MEMOIR: REGION-LEVEL GRAPH MEMORY FOR LONG-TERM VIDEO OBJECT SEGMENTATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Long-term video object segmentation must preserve target identity across prolonged absence, severe occlusion, and similar distractors, a requirement that ties each prediction to historical appearance evidence far beyond the recent observations. Existing memory designs retain this history as whole frames within a bounded window and expose all of it to every prediction. Informative partial appearances therefore vanish with the discarded frames, and every prediction attends to the entire store regardless of what it requires. To address this limitation, we propose Memory with Evidence Modeled On Interconnected Regions (MEMOIR), a graph memory that stores, maintains, and exposes historical evidence at the level of spatially coherent regions. MEMOIR organizes the region-level memories of a frozen SAM-style base model into a graph whose edges encode spatial, similarity, and temporal relations. Retrieval exposes long-term evidence only where recent observations fail to support the prediction, and management maintains the graph within explicit node and token budgets according to temporal support and retrieval utility rather than age. The base model itself remains unchanged, and no training is required. Across seven video object segmentation benchmarks, MEMOIR improves on the SAM 3 base model with the largest gains on LVOSv2 and MOSEv2, whose long and crowded videos stress memory most, and exceeds SAM 3 on all four reappearance splits, where historical evidence determines the prediction.

1 INTRODUCTION

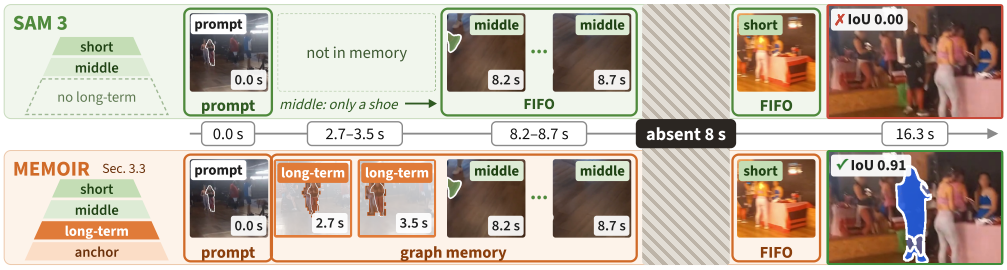

Figure 1: MEMOIR versus SAM 3 at target reappearance. After an 8 s absence, SAM 3, which has no long-term memory, retains only stale frames from the exit at 8.2–8.7 s that show merely a shoe, and the prediction fails (IoU = 0.00). MEMOIR instead retrieves region-level evidence stored at 2.7–3.5 s from its graph memory and recovers the target upon its return (IoU = 0.91).

Consistent object masks enable downstream systems to track individual entities and reason about their motion, interactions, and scene context over time, a capability that autonomous navigation, robotic manipulation, and video analytics all require. Given that such reasoning holds only while successive masks refer to the same object, long-term video object segmentation must preserve target identity across extended sequences, even when similar distractors render recent observations ambiguous or the target remains absent for prolonged intervals. Reliable propagation therefore depends on historical appearance evidence that remains informative when recent observations no longer provide sufficient support. However, the continuing accumulation of observations creates a dilemma: unrestricted retention introduces redundancy, whereas bounded memory capacity forces selective preservation. Historical frames do not constitute uniform units of evidence, since useful information may reside in only certain regions, other regions may remain unsupported or redundant, and relations among partial object appearances evolve across time. Retrieval poses a further obstacle. Repeated or locally dense features render direct matching ambiguous, and a purely local match may miss relevant evidence connected through relations across space and time. The central challenge therefore couples bounded memory growth with the preservation of temporal relations among informative partial object appearances and the retrieval of query-relevant evidence despite local matching ambiguity.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

Existing memory designs meet this challenge only in part. SAM-style segmentation models (Ravi et al., 2024; Carion et al., 2025) maintain a streaming memory that conditions each prediction on the annotated frame and a fixed window of recent object-present observations, a design that meets bounded capacity by discarding older history rather than preserving informative evidence. Recent SAM-based methods reinforce this framework against specific failure modes, from error accumulation (Ding et al., 2024) and distractor confusion (Videnovic et al., 2024) to complex scenes (Zhang et al., 2025), scene transitions (Hu et al., 2025), and unreliable memory admission (Alansari et al., 2026). Each design improves robustness under its targeted failure, while the retained history remains whole-frame observations or predefined memory roles. Both forms treat an entire observation as the unit of evidence. Frame-level units cannot distinguish the informative partial appearances within an observation from redundant or unsupported regions, and the relations among partial appearances that evolve across time remain unmodeled. Retrieval likewise remains coupled to retention, since each prediction attends to the entire store rather than evidence selected for the query. The joint organization, maintenance, and query-conditioned exposure of diverse historical appearances over long-term propagation therefore remain underexplored across current memory-based designs.

The gaps above narrow once the unit of evidence is reconsidered. A stored frame is not an indivisible piece of history: the appearance evidence that later predictions rely on resides in particular regions, and a region that stays coherent in space and semantics can be preserved on its own, at a granularity that neither whole-frame storage nor scattered salient tokens provide. Historical appearances are also far from independent of one another. A partial appearance gains meaning from where it sits in the scene, from the appearances it resembles, and from how it evolves across time, and the relations among appearances carry information that frame-level storage discards. Whether an appearance deserves to stay in memory over a long horizon is a separate question from whether it should inform the prediction at hand, and once the two are separated, a bounded memory no longer forces a choice between hoarding and premature discarding. These observations turn the organization, maintenance, and exposure of historical appearances from an open difficulty into a set of design decisions over regions and their relations, and they are the reason the challenge above admits a better answer.

To act on these decisions, we propose Memory with Evidence Modeled On Interconnected Regions (MEMOIR), a memory design in which preserving an appearance for the long term no longer requires presenting it to every prediction: history stays diverse within a fixed budget and is consulted when recent observations lose track of the target. MEMOIR realizes this through three components. (1) Structured Graph Memory (SGM) stores historical evidence as a region-level graph whose relations preserve how appearances relate across space and time. (2) Query-Conditioned Graph Retrieval (QCGR) admits long-term evidence into the native memory attention only when the current query requires it. (3) Budget-Aware Memory Management (BAM) retires evidence under explicit node and token budgets by temporal support and accumulated retrieval relevance rather than by age. Across seven video object segmentation settings, MEMOIR improves on the SAM 3 base model in five, matches it in one, and trails it by 0.4 points on DAVIS 2017, with gains of 1.6 points on LVOSv2 and 2.1 on MOSEv2 (Table 1), and exceeds SAM 3 on all four reappearance splits by 0.8 to 4.0 points (Table 2), where prolonged absence renders historical evidence decisive. Our contributions are:

- We locate the long-term propagation bottleneck in the unit rather than size of memory: whole frames bind valuable and irrelevant evidence, and scattered tokens lose coherence. SGM preserves, replaces, and consults each partial appearance independently of its frame.
- We separate two questions that existing memories treat as one: what to retain and what to expose. QCGR assigns retention to long-term value and admission to the query, and a growing history no longer crowds the memory-attention interface at every prediction.
- We ground retention in utility rather than in recency, since newness alone is unable to mark the appearances that may matter after a prolonged absence. BAM therefore prevents bounded capacity from suppressing the appearance diversity that long videos accumulate.

2 PRELIMINARY

Semi-supervised video object segmentation delineates a designated target throughout a video, given its annotated mask on a conditioning frame. SAM-style models (Ravi et al., 2024; Carion et al., 2025) approach this task with a streaming-memory framework that processes frames in temporal order and conditions each prediction on memories from earlier frames. Each prediction follows a native recurrence of image encoding, memory attention, mask decoding, and memory encoding. At frame t, the image encoder $\mathcal{E}_{\mathrm{img}}$ maps frame I_t to its visual representation, the mask decoder $\mathcal{D}_{\mathrm{mask}}$ produces the prediction $\hat{Y}_t$ once the native memory attention $\mathcal{A}_{\mathrm{mem}}$ has conditioned it on the stored history,

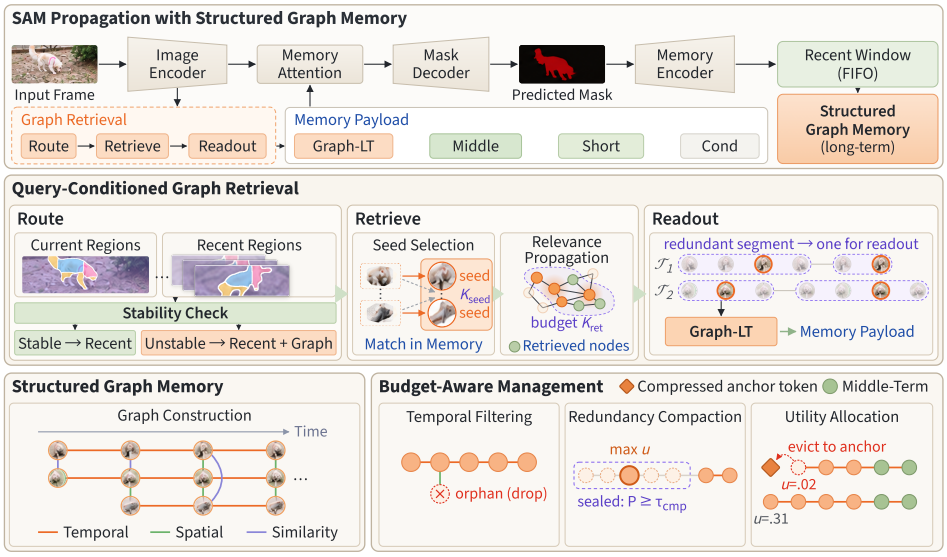

Figure 2: **MEMOIR overview. Top:** the frozen SAM-style base model follows its native prediction path, while QCGR replaces the memory payload memory attention reads, and each predicted frame enters the recency window on its way into graph memory. **Middle:** QCGR selects long-term evidence in one pass of *Route*, *Retrieve*, and *Readout*. **Bottom left:** SGM organizes region-level memories into a graph with temporal, spatial, and similarity edges. **Bottom right:** BAM maintains the graph within budgets through temporal filtering, redundancy compaction, and utility-guided allocation.

and the memory encoder $\mathcal{E}_{\text{mem}}$ encodes the predicted frame into a recency-bounded bank as follows:

$$F_t = \mathcal{E}_{\text{img}}(I_t), \quad \hat{Y}_t = \mathcal{D}_{\text{mask}}(\mathcal{A}_{\text{mem}}(F_t; \mathcal{B}_t)), \quad M_t = \mathcal{E}_{\text{mem}}(F_t, \hat{Y}_t), \quad \mathcal{B}_t = \{M_0, M_{r_1}, \ldots, M_{r_K}\}, \quad (1)$$

where $\mathcal{B}_t$ denotes the historical memory available at frame t, $M_t = \{x_{t,p}\}_{p=1}^{H_m W_m}$ collects the memory tokens on the $H_m \times W_m$ grid, and $r_1, \ldots, r_K < t$ index the K most recent frames. Stored history reaches $\hat{Y}_t$ only through the $\mathcal{B}_t$ that $\mathcal{A}_{\text{mem}}$ reads. The conditioning memory M_0 is encoded from the annotated pair (I_0, Y_0) rather than from a prediction, and M_t enters the bank only for subsequent frames, which keeps the recurrence causal. Retention therefore follows recency alone: apart from M_0, only the K newest memories reach $\mathcal{A}_{\text{mem}}$, and once the target stays absent long enough, the appearances that could support its re-identification have already left this window.

