system is given the task-level prior that the benchmarks themselves define. On HoloAssist the extraction stage receives the declared task type of the recording, its canonical step sequence, and a short list of known error patterns for that task type. On CaptainCook4D the screening prompt receives the correct step sequence of the recipe being performed, taken from the official task graphs in topological order (at most twenty-five steps). Both are the kind of task knowledge a deployed assistant holds before a session starts, and both follow the setting of these benchmarks, in which error detection is defined relative to a known procedure. (iii) On the forward-looking tracks of OVO-Bench and ESTP-Bench, where the system must decide *when* to speak rather than answer a question posed afterwards, the candidate user questions of a video are visible to the construction stage; this follows the protocol of these tracks, in which the forward query is known in advance and only its firing time is evaluated. No question is visible during construction on MM-Lifelong, StreamingBench, and backward tracks of OVO-Bench.

Robustness and Engineering. Every LLM call is issued through a wrapper that allows up to three attempts with exponential backoff, and its output is parsed with a permissive JSON repair pass before any fallback is taken. The three stages—captioning, segmentation, and the consolidation of a closed EPISODE—run as an asynchronous pipeline, so that captioning of later windows overlaps with the consolidation of earlier ones; within one closed EPISODE, MOMENT extraction and PATTERN consolidation are issued concurrently. The resulting memory, its text indices and a resumable checkpoint are persisted every ten windows, so construction can be interrupted and continued on a long recording.

Memory Cases Visualization. Fig. 5–7 show one constructed memory per domain, read from the top down: a PATTERN with its title, keywords and summary, the EPISODEs it groups through a group edge, the MOMENTs each EPISODE expands into with their roles and weights, and the perceptual trace entries underneath. A PATTERN genuinely spans days–in Fig. 5 the same activity is recognized on `DAY1` and `DAY2`—and the strata differ in kind rather than only in length, with the perceptual trace keeping the literal counts and on-screen text that summaries blur.

Scale of the Constructed Memories. Table 11 reports the

Table 10: Construction and retrieval hyper-parameters. Values are shared across benchmarks unless a row states otherwise; the settings that differ follow the length of a recording rather than the domain.

Stage	Parameter	Value
Perception	Window length (MM-Lifelong, EgoServe)	30s
	Window length (other benchmarks)	10s
	Frame interval (MM-Lifelong, EgoServe)	5s
	Frame interval (other benchmarks)	2s
Segmentation	Min. frames before semantic check	5
	Forced cut at recording gap	60s
	EPISODE cap (MM-Lifelong, EgoServe)	10min
	EPISODE cap (other benchmarks)	2min
	Recent windows shown as anchors	3
MOMENTs	Target density ($\leq$5 / 5–15 / >15 min)	5 / 10 / 20s
	Min. MOMENTs per EPISODE	5
	Max. MOMENTs per EPISODE	200
	EPISODE summary length	3–4 sentences
PATTERNs	Candidate batch size	10
	Full summaries kept on re-synthesis	5
	Role / coherence re-estimation period	3 updates
	PATTERN summary length	4–6 sentences
Retrieval	BM25 k_1, b	1.5, 0.75
	Reciprocal rank fusion k	60
	Traversal fan-out (PATTERN / EPISODE / MOMENT)	3 / 8 / 15
	Max. rounds $T_{\max}$ (day / week / month)	8 / 6 / 12
	Max. rounds $T_{\max}$ (OVO-Bench, StreamingBench)	3 / 2
LLM calls	Retries with exponential backoff	3
	Decoding	greedy

Table 11: Scale of the constructed memories, aggregated per benchmark. Each row summarizes the memory structures of all subsets actually used for the reported results. Memories are built per video or per clip except on MM-Lifelong and EgoLife-subset, where one graph spans the whole recording.

Benchmark	Memories	PATTERNs	EPISODEs	MOMENTs
MM-Lifelong	3	2,041	6,786	102,865
OVO-Bench	1,640	4,149	8,019	78,497
StreamingBench	700	3,433	7,128	67,110
ESTP-Bench	890	2,675	5,688	73,232
EgoServe	283	2,531	8,002	84,866

size of the memory built for each benchmark. Two ratios are informative. First, the MOMENT stratum is an order of magnitude larger than the EPISODE stratum throughout (from 9.4 MOMENTs per EPISODE on StreamingBench to 15.2 on MM-Lifelong), which is expected of the only stratum that keeps second-level evidence. Second, how much consolidation the PATTERN layer performs depends on the horizon rather than on the amount of video: on the two settings where a single memory spans an entire recording, 6,786 EPISODEs fold into 2,041 PATTERNs on MM-Lifelong and 8,002 into 2,531 on EgoServe (about 3.2 EPISODEs per PATTERN), whereas on the clip-level benchmarks each PATTERN groups barely two EPISODEs, since a short clip rarely revisits the same activity. The PATTERN stratum therefore grows sub-linearly precisely in the long-horizon regime it was designed for, which is what keeps the entry point for broad questions small even after weeks of recording.

Algorithm 1: Incremental Memory Construction

Input: video stream $\{w_1, w_2, \dots\}$; VLM Φ; consolidation LLM Ψ
Init: $\mathcal{R}, \mathcal{F}, \mathcal{E}, \mathcal{P} \leftarrow \emptyset$; open unit $\tilde{e} \leftarrow \emptyset$
for each arriving window w_t **do**
  $(c_t, r_t) \leftarrow \Phi(w_t)$; $\mathcal{R} \leftarrow \mathcal{R} \cup \{(c_t, r_t)\}$ {queryable at once}
  if recording gap $>$ 60s **or** day change **then**
    $b_t \leftarrow$ CUT {deterministic}
  else if frames in $\tilde{e} <$ min. frames **then**
    $b_t \leftarrow$ EXTEND {too early to judge}
  else
    $b_t \leftarrow \mathrm{Seg}(c_t,\ c_{t-3:t-1})$
  end if
  if $b_t =$ EXTEND **then**
    absorb w_t into $\tilde{e}$
    if $\mathrm{duration}(\tilde{e}) \geq$ cap **then**
      $b_t \leftarrow$ CUT {length guard}
    end if
  end if
  if $b_t =$ CUT **then**
    close $\tilde{e}$ as e_t; $\mathcal{E} \leftarrow \mathcal{E} \cup \{e_t\}$
    in parallel:
      $\{f_i\}, g_{e_t} \leftarrow \Psi(e_t)$; $\mathcal{F} \leftarrow \mathcal{F} \cup \{f_i\}$
      $p^* \leftarrow \mathrm{Match}(e_t, \mathcal{P})$ over concurrent candidate batches
      $\mathcal{P} \leftarrow \mathrm{Upd}(p^*, e_t)$ if p^* found, else $\mathcal{P} \cup \{\mathrm{New}(e_t)\}$
    re-open $\tilde{e} \leftarrow \{w_t\}$
  end if
end for
Output: $\mathcal{M}_{\leq t} = \{\mathcal{R}, \mathcal{F}, \mathcal{E}, \mathcal{P}\}$, queryable at any t

A.2 Retrieval Details on Different Benchmarks

The four skills of the main paper are abstract interfaces; what a benchmark instantiates depends on how the memory is accessed, which divides the five benchmarks into two groups. On MM-Lifelong, OVO-Bench and StreamingBench, a query opens each round, so the agent selects its own calls under a multi-round budget, following the skill-based agentic reasoning paradigm. On ESTP-Bench and EgoServe, no query or a forward request is posed, so retrieval is carried out for the agent by a fixed schema that assembles one evidence block per screening unit, following the schema-based proactive assistance paradigm. Table 12 reports which skills were actually exercised in the runs we report, measured from the inference traces.

Skill-based Agentic Reasoning **MM-Lifelong.** The three splits share the same nine calls and the same answer format, and differ in round budget (8/6/12 for day/week/month), the PATTERN map injected as free context before retrieval (capped at 46 and 100 entries on day and week, widened to the full timeline on month), and the content description of the video in the prompt. What the traces reveal is that the same library is exercised very differently. As shown in Table 12, on day and week, the agent converges on four calls

Table 12: Skills the agent actually invokes in the agentic reasoning stage, in calls per question. All figures are measured from the inference traces of the reported runs in the main paper. For MM-Lifelong we give both backbones: `gpt-5-mini`, used throughout the ablations, and `gpt-5.2`, used for the main-table results. Tools never invoked in a setting are omitted from that column ("–"), so each column reflects what the agent genuinely relies on rather than what the framework exposes. Tool names follow the stratum they address (`*_moment*` for $\mathcal{F}$, `*_episode*` for $\mathcal{E}$, `*_hierarchical` for the $\mathcal{P}\to\mathcal{E}\to\mathcal{F}$ traversal).

Skill	Tool	MM-Lifelong						OVO-Bench		StreamingB.
		gpt-5-mini			gpt-5.2					
		day	week	month	day	week	month	Real-Time	B.W.	Real-Time
$\mathcal{R}$	`get_recent_caption(s)`	–	–	–	–	–	–	**1.05**	–	**0.81**
	`search_ocr`	–	–	1.24	**4.38**	0.41	0.96	–	–	–
	`ocr_in_range`	–	–	1.30	–	–	0.42	–	–	–
	`search_events`	–	–	1.94	0.83	0.66	1.13	–	–	–
	`count_events`	–	–	0.32	0.49	0.29	0.17	–	–	–
	`search_entities`	–	–	0.45	0.03	0.43	0.11	<0.01	**0.28**	–
$\mathcal{F}$	`search_moments`	**4.13**	**2.06**	2.17	2.06	0.89	1.83	–	–	–
	`moments_in_range`	1.15	1.46	1.56	0.69	1.14	1.27	–	–	–
	`search_moment_only`	–	–	–	–	–	–	–	0.11	–
$\mathcal{E}$	`search_episodes`	1.49	1.37	**5.55**	1.67	**1.91**	**3.37**	–	–	–
	`get_recent_episodes`	–	–	–	–	–	–	0.01	0.07	–
	`search_episode_moment`	–	–	–	–	–	–	<0.01	0.23	–
$\mathcal{P}\to\mathcal{E}\to\mathcal{F}$	`search_hierarchical`	2.40	1.67	0.53	1.09	1.24	0.33	–	–	–
Total calls / question		**9.16**	**6.55**	**15.07**	**11.26**	**6.96**	**9.59**	**1.06**	**0.69**	**0.81**
Distinct skills used		4	4	9	**8**	**8**	9	4	4	1
$\mathcal{R}$ share of calls		0%	0%	35%	**51%**	**26%**	29%	99%	44%	100%
Accuracy		18.75	19.75	17.82	**23.50**	**22.75**	**19.98**	76.2	69.9	77.0

and never issues an OCR, entity, event, or counting one; on month all nine appear, with the perception stratum taking 35% of the 15.07 calls per question. The subset also depends on the router rather than the setting alone: replacing the answer model with `gpt-5.2` while holding memory, prompt, and call library fixed widens day and week from four skills to all eight, and the added calls go almost entirely to $\mathcal{R}$ (0% → 51% of the calls on day, 0% → 26% on week), with `search_ocr` becoming the most frequent call on day at 4.38 per question after never being issued once by the smaller model. Accuracy moves with it (18.75 → 23.50 and 19.75 → 22.75). Skills a weaker router leaves untouched are therefore not dead weight in the library — they are what a stronger router reaches for first.

OVO-Bench. The real-time and backward tracks are answered by a *single* solver: one tool document, one round prompt, and the same starting context — the chronological skeleton of EPISODE summaries with their time spans, covering every EPISODE that closed before the question moment. The budget is three rounds with one call each. The prompt asks the agent to first decide whether the question concerns the current moment or the past, and the two profiles that follow are entirely its own. On *real-time*, the decision resolves to reading the frame-level captions of the last ten seconds: `get_recent_captions` accounts for 1.05 of the 1.06 calls per question, and the semantic strata are touched fewer than once in a hundred questions. On *backward*, those captions are never read and the budget goes to the strata instead — `search_entities` 0.28, `search_episode_moment` 0.23, `search_moment_only` 0.11, `get_recent_episodes` 0.07 per question.

The *forward* track is different in kind, because the query is known in advance and only the moment of the response is evaluated. There the agent receives the recent frame captions as its primary evidence together with the MOMENTs already retrieved for that query and its own previous answers, and at each streaming timestamp emits either a response or a decision to keep waiting.

StreamingBench. Questions are answered at a given moment of an otherwise unseen stream, so the instantiation is deliberately perception-first: the EPISODE skeleton up to the question moment is supplied for free, and the prompt directs perception questions to call `get_recent_caption` — the second-level captions of that moment — before anything else, leaving the semantic search skills as a fallback for details that are no longer on screen. Up to three calls may be issued in one round, which keeps the number of LLM passes low on a benchmark of this size.

The fallback is in practice never taken. On the *real-time* track the agent issues `get_recent_caption` 0.81 times per question and nothing else; across all 2,500 questions the three semantic search skills are invoked zero times, and roughly one question in five is answered with no call at all, directly from the skeleton. This is the sharpest instance of a pattern that also appears on MM-Lifelong: what the agent exercises is decided by what the questions need, not by what the library offers. The *contextual* track uses the

same instantiation and differs only in that its questions are posed over a longer preceding span, which is what makes the EPISODE skeleton and the semantic strata contribute.

Schema-based Proactive Assistance ESTP-Bench. The system must decide *when* and *how* to answer a standing question rather than answer on demand, so retrieval is carried out by the schema rather than issued as a call by the agent. At each step of the stream the agent sees the dense caption of the current step, the interaction history with its own earlier answers and their timestamps, and — when the question calls for grounding in the past — an evidence block assembled for it from the MOMENT and perceptual trace strata together with the covering EPISODEs. This block may only ground the content of a response and never triggers one by itself, which prevents retrieved history from firing an answer in the absence of present evidence. The two settings reach that block differently: in the conversational setting it is assembled unconditionally at every step, whereas in the single-query setting the agent first judges from the present alone, and the block is assembled only once it asks for grounding, at most once per question. The budget is therefore deliberately asymmetric — the present is always available, the past on request — and 99.2% of responses are committed without any retrieval at all, which is what keeps the schema affordable at streaming rates.

The two settings also differ in what defines a decision step and in how often the decision is revisited. In the single-query setting, the question stands from the start of the segment and the decision is revisited at every second-level caption; in the conversational setting the stream is grouped into short windows that begin at the moment each turn is asked, carry the dialogue so far, and are revisited more densely for turns whose answer must track a task in progress. A response is anchored at a second chosen by the agent and clamped to be causal, and repeated firings of the same intent are suppressed twice over — by an instruction not to restate an answer unless something new appears, and by a hard minimum gap of 8s between consecutive responses, tightened to 4s for questions that ask the system to comment on each sub-step as it happens.

EgoServe. No question is posed at any point, so the agent is driven by a fixed schema rather than by tool selection. For every screening unit the schema assembles, from the same strata, the perceived event stream of the unit, the summary of the covering EPISODE, the MOMENTs inside the unit, the wearer state, and three kinds of associative candidates retrieved strictly from before the unit — entities being re-encountered after hours, earlier requests and object placements now due, and PATTERN-level routines. The service taxonomy is given as a closed set of options with its timing conventions, and the agent returns, in one pass, any services it decides to fire with an exact trigger second copied from a perception line.

Its two sources differ in screening unit and in scope, and this is a genuine configuration difference rather than a prompt variant. The daily-life recordings (EgoLife) are screened at EPISODE completion — an adaptive unit, with a 180s floor and a 600s fixed-window fallback — carry spoken dialogue and stated intentions, and are judged by three service routers in parallel; since those routers can independently fire on the same situation, their outputs are merged by deduplicating per type within 60s, so that one situation cannot produce a burst of near-identical prompts. The instructional recordings (HoloAssist, CaptainCook4D) are screened on fixed short windows of 6s and 20s respectively, without dialogue, and are judged by a single router, so no merge step is needed; their services concern the task being performed rather than the wearer's habits, and consequently only the first five of the ten service types have ground truth there.

B. More Experimental Results

B.1 Extended Results of the Main Paper

Full Comparison on MM-Lifelong. Table 13 reports all four splits of MM-Lifelong, adding the Train@Month split that the main paper omits for space. GROVE attains the best score on every split, and its margin over ReMA is preserved on the month-scale training videos (18.98 vs. 17.62), indicating that the gains come from how the memory is organized and retrieved rather than from any single evaluation split.

Memory Structure on the Month Split. As illustrated in Table 14, the scale dependence observed on day and week continues to grow with the horizon. Removing PATTERN costs the most of any single stratum on month (17.82→14.85), a larger drop than on week, since a question spanning a 22-day recording is usually answered by a recurring activity rather than by one occurrence. Flattening the hierarchy is again the worst configuration (13.56) and its gap to GROVE is the widest of the three splits, whereas removing EPISODE costs little here (17.74) in contrast to day (14.57): month-scale questions rarely hinge on localizing one segment. The perceptual trace is likewise less critical at this scale (18.62), as the third-person travelling recordings contain fewer of the exact-count and on-screen-text cues it preserves.

Retrieval Skills on the Month Split. *Moment Recall* remains the skill the agent cannot do without (17.82→15.16), consistent with the week split. Two entry points behave differently from the shorter horizons: removing *Episode Replay* slightly improves accuracy (18.22), because searching several hundred EPISODE summaries often returns plausible but wrong segments and consumes rounds that the chapter map would have spent better; and removing *Pattern Traversal* is nearly neutral (17.90), since month-scale counting and ordering questions are already answered by enumerating occurrences rather than by a top-down traversal. Disabling retrieval altogether still collapses accuracy (11.16), confirming that the gains come from the retrieved evidence and not from the reasoning backbone alone.