3 Methodology

3.1 Problem Formulation

This paper concerns propagating an annotated mask through videos of unconstrained length, in which the target does not stay visible throughout: it may leave the view, remain absent for prolonged intervals, and later return. The task provides a video sequence $\mathcal{V} = \{I_t\}_{t=0}^{T-1}$ and the ground-truth mask Y_0 of the target on the conditioning frame I_0, and requires a predicted mask $\hat{Y}_t$ on every remaining frame $t = 1, \ldots, T-1$, whether or not the target is visible. This formulation places the problem on the memory side of Eq. (1) rather than on its predictor. Over such a video, the evidence a prediction demands may lie arbitrarily far behind the recency window of Eq. (1), while the memory that holds it must stay within a bounded footprint. The problem therefore divides into what to keep within that footprint, how to organize what is kept, and what to expose to $\mathcal{A}_{\text{mem}}$ at each prediction.

3.2 Theoretical Properties

The two properties below separate our memory from the native design: which history stays reachable, and what reading it costs. Both statements concern the frames whose memory tokens reach the memory attention. Let $\mathcal{S}_t$ denote this set at frame t, and let $\mathcal{O}_t = \{t-1\} \cup \{\, s \in [1, t-1) : q_s > \tau_{\text{mf}} \,\}$ denote the set retained by SAM3's memory selection, where q_s denotes the object-score-weighted predicted IoU of frame s and $\tau_{\text{mf}} = 0.01$ denotes the selection threshold of SAM3. $\text{Top}_k(\cdot)$ returns the k most recent frames of a set, and a graph region is *accessible* at frame t if its tokens are read at t or the region belongs to the retrieval candidate set at t. Please note that Section E provides the proofs.

Proposition 1 (Recency bound vs. retention) *(a) For native SAM3, $\mathcal{S}_t = \{0\} \cup \text{Top}_K(\mathcal{O}_t)$ with $K = 6$. In particular, $s \notin \mathcal{S}_t$ for all $s \in (0, t)$ with $|\mathcal{O}_t \cap (s, t)| \geq K$. (b) For our method, $|\mathcal{V}_t| \leq N_{\max}$ for all t. Let v denote a temporally validated region of frame s, and let t_v denote its*

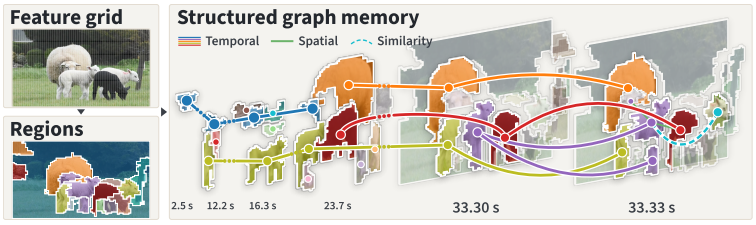

Figure 4: **SGM**. Regions become nodes, linked by spatial and similarity edges within an observation and temporal edges across observations. Older observations keep fewer regions under memory budget.

removal frame, with $t_v = \infty$ if v is never removed. Then v is accessible at every $t \in (s, t_v)$, and only utility-guided eviction or redundancy compaction removes v, when a management trigger fires.

Proposition 2 (Length-independent cost) *Let C_t denote the cost of graph retrieval at frame t, and let $N_t = |\mathcal{V}_t|$. Then $C_t = O(N_t N_{\text{pool}} d\,(H_m W_m + d) + I\,N_t^2)$ and $\sup_t C_t \le O(N_{\max} N_{\text{pool}} d\,(H_m W_m + d) + I\,N_{\max}^2)$, where neither bound depends on the video length.*

In these bounds, $N_{\max} = T_N + (K_{\text{prot}} + 1)R_{\max}$ denotes the largest node count management permits, where T_N is the graph-node trigger, I is the PPR iteration count, and d is the memory-attention width. Proposition 1 (a) formalizes the recency window of Section 2: once K retained frames follow a frame, the native memory no longer reads its appearance, regardless of its later relevance. Proposition 1 (b) states that our memory holds such appearances accessible within a fixed budget until management removes them, and Proposition 2 that accessing them costs no more as the video grows. Without the budget, N_t grows linearly in practice, and Fig. 3 traces the resulting quadratic PPR cost.

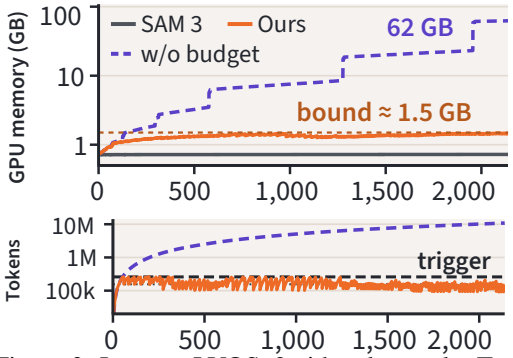

Figure 3: Longest LVOSv2 video, log scale. Top: GPU memory beyond weights and frames. Bottom: graph tokens stored and attended per frame.

3.3 Overview

Fig. 2 summarizes the framework. MEMOIR answers the three questions of Section 3.1 by settling organization first, since both retention and exposure act on the unit that organization defines. **Structured Graph Memory (SGM)** defines this unit as a spatially coherent region of the base model's own memory tokens, and each piece of appearance evidence persists and is consulted independently of its frame. SGM organizes the regions into a graph whose typed edges relate them within and across observations. Over this graph, **Query-Conditioned Graph Retrieval (QCGR)** decides what to expose for the current query, whereas **Budget-Aware Graph Memory Management (BAM)** decides what to keep under node and token budgets by temporal support and retrieval utility. Evidence beyond the recency window thus remains within the bounded footprint without being exposed at every frame. The most recent frames stay complete in the recency window, recent object-present frames remain complete as middle-term, older evidence survives only as compressed regions in the graph, and the annotated frame M_0 conditions the sequence throughout: the older the evidence, the wider its span and the more compressed its form. At each frame t, QCGR selects what to expose to the memory attention through a single pass of *Route*, *Retrieve*, and *Readout*. *Route* confines retrieval to the query regions that lack recent support, *Retrieve* propagates relevance over the graph from directly matched nodes, and *Readout* consolidates and assembles the selected evidence with the native memories into $\mathcal{B}_t$. After the prediction, the frame's encoded memory M_t enters the graph and BAM enforces the node and token budgets. MEMOIR therefore replaces only the construction of $\mathcal{B}_t$ in Eq. (1), while its prediction interface and the memory attention $\mathcal{A}_{\text{mem}}$ remain those of the frozen base model.

3.4 Structured Graph Memory

Structured Graph Memory derives the structure of its graph memory from the choice of memory unit, which must be coarse enough to hold a coherent appearance and fine enough to separate that appearance from the rest of its frame. A frame-level unit satisfies only the first requirement, since it retains or discards the evidence of a whole frame together even when a single part of that frame remains informative. A grid-level unit satisfies only the second, as a content-agnostic grid splits an appearance across several units that no longer describe it as a whole. A region-level unit satisfies both requirements, and SGM defines it directly on the native memory M_t of the base model.

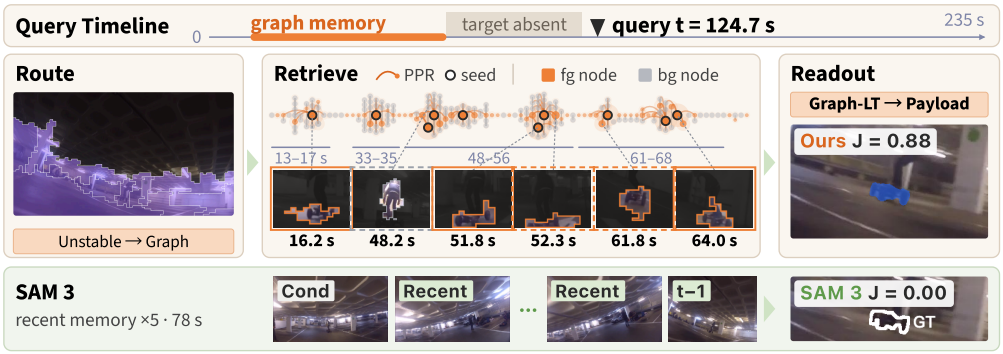

Figure 5: **QCGR at target reappearance** (LVOSv2, absent 72.8–114.2 s, query at 124.7 s). **Route** selects unstable query regions, **Retrieve** spreads PPR relevance from ringed seeds (solid) to target and background regions 60–108 s old (dashed crops), and **Readout** feeds them to the native memory attention. MEMOIR recovers the target ($\mathcal{J}$=0.88), unlike SAM 3's recency window ($\mathcal{J}$=0.00).

The partition $\mathcal{R}_t = \{\Omega_{t,1}, \ldots, \Omega_{t,R_t}\}$ is computed on the frame-t feature map F_t (Section A.1.1), and M_t reuses it on its aligned $H_m \times W_m$ grid. The memory of region $\Omega_{t,r}$ is then defined as $X_{t,r} = \{\, x_{t,p} \in M_t : p \in \Omega_{t,r} \,\}$, where $x_{t,p}$ denotes the memory token of M_t at grid position p. Given that $\mathcal{R}_t$ partitions the whole map, the region memories $X_{t,1}, \ldots, X_{t,R_t}$ divide M_t into disjoint spatial subsets, and every token, background included, belongs to exactly one of them. Each region memory forms one node $v \in \mathcal{V}$ of the graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ and retains its native tokens together with their spatial positional encodings rather than a synthesized feature. As these tokens originate from the base model itself, the frozen memory attention reads a retrieved node in the same form as its native memory. SGM organizes every memory M_t encoded from a prediction $\hat{Y}_t$ in this way along the update path of Fig. 2, whereas the conditioning memory M_0 remains unchanged during propagation.

Relations among these regions complete the graph memory. Within an observation, two edge types relate regions at different ranges. Spatial edges $\mathcal{E}_{\text{spa}}$ preserve local object structure by connecting nodes whose regions Ω_i and Ω_j are neighbors on the four-connected grid, denoted $\Omega_i \sim_4 \Omega_j$. Similarity edges $\mathcal{E}_{\text{sim}}$ capture associations beyond local adjacency by connecting appearance-related regions anywhere in the same observation, namely pairs whose affinity exceeds the similarity threshold τ_{sim}. Across observations, temporal edges $\mathcal{E}_{\text{temp}}$ connect corresponding appearances and form the temporal trajectories along which an appearance persists. The region affinity that weights these edges and the linking rules of each edge type are detailed in Section A.1.2. For two nodes v_i and v_j on one trajectory, the temporal path affinity $C(i, j)$ denotes the product of the temporal edge weights between them and measures how continuously an appearance persists from v_i to v_j. With regions as nodes and affinity-weighted typed relations as edges, SGM settles the organization question of Section 3.1, and the node budget that BAM enforces bounds the complexity of graph operations.

3.5 Query-Conditioned Graph Retrieval

Query-Conditioned Graph Retrieval answers the exposure question of Section 3.1 with the current query as its condition and the graph of Section 3.4 as its search space. Only the long-term share of the payload depends on the current query, and exposing it unconditionally would place older appearances beside recent ones even for the regions that the recency window already covers. *Route* therefore confines long-term retrieval to the query regions that lack recent support and quantifies this support through the *temporal stability* of each $\Omega_{t,r} \in \mathcal{R}_t$, namely how consistently the region recurs among the regions of recent frames. This stability and the set of routed regions are defined as follows:

$$S_{t,r} = \underset{\delta \in \mathcal{H}_t}{\text{median}} \; \max_{\Omega' \in \mathcal{R}_{t-\delta}} \text{IoU}\left(\Omega_{t,r}, \Omega'\right), \qquad \mathcal{R}_t^{\text{LT}} = \left\{\, \Omega_{t,r} \in \mathcal{R}_t : S_{t,r} < \tau_{\text{route}} \,\right\}, \tag{2}$$

where δ ranges over the recent window $\mathcal{H}_t$ of frame offsets, Ω' ranges over the partition $\mathcal{R}_{t-\delta}$ of frame $t - \delta$, IoU denotes the intersection over union of two regions as sets of grid locations, and τ_{route} denotes the routing threshold. Since SGM aligns every partition with the same $H_m \times W_m$ grid, regions of different frames overlap on common grid locations, and the stability compares their spatial extents directly. As $S_{t,r}$ takes the median of these per-frame best overlaps, a region remains stable while at least half of the recent frames supply a counterpart that overlaps it by at least τ_{route}. A region of a target that returns after a prolonged absence, by contrast, finds such counterparts in few recent frames. Its stability therefore falls below τ_{route}, and the region enters $\mathcal{R}_t^{\text{LT}}$.

Retrieve gathers the long-term evidence for the routed regions, and the evidence a routed region requires extends beyond what direct matching can locate. Matching finds the nodes that resemble the region's current appearance, while adjacent parts of the same object, related appearances elsewhere in an observation, and corresponding appearances in other observations connect to those nodes through the spatial, similarity, and temporal edges of Section 3.4. Retrieve therefore seeds the search with direct matches and spreads their relevance over $\mathcal{G}$ through *relevance propagation*. For a node v with region memory X_v, the direct relevance and the propagated relevance are defined as follows:

$$s_v = \max_{\Omega \in \mathcal{R}_t^{\mathrm{LT}}} \frac{1}{|X_v|} \sum_{m \in X_v} \max_{q \in \Omega} \mathrm{sim}\left(\phi_Q(f_{t,q}), \phi_K(m)\right), \qquad \mathbf{p} = \alpha\, \mathbf{r} + (1-\alpha)\, P^\top \mathbf{p}, \tag{3}$$

where Ω ranges over the routed regions, q ranges over the grid locations of Ω, m ranges over the memory tokens of v, $f_{t,q}$ denotes the feature of F_t at q, ϕ_Q and ϕ_K denote the frozen query and key projections of the first memory-attention layer, sim denotes the cosine similarity of two projected tokens, $\mathbf{r}$ denotes the restart distribution, P denotes the transition matrix of the normalized edge weights $w(v_i, v_j)$ of $\mathcal{G}$, and α denotes the restart probability. The direct relevance s_v favors a node whose tokens find close counterparts within a single routed region and assigns the node the local compatibility of its best-matching region. As the query side passes through ϕ_Q and the memory side through ϕ_K, the direct relevance measures the match in the attention space of the frozen base model, where the memory attention later reads the retrieved tokens. The K_{seed} nodes with the highest direct relevance form the seed set $\mathcal{S}_t$, and $\mathbf{r}$ is uniform over $\mathcal{S}_t$ and zero elsewhere. Relevance propagation instantiates Personalized PageRank (Page et al., 1999; Jeh & Widom, 2003) on $\mathcal{G}$. Ranking by s_v alone would reduce Retrieve to a similarity top-k search over resembling regions, whereas propagation also recovers the evidence that the graph relations attach to them. Retrieve finally ranks the nodes by $\mathbf{p}$ and collects the highest-ranked ones into $\mathcal{V}_t^{\mathrm{ret}}$ until the retrieval budget of K_{ret} nodes or the token budget B_T is exhausted. These budgets bound the retrieved share of $\mathcal{B}_t$ regardless of the graph's size.

Readout assembles the memory payload and removes redundancy from the retrieved evidence before that payload reaches the frozen memory attention. As relevance travels along temporal edges into consecutive observations of one appearance, $\mathcal{V}_t^{\mathrm{ret}}$ can contain several nodes of one trajectory that expose nearly identical evidence. Trajectory-level *consolidation* therefore retains only the representatives that cover the other retrieved nodes of their trajectory under the path affinity $C(i,j) \geq \tau_{\mathrm{con}}$ of Section 3.4. The payload $\mathcal{B}_t$ assembled in place of Eq. (1) is defined as follows:

$$\mathcal{B}_t = M_0 \oplus \mathcal{B}_t^{\mathrm{short}} \oplus \mathcal{B}_t^{\mathrm{mid}} \oplus \mathcal{B}_t^{\mathrm{LT}} \oplus \mathcal{B}_t^{\mathrm{compress}}, \tag{4}$$

where $\oplus$ denotes concatenation into the native memory payload, M_0 denotes the conditioning memory, $\mathcal{B}_t^{\mathrm{short}}$ denotes the recency window of Eq. (1), $\mathcal{B}_t^{\mathrm{mid}}$ denotes the middle-term frames, $\mathcal{B}_t^{\mathrm{LT}}$ denotes the long-term representatives that consolidation retains, and $\mathcal{B}_t^{\mathrm{compress}}$ denotes the compressed history that BAM maintains for the trajectories most relevant under $\mathbf{p}$. Section A.2 details the consolidation and the payload assembly. QCGR thus settles the exposure question of Section 3.1.

3.6 Budget-Aware Graph Memory Management

BAM answers the retention question of Section 3.1: it governs which historical evidence remains within finite storage according to temporal support and retrieval utility. BAM bounds the stored graph by a node budget B_N^{mem} on retained regions and a token budget B_T^{mem} on their native tokens, whereas the retrieval budgets K_{ret} and B_T of Section 3.5 constrain only a single frame's payload. Temporal validation applies once a region leaves the protected interval of the latest K_{prot} observations, regardless of budget occupancy. A region without any temporal edge is removed at that point, which excludes unverified observations from long-term memory. A budget violation then triggers redundancy compaction, followed by utility-guided budget allocation if the node budget remains exceeded.

Redundancy compaction groups consecutive units of a temporal trajectory into a redundant segment under the path-affinity threshold τ_{cmp} and retains the segment's highest-utility unit as its representative. Utility-guided budget allocation removes low-utility storage units from the oldest end of each trajectory. When the token budget remains exceeded, token downsampling reduces lower-utility background regions without changing their graph identity. Anchor compression summarizes each utility-evicted trajectory prefix into a bounded anchor before that prefix is discarded. Readout exposes these anchors as the compressed component $\mathcal{B}_t^{\mathrm{compress}}$ of Eq. (4), within the token budget B_T^{compress}. Older evidence thus maps to fewer and more compressed entries that stand for wider temporal spans, while the tokens of every retained graph region remain native to a single observation. Complete definitions, the utility statistic, and the full maintenance procedure appear in Section A.3.

Table 1: Performance comparison with prior work on standard video object segmentation benchmarks. **Bold** indicates the best performance, and underline indicates the second-best performance. † Reported by SENTRY Alansari et al. (2026). ‡ Reported in MOSEv2 Ding et al. (2025).

Method	$\mathcal{J}\&\mathcal{F}$					$\mathcal{G}$	$\mathcal{J}\&\dot{\mathcal{F}}$
	SA-V val	SA-V test	LVOS v2 val	MOSE v1 val	DAVIS 2017 val	YTVOS 2019 val	MOSE v2 val
STCN	61.0	62.5	60.6	52.5	85.4	82.7	29.7
RDE	51.8	53.9	62.2	46.8	84.2	81.9	32.0
XMem	60.1	62.3	64.5	59.6	86.0	85.6	36.3
Cutie-base	60.7	62.7	–	69.9	87.9	87.0	42.8
SAM 2.1	78.6	79.6	84.1	74.5	90.6	88.7	49.5
SAM2.1Long	81.1	81.2	85.9	75.2	91.4	88.7	51.5
SAMURAI	79.8	80.0	84.2	72.6	89.9	88.3	51.1
DAM4SAM ‡	80.5	81.0	85.5	75.6	–	–	51.2
SAMITE †	80.8	81.3	85.8	76.8	–	–	–
HiM2SAM †	81.0	81.5	86.1	77.0	–	–	–
SeC	82.7	81.7	86.5	75.3	91.3	88.6	53.8
SENTRY-S2	79.4	80.1	84.9	75.8	–	–	–
SAM 3	83.5	84.4	88.5	**78.4**	**92.2**	89.7	60.3
Ours	**84.3**	**85.0**	**90.1**	**78.4**	91.8	**90.0**	**62.4**

4 Experimental Results

4.1 Experimental Setups

Benchmarks and Metrics. We evaluate MEMOIR on seven semi-supervised video object segmentation benchmarks that together cover standard short sequences and the long, crowded scenarios that our design targets. DAVIS 2017 val (Pont-Tuset et al., 2017) and YouTube-VOS 2019 val (Xu et al., 2018) represent the established short-video setting. SA-V val and test (Ravi et al., 2024) contribute large-scale evaluation with frequent partial visibility, LVOSv2 val (Hong et al., 2024) provides minute-long videos for long-term evaluation, and MOSEv1 val (Ding et al., 2023) and MOSEv2 val (Ding et al., 2025) emphasize crowded scenes, severe occlusion, and disappearance–reappearance cases. Following standard protocols, we report $\mathcal{J}\&\mathcal{F}$ on SA-V, LVOSv2, MOSEv1, and DAVIS 2017, $\mathcal{G}$ on YouTube-VOS 2019, and $\mathcal{J}\&\dot{\mathcal{F}}$ on MOSEv2. To measure behavior under target absence directly, we further report the disappearance and reappearance scores $\mathcal{J}\&\dot{\mathcal{F}}_{\mathrm{d}}$ and $\mathcal{J}\&\dot{\mathcal{F}}_{\mathrm{r}}$ introduced with MOSEv2 on SA-V val and test, LVOSv2 val, and MOSEv2 val. The scenario and ablation tables report $\mathcal{J}\&\dot{\mathcal{F}}$ on every split, and their absolute values therefore differ from the $\mathcal{J}\&\mathcal{F}$ entries in Table 1. Evaluation on five visual object tracking benchmarks appears in the appendix.

Baselines. We compare MEMOIR with two families of methods. The first family manages memory at the frame level, and the classical memory-based line includes STCN (Cheng et al., 2021), RDE (Li et al., 2022), XMem (Cheng & Schwing, 2022), and Cutie (Cheng et al., 2024). The second family builds on SAM-style streaming memory, and the recent extensions of this design include SAM 2.1 (Ravi et al., 2024), SAM2.1Long (Ding et al., 2024), SAMURAI (Yang et al., 2024), DAM4SAM (Videnovic et al., 2024), SAMITE (Xu et al., 2025), HiM2SAM (Chen et al., 2025), SeC (Zhang et al., 2025), and SENTRY (Alansari et al., 2026). SAM 3 (Carion et al., 2025) serves as both the strongest baseline and the base model of MEMOIR, and the comparison between the two therefore isolates the effect of the proposed graph memory on an identical frozen backbone.

4.2 Results on Video Object Segmentation Benchmarks

Table 1 reports the comparison with both baseline families on the seven benchmarks. The improvements concentrate exactly where long-term memory matters: MEMOIR attains the strongest results on the long and crowded benchmarks such as LVOSv2 and MOSEv2, while remaining on par with SAM 3 on the short standard sets where recent observations already suffice. This distribution follows directly from the routing design, since long-term retrieval intervenes only for the regions whose recent support fails, and the gains therefore stem from recovered historical evidence rather than from a general recalibration of the base model. SeC, the strongest competitor in the SAM 2.1 line, illustrates a complementary design point. A concept-level summary establishes what the target is, whereas the crowded and minute-long benchmarks

Table 2: Comparison with SAM 3. All / D / R denote $\mathcal{J}\&\dot{\mathcal{F}}$, $\mathcal{J}\&\dot{\mathcal{F}}_{\mathrm{d}}$ (disappearance), $\mathcal{J}\&\dot{\mathcal{F}}_{\mathrm{r}}$ (reappearance). †: reproduced with the official code under the same protocol.

Method	SA-V val			SA-V test			LVOSv2			MOSEv2		
	All	D	R	All	D	R	All	D	R	All	D	R
SAM 3†	81.8	82.6	61.5	83.0	83.2	62.8	87.9	**67.2**	75.4	60.0	64.4	42.7
Ours	**82.9**	**84.2**	**63.9**	**83.5**	**84.0**	**63.6**	**89.2**	67.1	**79.1**	**62.4**	**65.2**	**46.7**

Table 3: Ablation on memory unit representation.