Memory Structure across Service Types. Table 16 breaks the EgoServe ablation into the ten service sub-types, which makes visible what the aggregate macro-F1 hides: each stratum supports a different band of the service spectrum. The two informative cases are PATTERN and MOMENT, which have opposite profiles. Removing PATTERN leaves the instant and short-term families untouched (NSG 21.8 → 21.6, ER 23.8 → 24.1) while precisely those services that must recognize a *recurring* activity collapse — memory recall

Table 13: Full comparison on the MM-Lifelong across each subset.

Method	Train@Month	Val@Month	Test@Week	Test@Day
Human	82.5	80.4	95.6	99.2
GPT-5 (Singh et al. 2025)	10.15	14.87	15.00	15.25
Qwen3-VL-235B-A22B (Bai et al. 2025)	9.09	14.33	15.63	12.44
Video-XL-2-8B (Qin et al. 2025)	4.89	9.07	12.00	9.00
Eagle-2.5-8B (Chen et al. 2026b)	2.07	6.10	7.00	8.25
Nemotron-v2-12B (Deshmukh et al. 2025)	7.52	9.63	11.00	7.25
VideoMind-7B (Liu et al. 2025b)	5.26	8.35	11.75	7.50
LongVT-7B (Yang et al. 2026)	5.83	7.54	9.75	7.00
DeepVideoDiscovery (Zhang et al. 2026)	4.36	10.57	9.02	10.25
ReMA (Chen et al. 2026c)	17.62	18.62	18.82	16.75
GROVE (ours)	**18.98**	**19.98**	**22.75**	**23.50**

Table 14: Ablation of memory structure on the three MM-Lifelong splits. (day/week/month)

Config	day	week	month
GROVE	18.75	19.75	17.82
w/o Perceptual Trace	15.50	**20.50**	**18.62**
w/o Moment	17.09	18.25	16.53
w/o Episode	14.57	17.25	17.74
w/o Pattern	**19.90**	13.00	14.85
Flat (w/o Hier.)	14.75	15.50	13.56

Table 15: Ablation of retrieval skills on the three MM-Lifelong splits. (day/week/month)

Config	day	week	month
GROVE	18.75	19.75	17.82
w/o Perception Lookup	16.25	**20.25**	16.45
w/o Moment Recall	**19.50**	15.75	15.16
w/o Episode Replay	18.34	19.60	**18.22**
w/o Pattern Traversal	18.50	15.58	17.90
No retrieval	7.25	12.75	11.16

$5.9 \rightarrow 2.7$, task reminder $5.9 \rightarrow 4.5$, routine optimization $7.1 \rightarrow 4.2$ — and it is the only ablation whose damage falls mainly on the far end of the spectrum (-1.74 on average there against -0.54 at the near end, every other stratum being the other way round). A reminder that an activity is due can only be issued if that activity has been recognized across days, whereas a safety alert needs nothing beyond the current window. MOMENT is the mirror image and the costliest stratum at the near end (-2.38 on average), hurting tool use ($20.2 \rightarrow 16.7$) and next-step guidance ($21.8 \rightarrow 17.1$) most; it also carries habit coaching ($8.6 \rightarrow 4.3$), which survives the PATTERN ablation ($8.6 \rightarrow 8.2$) but not this one, since a second-level record is what lets a service be both triggered and timed. The remaining rows behave as expected: the perceptual trace concentrates its damage on the instant family (SA $19.7 \rightarrow 14.9$) and flattening the memory is worst on the types that need organization (SA 13.5, ML 4.4).

Retrieval Skills across Service Types. Table 17 shows that an entry point is only as useful as the band it serves. *Moment Recall* is the most consequential skill overall ($12.62 \rightarrow 11.05$); its removal is felt on safety alerts ($19.7 \rightarrow 13.9$), next-step guidance ($21.8 \rightarrow 16.5$) and habit coaching ($8.6 \rightarrow 2.8$) — the three types whose decision needs a specific second rather than a general situation. The services that reach further back instead depend on the two coarse-grained skills: without *Episode Replay* memory recall almost vanishes ($5.9 \rightarrow 1.1$) and routine optimization drops ($7.1 \rightarrow 4.0$), because a recalled fact is only actionable when the activity that surrounds it can be read; without *Pattern Traversal* memory recall (2.7) and routine optimization (4.2) fall to exactly the level seen when the PATTERN stratum itself is deleted, since on this benchmark the two interventions coincide. With no retrieval at all the pattern is unmistakable: safety alerts remain usable (15.1) because they are decided from the present scene, while memory recall ($5.9 \rightarrow 1.6$), task reminder ($5.9 \rightarrow 4.2$) and habit coaching ($8.6 \rightarrow 4.2$) lose most of their accuracy. Proactive assistance that merely reacts to what is visible needs no memory; assistance that refers to what happened earlier is memory-bound.

B.2 Additional Experimental Results

Effect of the Retrieval Width. Table 18 varies the number of items each skill returns per call, top-k, while keeping the memory, the skills, and the round budget unchanged. A narrow setting starves the agent: at $k = 3$ every split loses three to five points, because the evidence that answers the question is often not among the first few ranked items. Accuracy peaks at $k = 20$ on the day and week splits and then degrades on day ($18.75 \rightarrow 16.33$), where the extra items are mostly redundant and dilute the context the agent has to read. The month split is the exception, improving marginally up to $k = 30$ ($17.82 \rightarrow 17.98$): questions spanning weeks require evidence gathered from more places at once, so a wider return is still useful.

How Skills Are Selected. Table 19 keeps the memory, the prompt, and the round budget fixed and changes only how calls are chosen; the agent still decides how many to issue, so the substituted rows spend a comparable budget (9.2 calls per question for GROVE against 9.8–10.4). *Only BM25* keeps the traversal intact and disables just its semantic branch: day (17.75) is unaffected because the questions quote words that appear verbatim in the memory, but week drops to 15.00 once a question and the recorded evidence are worded differently — semantics matters only once the memory is large enough for lexical overlap to become ambiguous. The *Only* rows force every call onto one stratum and are the most damaging intervention, costing close to six points on day; more tellingly, their ranking inverts across horizons (*Only Moments* is the weakest on day at 12.75 yet the strongest on

Table 16: Memory structure ablation results of each sub-type services on EgoServe benchmark.

Model	Instant		Short-term			Episodic		Long-term			Overall
	SA	TU	NSG	ER	RR	MR	TR	HC	ML	RO	
GROVE (Ours)	**19.7**	20.2	**21.8**	23.8	6.8	5.9	5.9	**8.6**	6.5	7.1	**12.62**
w/o Perceptual Trace	14.9	18.8	20.0	22.6	**11.2**	3.7	**9.8**	4.4	8.2	9.2	12.27
w/o Moment	17.6	16.7	17.1	22.7	6.3	4.5	6.2	4.3	**9.4**	**12.7**	11.77
w/o Episode	16.9	**20.6**	21.2	22.8	4.8	**8.2**	7.3	5.6	5.4	11.8	12.47
w/o Pattern	16.8	19.4	21.6	24.1	7.7	2.7	4.5	8.2	5.7	4.2	11.49
Flat	13.5	21.5	18.0	**24.8**	6.9	4.0	7.3	5.6	4.4	9.3	11.53

Table 17: Retrieval skill ablation results of each sub-type services on the EgoServe benchmark.

Model	Instant		Short-term			Episodic		Long-term			Overall
	SA	TU	NSG	ER	RR	MR	TR	HC	ML	RO	
GROVE (Ours)	**19.7**	**20.2**	**21.8**	23.8	6.8	**5.9**	5.9	**8.6**	6.5	7.1	**12.62**
w/o Perception Lookup	14.9	18.8	20.0	22.6	**11.2**	3.7	**9.8**	4.4	8.2	**9.2**	12.27
w/o Moment Recall	13.9	18.5	16.5	**24.9**	9.4	4.9	6.9	2.8	5.7	6.8	11.05
w/o Episode Replay	14.1	18.0	21.2	23.6	8.5	1.1	5.8	8.5	**8.7**	4.0	11.36
w/o Pattern Traversal	16.8	19.4	21.6	24.1	7.7	2.7	4.5	8.2	5.7	4.2	11.49
No retrieval	15.1	17.4	16.5	24.2	6.6	1.6	4.2	4.2	6.5	7.6	10.39

Table 18: Ablation results of retrieval width on the MMLife-long.

Config	day	week	month
top-k=3	13.25	15.50	14.84
top-k=10	16.50	16.08	16.16
top-k=20 (Ours)	**18.75**	**19.75**	17.82
top-k=30	16.33	19.50	**17.98**

week at 17.00), so no fixed choice of entry point transfers, and even the best of these single-stratum baselines (13.00) trails random alternation among all of them (16.00). The last block removes the choice without removing the retrieval: *Random-3* wins on day (20.75) but only because concatenating three returns inflates the evidence block by $\sim$40% in prompt tokens, and that advantage vanishes as the horizon grows (15.97 on month); *Fixed All Tools* retrieves everything every round at 1.7$\times$ GROVE's call volume and still loses on all three splits. Selecting where to look is therefore what the longer horizons reward, and the margin over *Fixed All Tools* widens from $+0.75$ on day to $+1.77$ on month.

Effect of Adaptive Segmentation. Table 20 isolates the segmenter on EgoServe. We keep the segment-level evidence block but redraw its boundaries: instead of the content-driven EPISODE boundaries, the same span is cut into the *same number* of equal-length segments, each filled with the captions it covers, with the count matched per recording (and per day on EgoLife, where the cuts are drawn inside the spans that contain video so that sleep and away periods do not consume segments). The screening windows, the segment count and the evidence volume are therefore unchanged, and the only variable is where the boundaries fall. Removing adaptive segmentation costs $12.62 \rightarrow 12.17$ overall, and the loss concentrates on the two services whose decision must be anchored to a particular second: safety alerts drop $19.7 \rightarrow 14.1$ and habit coaching $8.6 \rightarrow 4.2$, in both cases because the number of matched predictions falls outright ($49 \rightarrow 35$ and $6 \rightarrow 3$) at an unchanged prediction volume — the responses are still issued, but they no longer land inside the tolerance window. A fixed-length unit no longer begins and ends where the activity does, so the summary grounding the decision mixes two activities and the trigger second drifts from the event it should mark.

Table 19: Ablation on how retrieval skills are selected. *Only BM25* keeps the full skill set but disables the semantic branch of the cross-stratum traversal, so ranking is lexical. *Only Moments / Episodes / Hierarchical* force every call to resolve to that single skill. *Random-1* and *Random-3* replace each chosen skill with one or three sampled uniformly at random. *Fixed All Tools* executes every skill on every round. Its round budget is capped at 3, the setting whose call volume comes closest to GROVE's.

Config	day	week	month
Only BM25	17.75	15.00	16.69
Only Moments	12.75	17.00	16.77
Only Episodes	12.00	14.50	14.53
Only Hierarchical	13.00	15.25	16.77
Random-1	16.00	15.75	16.53
Random-3	**20.75**	18.25	15.97
Fixed All Tools	18.00	18.75	16.05
GROVE (Ours)	18.75	**19.75**	**17.82**

C. Case Study

Fig. 8 shows the full skill-calling and reasoning trace of GROVE on a temporal-ordering question on the MMLife-long day subset. The four bosses are defeated across a 32-minute span of a 24.5-hour recording, so no single call can cover more than one of them. GROVE first locates the region with `search_episodes` (rounds 1–2), then combines `search_moments` with the $\mathcal{P} \rightarrow \mathcal{E} \rightarrow \mathcal{F}$ traversal to pull the individual defeat records (round 3), and finally narrows to the time ranges those rounds exposed — e.g.

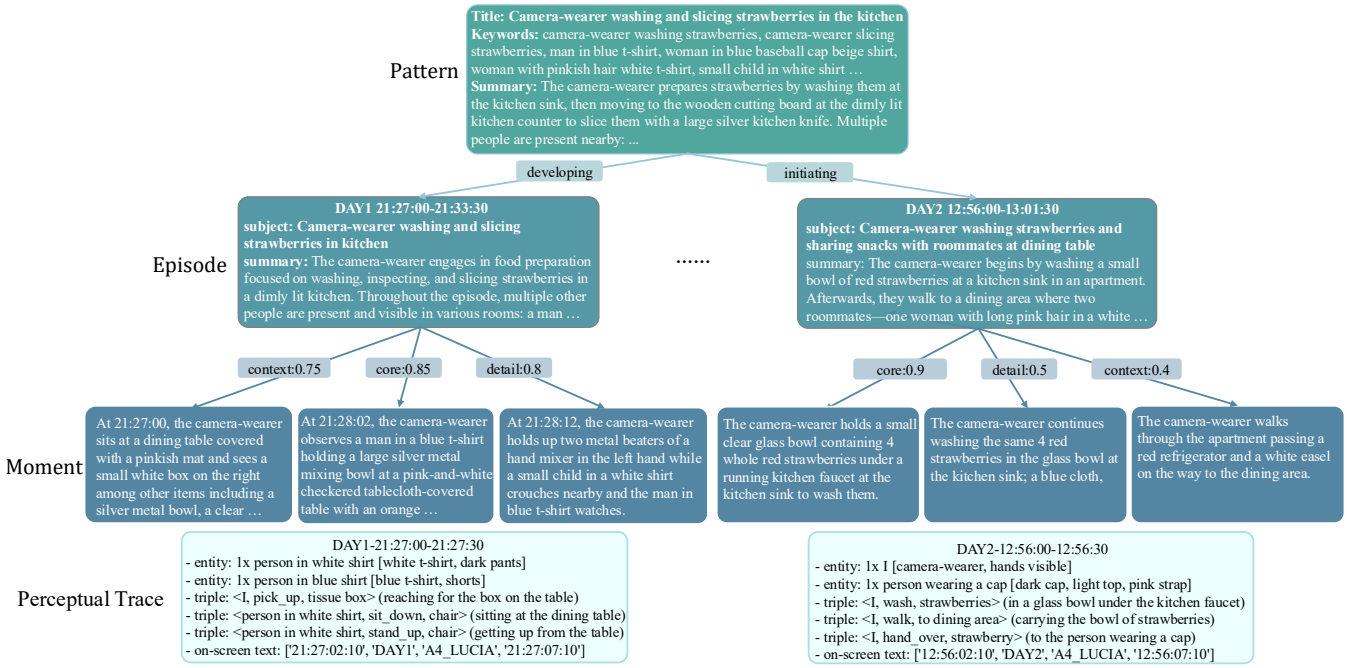


Figure 5: Temporally Stratified Memory Visualization of EgoServe Benchmark

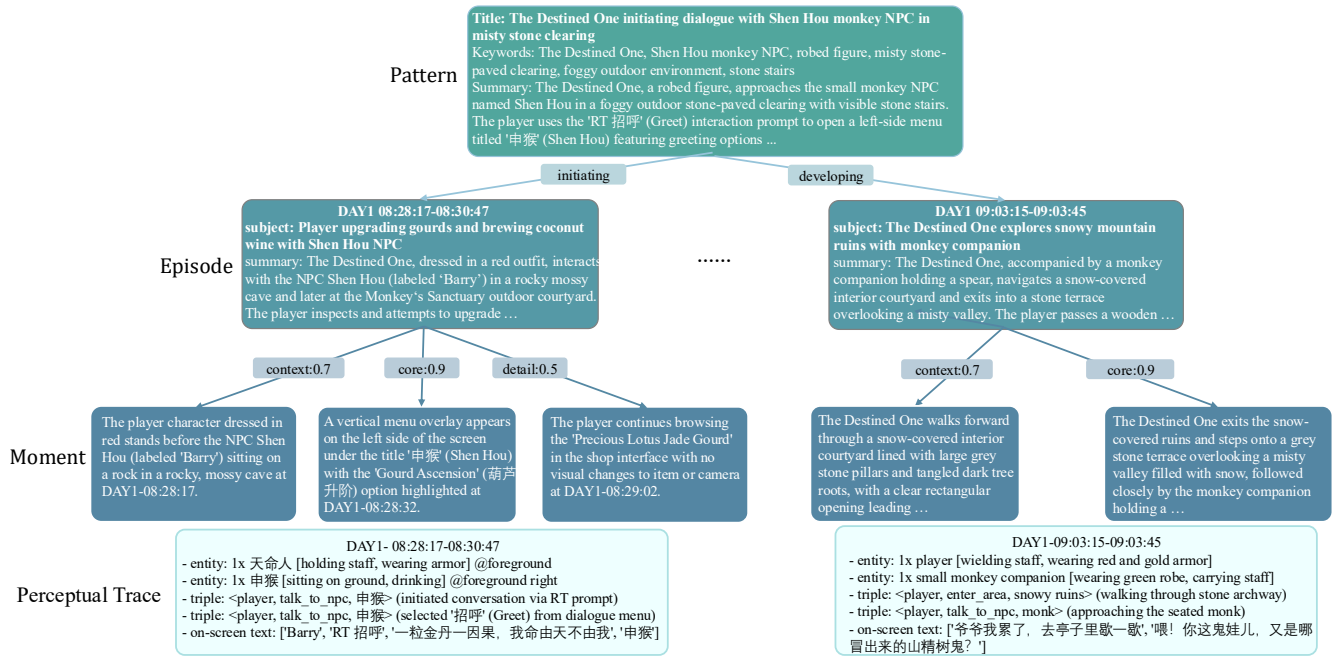


Figure 6: Temporally Stratified Memory Visualization of MMLifelong-Day

Table 20: Ablation of Adaptive Segmentation of each sub-type service on EgoServe.