Repr.	SA-V val			SA-V test			LVOSv2			MOSEv2		
	All	D	R	All	D	R	All	D	R	All	D	R
Frame	81.8	**84.7**	60.5	83.0	83.9	62.5	88.4	67.5	76.7	60.8	64.8	44.2
Grid (6×6)	82.4	84.3	62.1	**83.5**	**84.0**	**63.7**	88.8	**67.9**	77.9	61.6	63.9	45.7
Region	**82.9**	84.2	**63.9**	**83.5**	**84.0**	63.6	**89.2**	67.1	**79.1**	**62.4**	**65.2**	**46.7**

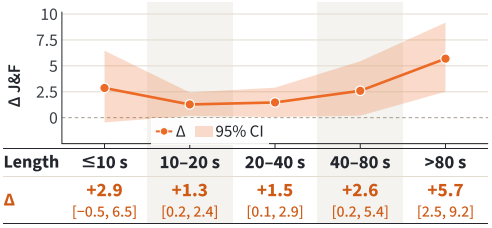

Figure 6: Gain as video length grows.

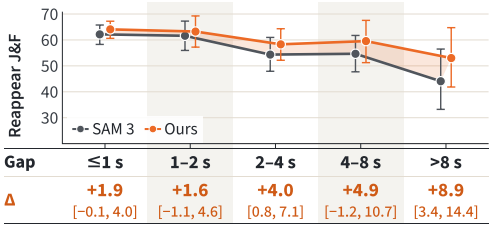

Figure 7: Gain as disappearance duration grows.

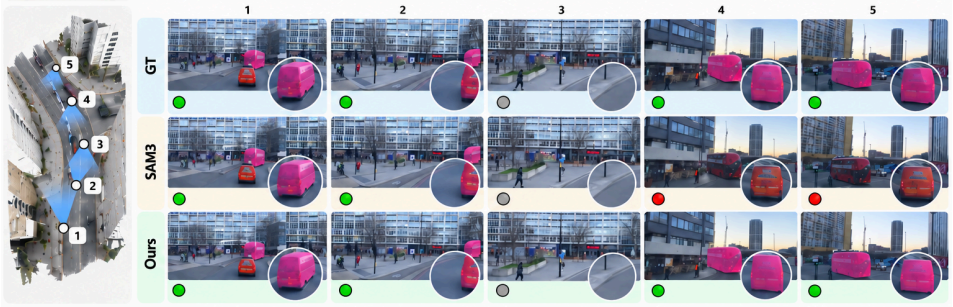

Figure 8: **Qualitative Results.** Long disappearance duration scenario.

reward a memory that returns the specific partial appearance each query needs, the granularity that MEMOIR preserves. Fig. 6 confirms this gradation within single videos: the margin over SAM 3 rises with video length, and beyond 80 s the gain stays positive over the entire confidence interval. Table 2 evaluates the disappearance–reappearance regime in which historical evidence becomes the only usable cue. The reappearance scores show the clearest advantage of MEMOIR across all four splits, while the disappearance scores change far less, between -0.1 and $+1.6$ points. This asymmetry traces back to the mechanism. Once a target returns after a prolonged absence, the recent window contains no usable evidence of it, and retrieval from the region-level history supplies the appearance evidence that the prediction lacks. In the final configuration, the handling of absence stays close to the base model's. The gains therefore reflect an improved ability to recover the target after absence rather than a shifted balance between reporting and suppressing the target. Fig. 7 extends this reading along the axis of absence duration: the margin over SAM 3 widens as the gap grows and remains positive throughout the confidence interval beyond 8 s, where it reaches $+8.9$ points.

4.3 Qualitative Results

Fig. 8 contrasts MEMOIR with SAM 3 in the scenario that stresses long-term memory most, prolonged target absence. The target leaves the view for an extended interval, and SAM 3 fails to reacquire it upon return, while MEMOIR re-identifies it from the preserved region-level evidence and continues the propagation. Whenever recent observations no longer identify the target, the retrieved history supports the prediction, which is the qualitative counterpart of the reappearance gains in Table 2. An additional example combining a crowded scene with severe occlusion is provided in Section B.1.

4.4 Ablation Study

4.4.1 Memory Representation Study

Table 3 varies the granularity of the stored evidence while the remaining components stay fixed. The unit of evidence itself governs long-term memory quality: region-level storage attains the best aggregate score on every benchmark, and the score rises with each refinement step on SA-V val, LVOSv2, and MOSEv2 while saturating between grid and region on SA-V test. Partitioning alone does not supply this quality, since regions must follow appearance boundaries for the stored tokens to remain coherent evidence, and the gap between the content-agnostic grid and region-level storage on the same three benchmarks reflects exactly this requirement. The same coherent regions supply the partial appearances from which a returning target is recognized, and the reappearance scores accordingly benefit most from the finer granularity. The region is therefore the granularity at which stored evidence stays both coherent and recognizable.

Table 4: Component ablation. Top: retrieval stages; bottom: memory management. All / D / R denote $\mathcal{J}\&\dot{\mathcal{F}}$, $\mathcal{J}\&\dot{\mathcal{F}}_{\mathrm{d}}$, $\mathcal{J}\&\dot{\mathcal{F}}_{\mathrm{r}}$. ‡: FIFO replaces utility-based allocation.

Variant	SA-V val			SA-V test			LVOSv2			MOSEv2		
	All	D	R	All	D	R	All	D	R	All	D	R
w/o Route	82.7	**84.6**	62.5	83.4	**84.3**	63.7	88.6	66.6	77.5	61.9	63.7	45.9
w/o Consol.	81.9	83.0	62.9	82.8	81.9	63.0	87.5	64.7	78.0	60.6	61.5	45.1
w/o Graph†	80.5	80.5	61.1	81.6	79.1	62.1	86.0	60.1	77.0	59.7	61.3	43.5
w/o Temp. Val.	82.3	84.1	62.9	83.1	83.0	63.8	88.5	66.4	77.8	61.9	64.5	45.8
w/o Redund.	82.6	84.3	63.1	83.4	84.2	**63.9**	88.8	**67.5**	78.5	61.9	64.3	45.9
FIFO‡	82.8	84.4	62.6	82.5	82.6	62.0	88.6	66.6	78.2	61.8	64.6	45.5
Full	**82.9**	84.2	**63.9**	**83.5**	84.0	63.6	**89.2**	67.1	**79.1**	**62.4**	**65.2**	**46.7**

4.4.2 Retrieval Study

Table 4 removes each retrieval stage in turn. Relational propagation extends retrieval beyond local appearance matching, and the largest degradation accordingly follows the removal of graph relations. When restricted to local similarity alone, retrieval misses the supporting evidence that relations connect and mistakes similar distractors for the target during its absence. Consolidation protects the bounded payload from redundancy, and its removal contributes the next largest share. Without it, repeated selections from the same trajectory could crowd the fixed budget that memory attention receives. The smallest share belongs to routing, whose role lies in confinement rather than accuracy. Routing determines when long-term evidence is consulted rather than what is retrieved, and confining that access to the predictions that need it preserves the base model's behavior elsewhere.

4.4.3 Management Study

Table 4 disables each maintenance mechanism under the same memory budget. Temporal support serves as a usable proxy for the quality of what enters. Removing temporal validation admits unverified observations into long-term memory and lowers the aggregate score on every benchmark. Even the observations that pass this gate can waste capacity once they recur. Redundancy compaction reclaims the space that repeated appearances occupy, and its removal drops the aggregate most on MOSEv2. The budget eventually forces deletion, and the appearances that matter after a long absence are rarely the newest ones. Replacing utility-based allocation with FIFO deletion then discards evidence by age alone and produces the clearest drop in the reappearance scores. The three mechanisms therefore guard the same memory in sequence, against contamination at entry, waste while stored, and premature deletion at exit. Section B.2 reports these studies on all seven benchmarks, together with further ablations on graph construction and the memory layers.

5 Related Work

Space-time memory networks (Oh et al., 2019; Cheng et al., 2021) segment the current frame by densely matching its features against stored past frames, and subsequent work refines this recipe (Yang et al., 2021; Yang & Yang, 2022; Seong et al., 2021; Cheng et al., 2024) as well as the organization (Cheng & Schwing, 2022), reading (Xie et al., 2021), and retention (Zhou et al., 2024; Liu et al., 2022; 2025; Qiao et al., 2026; Dang et al., 2025) of the memory. SAM-style models (Ravi et al., 2024; Carion et al., 2025) adapt this design to promptable streaming segmentation. Their extensions adjust propagation (Ding et al., 2024), screen what enters the bank (Videnovic et al., 2024; Alansari et al., 2026), supply context beyond the stored frames (Zhang et al., 2025; Hu et al., 2025), or refine the bank's writing rules (Yang et al., 2024; Xu et al., 2025; Chen et al., 2025; Kim & Lim, 2026), and a recent study surveys these strategies (Alansari et al., 2025). Across both lines, the unit of storage and management remains an intact observation or its per-location features, and the relations among the target's partial appearances stay outside the model. Within segmentation, EGMN (Lu et al., 2020) arranges stored frames in a fully connected graph with learned read and write controllers, and DTM-VOS (Fan et al., 2024) links local regions of neighboring frames within a short window for feature refinement. Outside segmentation, HippoRAG (Gutiérrez et al., 2024; 2025) propagates the query signal with Personalized PageRank, and SelectStream (Ge et al., 2026) conditions retrieval on the query over a bounded latent graph. MEMOIR instead stores relation-linked regions of the frozen segmenter's own memory tokens and separates their retention by BAM from their query-conditioned exposure by QCGR, without training. Section F provides an extended discussion of each line.

6 Conclusion

This paper addressed long-term video object segmentation, in which a bounded memory must preserve the historical appearance evidence that identifies a target through prolonged absence and distraction. We located the bottleneck of long-term propagation in the unit of memory and proposed MEMOIR, which stores history as spatially coherent regions of the frozen base model's own memory tokens. SGM organized these regions into a graph whose spatial, similarity, and temporal relations preserve how appearances relate across space and time. QCGR exposed long-term evidence only where recent observations fail to support the prediction. BAM maintained the graph within explicit node and token budgets according to temporal support and retrieval utility rather than age. Across seven video object segmentation benchmarks, MEMOIR improved on the SAM 3 base model with the largest margins on the long and crowded settings, and it exceeded SAM 3 on all four reappearance splits, which probe the regime that motivates long-term memory. These results indicate that once the unit of memory follows related partial appearances rather than whole frames, a bounded history remains both diverse and selectively accessible, and MEMOIR attains this without retraining or altering the base model.

AI USE STATEMENT

In this work, we used generative AI tools to formulate mathematical claims, implement methods, provide feedback on the research methodology and experimental design, help develop the theoretical model and conceptual framework, assist in writing proofs, and assist with translation. We have not used generative AI tools to provide critical ingredients for proving mathematical claims, propose or refine hypotheses, clean or reformat datasets, or interpret results; generating synthetic datasets and qualitative or thematic data analysis are not applicable to this work. Additionally, we used generative AI tools to debug and edit software code, create and modify scientific figures, edit the manuscript to improve readability, identify relevant literature, and format references. We have reviewed all AI-assisted work: all mathematical claims and proofs were verified step by step by the authors, AI-generated code was tested for correctness, all cited literature was checked against the original sources, and AI-suggested ideas were checked for potential overlap with prior work through a manual literature survey. We take responsibility for the final content of this work, including text, claims, and artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

Code is provided in the supplementary material. Implementation details and hyperparameters are given in Section C, and proofs are in Section E.

REFERENCES

Mohamad Alansari, Muzammal Naseer, Hasan Al Marzouqi, Naoufel Werghi, and Sajid Javed. Rethinking memory design in sam-based visual object tracking. *arXiv preprint arXiv:2512.22624*, 2025.