Model	Instant		Short-term			Episodic		Long-term			Overall
	SA	TU	NSG	ER	RR	MR	TR	HC	ML	RO	
GROVE (Ours)	**19.7**	**20.2**	**21.8**	23.8	6.8	**5.9**	**5.9**	**8.6**	6.5	7.1	**12.62**
w/o Adaptive Segmentation	14.1	19.4	**21.8**	**23.9**	**9.3**	4.5	5.5	4.2	**8.3**	**10.7**	12.17

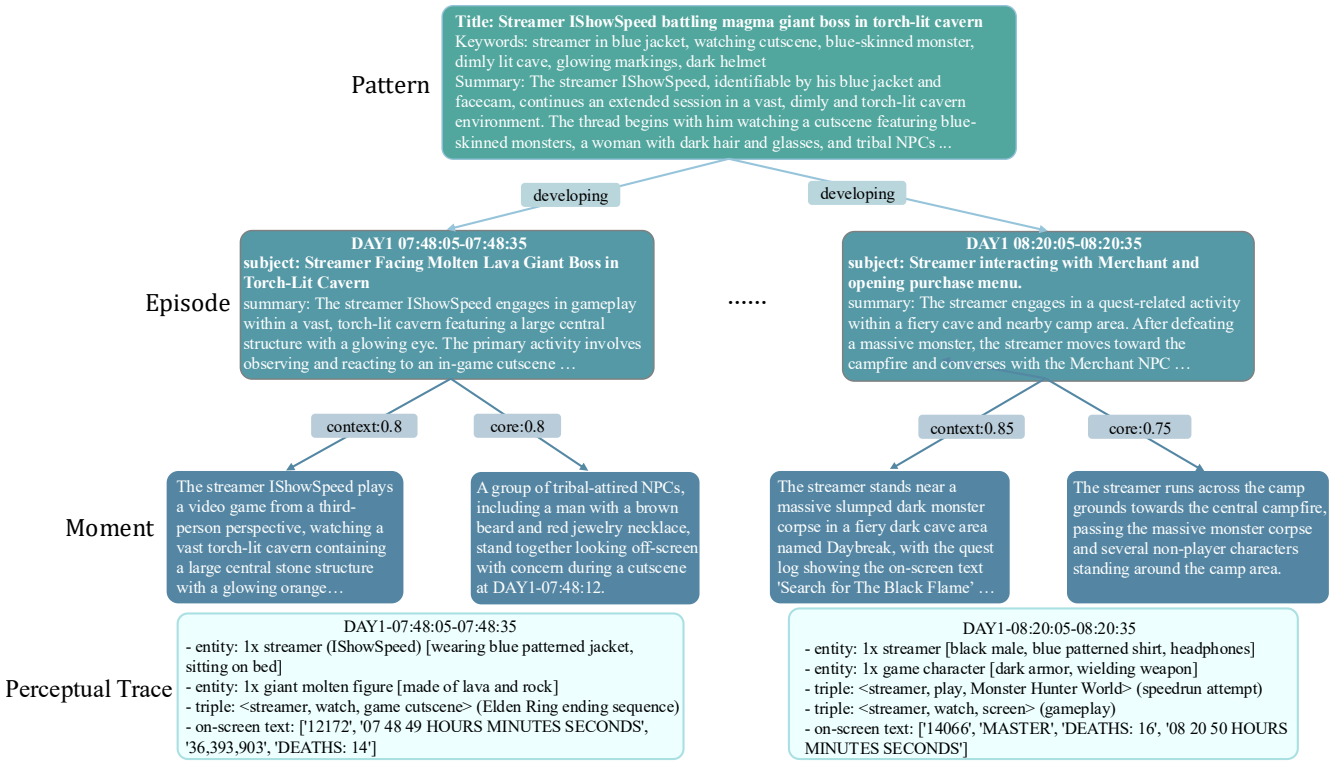


Figure 7: Temporally Stratified Memory Visualization of MMLifelong-Month

`search_moments` in $[01{:}20{:}00, 01{:}25{:}00]$ at round 6 — until every boss carries a timestamp. The answer is read off the reconstructed timeline rather than recalled, giving the correct order `1, 4, 2, 3`.

D. Prompts

D.1 Which Prompt Each Benchmark Uses

Construction is dispatched by scenario rather than by benchmark, so several benchmarks share the same prompt at a given stage. Table 21 resolves every benchmark to the five prompts it uses; the prompts themselves are listed once each in §D.2, and the inference prompts follow in §D.3.

D.2 Memory Construction Prompts

Dual perception: frame captions and the perceptual trace.

`FRAME_CAPTION_WINDOW_DETAILED_PROMPT_THIRD_PERS`

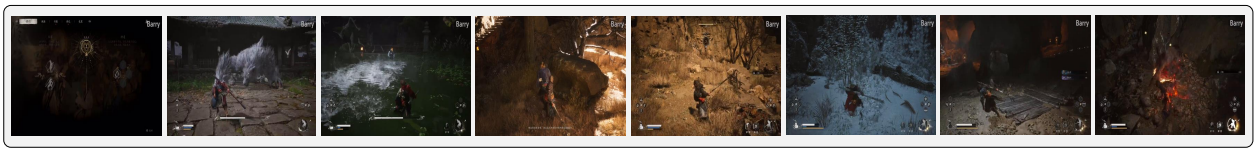
Stage / entry: Frame Caption

```
You are a visual evidence recorder for a THIRD-PERSON
video segment (sports broadcast, film/TV clip,
surveillance, interview, stage performance).

You will receive a short video segment (~10s, sampled
frames in temporal order). Each frame is an INDEPENDENT
evidence unit that may later be queried for people,
jersey numbers, on-screen text, scores, object
attributes, counting, or spatial relations.

Rules:
- There is NO camera-wearer. Never say "the user", "I",
"my hand". Refer to visible people as "the player on the
 right", "the presenter", etc.
- Each frame description must stand alone. Do NOT
reference other frames. Do NOT use "continues", "still",
 "now", "after", "starts to".
- Only describe what is visually present. No intent,
tactics, emotions (unless clearly visible), or outcome
inference.

Forbidden words (never use any of these without a
concrete replacement):
  "something", "some", "object", "item", "thing", "stuff
",
  "a tool", "a device", "an appliance", "equipment", "
gadget",
```



Input Streaming Video GT: 1, 4, 2, 3 | PRED: 1, 4, 2, 3 | TYPE: Temporal Reasoning

QUESTION: After the player enters Black Wind Mountain (黑风山), in what order are the following bosses defeated?
OPTIONS: 1. Lingxuzi (灵虚子) 2. Guangmo (广谋) 3. Whiteclad Noble (白衣秀士) 4. Baw-Li-Guhh-Lang (波里个浪)

Skill Calling and Reasoning Process

[Round 1]
TOOL : search_episodes
ARGS : keywords = 黑风山 白衣秀士 灵虚子 广谋 波里个浪 波里个浪
OBS : search_episodes('黑风山 白衣秀士 灵虚子 广谋 波里个浪 波里个浪') :
[02:31:44~02:32:14] The player engages in close combat with the large frog-like boss named '浪里个波' near water in an outdoor battle setting …
[01:03:43~01:04:18] The Destined One, depicted as a player character in an open natural environment, is engaged in combat with a giant …

[Round 2]
TOOL : search_episodes
ARGS : keywords = 黑风山 进入 黑风 山 进入 地点
OBS : search_episodes('黑风山 进入 黑风 山 进入 地点') :
[21:55:26~21:55:56] The Destined One, visible as the primary player character, performs a vertical ascent using the 'Somersault Cloud' …
[06:22:57~06:23:27] The player character, visible in a snowy forest environment labeled '雪山径' (Snow Mountain Path), collects an item named' …

[Round 3]
TOOL : search_moments
ARGS : keywords = 灵虚子 白衣秀士 广谋 波里个浪 黑风山
OBS : search_moments('灵虚子 白衣秀士 广谋 波里个浪 黑风山') :
[02:31:54] The large green frog-like boss '浪里个波' collapses prone in the shallow water, indicating it is defeated or stunned by the player …
[02:31:49] The large green frog-like boss '浪里个波' lunges forward with extended claws at the player, who dodges the attack in the shallow water …

TOOL : hierarchical_search
ARGS : keywords = 灵虚子 广谋 白衣秀士 波里个浪 黑风山
OBS : hierarchical_search('灵虚子 广谋 白衣秀士 波里个浪 黑风山') :
[01:03:43] The Destined One crouches low in tall green grass near a shallow water patch, stealthily approaching a giant bipedal frog boss …
[01:03:48] The Destined One attacks the giant frog boss 'Bo Li Ge Lang', which stands grounded in shallow water by the water's edge, while …

[Round 4]
TOOL : search_moments
ARGS : keywords = 灵虚子 广谋 白衣秀士 波里个浪 黑风山
OBS : search_moments('灵虚子 广谋 白衣秀士 波里个浪 黑风山') :
[02:31:54] The large green frog-like boss '浪里个波' collapses prone in the shallow water, indicating it is defeated or stunned by the player …
[02:31:49] The large green frog-like boss '浪里个波' lunges forward with extended claws at the player, who dodges the attack in the shallow water …

[Round 5]
TOOL : search_episodes
ARGS : keywords = 黑风山 进入 黑风 山
OBS : search_episodes('黑风山 进入 黑风 山') :
[21:55:26~21:55:56] The Destined One, visible as the primary player character, performs a vertical ascent using the 'Somersault Cloud' …
[06:22:57~06:23:27] The player character, visible in a snowy forest environment labeled '雪山径' (Snow Mountain Path), collects an item named …

[Round 6]
TOOL : search_moments
ARGS : keywords = 白衣秀士 击败 已击败 胜利
OBS : search_moments('白衣秀士 击败 已击败 胜利') in [01:20:00 ~ 01:25:00]:
[01:20:53] On-screen dialogue reads '小师父，天色已晚' (Little master, the sky is getting late) while the Destined One stands armed with a …
[01:20:58] The Destined One speaks with subtitle '到底是什么居心啊？ ' (What is the ulterior motive?) while the White-robed Swordsman stands ready with …

[Round 7]
TOOL : search_moments
ARGS : keywords = 灵虚子 波里个浪 广谋 白衣秀士 击败 已击败
OBS : search_moments('灵虚子 波里个浪 广谋 白衣秀士 击败 已击败') :
[00:41:21] The screen displays the status text '已击败' (Defeated) and the boss's full title '头目 · 广智' (Head of the Pack • Guangzhi) immediately …
[02:59:29] A large black calligraphic 'Victory' symbol (得胜) fills the screen with overlaid text '已击败' (Defeated) and '头目·沙国王子子' …

[Round 8] ANSWER
{"action":"answer","answer":"1, 4, 2, 3","reasoning":"Timeline evidence: 灵虚子 defeated ~00:52, 波里个浪 fight ~01:03–01:04, 广谋 defeated ~01:13–01:14, 白衣秀士 fight ~01:20–01:24, so order is 灵虚子 → 波里个浪 → 广谋 → 白衣秀士."}

Figure 8: Reasoning and Skill Scale Calling Example on the MMLifelong (day).

Benchmark	Frame caption	Episode boundary	Episode summary	Moment extraction	Pattern aggregation
MMLifelong-day	DETAILED_THIRD_PERSON	GAME	THIRDPERSON	PROMPT_QA	MATCH_GAME
MMLifelong-week	EGOLIFE	FIRST_PERSON	FIRST_PERSON	PROMPT_QA	MATCH_LIFE
MMLifelong-month	DETAILED_THIRD_PERSON	GAME	THIRDPERSON	PROMPT_QA	MATCH_GAME
EgoLife	EGOLIFE	FIRST_PERSON	FIRST_PERSON	PROMPT	CREATE_FIRST_PERSON
HoloAssist	HOLOASSIST	HOLOASSIST	HOLOASSIST	PROMPT_HOLOASSIST	CREATE_FIRST_PERSON
CaptainCook4D	HOLOASSIST	HOLOASSIST	HOLOASSIST	PROMPT_HOLOASSIST	CREATE_FIRST_PERSON
OVO-Bench	OVOBENCH	OVOBENCH	OVOBENCH	PROMPT_OVOBENCH	CREATE_OVOBENCH
OVO-Forward	FORWARD_EGO	OVOBENCH	FORWARD	PROMPT_FORWARD	CREATE_OVOBENCH
StreamingBench	V2	PROMPT	PROMPT	PROMPT_QA	CREATE
ESTP-Bench	ESTP_QAWARE_V2	ESTP	ESTP	PROMPT_ESTP	CREATE_ESTP

Table 21: Prompt used by each benchmark at each construction stage. Common prefixes (FRAME_CAPTION_WINDOW_, EPISODE_BOUNDARY_STREAM_, EPISODE_SUMMARY_, FACT_EXTRACTION_WITH_ROLES_, TOPIC_) and the _PROMPT suffix are omitted.

```
  "blue object", "red item", "large thing",
  "someone" (when the person is visible -- describe them
).

If an object is unclear, use its closest specific
category + visible attributes:
  GOOD: "cylindrical silver microphone on black stand",
"rectangular LED scoreboard on wall".
  BAD:  "a device", "a black item", "some equipment".
Every concrete noun must carry at least one visible
attribute (color / material / shape / number).

For EACH frame, record all that is visible:

1. PEOPLE (highest priority)
   - Count every visible person.
   - For each prominent one: role cue ("player on the
right", "referee in black"), clothing colors/pattern/
jersey number/logos/flags, body orientation, pose, hands
 (what is held/touched), facial expression only if
clearly visible.
(The remaining content is omitted for brevity.)
```

FRAME_CAPTION_WINDOW_EGOLIFE_PROMPT

Stage / entry: Frame Caption

```
You are a visual evidence recorder for a first-person
daily-life video segment (EgoLife dataset, camera-wearer
's point of view).

Input: {num_frames} frames sampled every {interval_sec}s
 from a ~{window_seconds}s window.
Absolute time of this window: {abs_start} -> {abs_end}
(format: DAYd-HH:MM:SS)

You MUST produce BOTH:
  (A) Per-frame captions -- one for each frame, EACH
prefixed with its absolute timestamp.
  (B) A single overall `description` covering everything
 that appears in any per-frame caption.

------------------------------------------------------------

First-person Viewpoint
------------------------------------------------------------
The camera is worn by the user. Refer to them as "the
camera-wearer" (first person allowed).
Describe other people by visible attributes / role (e.g.
 "a woman in a pink shirt", "the person across the table
").

------------------------------------------------------------

Anti-Repetition Rule (READ THIS -- hard requirement)
------------------------------------------------------------


Each frame caption MUST describe what is UNIQUE in that
frame. If frame 3 and frame 5 show
the same posture, still write them differently --
describe different objects noticed, different
gaze direction, different hand micro-actions, different
background details.

NEVER use these filler phrases -- they are banned:
(The remaining content is omitted for brevity.)
```

FRAME_CAPTION_WINDOW_V2_PROMPT

Stage / entry: Frame Caption

```
You are a visual evidence recorder for a short video
segment.

Input: {num_frames} frames sampled uniformly (~2 seconds
 apart). Each frame is an INDEPENDENT visual evidence
unit.

You MUST produce BOTH:

  (A) Per-frame captions -- one concrete caption per
sampled frame, independent of other frames.
  (B) A single overall description of the whole ~{
window_seconds} second window -- which MUST cover every
key point appearing in ANY per-frame caption (subjects,
actions, objects, on-screen text, scores, state
transitions).

Whatever appears in the per-frame captions, an
aggregated form must also appear in the description.
Downstream episode boundary detection ONLY reads the
description, so any detail dropped from it is
effectively invisible.
```

```
------------------------------------------------------------

Viewpoint & Forbidden Words
------------------------------------------------------------



No camera-wearer framing: never say "the user", "I", "my
 hand". Refer to visible people by role/attribute ("the
player on the right", "the host in the black suit", "the
 teacher at the whiteboard").

## Naming Policy (STRICT -- applies to BOTH frame
captions AND description)

NEVER invent or guess a person's name, team, nationality
, or identity.
You may use a proper name for a person ONLY when that
exact name is LEGIBLY VISIBLE on-screen in the same
window -- for example:
  * a jersey name patch ("Ma Long" printed on the back
of the shirt)
  * a scoreboard entry ("FAN ZHENDONG 2 | MA LONG 1")
  * a name tag / lower-third caption ("Sarah Chen, Head
Chef")
(The remaining content is omitted for brevity.)
```

FRAME_CAPTION_WINDOW_OVOBENCH_PROMPT

Stage / entry: Frame Caption

```
You are a visual evidence recorder for a short video
segment.

Input: {num_frames} frames sampled every {interval_sec}s
 from a ~{window_seconds}s window.
Absolute time of this window: {abs_start} -> {abs_end}
(format: DAYd-HH:MM:SS)

The segment may be either FIRST-PERSON (camera worn by
the actor; you see hands or
forearms reaching from the bottom edge of the frame; you
 never see the full body of
the person performing the action) OR THIRD-PERSON (
camera films a visible actor; you
can see the actor's whole body or upper body). Both
types share the same recording
rules below -- only the wording for the actor changes.

VIEWPOINT INFERENCE (judge once for the whole window):
  - Hands / forearms come from the bottom edge AND no
full body of that actor is
    visible -> first-person.
  - The actor's body is visible in the frame, filmed
from outside -> third-person.
  - Apply the SAME judgement to all frames in this
window (viewpoint does not
    change mid-window).

WORDING:
  - First-person: refer to the actor as "the camera-
wearer". Do NOT use "I", "my",
    "the user". Describe their hands / arms / lap / feet
 when visible.
  - Third-person: refer to the actor by visible
attributes ("the woman in the red
    apron", "the chef with grey hair", "the child in a
blue t-shirt") or role label
    ("the cook", "the teacher"). Use proper names ONLY
when legibly visible
    on-screen (name tag / subtitle / sign). Never invent
 identities, nationalities,
(The remaining content is omitted for brevity.)
```

FRAME_CAPTION_WINDOW_FORWARD_EGO_PROMPT

Stage / entry: Frame Caption

```
You are a visual evidence recorder for a first-person (
camera-wearer POV) video segment.

A user is watching this video and has posed the
following multi-choice question(s).
This is OFFLINE evaluation: the question and ALL OPTIONS
 are already shown to you.
Your captions will later be used to pick the correct
option, so you MUST help
disambiguate between the options.

USER_QUESTIONS (with all options listed):
{questions_block}

OPTION-AWARE CAPTIONING (gentle guidance, not forced):
Read the listed options once at the start. If they hinge
 on specific verbs
or objects (e.g. "throw vs place", "block vs cup"), keep
 those words in
mind so you can pick the most precise verb when an
ambiguous action occurs.

Two principles:
  1. **Be precise when the action is genuinely ambiguous
 between options.**
     If the wearer might be "throwing" or "placing",
choose the verb that
     best matches the visual evidence (force /
trajectory / control). A
     quick clarifying clause is fine ("places the block,
 not a throw --
     controlled deposit").
  2. **Do NOT over-interpret clear or unrelated actions
.**
     If the wearer is clearly doing something unrelated
to all options
     (e.g. just walking, looking around), describe it
naturally. Do NOT
     force every frame to mention option verbs. Do NOT
add negations like
(The remaining content is omitted for brevity.)
```

FRAME_CAPTION_WINDOW_ESTP_QAWARE_V2_PROMPT

Stage / entry: Frame Caption

```
You are writing FIRST-PERSON egocentric captions for an
ESTP-Bench daily-life video. You ARE the camera-wearer:
always write as "I" ("I pick up the axe", "a white
bucket sits to my left"). NEVER write "the camera-wearer
", "the user" or "the person".

============================================================
```

```
ACTIVE USER QUESTION(S) FOR THIS VIDEO (known before the
 video starts - legitimate to use):
{questions_block}

FOCUS (not a filter): give EXTRA detail to any object,
hand action, on-screen text,
spatial relation, or state change relevant to answering
these questions - exact object
names with colour/material, exact readable text,
positions relative to me ("to my left",
"in front of me"), and before/after states. Still
describe every other clearly visible
thing objectively; never fabricate; do NOT answer the
questions and do NOT mention them.
============================================================


Input: {num_frames} frames sampled every {interval_sec}s
 from a ~{window_seconds}s window.
Absolute time of this window: {abs_start} -> {abs_end}
(format: DAYd-HH:MM:SS)

OUTPUT STRUCTURE (both parts, dense then summary):
1. "dense_caption": one entry PER FRAME keyed by its
absolute timestamp (DAYd-HH:MM:SS),
   each a SHORT factual first-person sentence of what I
am doing / what is visible AT
   THAT SECOND. These per-second anchors are used for
precise answer timing - make the
   MOMENT an action starts / an object appears / a state
 changes explicit at the right key.
2. "description": ONE short first-person sentence
summarising the window.

STYLE (GT-aligned, STRICT):
- Short factual sentences. One action or observation per
 sentence.
(The remaining content is omitted for brevity.)
```

FRAME_CAPTION_WINDOW_HOLOASSIST_PROMPT

Stage / entry: Frame Caption

```
You are a visual evidence recorder for a first-person
tool-operation video segment (HoloAssist dataset).

Input: {num_frames} frames sampled every {interval_sec}s
 from a ~{window_seconds}s window.
Absolute time of this window: {abs_start} -> {abs_end}

You MUST produce BOTH (A) per-frame captions and (B) an
overall `description`. Captions must be
maximally informative -- every observable detail of
hands / tools / parts / contact / posture
that could later be queried.

------------------------------------------------------------

First-person Viewpoint
------------------------------------------------------------


The camera is worn by the user (the "camera-wearer") who
 is performing a hands-on assembly /
disassembly / repair / setup task. Refer to them as "the
 camera-wearer" (first person allowed).
Describe an instructor or second person, if visible, by
attributes ("the instructor on the
right in a blue shirt", "a hand reaching from the left
side wearing a black wristband").

------------------------------------------------------------

Anti-Repetition Rule
------------------------------------------------------------


Each frame caption MUST describe a DIFFERENT micro-state
 of the task. Even a 2-second interval
usually contains a distinguishable change in grip /
angle / position / contact. Banned filler
phrases: "still holding", "same as before", "unchanged",
 "continues to", "remains".
(The remaining content is omitted for brevity.)
```

Adaptive episode segmentation.

EPISODE_BOUNDARY_STREAM_GAME_PROMPT

Stage / entry: Episode Boundary

```
You are detecting episode boundaries in a third-person
GAMEPLAY video stream (Black Myth: Wukong, a Chinese
action-RPG playthrough), one 30-second chunk at a time.

An "episode" is ONE coherent GAMEPLAY UNIT. The natural
units in this game are:
  * A BATTLE: from engaging/locking onto an enemy or a
boss HP-bar appearing, through the fight, until that
enemy/boss is DEFEATED (or the player permanently
abandons the area). Player DEATH + respawn + re-
challenging the SAME boss is the SAME battle episode --
NEVER split on death/revival; a hard boss fight with 10
deaths is still ONE episode. Brief phase-transition
cinematics or dialogue lines DURING a fight are part of
the fight. One whole fight = ONE episode, even if it
lasts many minutes. Do NOT split an ongoing fight.
  * A STORY / DIALOGUE / CUTSCENE: an NPC conversation,
a narrated scroll, a chapter-title card, a
transformation cutscene. One continuous conversation/
cutscene = ONE episode -- consecutive dialogue windows
with the same NPC(s) are the SAME unit even when camera
framing changes between close-up and wide shots.
  * An EXPLORATION / TRAVERSAL segment: moving through
an area, looting chests, finding shrines, with no
sustained combat.
  * A MENU / SYSTEM segment: skill tree, equipment,
forging, crafting, spirit/gourd management.

You make a BINARY decision about the NEW CHUNK as a
whole:
  * `should_end=false` -- NEW CHUNK continues the SAME
gameplay unit in CURRENT BUFFER.
  * `should_end=true`  -- NEW CHUNK starts a DIFFERENT
gameplay unit (finalize current buffer).

## Decision Procedure

STEP 1 -- CONTINUITY CHECK (default = false)
Is the NEW CHUNK the same gameplay unit as CURRENT
BUFFER?
  * Same boss/enemy fight still ongoing (same boss name
in HP-bar, same arena) -> false.
```

```
  * Same conversation/cutscene continuing (same NPC,
same dialogue thread) -> false.
  * Still exploring the same area with no unit change ->
 false.
  * Still in menus -> false.
Brief camera swings, spell effects, dodges, HP-bar
flicker, subtitle changes WITHIN the same fight/
conversation are NOT boundaries.

STEP 2 -- BOUNDARY CHECK (only if the unit genuinely
changed)
Output `should_end=true` only when one clearly holds:
  (a) COMBAT TRANSITION: a fight TRULY ended (boss
DEFEATED / "[zh]" shown) and the NEW CHUNK is now
exploration, menu, dialogue, or a DIFFERENT boss/enemy
engagement. Death+respawn against the SAME boss is NOT a
 transition.
(The remaining content is omitted for brevity.)
```

EPISODE_BOUNDARY_STREAM_FIRST_PERSON_PROMPT

Stage / entry: Episode Boundary

```
You are detecting episode boundaries in a first-person (
camera-wearer POV) video stream, one chunk at a time.

An "episode" is a coherent ACTIVITY of the camera-wearer
 in a coherent LOCATION. An episode ends ONLY when the
wearer's primary activity OR room/location fundamentally
 changes.

You make a BINARY decision about the NEW CHUNK as a
whole:
  * `should_end=false` -- NEW CHUNK continues the
current episode in CURRENT BUFFER.
  * `should_end=true`  -- NEW CHUNK starts a new episode
 (finalize current buffer).

## Inputs You Will See (ONLY frame-level captions; no
summaries)

1. CURRENT BUFFER -- recent first-person frame captions,
 covering the last {recent_k} windows
   of the in-progress episode.
2. NEW CHUNK -- first-person frame captions for the
incoming chunk.

Frame captions are camera-wearer POV: hands / held
objects / posture / gaze; other people by
visible role + attributes; key objects with attributes;
concrete room/location phrase.

## Critical Principle: Judge by STABLE ANCHORS Across
Frames

In first-person footage, stable anchors persist for
minutes -- the wearer rarely teleports.
Focus on anchors that persist:
  * ROOM / LOCATION (kitchen / living-room sofa /
bedroom desk / hallway / outdoor street / car /
restaurant)
  * HIGH-LEVEL ACTIVITY (cooking / eating / drinking /
using laptop for work / studying / reading /
    watching TV / playing a video game / scrolling phone
 / chatting / cleaning / commuting)
  * KEY OBJECTS / DEVICES IN USE (the same laptop, mug,
console, tool)
(The remaining content is omitted for brevity.)
```

EPISODE_BOUNDARY_STREAM_PROMPT

Stage / entry: Episode Boundary

```
You are detecting episode boundaries in a streaming
video, one 30-second chunk at a time.

An "episode" is a semantically coherent activity segment
. An episode ends ONLY when the primary activity,
location, subject(s), or event thread fundamentally
changes.

You make a BINARY decision about the NEW CHUNK as a
whole:
  * `should_end=false` -- NEW CHUNK continues the
current episode in CURRENT BUFFER.
  * `should_end=true`  -- NEW CHUNK starts a new episode
 (finalize current buffer).

## Inputs You Will See (ONLY frame-level captions; no
summaries)

1. CURRENT BUFFER -- recent per-2s frame captions,
covering the last {recent_k} windows
   of the in-progress episode.
2. NEW CHUNK -- per-2s frame captions for the incoming
10s window.

Frame captions are rich and specific (people, clothing,
objects, on-screen text, setting).
Use them directly.

## Critical Principle: Judge by STABLE ANCHORS Across
Frames

Broadcasts / livestreams / variety shows frequently cut
between angles:
  wide shot -> close-up -> scoreboard -> crowd ->
commentator -> back to action.
Individual frames from different angles can read very
differently, but the
underlying episode is the same. Focus on anchors that
persist across frames:
  * scene elements (court / studio / classroom / kitchen
 / venue signage)
  * on-screen text (scoreboards, hashtags, channel bugs,
 subtitles, name tags)
(The remaining content is omitted for brevity.)
```

EPISODE_BOUNDARY_STREAM_OVOBENCH_PROMPT

Stage / entry: Episode Boundary

```
You are detecting episode boundaries in a video stream,
one chunk at a time.

The video may be either FIRST-PERSON (camera worn by an
actor; you see their hands /
forearms reaching from the bottom edge; no full body of
the actor visible; sources
include head-mounted Ego4D and robot-navigation OpenEQA)
 OR THIRD-PERSON (camera
```

```
films a visible actor performing a procedural task such
as cooking, crafting,
demonstrating, or playing; sources include cooking
tutorials, classroom
demonstrations, and home videos). You must judge
viewpoint from the frame captions
themselves and apply the same rules to both.

An "episode" is a coherent ACTIVITY in a coherent
LOCATION. An episode ends ONLY
when the primary activity OR room/location fundamentally
 changes.

You make a BINARY decision about the NEW CHUNK as a
whole:
  * `should_end=false` -- NEW CHUNK continues the
current episode in CURRENT BUFFER.
  * `should_end=true`  -- NEW CHUNK starts a new episode
 (finalize current buffer).

## Inputs You Will See (ONLY frame-level captions; no
summaries)

1. CURRENT BUFFER -- recent frame captions, covering the
 last {recent_k} windows
   of the in-progress episode.
2. NEW CHUNK -- frame captions for the incoming chunk.

## Critical Principle: Judge by STABLE ANCHORS Across
Frames
(The remaining content is omitted for brevity.)
```

EPISODE_BOUNDARY_STREAM_ESTP_PROMPT

Stage / entry: Episode Boundary

```
You are detecting episode boundaries in a FIRST-PERSON (
camera-wearer POV) egocentric video stream (ESTP-Bench),
 one chunk at a time.

This is STRICTLY first-person: the camera is worn by the
 actor; you see their
hands/forearms from the bottom edge, never their full
body. The whole clip often
takes place in ONE physical scene, so DO NOT split on
visual scene appearance alone.

An "episode" here is a coherent **ACTION / TASK / SUB-
GOAL** of the camera-wearer.
An episode ends when the camera-wearer SWITCHES to a
different action, task, or
sub-goal -- NOT merely when the camera pans, lighting
changes, or a new object enters
view within the same ongoing task.

You make a BINARY decision about the NEW CHUNK as a
whole:
  * `should_end=false` -- NEW CHUNK continues the SAME
task/sub-goal in CURRENT BUFFER.
  * `should_end=true`  -- NEW CHUNK starts a DIFFERENT
task/sub-goal (finalize buffer).

## Inputs You Will See (ONLY frame-level captions; no
summaries)

1. CURRENT BUFFER -- recent frame captions, covering the
 last {recent_k} windows of
   the in-progress episode.
2. NEW CHUNK -- frame captions for the incoming chunk.

## Critical Principle: Judge by the CAMERA-WEARER'S TASK
, not by visuals

Anchors that persist for an episode:
  * THE CURRENT SUB-GOAL / TASK (e.g. "making coffee", "
assembling the bracket",
(The remaining content is omitted for brevity.)
```

EPISODE_BOUNDARY_STREAM_HOLOASSIST_PROMPT

Stage / entry: Episode Boundary

```
You are detecting episode boundaries in a first-person
tool-operation video stream (HoloAssist), one chunk at a
 time.

An "episode" here is ONE coherent SUB-STEP of a hands-on
 assembly / disassembly / repair /
setup task. Sub-steps are SHORT (typically 20-90 seconds
). A 3-5 minute video usually contains
6-12 sub-step episodes. UNDER-splitting (giant 3-min
episodes covering many sub-steps) is much
worse than over-splitting for fact extraction quality.

You make a BINARY decision about the NEW CHUNK as a
whole:
  * `should_end=false` -- NEW CHUNK continues the SAME
sub-step from CURRENT BUFFER.
  * `should_end=true`  -- NEW CHUNK starts a NEW sub-
step (finalize current buffer).

## Inputs You Will See

1. CURRENT BUFFER -- recent frame captions covering the
last {recent_k} windows of the
   in-progress sub-step.
2. NEW CHUNK -- frame captions for the incoming chunk.

Each caption describes hand pose / tool / part / contact
 / motion of the camera-wearer.

## What Counts as a Sub-Step Boundary (output `
should_end=true`)

Trigger a boundary when ANY of these is clearly visible
in NEW CHUNK relative to CURRENT BUFFER:

(A) TOOL SWITCH -- wearer puts down the previously-held
tool and picks up a different tool
    (e.g. screwdriver -> tweezers; hand -> wrench; tool
-> bare hand).
(The remaining content is omitted for brevity.)
```

Episode summary and description.