Mohamad Alansari, Yonathan Michael, Hasan AlMarzouqi, Muzammal Naseer, Naoufel Werghi, and Sajid Javed. SENTRY: SAM2-enhanced neighbor-aware and temporally reasoned memory for visual tracking. *arXiv preprint arXiv:2606.24449*, 2026.

Nicolas Carion, Laura Gustafson, Yuan-Ting Hu, Shoubhik Debnath, Ronghang Hu, Didac Suris, Chaitanya Ryali, Kalyan Vasudev Alwala, Haitham Khedr, Andrew Huang, Jie Lei, Tengyu Ma, Baishan Guo, Arpit Kalla, Markus Marks, Joseph Greer, Meng Wang, Peize Sun, Roman Rädle, Triantafyllos Afouras, Effrosyni Mavroudi, Katherine Xu, Tsung-Han Wu, Yu Zhou, Liliane Momeni, Rishi Hazra, Shuangrui Ding, Sagar Vaze, Francois Porcher, Feng Li, Siyuan Li, Aishwarya Kamath, Ho Kei Cheng, Piotr Dollár, Nikhila Ravi, Kate Saenko, Pengchuan Zhang, and Christoph Feichtenhofer. SAM 3: Segment anything with concepts. *arXiv preprint arXiv:2511.16719*, 2025.

Ruixiang Chen, Guolei Sun, Yawei Li, Jie Qin, and Luca Benini. HiM2SAM: Enhancing SAM2 with hierarchical motion estimation and memory optimization towards long-term tracking. *arXiv preprint arXiv:2507.07603*, 2025.

Ho Kei Cheng and Alexander G. Schwing. Xmem: Long-term video object segmentation with an atkinson-shiffrin memory model. In *Computer Vision – ECCV 2022*, pp. 633–649, 2022.

Ho Kei Cheng, Yu-Wing Tai, and Chi-Keung Tang. Rethinking space-time networks with improved memory coverage for efficient video object segmentation. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

Ho Kei Cheng, Seoung Wug Oh, Brian Price, Joon-Young Lee, and Alexander Schwing. Putting the object back into video object segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3151–3161, 2024.

Alexis Conneau, Guillaume Lample, Marc'Aurelio Ranzato, Ludovic Denoyer, and Hervé Jégou. Word translation without parallel data. In *International Conference on Learning Representations (ICLR)*, 2018. URL https://openreview.net/forum?id=H196sainb.

Jisheng Dang, Huicheng Zheng, Xiaohao Xu, Longguang Wang, Q. Hu, and Yulan Guo. Adaptive sparse memory networks for efficient and robust video object segmentation. *IEEE Transactions on Neural Networks and Learning Systems*, 36(2):3820–3833, 2025. doi: 10.1109/tnnls.2024.3357118.

Henghui Ding, Chang Liu, Shuting He, Xudong Jiang, Philip H. S. Torr, and Song Bai. MOSE: A new dataset for video object segmentation in complex scenes. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023.

Henghui Ding, Kaining Ying, Chang Liu, Shuting He, Xudong Jiang, Yu-Gang Jiang, Philip H. S. Torr, and Song Bai. MOSEv2: A more challenging dataset for video object segmentation in complex scenes. *arXiv preprint arXiv:2508.05630*, 2025.

Shuangrui Ding, Rui Qian, Xiaoyi Dong, Pan Zhang, Yuhang Zang, Yuhang Cao, Yuwei Guo, Dahua Lin, and Jiaqi Wang. SAM2Long: Enhancing SAM 2 for long video segmentation with a training-free memory tree. *arXiv preprint arXiv:2410.16268*, 2024.

Jiaqing Fan, Shenglong Hu, Long Wang, Kaihua Zhang, and Bo Liu. Dual temporal memory network with high-order spatio-temporal graph learning for video object segmentation. *Image and Vision Computing*, 150:105208, 2024. doi: 10.1016/j.imavis.2024.105208.

Haonan Ge, Yiwei Wang, Hang Wu, and Yujun Cai. What should a streaming video model remember?, 2026. URL https://arxiv.org/abs/2606.16353.

Bernal Jiménez Gutiérrez, Yiheng Shu, Yu Gu, Michihiro Yasunaga, and Yu Su. Hipporag: Neurobiologically inspired long-term memory for large language models. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 59532–59569. Curran Associates, Inc., 2024. doi: 10.52202/079017-1902. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/6ddc001d07ca4f319af96a3024f6dbd1-Paper-Conference.pdf.

Bernal Jiménez Gutiérrez, Yiheng Shu, Weijian Qi, Sizhe Zhou, and Yu Su. From RAG to memory: Non-parametric continual learning for large language models. In Aarti Singh, Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien, Felix Berkenkamp, Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu (eds.), *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 21497–21515. PMLR, 13–19 Jul 2025. URL https://proceedings.mlr.press/v267/gutierrez25a.html.

Lingyi Hong, Zhongying Liu, Wenchao Chen, Chenzhi Tan, Yuang Feng, Xinyu Zhou, Pinxue Guo, Jinglun Li, Zhaoyu Chen, Shuyong Gao, Wei Zhang, and Wenqiang Zhang. LVOS: A benchmark for large-scale long-term video object segmentation. *arXiv preprint arXiv:2404.19326*, 2024.

Hengrui Hu, Kaining Ying, and Henghui Ding. Segment anything across shots: A method and benchmark. *arXiv preprint arXiv:2511.13715*, 2025.

Glen Jeh and Jennifer Widom. Scaling personalized web search. In *Proceedings of the 12th International Conference on World Wide Web (WWW)*, pp. 271–279. Association for Computing Machinery, 2003. doi: 10.1145/775152.775191.

JeongRae Kim and Changwon Lim. Sam2dual: Training-free, dual memory for long-term video object segmentation. *arXiv preprint arXiv:2608.18640*, 2026.

Mingxing Li, Li Hu, Zhiwei Xiong, Bang Zhang, Pan Pan, and Dong Liu. Recurrent dynamic embedding for video object segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022.

Yong Liu, Ran Yu, Fei Yin, Xinyuan Zhao, Wei Zhao, Weihao Xia, and Yujiu Yang. Learning quality-aware dynamic memory for video object segmentation. In *Computer Vision – ECCV 2022*, pp. 408–425, 2022.

Yong Liu, Ran Yu, Fei Yin, Xinyuan Zhao, Wei Zhao, Weihao Xia, Jiahao Wang, Yitong Wang, Yansong Tang, and Yujiu Yang. Learning high-quality dynamic memory for video object segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 47(5), 2025. doi: 10.1109/tpami.2025.3532306.

Xiankai Lu, Wenguan Wang, Martin Danelljan, Tianfei Zhou, Jianbing Shen, and Luc Van Gool. Video object segmentation with episodic graph memory networks. In *Computer Vision – ECCV 2020*, volume 11215 of *Lecture Notes in Computer Science*, pp. 661–679. Springer International Publishing, 2020. doi: 10.1007/978-3-030-58580-8_39.

Seoung Wug Oh, Joon-Young Lee, Ning Xu, and Seon Joo Kim. Video object segmentation using space-time memory networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 9225–9234, 2019.

Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. The PageRank citation ranking: Bringing order to the web. Technical Report 1999-66, Stanford InfoLab, 1999. URL `http://ilpubs.stanford.edu:8090/422/`.

Jordi Pont-Tuset, Federico Perazzi, Sergi Caelles, Pablo Arbeláez, Alex Sorkine-Hornung, and Luc Van Gool. The 2017 DAVIS challenge on video object segmentation. *arXiv preprint arXiv:1704.00675*, 2017.

Shengye Qiao, Changqun Xia, Yanjie Liang, Xiaowu Chen, and Jia Li. Learning when and how to update memory for video object segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 48(9), 2026. doi: 10.1109/tpami.2026.3684742.

Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloe Rolland, Laura Gustafson, Eric Mintun, Junting Pan, Kalyan Vasudev Alwala, Nicolas Carion, Chao-Yuan Wu, Ross Girshick, Piotr Dollár, and Christoph Feichtenhofer. SAM 2: Segment anything in images and videos. *arXiv preprint arXiv:2408.00714*, 2024.

Hongje Seong, Seoung Wug Oh, Joon-Young Lee, Seongwon Lee, Suhyeon Lee, and Euntai Kim. Hierarchical memory matching network for video object segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021.

Jovana Videnovic, Alan Lukezic, and Matej Kristan. A distractor-aware memory for visual object tracking with SAM2. *arXiv preprint arXiv:2411.17576*, 2024.

Haozhe Xie, Hongxun Yao, Shangchen Zhou, Shengping Zhang, and Wenxiu Sun. Efficient regional memory network for video object segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

Ning Xu, Linjie Yang, Yuchen Fan, Dingcheng Yue, Yuchen Liang, Jianchao Yang, and Thomas Huang. YouTube-VOS: A large-scale video object segmentation benchmark. *arXiv preprint arXiv:1809.03327*, 2018.

Qianxiong Xu, Lanyun Zhu, Chenxi Liu, Guosheng Lin, Cheng Long, Ziyue Li, and Rui Zhao. SAMITE: Position prompted SAM2 with calibrated memory for visual object tracking. *arXiv preprint arXiv:2507.21732*, 2025.

Cheng-Yen Yang, Hsiang-Wei Huang, Wenhao Chai, Zhongyu Jiang, and Jenq-Neng Hwang. SAMURAI: Adapting segment anything model for zero-shot visual tracking with motion-aware memory. *arXiv preprint arXiv:2411.11922*, 2024.

Zongxin Yang and Yi Yang. Decoupling features in hierarchical propagation for video object segmentation. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

Zongxin Yang, Yunchao Wei, and Yi Yang. Associating objects with transformers for video object segmentation. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

Zhixiong Zhang, Shuangrui Ding, Xiaoyi Dong, Songxin He, Jianfan Lin, Junsong Tang, Yuhang Zang, Yuhang Cao, Dahua Lin, and Jiaqi Wang. Advancing complex video object segmentation via progressive concept construction. *arXiv preprint arXiv:2507.15852*, 2025.

Junbao Zhou, Ziqi Pang, and Yu-Xiong Wang. Rmem: Restricted memory banks improve video object segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

Appendix

TABLE OF CONTENTS

A Additional Method Details **14**

A.1 Structured Graph Memory Details 14

A.2 Query-Conditioned Graph Retrieval Details 15

A.3 Budget-Aware Graph Memory Management Details 16

B Supplementary Experiments **17**

B.1 Qualitative Results 17

B.2 More Ablations 18

B.3 Results on Visual Object Tracking 19

B.4 Performance on SAM2 20

C Implementation Details **21**

D Cost Analysis **21**

E Proofs of Propositions **23**

E.1 Proof of Proposition 1 23

E.2 Proof of Proposition 2 24

F Extended Related Work **25**

G Limitations **26**

A Additional Method Details

A.1 Structured Graph Memory Details

A.1.1 Region Partitioning Details

This section details the feature-guided agglomeration that produces the partition $\mathcal{R}_t$ in Section 3.4. SGM requires each region to hold one coherent appearance of the frame, and the partition is therefore derived from the content of the current representation $F_t = \{f_{t,p}\}$ rather than from a fixed layout. Feature-guided agglomeration operates on the four-connected feature grid, on which every location is initially an individual region. For a region A, its mean feature and intra-region variance are defined as follows:

$$\mu_A = \frac{1}{|A|} \sum_{p \in A} f_{t,p}, \qquad \sigma_A^2 = \frac{1}{|A|} \sum_{p \in A} \left\| f_{t,p} - \mu_A \right\|_2^2, \tag{5}$$

where $|A|$ counts the grid locations of A. For two spatially adjacent regions A and B, the feature distance and the merge criterion are defined as follows:

$$d(A,B) = \sqrt{\|\mu_A - \mu_B\|_2^2 + \sigma_A^2 + \sigma_B^2}, \qquad d(A,B) < \tau_g + \frac{\kappa}{\min(|A|,|B|)}, \tag{6}$$

where τ_g denotes the merge threshold on feature distance and κ scales the size-dependent term. The variance terms admit a merge only when both regions are internally homogeneous and their means agree. The size-dependent term relaxes the threshold for small regions and absorbs isolated fragments into their neighbors, whereas for two large regions the criterion approaches $d(A,B) < \tau_g$.