EPISODE_SUMMARY_THIRDPERSON_PROMPT

Stage / entry: Episode Summary

```
You are summarizing a video episode (OVO-Bench).

This episode is STRICTLY THIRD-PERSON: a visible
protagonist/player/streamer performs the activity,
```

```
observed by an external camera. NEVER use 'I', 'me', 'my
', or 'camera-wearer'. Always describe the subject in
third person (e.g. 'the player', 'the streamer', 'the
Destined One').

## Your Task

Generate a structured summary whose four fields serve
DIFFERENT retrieval purposes.
Do not repeat content across fields.

- `subject`     -> searchable title for exact-match
retrieval.
- `summary`     -> topic-matching field; matches this
episode to a broader thread.
                  Focus on WHO + WHAT + CONTEXT at a
high level.
- `keywords`    -> BM25 / embedding recall anchors.
- `description` -> full chronological factual replay;
this is what downstream QA
                  searches for specifics.

## Output

Respond with a JSON object:
'''json
{{
    "subject": "Short title (5-10 words) naming the
specific event. Examples: 'Camera-wearer chopping
vegetables on kitchen counter', 'Cook adding seasoning
to simmering pan', 'Camera-wearer assembling shelf with
allen wrench', 'Person dealing cards on a black table
'.",
    "summary": "{summary_min_sentences}-{
summary_max_sentences} sentences. WHO is acting (camera-
wearer for first-person, or describe the visible actor
by clothing/role for third-person), WHAT primary
activity, WHERE / in WHAT setting, and KEY OBSERVABLE
EVENTS (object placements, state changes, on-screen text
 transitions, camera transitions to a new task step).",
    "keywords": ["keyword1", "keyword2", ...],
    "description": "Chronological narrative of every
observable action / state / transition. Include specific
 object names with attributes (color/material/state),
the actor's role/identity, the setting, and timestamps.
Preserve visible cause-effect chains within this episode
."
(The remaining content is omitted for brevity.)
```

EPISODE_SUMMARY_FIRST_PERSON_PROMPT

Stage / entry: Episode Summary

```
You are summarizing a first-person (camera-wearer POV)
video episode -- a semantically coherent activity
segment of one wearer.

This episode may be from any of these first-person scene
 types (and many others):
  daily life / cooking / eating / commuting / shopping /
 working at a computer / studying .
  gaming / watching TV / phone use . social interaction
with family or colleagues .
  sleeping or going to bed . tool-operation tasks (
assembly, repair, calibration) .
  outdoor activity / sports practice

Do NOT assume a single scene type -- read the captions
and decide accordingly.

## Your Task

Generate a structured summary whose four fields serve
DIFFERENT retrieval purposes.
Do not repeat content across fields -- each one has a
specific downstream use:

- `subject`     -> searchable title for exact-match
retrieval.
- `summary`     -> **topic-matching field**: matches
this episode to a broader topic/scenario. WHO + WHAT +
WHERE at a high level. Do NOT exhaust details here.
- `keywords`    -> BM25 / embedding recall anchors.
- `description` -> **full factual replay**:
chronological record covering every observable
                  action, object, and transition with
timestamps -- this is what downstream QA
                  reads directly INSTEAD of the raw
frame captions. It is the primary entry-point.

## Output

Respond with a JSON object:
(The remaining content is omitted for brevity.)
```

EPISODE_SUMMARY_PROMPT

Stage / entry: Episode Summary

```
You are summarizing a video episode -- a semantically
coherent activity segment.

This episode may be from any of these scene types (and
many others):
  sports broadcast . film / TV clip . talk show . news /
 interview . live performance .
  cooking / tutorial / how-to . documentary / nature .
surveillance / dashcam .
  lecture / classroom . gaming livestream . vlog /
lifestyle . stage performance .
  museum tour . meeting / conference

Do NOT assume a single scene type -- read the captions
and decide accordingly.

## Your Task

Generate a structured summary whose four fields serve
DIFFERENT retrieval purposes.
Do not repeat the same content across fields -- each one
 has a specific downstream use:

- `subject`  -> searchable title for exact-match
retrieval.
- `summary`  -> **topic-matching field**: matches this
episode to a broader topic/scenario. Focus on WHO + WHAT
 + CONTEXT at a high level. Do NOT exhaust details here.
- `keywords` -> BM25 / embedding recall anchors.
- `description` -> **full factual replay**:
chronological record of every observable action, object,
```

```
 and transition -- this is what downstream QA searches
for specifics.

## Output

Respond with a JSON object:
'''json
{{
(The remaining content is omitted for brevity.)
```

EPISODE_SUMMARY_OVOBENCH_PROMPT

Stage / entry: Episode Summary

```
You are summarizing a video episode (OVO-Bench).

The episode may be FIRST-PERSON (camera-wearer
perspective; you see their hands /
arms; sources: Ego4D / OpenEQA) OR THIRD-PERSON (a
visible actor performs the
activity; sources: cooking tutorial, instructional demo,
 home video). Read the frame
captions and decide automatically -- both viewpoints
share the same recording rules.

## Your Task

Generate a structured summary whose four fields serve
DIFFERENT retrieval purposes.
Do not repeat content across fields.

- `subject`     -> searchable title for exact-match
retrieval.
- `summary`     -> topic-matching field; matches this
episode to a broader thread.
                  Focus on WHO + WHAT + CONTEXT at a
high level.
- `keywords`    -> BM25 / embedding recall anchors.
- `description` -> full chronological factual replay;
this is what downstream QA
                  searches for specifics.

## Output

Respond with a JSON object:
'''json
{{
    "subject": "Short title (5-10 words) naming the
specific event. Examples: 'Camera-wearer chopping
vegetables on kitchen counter', 'Cook adding seasoning
to simmering pan', 'Camera-wearer assembling shelf with
allen wrench', 'Person dealing cards on a black table
'.",
(The remaining content is omitted for brevity.)
```

EPISODE_SUMMARY_FORWARD_PROMPT

Stage / entry: Episode Summary

```
You are summarizing a video episode for FORWARD /
PROACTIVE question answering.

A user posed the following question(s) BEFORE this
episode. While summarizing,
also report each question's answer status as of the END
of this episode.

USER_QUESTIONS (active throughout the video):
{questions_block}

The episode may be FIRST-PERSON or THIRD-PERSON -- read
the frame captions and
follow their wording for the actor.

## Your Task

Generate a structured summary with FIVE fields:
- `subject`         -> short searchable title.
- `summary`         -> high-level WHO/WHAT/WHERE
narrative.
- `keywords`        -> BM25 anchors.
- `description`     -> full chronological factual replay
.
- `question_status` -> for each active question, the
answer status as of the END
                      of this episode.

## Output (STRICT JSON)

'''json
{{
(The remaining content is omitted for brevity.)
```

EPISODE_SUMMARY_ESTP_PROMPT

Stage / entry: Episode Summary

```
You are summarizing a FIRST-PERSON (camera-wearer POV)
egocentric video episode (ESTP-Bench) -- a coherent
action/task segment performed by the camera-wearer.

This is STRICTLY first-person: refer to the actor as "
the camera-wearer"; describe
object positions RELATIVE TO THE CAMERA-WEARER ("to my
left", "in front of me").

## Your Task

Generate a structured summary whose four fields serve
DIFFERENT retrieval purposes.
Do not repeat content across fields.

- `subject`     -> searchable title for exact-match
retrieval.
- `summary`     -> topic-matching field; matches this
episode to a broader thread.
                  Focus on the camera-wearer's TASK +
KEY EVENTS at a high level.
- `keywords`    -> BM25 / embedding recall anchors.
- `description` -> full chronological factual replay;
this is what downstream QA
                  searches for specifics (when did I X,
before/after, state changes).

## Output

Respond with a JSON object:
'''json
{{
    "subject": "Short title (5-10 words) naming the
specific egocentric task. Examples: 'Camera-wearer
```

```
brewing pour-over coffee at counter', 'Camera-wearer
assembling a wooden bracket at workbench', 'Camera-
wearer filling out a paper form at desk'.",
    "summary": "{summary_min_sentences}-{
summary_max_sentences} sentences. WHAT task the camera-
wearer is doing, the ordered KEY SUB-ACTIONS, the KEY
OBJECTS (with attributes) and their position relative to
 me, any OBJECT STATE CHANGES, any OBJECT FUNCTION/USE
revealed, and any on-screen text.",
    "keywords": ["keyword1", "keyword2", ...],
(The remaining content is omitted for brevity.)
```

EPISODE_SUMMARY_HOLOASSIST_PROMPT

Stage / entry: Episode Summary

```
You are summarizing a first-person tool-operation
episode (HoloAssist).

## Your Task

Produce a structured summary focused on HANDS + TOOL +
PART + OUTCOME + IRREGULARITY.

Return JSON:
```json
{{
    "subject": "Short title (5-10 words) naming the sub-
step. Examples: 'Inserting battery into GoPro body', '
Aligning lens cap onto camera front'.",
    "summary": "{summary_min_sentences}-{
summary_max_sentences} sentences. Cover: TOOL(S) used,
PART(S) operated on, ACTION(S) performed in sequence,
OUTCOME (sub-step completed / abandoned / in-progress),
and an explicit IRREGULARITY flag for the whole episode
-- either 'no irregularity observed' or one-line
description of the observed error / hesitation / re-
attempt.",
    "keywords": ["tool names", "part names", "action
verbs", "error-type-keyword-if-any", ...],
    "description": "Chronological narrative of every
action. For each: timestamp [DAYd-HH:MM:SS], hand(s)
involved, tool, part, action verb, and whether action
appears correct or incorrect."
}}
```

Hard Requirements

1. `summary` MUST explicitly include an irregularity
statement. Examples:
     "... observed irregularity: wrong screw inserted at
 00:02:14, corrected 3 seconds later."
     "... observed irregularity: none."

2. Use first-person / camera-wearer framing. ("The
camera-wearer picks up the black screwdriver ...")

3. No invented named entities. Tool / part names come
from what is visible (shape, colour, on-body label).
(The remaining content is omitted for brevity.)
```

Moment extraction with roles and weights.

FACT_EXTRACTION_WITH_ROLES_PROMPT_QA

Stage / entry: Moment/Fact Extraction

```
You are extracting atomic facts from a video episode AND
 assigning each fact a role + importance weight in a
single pass.

These facts will be used to answer questions about video
 content:
- WHO did WHAT (actions, interactions)
- WHAT objects are present and their attributes (color,
shape, material, state)
- WHERE things are (spatial relations, location)
- WHEN things happened (timestamps)
- WHAT text / scores / graphics are on screen
- Counting, identity, sequence

## Extraction Principles

Each fact must be:
- **Self-contained**: WHO, WHAT, WHEN, WHERE packed into
 the fact content itself.
- **Concretely attributed**: every concrete noun carries
 a visible attribute (color / material / shape / number
/ role).
- **Timestamped (REQUIRED)**: the `temporal` field is
MANDATORY -- always emit a non-empty string in format `
DAY{{N}}-HH:MM:SS`, pointing to the most specific
visible moment for this fact. If a fact spans a range,
use the start time. Do not omit or leave empty.
- **Visually grounded**: do NOT invent. If not
observable in the frames, do NOT output it.
- **Scoped to THIS episode**: never reference events
from prior episodes; each fact is about something that
happened within the frame captions provided below.

### Two-pass coverage (apply mentally)
- **Pass 1**: per-frame actions and observations.
- **Pass 2**: connected facts spanning multiple frames (
e.g. object placed -> object retrieved).

### What to prioritize (any of these that apply):
- Actions with clear actors (by specific role, e.g. "
player on the right wearing blue jersey number 3").
(The remaining content is omitted for brevity.)
```

FACT_EXTRACTION_WITH_ROLES_PROMPT_OVOBENCH

Stage / entry: Moment/Fact Extraction

```
You are extracting atomic facts from a video episode (
OVO-Bench) AND assigning each fact a role + importance
weight in a single pass.

These facts will be used to answer Backward Tracing
questions:
  - EPM (episodic memory): "what colour is the X on the
Y?" / "where did I put Z?"
  - ASI (action sequence): "what did the person do
before/after X?"
  - HLD (hallucination detection): selecting "Unable to
answer" when the asked-about
    thing is not actually present in the video.

The episode may be FIRST-PERSON (camera-wearer
perspective; sources Ego4D /
OpenEQA) OR THIRD-PERSON (visible actor; sources cooking
 tutorial, instructional
```

```
demo, home video). Read the frame captions and decide
automatically -- both share
the same extraction rules, only the wording for the
actor changes.

## Extraction Principles

Each fact must be:
- **Self-contained**: WHO + WHAT + WHEN + WHERE in the
fact content itself.
- **Concretely attributed**: every concrete noun carries
 a visible attribute
  (color / material / shape / number / state / role).
- **Timestamped (REQUIRED)**: `temporal` is MANDATORY,
format `DAY{{N}}-HH:MM:SS`,
  copied from the most specific frame caption supporting
 the fact.
- **Visually grounded (CRITICAL for HLD)**: do NOT
invent. If something appears
  for only 1-2 frames as a passerby / background blur
and is unrelated to the
  primary activity, do NOT emit a fact for it -- those
generate hallucination
  errors when the question asks about absent objects.
(The remaining content is omitted for brevity.)
```

FACT_EXTRACTION_WITH_ROLES_PROMPT_FORWARD

Stage / entry: Moment/Fact Extraction

```
You are extracting atomic facts from a video episode AND
 assigning each fact a role + importance weight in a
single pass.

These facts will be used for FORWARD / PROACTIVE
question answering: a user
posed one or more questions BEFORE this episode, and the
 system needs to track
visual evidence over time so it can:
  - Answer evolving questions ("what is the white object
 being handled now")
    at multiple timestamps as the visual state changes.
  - Detect proactive trigger conditions ("when
scoreboard shows 40, output 40").
  - Track step / count progression ("which tutorial step
 is current", "how many
    times has X happened").

USER_QUESTIONS (active throughout the video; refer to by
 qid):
{questions_block}

The episode is {viewpoint_hint} (third-person broadcast
/ movie / tutorial OR
first-person camera-wearer POV -- read the frame
captions and follow their
wording for the actor).

## Extraction Principles

Each fact must be:
- **Self-contained**: WHO + WHAT + WHEN + WHERE in the
fact content itself.
- **Concretely attributed**: every concrete noun carries
 a visible attribute
  (color / material / shape / number / state / role).
- **Timestamped (REQUIRED)**: `temporal` is MANDATORY,
format `DAY{{N}}-HH:MM:SS`,
(The remaining content is omitted for brevity.)
```

FACT_EXTRACTION_WITH_ROLES_PROMPT_ESTP

Stage / entry: Moment/Fact Extraction

```
You are extracting atomic facts from a FIRST-PERSON (
camera-wearer POV) egocentric video episode (ESTP-Bench)
 AND assigning each fact a role + importance weight in a
 single pass.

This episode is STRICTLY FIRST-PERSON: the camera is
worn on the actor's head/body,
you see their hands/forearms reaching from the bottom
edge, never their full body.
ALWAYS refer to the actor as "the camera-wearer".
Spatial relations are described
RELATIVE TO THE CAMERA-WEARER ("to my left", "directly
in front of me", "an arm's
length away on the right") -- this egocentric framing is
 what downstream questions
test.

These facts will be used to answer egocentric questions
such as:
  - Temporal: "What was I doing right before I picked up
 the kettle?" / "What did I
    do immediately after opening the drawer?" -- answers
 depend on the EXACT order
    and timestamp of my actions.
  - State Change: "What changed about the cup after I
poured the water?" -- answers
    depend on a recorded BEFORE->AFTER transition of an
object's position / form /
    open-closed / on-off / fill-level / cleanliness.
  - Object Function / Information Function: "What is the
 blue handle on the machine
    used for?" / "What information does the display show
 me?" -- answers depend on
    the recorded USE / PURPOSE of an object or the
meaning of on-screen content.
  - Task Understanding: "What was the overall goal of
what I was doing?" -- answers
    depend on facts that chain sub-actions toward a goal
.
  - Spatial Reasoning: "Where is the red bottle relative
 to me?" -- answers depend on
    egocentric position facts.
  - Text-Rich Understanding: questions about labels /
displays / signage -- answers
    depend on exactly transcribed on-screen text.
(The remaining content is omitted for brevity.)
```

FACT_EXTRACTION_WITH_ROLES_PROMPT

Stage / entry: Moment/Fact Extraction

```
You are extracting atomic facts from egocentric video
episodes for a proactive AI assistant, AND assigning
each fact a role + importance weight in a single pass.

These facts will be used to:
- Recall what the user did and when (episodic memory)
- Detect patterns (routines, habits)
```

```
- Trigger proactive services (safety warnings, reminders
, health coaching)

## Extraction Guidelines

### Two-Pass Strategy (apply mentally)
- **Pass 1**: Extract individual facts from each frame/
action (forms the foundation)
- **Pass 2**: Extract connected facts spanning multiple
frames (supplement, don't replace Pass 1)

### What to Extract
Each fact must be:
- **Self-contained**: Include WHO, WHAT, WHEN, WHERE in
the fact itself
- **Specific**: Use exact object names, precise
timestamps, actual locations
- **Timestamped**: Always include the timestamp in
format "DAY{{N}}-HH:MM:SS"

### Service Categories
Classify each fact into one of these categories based on
 its relevance to proactive services:

| Category | Description | Examples |
|----------|-------------|----------|
| `safety` | Physical hazards, dangerous situations |
Hand near flame, wet floor, unstable objects |
(The remaining content is omitted for brevity.)
```

FACT_EXTRACTION_WITH_ROLES_PROMPT_HOLOASSIST

Stage / entry: Moment/Fact Extraction

```
You are extracting atomic facts from a first-person tool
-operation video episode (HoloAssist) AND assigning each
 fact a role + importance weight.

The episode is one sub-step of a hands-on assembly /
disassembly / repair / setup task. Your facts will be
used BOTH for retrieval (answering questions about what
happened) AND for proactive service (detecting mistakes,
 hesitations, retry patterns).

IMPORTANT -- INPUT MODALITY:
The captions describe ONLY what is visible from the
camera-wearer's first-person view.
There is NO instructor in the captions and NO audio.
Therefore "correctness" must be inferred PURELY from
visual evidence in the wearer's
own actions and tool/part interactions (misalignment,
slip, retry, abandoned action,
wrong-tool-for-step, out-of-order step, etc.). Never
write facts that reference an
instructor, verbal correction, demonstration, voice, or
any party other than the
camera-wearer.

------------------------------------------------------------

Fact Schema (STRICT)
------------------------------------------------------------


Each fact is a JSON object with these keys:

  content          : one self-contained natural-language
 sentence of what happened.
                     MUST be concrete (tool + part +
action), no pronouns, no "the user" filler
                     -- refer to the actor as "the
camera-wearer".
  temporal         : absolute timestamp "DAYd-HH:MM:SS"
at which the action occurs.
  spatial          : short location phrase if
distinguishable (e.g. "at the workbench",
                     "on the left side of the table").
null if not clear.
(The remaining content is omitted for brevity.)
```

Pattern consolidation.

TOPIC_MATCH_GAME_PROMPT

Stage / entry: Pattern/Topic Aggregation

```
You are matching a new gameplay episode to existing
topics in a video memory system for a Black Myth: Wukong
 playthrough.

A "topic" is a **specific chapter / location arc / story
 thread** of the playthrough -- e.g.
"Black Wind Mountain ([zh]) chapter: fights and
exploration up to boss [zh]", "Yellow Wind
Ridge exploration and [zh] boss arc", "Forging & skill-
tree upgrade session". A topic is NOT
the broad category "boss battles" or "combat".

## Match criteria (output `match=true` when)
- The episode is the SAME chapter/location arc
CONTINUING (e.g. still progressing through [zh]:
  more exploration or the next enemy in that same area/
chapter).
- A natural next phase of the SAME arc (exploring an
area -> the area's mini-boss -> the area's main
  boss are the SAME arc if in the same location/chapter)
.
- Returns to the same arc after a menu/cutscene detour.

## Anti-patterns (output `match=false` when)
- Only "both are boss fights" or "both are combat"
matches -- that is the broad category, NOT a
  topic. Different bosses in DIFFERENT locations/
chapters are DIFFERENT topics.
- The location/biome/chapter fundamentally changed ([zh]
 -> [zh] -> [zh]): new topic.
- A menu/forging/skill session is its own topic, not
part of a fight arc.
- **Vague / overly broad existing topic** (title like "
battles and transformations", "boss
  fights", "various combat") -- that topic was poorly
defined; prefer `match=false` so the new
  episode gets a specific chapter/location topic.

When in doubt, output false -- a new specific chapter/
location topic beats polluting a giant
catch-all "boss battles" topic.
(The remaining content is omitted for brevity.)
```

TOPIC_MATCH_LIFE_PROMPT

Stage / entry: Pattern/Topic Aggregation

```
You are matching a new episode to existing topics in a
```

```
video memory system for a FIRST-PERSON daily-life
recording (EgoLife: camera-wearer sharing a house with
roommates over multiple days).

A "topic" is ONE specific, bounded ACTIVITY THREAD -- e.
g. "Making dinner with roommates on Day2 evening", "
Assembling the white storage shelf in the living room",
"Grocery shopping trip at Hema supermarket", "Playing
the tabletop card game after lunch". A topic is NOT a
location ("kitchen stuff"), NOT a whole day, and NOT a
recurring chore category ("cleaning").

## Match criteria (`match=true`)
- The episode DIRECTLY CONTINUES the same bounded
activity (same meal still being cooked/eaten; same shelf
 still being assembled; same shopping trip).
- Returns to the same activity after a short
interruption (phone call, bathroom break) WITHIN the
same session.

## Anti-patterns (`match=false`)
- Same LOCATION but different activity (kitchen: cooking
 dinner vs washing dishes next morning = DIFFERENT).
- Same CATEGORY on a different occasion (today's lunch
vs yesterday's lunch = DIFFERENT topics; a second repair
 session hours later = DIFFERENT unless visibly the same
 unfinished job resumed).
- A very long-running vague topic (one topic absorbing
hours of unrelated episodes) -- if the existing topic
already spans many episodes and this new episode is a
different session/occasion, prefer `match=false`.
- Vague existing topic title ("daily activities", "
working on stuff") => prefer false.

When in doubt, output false -- specific bounded activity
 threads beat giant catch-alls.

## Output
```json
{{
    "matches": {{
        "topic_id_1": {{"match": true/false, "reason": "
same/different bounded activity session"}},
        "topic_id_2": {{"match": true/false, "reason":
"..."}}
    }}
}}
```
(The remaining content is omitted for brevity.)
```

TOPIC_CREATE_PROMPT

Stage / entry: Pattern/Topic Aggregation

```
You are creating a new topic for a video memory system.

A topic is a **specific, identifiable event thread or
activity line** -- ONE concrete thing happening, NOT a
broad category.

Videos in this system come from many scenes:
  sports broadcast . film / TV clip . talk show . news /
 interview . live performance .
  cooking / tutorial / how-to . documentary .
surveillance / dashcam .
  lecture / classroom . gaming livestream . vlog /
lifestyle . museum / tour .
  stage performance . meeting / conference

Do NOT assume a single scene type -- infer from the
episodes.

## Your Task

Create a topic that captures the ONE specific event
thread across the given episodes.

## Specificity (CRITICAL)

Good topic names -- specific to ONE event thread (
examples across scene types):
  * "Fan Zhendong vs Ma Long Men's Singles semifinal at
WTT Singapore Smash"  (sports)
  * "Chef Sarah Chen demonstrating pasta dough kneading
on cooking show"       (cooking)
  * "Host interviewing director Chris Nolan about
upcoming sci-fi film"       (interview)
  * "Professor Liu explaining Pythagorean theorem with
whiteboard diagrams"    (lecture)
  * "News anchor reading Q3 earnings report at studio
desk"                     (news)
  * "Stand-up comedian performing set at Comedy Club's
open mic night"         (performance)
(The remaining content is omitted for brevity.)
```

TOPIC_CREATE_OVOBENCH_PROMPT

Stage / entry: Pattern/Topic Aggregation

```
You are creating a new topic for a video memory system (
OVO-Bench).

A topic is a **specific, identifiable event thread or
activity line** -- ONE concrete
thing happening in this video, NOT a broad category.

Episodes in this system come from short videos that may
be FIRST-PERSON
(camera-wearer perspective; sources Ego4D / OpenEQA) OR
THIRD-PERSON (visible
actor performs the activity; sources cooking tutorial,
instructional demo, home
video, gaming clip). Read the episodes' subjects/
summaries to decide automatically.

## Your Task

Create a topic that captures the ONE specific event
thread across the given episodes.

## Specificity (CRITICAL)

Good topic names -- specific to ONE event thread (
examples across viewpoint types):
  * "Camera-wearer chopping vegetables and frying onions
 on kitchen stovetop"          (first-person cooking)
  * "Camera-wearer assembling a wooden shelf with allen
wrench at workbench"           (first-person tool-use)
  * "Camera-wearer browsing a recipe app on smartphone
while seated on living-room sofa" (first-person device
use)
  * "Chef in red apron demonstrating pasta dough
```

```
kneading on marble counter"           (third-person
cooking demo)
  * "Instructor in blue shirt explaining motherboard
installation to camera at workbench" (third-person
tutorial)
  * "Person in striped shirt dealing cards and managing
stacks at black gaming table"  (third-person home video)

Bad topic names (too broad -- anti-patterns):
(The remaining content is omitted for brevity.)
```

TOPIC_CREATE_ESTP_PROMPT

Stage / entry: Pattern/Topic Aggregation

```
You are creating a new topic for a FIRST-PERSON (camera-
wearer POV) egocentric video memory system (ESTP-Bench).

A topic is a **specific, identifiable task/action thread
** of the camera-wearer -- ONE
concrete thing the camera-wearer is doing in this clip,
NOT a broad category.

This is STRICTLY first-person: refer to the actor as "
the camera-wearer"; describe
object positions relative to the camera-wearer where
relevant.

## Your Task

Create a topic that captures the ONE specific egocentric
 task thread across the given
episodes.

## Specificity (CRITICAL)

Good topic names -- specific to ONE egocentric task
thread:
  * "Camera-wearer brewing pour-over coffee at the
kitchen counter"
  * "Camera-wearer assembling a wooden bracket with a
screwdriver at the workbench"
  * "Camera-wearer filling out and stamping a paper form
 at the desk"

Bad topic names (too broad -- anti-patterns):
  * "Kitchen activities", "Daily life", "Tool use", "
Various tasks", "Routine"

Separate different aspects into different topics:
  * "Making the coffee" vs "Washing the dishes" ->
DIFFERENT topics; task changed.
(The remaining content is omitted for brevity.)
```

TOPIC_CREATE_FIRST_PERSON_PROMPT

Stage / entry: Pattern/Topic Aggregation

```
You are creating a new topic for a first-person (camera-
wearer POV) video memory system.

A topic is a **specific, identifiable event thread or
activity line** in the wearer's day --
ONE concrete thing the wearer is doing, NOT a broad
category.

Episodes in this system come from first-person scenes, e
.g.:
  cooking / eating / drinking / commuting / shopping /
working at a computer / studying .
  gaming / watching TV / phone use . social interaction
with family or colleagues .
  sleeping or going to bed . tool-operation tasks (
assembly, repair, calibration) .
  outdoor activity / sports practice

## Your Task

Create a topic that captures the ONE specific event
thread across the given episodes.

## Specificity (CRITICAL)

Good topic names -- specific to ONE first-person event
thread:
  * "Camera-wearer making pour-over coffee in the
kitchen on Saturday morning"
  * "Camera-wearer playing Mario Kart on the living-room
 sofa with two friends"
  * "Camera-wearer assembling an IKEA chair on the
bedroom floor with the instructor"
  * "Camera-wearer commuting on the subway with a
backpack on the lap"
  * "Camera-wearer working on a Python script at the
living-room dining table"
  * "Camera-wearer disassembling a GoPro to insert a new
 battery and SD card"
(The remaining content is omitted for brevity.)
```

D.3 Inference Prompts

MM-Lifelong (day, week).

TOOLS_DOC_V2

Stage / entry: MM-Lifelong (day, week)

```
You answer a question about a ~25h gameplay video (Black
 Myth: Wukong) using a \
searchable HYPERGRAPH memory (no video access). Time is
the video's cumulative clock HH:MM:SS \
from 00:00:00.

You are given a CHAPTER MAP below (topic timeline) so
you already know the global structure --
use it to pick the rough time region BEFORE searching,
then drill down.

Tools (call with JSON). Prefer the hierarchy: chapter
map -> search_episodes (get precise \
time span) -> facts_in_range / count_events (drill into
that span).
  search_episodes(keywords)                    - search
2-min episode summaries; returns each \
episode's [t0~t1] TIME SPAN. Best FIRST step to locate
WHERE something happens and get t_start/t_end.
  facts_in_range(t_start, t_end)               - ALL
dense per-5s facts in a time range, in order \
(not keyword-filtered). Use after you fixed a range to
read the full story of that segment.
  search_events(keywords, t_start?, t_end?)    -
timestamped atomic action records. WHAT HAPPENED WHEN.
```

```
  count_events(keywords, t_start?, t_end?)    - count
deduplicated event instances in range. \
USE for "how many times" AFTER fixing the range.
  search_entities(keywords, t_start?, t_end?) -
inventory records with EXACT counts ("3x [zh]").
  search_ocr(keywords, t_start?, t_end?)      - verbatim
 on-screen/UI text (boss names, [zh], \
item toasts, numeric counters).
  search_facts(keywords, t_start?, t_end?)    - keyword
search over dense facts (with structured \
entity/action fields). Good coverage.

Strategy:
1. LOCATE via hierarchy: read the CHAPTER MAP; call
search_episodes(concept) to get the exact \
[t0~t1] of the relevant segment; then restrict later
calls with those t_start/t_end.
(The remaining content is omitted for brevity.)
```

ROUND_PROMPT_V2

Stage / entry: MM-Lifelong (day, week)

```
{tools}

============================================================

CHAPTER MAP (topic timeline -- global structure of the
video)
============================================================

{chapters}

Question: {question}

============================================================

RETRIEVED_EVIDENCE (from your tool calls so far)
============================================================

{evidence}

Round {rnd}/{max_rounds}. Decide:
- gather info: {{"action":"tools","calls":[{{"tool":"<
name>","args":{{"keywords":"...","t_start":"HH:MM:SS or
null","t_end":"HH:MM:SS or null"}}}}, ...]}} (max 3
calls)
- answer:      {{"action":"answer","answer":"<final
answer string>","reasoning":"<one sentence>"}}
- On the final round you MUST answer.
Output STRICT JSON only.
```

MM-Lifelong (month).

TOOLS_DOC_V3

Stage / entry: MM-Lifelong (month)

```
You answer a question about a MONTH-LONG travel vlog:
the streamer IShowSpeed's \
multi-week trip across China, recorded as ~22 separate
livestream videos (third-person view of \
Speed; questions refer to him as "the streamer" / "he").
 You use a searchable HYPERGRAPH memory \
(no video access). Time is the cumulative clock HH:MM:SS
 from 00:00:00 spanning the WHOLE trip. \
A CHAPTER MAP (topic timeline of the whole trip, city by
 city / stream by stream) is provided \
below -- because the trip is very long and questions
often span the ENTIRE trip, you SHOULD use the \
CHAPTER MAP first to locate WHICH parts of the trip are
relevant, then drill down with tools.

TOOLS (call with JSON). Each notes WHEN to use / WHEN
NOT:
  search_ocr(keywords, t_start?, t_end?)      - verbatim
 on-screen/UI text with EXACT timestamps \
(boss names, location labels [zh], item toasts, numeric
counters [zh]/[zh]/attack values). \
BEST for: exact numbers, UI readings, and ORDERING/
TIMELINE questions (its timestamps are the \
most precise).
  count_events(keywords, t_start?, t_end?)    - count
DEDUPLICATED event instances in a range. \
BEST for: "how many times did X happen" AFTER you fixed
the range. Do NOT answer a count from \
episode summaries (2-min granularity misses/merges
instances).
  search_events(keywords, t_start?, t_end?)   -
timestamped atomic action records (WHAT happened WHEN).
  search_entities(keywords, t_start?, t_end?) -
STRUCTURED OBJECT INVENTORY: every record is a \
per-moment census "COUNT x NAME [attributes] @location"
(e.g. "3x laptop [silver, open] @dining table") \
extracted frame-by-frame. BEST for: "how many X" about
OBJECTS (laptops/cups/bags/items), object \
attributes (color/state), and "what objects were at
PLACE". For object-counting questions CALL THIS \
FIRST with the object name (English AND Chinese) before
anything else -- it is the only source with \
exact per-frame object counts; captions/facts
underestimate counts.
  search_facts(keywords, t_start?, t_end?)    - keyword
search over dense per-5s facts (structured \
entity/action fields). Good general coverage; fine-
grained backstop for counts and details.
(The remaining content is omitted for brevity.)
```

ROUND_PROMPT_V3

Stage / entry: MM-Lifelong (month)

```
{tools}
{day_hint}
============================================================

CHAPTER MAP (optional global reference -- topic timeline
)
============================================================

{chapters}

Question: {question}

============================================================

RETRIEVED_EVIDENCE (from your tool calls so far)
============================================================

{evidence}

Round {rnd}/{max_rounds}. Decide:
```

```
- gather info: {{"action":"tools","calls":[{{"tool":"<
name>","args":{{"keywords":"...","t_start":"HH:MM:SS or
null","t_end":"HH:MM:SS or null"}}}}, ...]}} (max 3
calls)
- answer:      {{"action":"answer","answer":"<final
answer string>","reasoning":"<one sentence>"}}
- On the final round you MUST answer.
Output STRICT JSON only.
```

OVO-Bench (real-time, backward). Both tracks are answered by a single solver — one tool document and one round prompt, with no routing by task type. The prompt asks the agent to first decide whether the question concerns the current moment or the past.

TOOLS_DOC

Stage / entry: OVO-Bench (real-time, backward)

```
\
You are given (below) the VIDEO SCALE and the full
EPISODE SKELETON (each episode's short
summary, numbered #1..#N). Tools:
  get_recent_captions   - second-level frame captions of
 the last ~10s before the question moment,
        plus the window-level summary of that segment (
the literal CURRENT on-screen scene,
        finest detail: objects, attributes, on-screen
text, spatial layout). (No args.)
  get_recent_episodes   - full detail (summary+
description) of the LAST few episodes (the
        most recent activity before the question moment)
. (No args.)
  get_global_captions   - minute-level window summaries
of the WHOLE video so far (coarse global
        context; useful for prediction / questions
needing the broader storyline). (No args.)
  search_fact_only(semantic_query, keywords)     -
global search over atomic facts (precise detail).
  search_episode_fact(semantic_query, keywords)  -
locate the scene first, then its facts.
  search_hierarchical(semantic_query, keywords)  - top-
down topic->episode->fact (broad/vague).
  search_entities(keywords)  - point-lookup over a
structured per-moment perception log: object
        inventory ("[time] Nx object [attributes]
@location"), on-screen TEXT, and atomic events.
        One option among the search tools; handy for a
concrete object's location / count / colour
        / state, or exact on-screen text at an earlier
moment. keywords = concrete object/text terms.

PLAN THE QUERY for the search tools (do NOT copy the
question verbatim). You MUST EXPAND the query using
(The remaining content is omitted for brevity.)
```

ROUND_PROMPT

Stage / entry: OVO-Bench (real-time, backward)

```
\
You answer a question about a video using a memory graph
 (Topic>Episode>Fact).

{tools}

Question type reference (what each kind of question asks
 for):
- Spatial layout / positions of visible things;
identifying visible objects and their
  relationships; colors, shapes, sizes, materials; the
action happening now or just before;
  reading on-screen text/signs (copy VERBATIM);
predicting what happens next - these concern
  the CURRENT moment.
- Questions about PAST events (what happened earlier, in
 what order, whether something ever
  appeared) - answer from episode details / retrieved
facts, not the current scene.

Question: {question}
{options_block}

============================================================

ALL_EPISODE_SUMMARIES (chronological skeleton #1..#N)
============================================================

{all_eps}

============================================================

RETRIEVED_EVIDENCE (from your earlier tool calls)
============================================================

{evidence}

ANSWER RULES (critical):
- "answer" MUST be EXACTLY one of the options listed
above - copy the full option text.
- NEVER output "Unable to answer"/"Cannot be determined
"/"no evidence" UNLESS that exact
  choice literally appears in the options. Incomplete
evidence is NOT a reason to refuse.
- BUT if one of the options IS "Unable to answer" (or an
 equivalent "none / cannot be
  determined" choice), that option is a REAL answer:
choose it when your retrieved evidence does
(The remaining content is omitted for brevity.)
```

StreamingBench. The real-time track is answered agentically: the tool block is assembled by `_tools_and_allowed(scheme=7)`, which exposes four skills and allows up to three calls in one round. The contextual track (ACU/MCU) is answered from the assembled EPISODE skeleton with the prompt below it.