Agglomeration repeatedly merges adjacent regions that satisfy Eq. (6) and stops once no adjacent pair satisfies it. The remaining regions form the partition $\mathcal{R}_t = \{\Omega_{t,1}, \ldots, \Omega_{t,R_t}\}$. Since each merge joins two adjacent regions, every region of $\mathcal{R}_t$ forms a four-connected set of grid locations with homogeneous features, which defines spatial coherence operationally in SGM. The number of regions R_t follows the content of the frame rather than a preset grid size.

The partition is computed once per frame, on F_t before the prediction, and first supplies the query regions of QCGR at frame t. After the prediction, SGM reuses the same partition to organize the encoded memory M_t: each region $\Omega_{t,r}$ selects the spatially corresponding tokens of M_t as its region memory $X_{t,r}$ defined in Section 3.4. The partition is thus determined in the image-encoder feature space, while the graph stores the native mask-memory tokens without constructing a new memory representation, and retrieval compares units of one kind on its query and memory sides.

Each region is further labeled as foreground or background: $\Omega_{t,r}$ is a foreground region if the fraction of its tokens inside the predicted mask $\hat{Y}_t$ on the $H_m \times W_m$ grid exceeds ρ_{fg}, and a background region otherwise. This label determines which regions token downsampling may reduce (Section A.3).

A.1.2 Graph Construction Details

This section details the region affinity and the linking rules of the three edge types in Section 3.4. For tokens a_i of region A and b_j of region B with cosine similarity c_{ij}, the corrected similarity and the region affinity are defined as follows:

$$\tilde{c}_{ij} = 2c_{ij} - r_A(i) - r_B(j), \qquad w(A,B) = \frac{1}{\max(|A|,|B|)} \sum_{(i,j) \in \mathcal{M}(A,B)} \max(c_{ij}, 0), \tag{7}$$

where $r_A(i)$ denotes the mean cosine similarity from a_i to its (up to) k nearest counterparts in B, $r_B(j)$ denotes the mean cosine similarity from b_j to its (up to) k nearest counterparts in A, and $\mathcal{M}(A,B)$ contains the token pairs that are mutual best matches under $\tilde{c}_{ij}$ with $\tilde{c}_{ij} > 0$. The correction is a pair-local form of cross-domain similarity local scaling (CSLS) Conneau et al. (2018) and lowers the scores of tokens that resemble many counterparts at once. It is used only to select reliable correspondences, while the affinity aggregates the original positive cosine evidence, so that $w(A,B) \in [0,1]$. Normalization by the larger region prevents a small region from appearing similar to a large one through a few matched tokens. To bound the cost of pairwise token matching, each region is spatially pooled to at most N_{pool} tokens before its affinities are computed, and $|A|$ and $|B|$

count the pooled tokens. If either region has fewer than two pooled tokens, CSLS is skipped and matches are selected by positive cosine.

Within an observation, similarity edges $\mathcal{E}_{\mathrm{sim}}$ connect appearance-related regions anywhere in the same observation, namely pairs whose affinity exceeds the similarity threshold τ_{sim}. Spatial edges $\mathcal{E}_{\mathrm{spa}}$ connect nodes whose regions Ω_i and Ω_j are neighbors on the four-connected grid, denoted $\Omega_i \sim_4 \Omega_j$, and that are not already connected by a similarity edge, so the two edge sets are disjoint.

Across observations, let $\mathcal{C}_i$ denote the regions of the tail units of all current trajectories, which serve as the temporal candidates of a new node v_i. SGM considers only the best-scoring candidate $v_j^* \in \mathcal{C}_i$, without falling back to other candidates, and accepts it either when $w(v_i, v_j^*)$ exceeds the temporal-edge threshold τ_{temp} or when it is the unique best match with positive affinity and a modified z-score above τ_z:

$$z_i(v_j) = \frac{0.6745\,\big(w(v_i, v_j) - \mathrm{med}_i\,\big)}{\mathrm{MAD}_i}, \tag{8}$$

where med_i and MAD_i denote the median and the median absolute deviation of the affinities between v_i and its candidates. The relative test complements the absolute threshold: it retains correspondences that stand out among the candidates even when the affinity itself remains modest.

Accepted matches organize the graph into storage units and trajectories. A storage unit U is the set of regions of a single frame that join the same trajectory: the new regions whose accepted matches fall in the same tail unit together form one new unit U_i, which becomes that trajectory's tail, while each region without an accepted match starts a new trajectory as a single-region unit. A trajectory is therefore a non-branching, time-ordered chain of units $U_1, U_2, \ldots$, and consecutive units need not come from adjacent frames, since the tail of a trajectory that is not continued remains matchable. Storage units are the atomic elements of memory management: they are never split and are created, deleted, or compressed only as a whole, whereas retrieval scores and seeds individual regions.

Spatial and similarity edges carry $w(v_i, v_j)$ directly as their weights. A temporal edge connects two consecutive units (U_{i-1}, U_i) of a trajectory and carries the mean affinity of the correspondences accepted between them, and during propagation, each region passes this weight evenly to the regions of the adjacent unit. When a unit is deleted, its surviving neighbors on the trajectory are re-linked with a recomputed edge weight.

A.2 Query-Conditioned Graph Retrieval Details

This section details how Readout turns the output of Route and Retrieve into the memory payload of Eq. (4). Route selects the query regions $\mathcal{R}_t^{\mathrm{LT}}$ that lack recent support, and Retrieve ranks the graph nodes by their propagated relevance $\mathbf{p}$ and collects $\mathcal{V}_t^{\mathrm{ret}}$ under the retrieval budgets.

Consolidation proceeds independently within each temporal trajectory. The retrieved node with the highest relevance in $\mathbf{p}$ is selected as a representative, and another retrieved node of the trajectory remains covered while its temporal path affinity $C(i, j)$ to the representative meets the consolidation threshold τ_{con}. A retrieved node outside this coverage is selected as a new representative, and each representative together with the nodes it covers forms a redundant segment of the trajectory, from which only the representative enters the payload. Consolidation never merges nodes of different trajectories. Within a covered observation, it suppresses only the region that corresponds temporally to the representative, and the other regions of that observation remain available for readout. Redundancy is thus removed at the level of regions rather than frames.

The retained regions are then assembled into the native memory payload without modifying their stored features. Each retrieved region directly reuses its native mask-memory tokens together with the spatial positional encoding of its original spatial support. No temporal positional encoding is added to long-term region tokens, while the short- and middle-term memories retain their native temporal semantics, and the middle-term frames enter in full to supply complete appearances of the target from its latest object-present observations. The resulting long-term payload is combined with the unchanged conditioning memory M_0, the short- and middle-term memories, and the compressed history described in Section A.3.

For SAM3, object pointers are selected separately from the region-level dense memory. We sum the PPR relevance of retrieved regions at the source-frame level, and the highest-ranked historical

frames contribute their original SAM3 object pointers to provide global object-level information. Of the up to $K_p = 15$ native object pointers of SAM3 (Section E), we keep the eight most recent, and graph retrieval contributes up to eight further pointers (Table 12). This preserves the native object representation while allowing region-level graph retrieval to determine which historical observations are exposed through the pointer channel. The temporal positional encoding of each selected object pointer is determined by the actual temporal distance between its source frame and the current frame, rather than by its retrieval rank or position in the graph payload. Thus, dense memory remains region-specific, while object pointers preserve the whole-frame object representation and temporal semantics of SAM3.

A.3 Budget-Aware Graph Memory Management Details

Budget-Aware Graph Memory Management answers the retention question of Section 3.1: it governs which historical evidence remains under finite storage capacity according to temporal support and retrieval utility. BAM maintains the graph history and compressed history from which Eq. (4) draws long-term evidence. Along the update path, it follows native memory encoding and graph insertion after prediction. The storage budgets B_N^{mem} and B_T^{mem} limit retained graph regions and retained native memory tokens, respectively. These limits bound graph-operation complexity and physical token storage, whereas K_{ret} and B_T constrain the retrieval payload. Temporal validation applies when a region leaves its protected interval regardless of budget occupancy. A budget violation triggers redundancy compaction, followed by utility-guided budget allocation if capacity remains exceeded.

Temporal validation establishes a temporal-support requirement for retention before capacity pressure determines which evidence survives. Regions from the latest K_{prot} graph observations remain protected from removal by validation and from reduction by compaction, allocation, or downsampling. Once a region leaves this interval, validation removes it if it has never participated in a temporal edge. The test uses the temporal correspondences established by SGM and concerns the region's participation history rather than its current temporal degree. This delayed check permits temporal evidence to accumulate before unsupported regions are removed. In contrast to Route's query-time stability test, validation determines whether stored regions satisfy the support requirement for continued retention. The short-term FIFO preserves complete recent frames, while the middle-term designation changes graph-region readout without changing the regions' lifecycle or maintenance.

For evidence eligible for reduction, historical retrieval utility supplies a common retention criterion for compaction and allocation. It measures the relevance received throughout a node's lifetime rather than its relevance to the current query alone. A removable storage unit U contains one or more regions along a temporal trajectory. The node utility $u(v)$ and unit utility $u(U)$ are defined as follows:

$$u(v) = \frac{1}{|\mathcal{A}_v|} \sum_{a \in \mathcal{A}_v} p_a(v), \qquad u(U) = \frac{1}{|U|} \sum_{v \in U} u(v), \tag{9}$$

where $\mathcal{A}_v$ contains the retrieval actions during the lifetime of node v, $p_a(v)$ denotes its Personalized PageRank relevance in action a, and $|U|$ counts the regions in unit U. Each retrieval action contributes its relevance to the lifetime average, with zero contribution when the action does not reach the node. Unreached actions remain in the denominator rather than disappearing from the utility estimate.

Redundancy compaction reclaims storage occupied by repeated appearances that have passed temporal validation. A recurring appearance can contribute a region to each observation in which it appears. Under budget pressure, compaction groups consecutive units of a temporal trajectory according to the existing path affinity $C(i, j)$ and the compaction threshold τ_{cmp}. Units are grouped while their cumulative path affinity meets this threshold, and only the unit with the highest historical utility remains as the segment's representative. That unit represents the segment with native tokens from its own observation. Later compaction does not reconsider previously compacted history. Unlike Readout consolidation in Section 3.5, which selects payload representatives by current relevance, compaction selects stored representatives by historical utility. Compaction permanently reduces stored regions, whereas consolidation affects the current payload.

Utility-guided budget allocation prioritizes historical usefulness when compaction alone cannot satisfy the node budget. Among removable units, allocation discards lower-utility units first, with removal proceeding from the oldest end of each temporal trajectory. Historical utility therefore determines the preference among eligible units, while trajectory age constrains the direction of

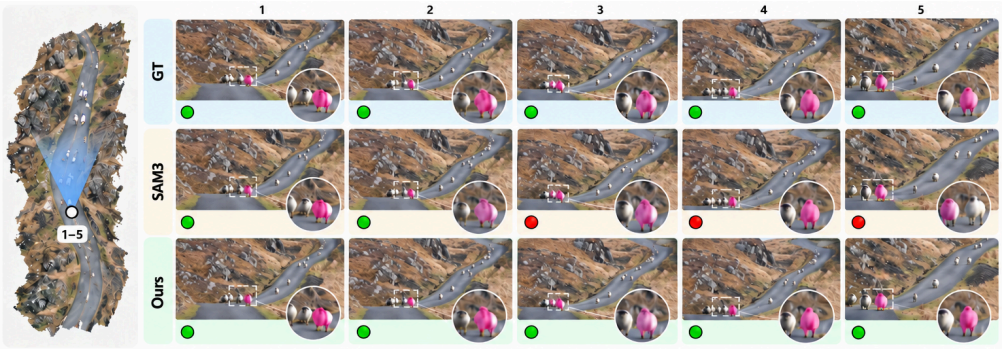

Figure 9: **Additional qualitative comparison with SAM3** on a sequence that combines a crowded scene with severe occlusion.

removal. The FIFO ablation replaces the utility priority with deletion by age alone. This distinction concerns accumulated retrieval relevance rather than whether a region remains reachable by the current query.