ROUND_PROMPT (scheme 7)

Stage / entry: StreamingBench (real-time)

```
\
You are a StreamingBench Real-Time Visual Understanding
QA assistant. Answer the MCQ from the evidence.
Task types: Object/Action/Attribute Perception, Spatial
Understanding, Counting, Event/Text-Rich
Understanding, Causal/Prospective Reasoning, Clips
Summarize. The answer is almost always in the
recent scene (RECENT_CAPTIONS) - what is on screen at
the question moment.

{tools_block}
```

```
QUESTION: {question}
OPTIONS:
{options_block}

{fixed}
------------------------------------------------------------

RETRIEVED_EVIDENCE (from your tool calls; empty until
you call one)
------------------------------------------------------------

{evidence}

DECISION RULES:
- ANSWER IMMEDIATELY when the evidence lets you pick
exactly one option (it usually does).
- Request a tool ONLY when a specific needed detail is
genuinely NOT in the evidence yet.
  DO NOT retrieve just to be safe - extra retrieval
usually HURTS.
- FINAL round: you MUST answer; best-supported guess
from partial evidence, never refuse. Never fabricate.

Round {round}/{max_rounds}. Output STRICT JSON only, one
 of:
  {{"decision":"answer","answer":"<the option LETTER A/B
/C/D...>","reasoning":"<one sentence>"}}
  {{"decision":"need_retrieval","calls":[{{"tool":"<name
>","args":{{...}}}}]}}
    (get_recent_caption / get_recent_episode /
get_recent_episodes args: {{}}; search_* args: {{"
semantic_query":"...","keywords":"..."}})
```

MCQ_PROMPT_DUMP_ALL

Stage / entry: StreamingBench

```
You are answering a multi-choice question about a third-
person video. The question is asked at TIME = {qtime} (
absolute video time). You can ONLY use information up to
 that time.

QUESTION: {question}

OPTIONS:
{options_block}

VIDEO MEMORY (chronological, truncated to TIME = {qtime
}):

EPISODES (subject + summary + description, only episodes
 that completed before {qtime}):
{episodes}

FACTS (atomic events <= {qtime}):
{facts}

RECENT FRAME CAPTIONS (boundary frames between the last
completed episode and {qtime}, useful for the very
latest visual state):
{frame_captions}

{prev_qa_block}

Pick the BEST option based ONLY on evidence at or before
 {qtime}. Do NOT use future content.

Output JSON: {{"answer": "A"|"B"|"C"|"D", "reasoning":
"<one sentence>"}}
Output ONLY the JSON.
```

Forward-looking tracks.

FORWARD_INFER_PROMPT

Stage / entry: Forward-looking tracks

```
You are answering a streaming / proactive question about
 a video.

A user asked the question(s) BEFORE this moment. You are
 now at TIME =
{time_now} seconds (since video start). You can ONLY see
 information from
[video_start, NOW] -- never assume future content.

QUESTION:
{question}

Task type: {task}
Task instruction: {task_description}

------------------------------------------------------------

PREVIOUS_ANSWERS for THIS question (chronological --
your own past replies):
------------------------------------------------------------

{prev_answers_block}

------------------------------------------------------------

RECENT_FRAME_CAPTIONS (last {recent_seconds}s before NOW
 -- PRIMARY signal):
------------------------------------------------------------

{recent_captions}

------------------------------------------------------------

RECENT_EPISODE_SUMMARIES (last {recent_eps_n} episodes
-- gives short-term
context of what just happened):
(The remaining content is omitted for brevity.)
```

FORWARD_INFER_FINAL_PROMPT

Stage / entry: Forward-looking tracks

```
You are giving the FINAL forward / proactive answer.
This is the last chance -- output the answer.

QUESTION:
{question}

Task type: {task} ({task_description})
TIME = {time_now} seconds.

PREVIOUS_ANSWERS for THIS question:
{prev_answers_block}

ORIGINAL RELEVANT_FACTS:
{relevant_facts}
```

```
ADDITIONAL FACTS retrieved this round:
{retrieved_facts}

RECENT_FRAME_CAPTIONS:
{recent_captions}

You MUST output the answer in the format expected for
the task type:
- REC: single integer.
- SSR: "Yes" or "No".
- CRR: "Yes" or "No".
(The remaining content is omitted for brevity.)
```

FORWARD_AGENTIC_TOOLS_DOC

Stage / entry: Forward-looking tracks

```
\
You are answering a STREAMING / PROACTIVE question at a
fixed moment in a video, using
a memory graph (Topic > Episode > Fact). You can ONLY
see [video_start, NOW] -- never the future.

You are ALWAYS given (below, as fixed context -- NOT
tool calls):
  - ALL_EPISODE_SUMMARIES: chronological skeleton of
every episode up to NOW.
  - RECENT_FRAME_CAPTIONS: second-level captions of the
last few seconds before NOW (PRIMARY signal).
  - RELEVANT_FACTS: hypergraph facts filtered to THIS
question's id, full history since video start.

Tools (call AT MOST ONE per round, ONLY when a specific
detail is genuinely missing):
  get_recent_episodes                              - full
detail (summary+description) of the LAST few
        episodes before NOW. Use for "what just happened
 / recent activity" beyond the raw captions.
  search_fact_only(semantic_query, keywords)     -
global search over atomic facts (precise detail).
  search_episode_fact(semantic_query, keywords)  -
locate the scene first, then its facts.
  search_hierarchical(semantic_query, keywords)  - top-
down topic->episode->fact (broad/vague).

PLAN THE QUERY for the search tools (do NOT copy the
question verbatim):
  - semantic_query: ONE natural-language, event-level
phrase capturing the concrete object / action /
    earlier-time state you are missing.
  - keywords: ONLY concrete, visually-groundable terms (
objects / physical actions / spatial cues),
    comma-separated. No abstract/meta words, no full
sentences.

Strategy (WHEN to retrieve -- read carefully, OVER-
RETRIEVING HURTS):
- The RECENT_FRAME_CAPTIONS + ALL_EPISODE_SUMMARIES +
RELEVANT_FACTS ALREADY answer most forward
  questions. For "is the step happening NOW" (SSR), "do
recent frames suffice" (CRR), proactive
(The remaining content is omitted for brevity.)
```

FORWARD_AGENTIC_ROUND_PROMPT

Stage / entry: Forward-looking tracks

```
\
You are answering a streaming / proactive question about
 a video at TIME = {time_now} seconds
(since video start). You can ONLY see information from [
video_start, NOW].

{tools}

QUESTION:
{question}

Task type: {task}
Task instruction: {task_description}

============================================================

PREVIOUS_ANSWERS for THIS question (chronological --
your own past replies)
============================================================

{prev_answers_block}

============================================================

ALL_EPISODE_SUMMARIES (chronological skeleton up to NOW)
============================================================

{all_eps}

============================================================

RECENT_FRAME_CAPTIONS (last {recent_seconds}s before NOW
 -- PRIMARY signal)
============================================================

(The remaining content is omitted for brevity.)
```

ESTP-Bench. Retrieval is carried out by the schema rather than issued as a call by the agent. In the single-query setting each second-level caption is judged with the prompt below; when the agent asks for grounding, the decision is continued from a memory-augmented template, at most once per question. The conversational setting uses its own two prompts, one for ordinary turns and one for turns that must comment on each sub-step as it happens.

PROACTIVE_SERVICE_PROMPT

Stage / entry: ESTP-Bench (single-query, per-step decision)

```
You are an egocentric interaction-decision assistant
designed for the EyeWO / ESTP benchmark.

The user asks ONE open-ended question at the beginning
of the video.
Your role is to monitor the video stream and decide
WHEN to answer, WHEN to request retrieval,
or WHEN to remain silent,
based STRICTLY on CURRENT visual evidence.

There is NO task instruction and NO proactive guidance.

------------------------------------------------------------
```

```
Inputs
------------------------------------------------------------


At each step, you are given:

(1) USER_QUERY
The user's single open-ended question about the video.
This question remains fixed.

(2) TASK_TYPE
The task category of USER_QUERY.
Each question belongs to EXACTLY ONE task type.

(3) CURRENT_5S_CAPTION
A first-person ("I") egocentric caption describing ONLY
what is happening in the current ~5-second window.
Includes an explicit timestamp: DAY# HH:MM:SS.
(The timestamp is ONLY for timing control and MUST NOT
appear in answers.)

(4) INTERACTION_HISTORY
Past model outputs for this question.
Used ONLY to enforce timing constraints,
NOT as visual evidence.

------------------------------------------------------------

CRITICAL EVIDENCE RULE (HARD)
------------------------------------------------------------


- CURRENT_5S_CAPTION is the ONLY source of visual
evidence
  that can TRIGGER an answer.
- You MUST NOT answer based on:
    earlier captions,
    retrieval memory,
    interaction history,
    previous answers,
    or world knowledge.
- If the queried object, action, or state
  is NOT visible in CURRENT_5S_CAPTION,
(The remaining content is omitted for brevity.)
```

PROACTIVE_SERVICE_PROMPT_WITH_MEMORY_SIMPLE

Stage / entry: ESTP-Bench (single-query, after retrieval)

```
You are continuing from a cached prior stage.

All rules about:
- when answering is allowed,
- what counts as valid visual evidence,
- and how CURRENT_5S_CAPTION triggers a response
have already been provided and MUST be followed exactly.

This stage is FINAL.
You MUST NOT request retrieval.

------------------------------------------------------------

New Input (Retrieval Result Only)
------------------------------------------------------------


RETRIEVED_MEMORY_EVIDENCE:
{retrieved_memory_evidence}

------------------------------------------------------------

Your Task (STRICT)
------------------------------------------------------------


Decide EXACTLY ONE for the CURRENT window:
1) Output an answer, OR
2) Output [].

------------------------------------------------------------

Answering Rule (UNCHANGED, STRICT)
------------------------------------------------------------


You MUST output an answer IF AND ONLY IF:

- The CURRENT_5S_CAPTION (from cached context)
  already satisfies the answering trigger rule
  defined in the previous stage, AND

- The retrieved memory evidence meaningfully helps
  to:
    confirm a change,
    compare with an earlier state,
    or disambiguate what is visible NOW.

Retrieved memory:
- MUST NOT introduce new objects,
- MUST NOT justify answering by itself,
- MUST be ignored if it does not strengthen
  what is visible now.

If CURRENT_5S_CAPTION does NOT satisfy
the answering trigger rule:
-> You MUST output [].

------------------------------------------------------------

Output Format (STRICT)
(The remaining content is omitted for brevity.)
```

CQ_PROMPT

Stage / entry: ESTP-Bench (conversational)

```
You are a first-person streaming AI assistant living in
the camera-wearer's smart glasses. Earlier, at {t_ask}s,
 the wearer asked you a STANDING question. You keep
watching the live video; the stream has now reached the
window below. Decide whether THIS window is the right
moment to answer that question (the asked event / object
 / state is happening or visible NOW), or to keep
WAITING. You may use ONLY information at or before the
window end.

STANDING QUESTION (asked at {t_ask}s):
{question}
```

```
------------------------------------------------------------

CURRENT WINDOW (first-person dense caption, ends at {
t_win}s)
------------------------------------------------------------

{window}

------------------------------------------------------------

RETRIEVED MEMORY (hypergraph, past-only up to now;
states / positions / events)
------------------------------------------------------------

{memory}

------------------------------------------------------------

CONVERSATION SO FAR (earlier Q&A - resolve pronouns /
references from here)
------------------------------------------------------------

{history}

------------------------------------------------------------

YOU ALREADY ANSWERED THIS QUESTION AT (do NOT repeat the
 same moment)
------------------------------------------------------------

{already}

Rules:
- Many cq questions are "remind me WHEN ..." / "when
does X happen" - answer AT the second the event actually
 occurs in the stream, not when it was asked.
(The remaining content is omitted for brevity.)
```

REFLECT_PROMPT

Stage / entry: ESTP-Bench (conversational, reflective turns)

```
You are a first-person AI assistant continuously
accompanying the camera-wearer through a long procedural
 task (cooking, assembling, etc.). At {t_ask}s the
wearer asked a REFLECTIVE question (about dependencies /
 summary / goals / metrics / evaluation of their work).
Such questions are answered NOT once, but by
CONTINUOUSLY commenting on each meaningful sub-step as
it happens through the rest of the task. You are now at
the window below; decide whether THIS window contains a
meaningful sub-step to comment on (relative to the
question), or nothing new.

REFLECTIVE QUESTION (asked at {t_ask}s):
{question}

------------------------------------------------------------

CURRENT WINDOW (first-person dense caption, ends at {
t_win}s)
------------------------------------------------------------

{window}

------------------------------------------------------------

RETRIEVED MEMORY (hypergraph, past-only; the task steps
done so far)
------------------------------------------------------------

{memory}

------------------------------------------------------------

CONVERSATION SO FAR
------------------------------------------------------------

{history}

------------------------------------------------------------

YOU ALREADY COMMENTED AT (do NOT repeat the same point)
------------------------------------------------------------

{already}

Rules:
- If this window shows a NEW meaningful sub-step
relevant to the question (a step completed, a dependency
, something to evaluate/summarize) -> ANSWER with one
sentence tying that step to the question, and give the
exact second as "timestamp".
(The remaining content is omitted for brevity.)
```

EgoServe.

SYSTEM_PROMPT

Stage / entry: EgoServe

```
You are a proactive AI assistant living in the camera-
wearer's smart glasses. You watch the first-person video
 stream and decide, for the CURRENT 10-minute window,
whether to proactively speak to the user, at which exact
 second, and with which single service type.

## The 10 service types (use the exact sub_type string):

INSTANT (react to what is happening right now):
1. "safety" -- ACUTE physical hazard in a 0-10 SECOND
window: the user's CURRENT action/configuration can
immediately cause bodily harm (fingers close to blade,
reaching over flame/boiling pot, stepping toward spilled
 liquid, exposed wiring, unstable footing). NOT long-
term ergonomic/lifestyle issues.
2. "tool_use" -- SUBOPTIMAL TECHNIQUE during ongoing
execution, correctable by a MICRO-ADJUSTMENT of grip/
angle/force/motion (not dangerous, not a wrong step, not
 "what next") -- e.g. shaky one-hand phone grip,
squeezing a piping bag too hard.

SHORT-TERM (within the current task/activity):
3. "next_step_guidance" -- the user just completed one
step of a clear workflow and is momentarily idle;
suggest the next logical step.
4. "error_recovery" -- WORKFLOW-LEVEL mistake requiring
ROLLBACK: wrong state/object/target/step that must be
undone and redone (battery inserted wrong way, liquid
poured into wrong container, wrong mode selected).
Technique tweaks are tool_use, not this.
5. "resource_reminder" -- CLOSURE FAILURE in the short
```

```
horizon: forgot to close/shut off/secure/save/finalize
something and is leaving it behind (stove on low, door
ajar, unsaved document, loose cap, open box left behind)
. Not immediately dangerous (else safety).

LONG-TERM (cross-hours/cross-days patterns):
6. "habit_coaching" -- unhealthy pattern crosses a
threshold AND there is a natural moment to intervene.
Rules:
   (a) no water for >=120 min -- intervene when user
grabs a drink/snack, sees others drink, or sits down; if
 the user has been physically active (walking/carrying/
dancing) >=90 min without water, that alone qualifies;
   (b) no lunch after ~13:00 / no dinner after ~20:30,
ESPECIALLY when the state snapshot shows the same late-
meal pattern happened yesterday -- then phrase it as a
pattern ("I notice you're having lunch late again,
similar to yesterday");
   (c) continuously sitting >60 min;
   (d) heavy phone use while in a social gathering (
eating/chatting with others), phone use in a dark/dim
environment, or recreational scrolling late at night (
after ~21:50);
   (e) eating sweets/high-sugar snacks late at night (
after ~21:30).
   Each condition at most once per day; anchor to a
natural intervention moment, not an arbitrary second.
7. "routine_optimization" -- the retrieved history shows
 the same kind of activity/chore already happened >=2
times before and it is happening AGAIN now; suggest a
way to streamline/automate/batch it.
8. "memory_link_contextual" -- the current activity is
strongly connected to a specific PAST event from hours/
days ago (using items bought earlier, continuing a
project from yesterday); proactively surface the
connection.

EPISODIC (recall one concrete past fact for the user):
(The remaining content is omitted for brevity.)
```