Token downsampling addresses residual token-storage pressure after region-level reduction. If token capacity remains exceeded, BAM downsamples lower-utility background regions outside the protected interval, while foreground regions retain their original resolution. Downsampling reduces the number of stored tokens without changing the region's graph identity or relations. This operation adjusts the storage occupied by a retained region without requiring its removal from the graph.

Anchor compression preserves temporally persistent information from each evicted historical prefix before the prefix is discarded. An anchor is defined as a bounded summary of such a prefix along one temporal trajectory and remains outside the graph. To extract persistent information, compression processes consecutive evicted units U_{i-1} and U_i through their native token sets $X(U_{i-1})$ and $X(U_i)$. Adjacent uncompressed units are aligned by their spatial centers. Token correspondences are then searched within a $w_{\text{anc}} \times w_{\text{anc}}$ neighborhood and retained when their cosine similarity exceeds τ_{anc}. Consecutive correspondences form appearance traces, and the mean of each trace forms an anchor token. The anchor capacity Q_i follows an exponential blend of the previous capacity Q_{i-1} and the latest unit's token count $|X(U_i)|$, with weighting controlled by β_{anc}. Compression favors temporally continuing traces and does not directly retain isolated appearances.

These anchors reconnect compressed history to retrieval through their associated live trajectories. The relevance of anchor A_j at frame t is defined as follows:

$$s_t(A_j) = \sum_{v \in \mathcal{T}_j} p_v, \tag{10}$$

where A_j denotes the anchor associated with trajectory j, $\mathcal{T}_j$ contains that trajectory's live graph nodes, and p_v denotes the entry of the current relevance vector $\mathbf{p}$ for node v. Readout exposes relevant anchors according to this trajectory relevance under the compression readout budget B_T^{compress}, which limits the exposed anchor tokens. The selected compressed history enters $\mathcal{B}_t^{\text{compress}}$ in Eq. (4). Together, region compaction and anchor compression express the progression described in Section 3.3: older evidence is represented by fewer entries that stand for wider temporal spans, in an increasingly compressed form, while the tokens of every retained graph region remain native to a single observation.

B Supplementary Experiments

B.1 Qualitative Results

Fig. 9 presents an additional comparison with SAM3 on a sequence that combines a crowded scene with severe occlusion. While the target is heavily occluded, several nearby objects resemble its visible part, and SAM3 drifts to a distractor as the recent observations become ambiguous. MEMOIR maintains the target identity through the occlusion by matching the query against retrieved historical appearances of the target, which remain distinguishable from the surrounding distractors.

Table 5: Ablation on graph node granularity. We compare frame-level, grid-level (6×6), and region-level memory nodes, all built on SAM 3. **Bold** indicates the best performance, and underline indicates the second-best performance.

Node Granularity	$\mathcal{J}\&\mathcal{F}$					$\mathcal{G}$	$\mathcal{J}\&\dot{\mathcal{F}}$
	SA-V val	SA-V test	LVOS v2 val	MOSE v1 val	DAVIS 2017 val	YTVOS 2019 val	MOSE v2 val
Frame	83.2	84.5	89.3	78.3	**91.8**	**90.0**	60.8
Grid (6×6)	83.8	84.9	89.7	**78.4**	**91.8**	89.9	61.6
Region	**84.3**	**85.0**	**90.1**	**78.4**	**91.8**	**90.0**	**62.4**

Table 6: Ablation on graph-based retrieval, built on SAM 3. We remove query routing (w/o Route), memory consolidation (w/o Consol.), and graph-based PPR retrieval (w/o Graph). **Bold** indicates the best performance, and underline indicates the second-best performance. † Retrieval uses feature similarity only, without the memory graph.

Variant	$\mathcal{J}\&\mathcal{F}$					$\mathcal{G}$	$\mathcal{J}\&\dot{\mathcal{F}}$
	SA-V val	SA-V test	LVOS v2 val	MOSE v1 val	DAVIS 2017 val	YTVOS 2019 val	MOSE v2 val
w/o Route	84.2	84.8	89.5	78.2	**91.8**	**90.0**	61.9
w/o Consol.	83.3	84.3	88.4	78.3	**91.8**	**90.0**	60.6
w/o Graph†	82.0	83.1	87.0	77.9	**91.8**	**90.0**	59.7
Full	**84.3**	**85.0**	**90.1**	**78.4**	**91.8**	**90.0**	**62.4**

B.2 More Ablations

In this section, we report all ablations under the standard evaluation protocol on all seven benchmarks, which cover both long-term and short-term settings. We further include two additional studies on graph construction and memory layer composition. All variants are built on SAM3, and each variant modifies a single component while keeping the rest of the framework unchanged. Across all studies, the variation on the short-term benchmarks is small. All variants equipped with graph memory obtain the same score on DAVIS 2017, and their scores differ by at most 0.2 points on YouTube-VOS 2019 and 0.6 points on MOSEv1. The differences between variants therefore concentrate on SA-V, LVOSv2, and MOSEv2.

B.2.1 Graph Representation Study

Performance increases monotonically from frame-level to grid-level to region-level memory on all four long-term settings. The margin of region-level over grid-level memory is largest on MOSEv2. We attribute this to the fact that fixed grid cells can cut across object parts or mix target and background tokens, which becomes more harmful in longer videos where more historical evidence must be retrieved from long-term memory.

B.2.2 Retrieval Study

The full retrieval pipeline achieves the best result on every benchmark. Similarity-only retrieval is the only variant that shows a clear drop on MOSEv1. Since MOSEv1 contains many crowded scenes with similar objects, this result suggests that direct feature matching can retrieve evidence belonging to distractors, whereas graph propagation favors memories that are supported by their relations to other relevant nodes.

B.2.3 Management Study

Table 7 reports the management study on all seven benchmarks. The full pipeline achieves the best or tied-best result on every benchmark, and the differences concentrate on the long-term settings, where evidence must survive over long durations before it becomes useful. Removing temporal validation lowers performance consistently across the long-term settings, whereas removing redundancy compaction has a generally smaller effect, since consolidation in readout already prevents repeated appearances from misleading retrieval. FIFO deletion lowers performance on most long-term settings, with the largest drop in the table on SA-V test.

Table 7: Ablation on memory management, built on SAM 3. We remove temporal validation (w/o Temp. Val.) and redundancy-aware selection (w/o Redund.), and replace utility-based deletion with FIFO. **Bold** indicates the best performance, and underline indicates the second-best performance. ‡ Utility-based deletion is replaced by first-in-first-out (FIFO) deletion.

Variant	$\mathcal{J}\&\mathcal{F}$					$\mathcal{G}$	$\mathcal{J}\&\dot{\mathcal{F}}$
	SA-V val	SA-V test	LVOS v2 val	MOSE v1 val	DAVIS 2017 val	YTVOS 2019 val	MOSE v2 val
w/o Temp. Val.	83.8	84.6	89.4	**78.4**	**91.8**	**90.0**	61.9
w/o Redund.	84.0	84.9	89.7	78.2	**91.8**	**90.0**	61.9
FIFO‡	**84.3**	84.0	89.5	78.3	**91.8**	**90.0**	61.8
Full	**84.3**	**85.0**	**90.1**	**78.4**	**91.8**	**90.0**	**62.4**

Table 8: Ablation on graph construction, built on SAM 3. We remove similarity edges (w/o Sim. Edge) and the modified z-score filtering (w/o Mod. z-score). **Bold** indicates the best performance, and underline indicates the second-best performance.

Variant	$\mathcal{J}\&\mathcal{F}$					$\mathcal{G}$	$\mathcal{J}\&\dot{\mathcal{F}}$
	SA-V val	SA-V test	LVOS v2 val	MOSE v1 val	DAVIS 2017 val	YTVOS 2019 val	MOSE v2 val
w/o Sim. Edge	84.1	84.7	**90.1**	78.3	**91.8**	**90.0**	62.2
w/o Mod. z-score	84.1	84.5	89.8	78.1	**91.8**	89.8	61.1
Full	**84.3**	**85.0**	**90.1**	**78.4**	**91.8**	**90.0**	**62.4**

B.2.4 Graph Construction Study

We evaluate two design choices in graph construction. The *w/o Sim. Edge* variant removes similarity edges while keeping spatial and temporal edges, and the *w/o Mod. z-score* variant accepts temporal edges only through the absolute affinity threshold.

Removing similarity edges consistently lowers performance on SA-V and MOSEv2, as these edges link appearance-related regions that are not spatially adjacent, such as target parts separated by occlusion. The drop is moderate because spatial and temporal edges already provide substantial connectivity, making similarity edges a complementary relation. Removing the modified z-score criterion causes a larger degradation, lowering performance on every benchmark except DAVIS 2017. Without it, distinctive correspondences with moderate absolute affinity, such as those under appearance change, are rejected. The resulting fragmented trajectories cause valid regions to be pruned as temporally isolated, and further weaken consolidation, redundancy compaction, and deletion order, which all rely on temporal edges.

B.2.5 Memory Layers Study

We evaluate the contribution of each memory layer. The *Short + Mid* variant corresponds to the native SAM3 memory design without graph memory. The *Short + Long* variant replaces middle-term memory with graph-based long-term memory, and the *w/o Compr.* variant uses all memory layers but disables anchor compression.

Short + Long performs below SAM3 on all benchmarks except YouTube-VOS 2019, indicating that partial regions retrieved from long-term memory cannot replace the complete recent appearance provided by middle-term memory. When the two layers are combined, performance surpasses SAM3 on all long-term settings, with the largest gains on LVOSv2 and MOSEv2. This shows that middle-term memory and long-term memory play complementary roles. Anchor compression brings further consistent improvements by retaining persistent appearance information from history that has been evicted from the graph.

B.3 Results on Visual Object Tracking

Setup. To examine whether our memory framework transfers beyond semi-supervised VOS, we further evaluate it on five visual object tracking benchmarks: LaSOT, LaSOT Extension, TNL2K, GOT-10k, and TrackingNet. We compare against SAM3 and the SAM3-based variants evaluated in *Rethinking Memory Design in SAM-Based Visual Object Tracking* Alansari et al. (2025), including

Table 9: Ablation on memory layers, built on SAM 3. Short + Mid corresponds to the original SAM 3 memory design, taken from the same reproduction as the SAM 3 baseline in Table 2. Full uses short-term, middle-term, long-term, and anchor memory; w/o Compr. uses the full layer design without long-term memory compression. **Bold** indicates the best performance, and underline indicates the second-best performance.

Memory Layers	$\mathcal{J}\&\mathcal{F}$					$\mathcal{G}$	$\mathcal{J}\&\dot{\mathcal{F}}$
	SA-V val	SA-V test	LVOS v2 val	MOSE v1 val	DAVIS 2017 val	YTVOS 2019 val	MOSE v2 val
Short + Mid (SAM 3)	83.5	84.4	88.5	**78.4**	**92.2**	89.7	60.0
Short + Long	82.8	84.0	87.5	77.8	91.8	89.8	59.3
w/o Compr.	83.7	84.6	89.9	78.3	91.8	**90.0**	62.1
Full	**84.3**	**85.0**	**90.1**	**78.4**	91.8	**90.0**	**62.4**

Table 10: Comparison with SAM3-based visual object trackers on five tracking benchmarks. We report success score (S), normalized precision (P_{norm}), and precision (P) on LaSOT, LaSOT$_{\text{ext}}$, TNL2K, and TrackingNet; and average overlap (AO), $\text{SR}_{0.50}$, and $\text{SR}_{0.75}$ on GOT-10k. The best result is shown in **bold**, and the second-best is underlined.