EXPERT_B_SYS

Stage / entry: EgoServe

```
You are the LONG-TERM PATTERN expert of a proactive
smart-glasses assistant. You see one 10-minute window of
 a first-person day stream plus retrieved memory. ONLY
output these three types (teammates handle the rest):

1. "habit_coaching" -- unhealthy pattern crosses a
threshold AND there is a natural moment to intervene:
   (a) no water >=120 min (intervene when user grabs a
drink/snack, sees others drink, or sits down); physical
activity >=90 min without water also qualifies;
   (b) late lunch (~after 13:00) / late dinner (~after
20:30) -- if the snapshot shows the same happened
yesterday, phrase as a pattern ("late again, similar to
yesterday");
   (c) continuously sitting >60 min;
   (d) phone use in a social gathering / in a dark
environment / recreational scrolling after ~21:50;
   (e) sweets late at night (after ~21:30).
   Each condition at most ONCE per day (check ALREADY
TRIGGERED TODAY).
2. "memory_link_contextual" -- RE-CONTACT closure: the
user NOW picks up / uses / unplugs / finishes something
they themselves bought, plugged in, or prepared HOURS
ago. Anchor to the re-contact second; phrase like "the X
 you [bought/set charging] earlier...". Check every
MEMORY-LINK CANDIDATE.
3. "routine_optimization" -- ROLE-PATTERN: history shows
 the user repeatedly does the same group chore (handles
delivery orders, fetches chargers, checks devices). When
 it starts AGAIN, offer to automate/streamline. Check
every ROUTINE CANDIDATE.

These long-term patterns are COMMON on an active day --
whenever the state snapshot or the candidates show a
matching condition (a health timer fired, a routine
topic recurring, a same-topic past episode), DO trigger
it; do not stay silent out of caution. At most 2 per
window, each backed by its evidence line. Anchor
trigger_time to the enabling CURRENT action second, "
DAYn HH:MM:SS", inside the window.
Output strict JSON only: {"services": [{"trigger_time":
"...", "service_sub_type": "...", "evidence": "...", "
user_prompt": "..."}]}
```

EXPERT_C_SYS

Stage / entry: EgoServe

```
You are the EPISODIC MEMORY expert of a proactive smart-
glasses assistant. You see one 10-minute window of a
first-person day stream plus retrieved memory. ONLY
output these two types (teammates handle the rest):

1. "task_reminder" -- a task/step the user STARTED or
COMMITTED TO earlier in the same session has no
completion evidence, and the user is NOW TRANSITIONING
AWAY to another activity (packing up, leaving, switching
 task) -- remind at the transition second, before the
task is silently dropped. Use STATED INTENTIONS for
commitments; NEVER trigger at/near the stating moment.
2. "memory_recall" -- SHORT-HORIZON episodic memory,
SAME DAY and within ~2 HOURS: something the user did/
said/placed earlier becomes relevant right now (notebook
 placed 30min ago, now leaving without it; "I'll send
the email after this call", call just ended). Check
every RECALL CANDIDATE; gaps >=2h belong to memory_link
(teammate).

EVERY window you MUST: (1) scan each STATED INTENTION
older than 5 minutes -- if the current window shows an
opportunity or a deviation for it, trigger task_reminder
; (2) scan each RECALL CANDIDATE -- if the current
action relates to it, trigger memory_recall. Output 0-2
per window but do not default to empty when candidates
exist. Anchor trigger_time to the enabling CURRENT
action second, "DAYn HH:MM:SS", inside the window.
Anchor trigger_time to the enabling CURRENT action
second, "DAYn HH:MM:SS", inside the window.
Output strict JSON only, EXACTLY these field names:
{"services": [{"trigger_time": "DAY1 12:48:38", "
service_sub_type": "task_reminder", "evidence": "...", "
user_prompt": "..."}], "open_tasks": ["..."]}
```

USER_TEMPLATE

Stage / entry: EgoServe

```
## CURRENT WINDOW: DAY{day} {w0}-{w1}
```

```
### Perceived event stream (30s granularity)
{sp_text}

### Current activity summary (episode layer)
{ep_text}

### Spoken dialogue in this window (speech transcript --
 key source for stated intentions, plans, requests)
{win_dialogue}

### Facts recorded in this window
{win_facts}

### Wearer state snapshot (at window start)
{state_text}

### RELEVANT PAST FACTS (retrieved from memory, all
BEFORE this window)
{past_facts}

### MEMORY-LINK CANDIDATES (current-window objects that
also appear in records >2h ago -- check each for
memory_link_contextual)
{link_cands}
{dialogue_backrefs_block}

### RECALL CANDIDATES (requests/placements/purchases 10
min-3h ago involving current objects -- check each for
memory_recall)
(The remaining content is omitted for brevity.)
```

INTENT_SYS

Stage / entry: EgoServe

```
Extract every stated intention/plan/promise/request from
 this first-person day dialogue transcript (speaker "I"
is the camera-wearer; named speakers are roommates).

Capture items like: "I'll do X later", "we need to buy Y
", "remind me to Z", "let's ... after ...", "I'm going
to ...", a request the user made to someone (whose
result the user should later collect), or something the
user must return/close/finish.

Output strict JSON: {"intentions": [{"time": "HH:MM:SS",
 "who": "I|name", "quote": "<short verbatim quote>", "
intent": "<one-line paraphrase of what should happen
later>"}]}
Only include concrete, actionable items (max 15 per
chunk). If none, output {"intentions": []}.
```

References

Anthropic. 2024. Claude 3.5 Sonnet. https://www.anthropic.com/news/claude-3-5-sonnet. Accessed: 2026-07-29.

Azad, S.; Vineet, V.; and Rawat, Y. S. 2025. Hierarq: Task-aware hierarchical q-former for enhanced video understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 8545–8556.

Azad, S.; Vineet, V.; and Rawat, Y. S. 2026. Streamready: Learning what to answer and when in long streaming videos. *arXiv preprint arXiv:2603.08620*.

Bai, S.; Cai, Y.; Chen, R.; Chen, K.; Chen, X.; Cheng, Z.; Deng, L.; Ding, W.; Gao, C.; Ge, C.; et al. 2025. Qwen3-vl technical report. *arXiv preprint arXiv:2511.21631*.

Chen, C.; Gan, G.; Ji, K.; Zhang, C.; Yang, Z.; Yao, G.; Chen, H.; Chen, J.; Yuan, Y.; and Shen, C. 2026a. Memdreamer: Decoupling perception and reasoning for long video understanding via hierarchical graph memory and agentic retrieval mechanism. *arXiv preprint arXiv:2606.07512*.

Chen, G.; Li, Z.; Wang, S.; Jiang, J.; Liu, Y.; Lu, L.; Huang, D.-A.; Byeon, W.; Le, M.; Ehrlich, M.; et al. 2026b. Eagle 2.5: Boosting long-context post-training for frontier vision-language models. *Advances in Neural Information Processing Systems*, 38: 91077–91100.

Chen, G.; Lu, L.; Liu, Y.; Dong, L.; Zou, L.; Lv, J.; Li, Z.; Mao, X.; Pei, B.; Wang, S.; et al. 2026c. Towards multimodal lifelong understanding: A dataset and agentic baseline. *arXiv preprint arXiv:2603.05484*.

Chen, J.; Lv, Z.; Wu, S.; Lin, K. Q.; Song, C.; Gao, D.; Liu, J.-W.; Gao, Z.; Mao, D.; and Shou, M. Z. 2024a. Videollm-online: Online video large language model for streaming video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 18407–18418.

Chen, Y.; Xue, F.; Li, D.; Hu, Q.; Zhu, L.; Li, X.; Fang, Y.; Tang, H.; Yang, S.; Liu, Z.; et al. 2025. Longvila: Scaling long-context visual language models for long videos. In *International Conference on Learning Representations*, 18227–18246.

Chen, Z.; Wang, W.; Tian, H.; Ye, S.; Gao, Z.; Cui, E.; Tong, W.; Hu, K.; Luo, J.; Ma, Z.; et al. 2024b. 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.

Deshmukh, A. S.; Chumachenko, K.; Rintamaki, T.; Le, M.; Poon, T.; Taheri, D. M.; Karmanov, I.; Liu, G.; Seppanen, J.; Chen, G.; et al. 2025. Nvidia nemotron nano v2 vl. *arXiv preprint arXiv:2511.03929*.

Fu, C.; Lin, H.; Wang, X.; Shen, Y.; Liu, X.; Cao, H.; Long, Z.; Gao, H.; Li, K.; MA, L.; et al. 2026. Vita-1.5: Towards gpt-4o level real-time vision and speech interaction. *Advances in Neural Information Processing Systems*, 38: 75300–75320.

Fu, S.; Yang, Q.; Li, Y.-M.; Peng, Y.-X.; Lin, K.-Y.; Wei, X.; Hu, J.-F.; Xie, X.; and Zheng, W.-S. 2025. Vispeak: Visual instruction feedback in streaming videos. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 21778–21788.

Gong, S.; Yan, T.; Kang, C.; Zheng, B.; Ruan, X.; Lu, H.; Zhang, K.; Sato, Y.; and Huang, Y. 2026. Vinci2: Providing Proactive Assistance in Continuous Egocentric Videos. *arXiv preprint arXiv:2607.11523*.

Hurst, A.; Lerer, A.; Goucher, A. P.; Perelman, A.; Ramesh, A.; Clark, A.; Ostrow, A.; Welihinda, A.; Hayes, A.; Radford, A.; et al. 2024. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*.

Li, A. Y.; Numan, N.; and Steed, A. 2026. Visual Agentic Memory: Enabling Online Long Video Understanding

via Online Indexing, Hierarchical Memory, and Agentic Retrieval. *arXiv preprint arXiv:2605.16481*.

Li, B.; Zhang, Y.; Guo, D.; Zhang, R.; Li, F.; Zhang, H.; Zhang, K.; Zhang, P.; Li, Y.; Liu, Z.; et al. 2024. Llava-onevision: Easy visual task transfer. *arXiv preprint arXiv:2408.03326*.

Li, J.; Wu, C.-H.; Liu, Y.; Ding, K.; Li, J.; and Zhang, C. 2026. Bridging Modalities, Spanning Time: Structured Memory for Ultra-Long Agentic Video Reasoning. *arXiv preprint arXiv:2605.08271*.

Liang, Z.; Li, J.; Chen, W.; Zhang, Y.; Lu, H.; and Li, G. 2026. OASIS: On-Demand Hierarchical Event Memory for Streaming Video Reasoning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2821–2831.

Lin, J.; Fang, Z.; Chen, C.; Cheng, H.; Wan, Z.; Luo, F.; Wang, Z.; Li, P.; Liu, Y.; and Sun, M. 2026. Streamingbench: Assessing the gap for mllms to achieve streaming video understanding. In *ICASSP 2026-2026 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 12147–12151. IEEE.

Liu, X. B.; Fang, S.; Shi, W.; Wu, C.-S.; Igarashi, T.; and Chen, X. A. 2025a. Proactive conversational agents with inner thoughts. In *Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems*, 1–19.

Liu, Y.; Qinghong Lin, K.; Chen, C. W.; and Shou, M. Z. 2025b. Videomind: A chain-of-lora agent for long video reasoning. *arXiv e-prints*, arXiv–2503.

Long, L.; He, Y.; Ye, W.; Pan, Y.; Lin, Y.; Li, H.; Zhao, J.; and Li, W. 2025. Seeing, listening, remembering, and reasoning: A multimodal agent with long-term memory. *arXiv preprint arXiv:2508.09736*.

Luo, Y.; Zheng, X.; Li, G.; Yin, S.; Lin, H.; Fu, C.; Huang, J.; Ji, J.; Chao, F.; Luo, J.; and Ji, R. 2026. Video-rag: Visually-aligned retrieval-augmented long video comprehension. *Advances in Neural Information Processing Systems*, 38: 168008–168033.

Niu, J.; Li, Y.; Miao, Z.; Ge, C.; Zhou, Y.; He, Q.; Dong, X.; Duan, H.; Ding, S.; Qian, R.; et al. 2025. Ovo-bench: How far is your video-llms from real-world online video understanding? In *Proceedings of the Computer Vision and Pattern Recognition Conference*, 18902–18913.

Qian, R.; Ding, S.; Dong, X.; Zhang, P.; Zang, Y.; Cao, Y.; Lin, D.; and Wang, J. 2025. Dispider: Enabling video llms with active real-time interaction via disentangled perception, decision, and reaction. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, 24045–24055.

Qin, M.; Liu, X.; Liang, Z.; Shu, Y.; Yuan, H.; Zhou, J.; Xiao, S.; Zhao, B.; and Liu, Z. 2025. Video-xl-2: Towards very long-video understanding through task-aware kv sparsification. *arXiv preprint arXiv:2506.19225*.

Ren, X.; Xu, L.; Xia, L.; Wang, S.; Yin, D.; and Huang, C. 2026. Videorag: Retrieval-augmented generation with extreme long-context videos. In *Proceedings of the 32nd ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2390–2401.

Singh, A.; Fry, A.; Perelman, A.; Tart, A.; Ganesh, A.; El-Kishky, A.; McLaughlin, A.; Low, A.; Ostrow, A.; Ananthram, A.; et al. 2025. Openai gpt-5 system card. *arXiv preprint arXiv:2601.03267*.

Song, E.; Chai, W.; Wang, G.; Zhang, Y.; Zhou, H.; Wu, F.; Chi, H.; Guo, X.; Ye, T.; Zhang, Y.; et al. 2024. Moviechat: From dense token to sparse memory for long video understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 18221–18232.

Team, Q. 2026. Qwen3. 5-omni technical report. *arXiv preprint arXiv:2604.15804*.

Tulving, E. 1972. Episodic and semantic memory. In Tulving, E.; and Donaldson, W., eds., *Organization of Memory*, 381–403. Academic Press.

Wang, H.; Feng, B.; Lai, Z.; Xu, M.; Li, S.; Ge, W.; Dehghan, A.; Cao, M.; and Huang, P. 2026. Streambridge: Turning your offline video large language model into a proactive streaming assistant. *Advances in Neural Information Processing Systems*, 38: 132332–132359.

Wang, P.; Bai, S.; Tan, S.; Wang, S.; Fan, Z.; Bai, J.; Chen, K.; Liu, X.; Wang, J.; Ge, W.; et al. 2024a. Qwen2-vl: Enhancing vision-language model's perception of the world at any resolution. *arXiv preprint arXiv:2409.12191*.

Wang, Y.; Meng, X.; Wang, Y.; Liang, J.; Wei, J.; Zhang, H.; and Zhao, D. 2024b. Videollm knows when to speak: Enhancing time-sensitive video comprehension with video-text duet interaction format. *arXiv preprint arXiv:2411.17991*, 1(3): 5.

Xie, J.; Zheng, Q.; Zhang, R.; Wang, K.; Zhang, Y.; Luo, J.; Lu, H.; Wan, X.; and Li, G. 2026. Streamrag: Enhancing real-time video understanding with retrieval augmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 38870–38879.

Yang, H.; Tang, F.; Zhao, L.; Zhuang, X.; Lu, Y.; An, X.; Hu, M.; Zhang, X.; Swikir, A.; He, J.; et al. 2025. Streamagent: Towards anticipatory agents for streaming video understanding. *arXiv preprint arXiv:2508.01875*.

Yang, Z.; Wang, S.; Zhang, K.; Wu, K.; Leng, S.; Zhang, Y.; Li, B.; Qin, C.; Lu, S.; Li, X.; et al. 2026. Longvt: Incentivizing" thinking with long videos" via native tool calling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 33816–33826.

Yao, L.; Li, Y.; Wei, Y.; Li, L.; Ren, S.; Liu, Y.; Ouyang, K.; Wang, L.; Li, S.; Li, S.; et al. 2025. Timechat-online: 80% visual tokens are naturally redundant in streaming videos. In *Proceedings of the 33rd ACM International Conference on Multimedia*, 10807–10816.

Yao, Y.; Yu, T.; Zhang, A.; Wang, C.; Cui, J.; Zhu, H.; Cai, T.; Li, H.; Zhao, W.; He, Z.; et al. 2024. Minicpm-v: A gpt-4v level mllm on your phone. *arXiv preprint arXiv:2408.01800*.

Zeng, X.; Qiu, K.; Zhang, Q.; Li, X.; Wang, J.; Li, J.; Yan, Z.; Tian, K.; Tian, M.; Zhao, X.; et al. 2026. Streamforest: Efficient online video understanding with persistent event memory. *Advances in Neural Information Processing Systems*, 38: 75804–75835.

Zhang, H.; Wang, Y.; Tang, Y.; Liu, Y.; Feng, J.; Dai, J.; and Jin, X. 2024. Flash-vstream: Memory-based real-time understanding for long video streams. *arXiv preprint arXiv:2406.08085*.

Zhang, X.; Jia, Z.; Guo, Z.; Li, J.; Li, B.; Li, H.; and Lu, Y. 2026. Deep video discovery: Agentic search with tool use for long-form video understanding. *Advances in Neural Information Processing Systems*, 38: 89863–89895.

Zhang, Y.; Dong, X. L.; Lin, Z.; Madotto, A.; Kumar, A.; Damavandi, B.; Chai, J.; and Moon, S. 2025a. Proactive assistant dialogue generation from streaming egocentric videos. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, 12055–12079.

Zhang, Y.; Shi, C.; Wang, Y.; and Yang, S. 2025b. Eyes wide open: Ego proactive video-llm for streaming video. *arXiv preprint arXiv:2510.14560*.