Method	**LaSOT**			**LaSOT$_{\text{ext}}$**			**TNL2K**			**GOT-10k**			**TrackingNet**		
	S	P_{norm}	P	S	P_{norm}	P	S	P_{norm}	P	AO	$\text{SR}_{0.50}$	$\text{SR}_{0.75}$	S	P_{norm}	P
SAM 3	74.7	82.0	78.8	63.8	78.5	75.5	64.2	84.4	71.9	84.5	94.6	82.2	86.6	91.2	88.2
SAM3Long	75.2	82.3	79.6	62.7	77.4	73.9	64.6	84.8	72.6	84.7	95.2	82.4	86.7	91.4	88.4
DAM4SAM-3	76.5	83.9	80.9	**66.0**	**81.3**	**78.6**	65.6	**86.0**	**73.7**	84.0	94.2	81.5	**86.8**	**91.5**	88.5
SAMURAI-3	75.2	82.3	79.4	62.6	77.1	74.3	64.9	85.0	72.7	84.4	94.4	81.9	86.5	91.1	88.2
SAMITE-3	**77.4**	**85.1**	**82.4**	65.3	80.2	77.3	65.1	85.4	72.8	83.9	94.2	81.2	86.7	**91.5**	**88.6**
HiM2SAM-3	75.6	82.8	79.6	62.9	77.1	74.0	64.8	84.9	72.5	84.1	94.1	81.8	86.6	91.2	88.2
Ours	76.9	84.0	81.1	63.9	75.3	73.7	**66.3**	83.3	73.6	**85.4**	**95.8**	**83.2**	86.5	91.1	87.9

SAM3Long, DAM4SAM-3, SAMURAI-3, SAMITE-3, and HiM2SAM-3, from which we obtain the corresponding baseline results. Following the same evaluation protocols, we report success (S), normalized precision (P_{norm}), and precision (P) on LaSOT, LaSOT Extension, TNL2K, and TrackingNet, and average overlap (AO), $\text{SR}_{0.50}$, and $\text{SR}_{0.75}$ on GOT-10k.

For tracking, the first frame supplies a ground-truth box rather than a mask, and the conditioning memory M_0 is therefore encoded from the predicted mask of that frame.

Results. Table 10 shows that the graph memory transfers to box-initialized tracking without any modification. Compared with SAM3, our method improves the success score on LaSOT, TNL2K, and GOT-10k, achieving the best results on all three GOT-10k metrics and the best success score on TNL2K, and ranks second on LaSOT behind SAMITE-3. These benchmarks contain long sequences and frequent occlusion or out-of-view events, where the target must be recovered from evidence beyond the recency window, which is the setting our memory is designed for. On TrackingNet, whose sequences are comparatively short, our method remains on par with SAM3, consistent with the behavior on the short VOS benchmarks, where recent observations already suffice. Precision on LaSOT$_{\text{ext}}$ lags behind the success score, as the gains come mainly from recovering the target after absence or occlusion, which the overlap-based success rewards more directly than the center-based precision.

B.4 Performance on SAM2

Results. To examine whether the graph memory depends on a particular base model, we apply MEMOIR to SAM 2.1, keeping the method unchanged and changing only the FGAL distance threshold ($\tau_g = 1.10$), while the 64×64 memory grid follows from the model. Table 11 shows that it improves SAM 2.1 on the long-term benchmarks, with the largest gains on SA-V val, LVOSv2, and MOSEv2, and achieves the best result on LVOSv2 among all SAM 2.1-based methods. These gains exceed those on SAM3, as the weaker native memory of SAM 2.1 leaves more long-term evidence unused and thus more room for retrieval to recover it. On the short-video benchmarks, where recent observations already suffice, the differences remain small.

Table 11: Results of MEMOIR applied to SAM 2.1 on standard video object segmentation benchmarks, compared with SAM 2.1-based methods. **Bold** indicates the best performance, and underline indicates the second-best performance. † Reported by SENTRY Alansari et al. (2026). ‡ Reported in MOSEv2 Ding et al. (2025).

Method	$\mathcal{J}\&\mathcal{F}$					$\mathcal{G}$	$\mathcal{J}\&\dot{\mathcal{F}}$
	SA-V val	SA-V test	LVOS v2 val	MOSE v1 val	DAVIS 2017 val	YTVOS 2019 val	MOSE v2 val
SAM 2.1	78.6	79.6	84.1	74.5	90.6	88.7	49.5
SAM2.1Long	81.1	81.2	85.9	75.2	**91.4**	88.7	51.5
SAMURAI	79.8	80.0	84.2	72.6	89.9	88.3	51.1
DAM4SAM ‡	80.5	81.0	85.5	75.6	–	–	51.2
SAMITE †	80.8	81.3	85.8	76.8	–	–	–
HiM2SAM †	81.0	81.5	86.1	**77.0**	–	–	–
SeC	**82.7**	**81.7**	86.5	75.3	91.3	88.6	**53.8**
SENTRY-S2	79.4	80.1	84.9	75.8	–	–	–
MEMOIR on SAM2	81.7	81.2	**87.3**	75.8	89.8	**88.9**	52.8

Comparison with SeC. SeC is the strongest competitor in this setting, and the two methods lead on different benchmarks. SeC builds a concept-level representation of the target with an LVLM and leads on SA-V and MOSEv2, whose crowded scenes and frequent distractors reward knowing what the target is. MEMOIR instead retains the specific partial appearances that each query needs and leads on LVOSv2, whose minute-long videos require recovering the target from appearances observed long before. The two methods also differ in cost. On SAM 2.1, each memory frame contains $64 \times 64 = 4{,}096$ tokens. SeC relies on a fine-tuned checkpoint with 22 memory slots, which amounts to $90{,}112$ memory tokens per frame, together with an additional LVLM. MEMOIR keeps the base model frozen and, without an LVLM, uses a nominal budget of $54{,}592$ memory tokens per frame: seven native memory frames ($28{,}672$ tokens) plus the same retrieval ($20{,}736$) and compressed ($5{,}184$) token budgets as on SAM3.

C Implementation Details

Base model. We build on the official SAM3 checkpoint (848M parameters), keeping all weights frozen. Following the official inference setting, frames are resized to 1008×1008 (72×72 memory tokens), with BF16 autocast and no post-processing. Only the memory supplied to the native memory attention, and which query regions may attend to the retrieved long-term memory, are changed. For Section B.4, we use the SAM2.1 Hiera-L checkpoint at 1024×1024. Segmentation experiments run on a single NVIDIA H200 GPU per run.

Hyperparameters. Table 12 lists all hyperparameters; the same values are used for every benchmark. For SAM2.1, all values are identical except the 64×64 memory feature grid and the FGAL distance threshold $\tau_g = 1.10$.

Evaluation. Since SAM3 does not report scenario-aware metrics, we reproduce it with its official code (commit `5dd401d`) and default settings for Table 2, tracking each object independently as in our method, and score it with exactly the same protocol as ours. MOSEv1 and MOSEv2 are evaluated on their official Codabench servers; all other benchmarks, including $\mathcal{J}\&\dot{\mathcal{F}}_d$ and $\mathcal{J}\&\dot{\mathcal{F}}_r$, are scored with the official MOSE toolkit (MOSE-api), which excludes the first and last frames of each video. Following MOSEv2, $\dot{\mathcal{F}}$ uses an object-adaptive boundary tolerance, and $\mathcal{J}\&\dot{\mathcal{F}}_d$ and $\mathcal{J}\&\dot{\mathcal{F}}_r$ are computed over the frames following each disappearance and reappearance of the target, respectively.

D Cost Analysis

Official speed protocol. We follow the official SAM3 speed protocol: a single NVIDIA H200 GPU, batch size 1, frames preloaded, three warm-up rounds, and FPS measured over ten full propagations of the same 50-frame, 1024×1024 synthetic clip with five objects. In the semi-supervised setting, both methods are prompted with the first-frame masks, use the tracker only, and track each object

Table 12: Hyperparameters of the graph memory, reported for SAM3. Management starts only when the node or token count exceeds its trigger and then reduces the memory to the corresponding budget. *Used for both region affinities and retrieval relevance.

Parameter	Symbol	Value
Hierarchical memory organization (Section 3.4, Section 3.6)		
Conditioning memory	M_0	1 frame
Short-term memory size	K_{short}	1 frame
Middle-term memory size	K_{mid}	5 frames
Middle-term admission	–	non-empty predicted mask
Protected interval	K_{prot}	7 observations
Graph-node budget	B_N^{mem}	750 nodes
Graph-node trigger	T_N	1,000 nodes
Token-storage budget	B_T^{mem}	155,520 tokens
Token-storage trigger	–	259,200 tokens
Region partitioning (FGAL, Section 3.4, Section A.1.1)		
Distance threshold	τ_g	1.25 (SAM2.1: 1.10)
Scale parameter	κ	0.3
Minimum region size	$N_{\min}$	30 tokens
Maximum number of regions	$R_{\max}$	100 per frame
Foreground-region ratio	ρ_{fg}	0.10
Relational graph construction (Section 3.4, Section A.1.2)		
Region pooling size*	N_{pool}	64 tokens
CSLS neighbourhood size	k	10
Similarity-edge threshold	τ_{sim}	0.70
Temporal-edge threshold	τ_{temp}	0.65
Modified z-score threshold	τ_z	3.5
Query-conditioned graph retrieval (Section 3.5)		
Recent temporal window	$\lvert\mathcal{H}_t\rvert$	10 frames
Route threshold	τ_{route}	0.8
Seed set size	K_{seed}	10 nodes
Restart probability	α	0.5
PPR iterations	I	10
Retrieval node budget	K_{ret}	200 nodes
Retrieval token budget	B_T	20,736 tokens
Consolidation threshold	τ_{con}	0.25
Object pointers (native + graph)	–	8 + 8
Budget-aware graph memory management (Section 3.6, Section A.3)		
Compaction threshold	τ_{cmp}	0.70
Downsampling window	w_{ds}	2×2 tokens
Anchor compression (Section A.3)		
Anchor matching window	w_{anc}	3×3 tokens
Anchor similarity threshold	τ_{anc}	0.80
Anchor capacity momentum	β_{anc}	0.90
Compressed readout budget	B_T^{compress}	5,184 tokens

independently as in our evaluation, since the graph memory is maintained per object. Both compile only the image encoder, since the fully compiled SAM3 memory attention does not accept the graph-memory interface. Under this protocol, SAM3 and our method run at 8.1 and 4.1 FPS for all five objects (40.3 and 20.7 FPS per object), with peak memory of 5.0 and 5.2 GB, respectively (Table 13). For reference, the official text-prompted SAM3 pipeline, which processes all objects jointly, runs at 18.3 FPS on the same GPU.

Cost over full benchmarks. On the short clip above, the graph memory never reaches its budget (at most 228 nodes), so the official protocol does not exercise memory management, the most expensive part of our method. We therefore also measure cost over all objects of DAVIS-17 val and LVOSv2 val (61 and 239 objects, up to 2,140 frames), following the common practice of reporting cost over full benchmark splits: each object is propagated once after a warm-up, and FPS is the total number

Table 13: Inference speed under the official SAM3 protocol (50-frame 1024×1024 synthetic clip, five objects, single NVIDIA H200 GPU; mean over ten timed rounds). Memory is the peak GPU memory allocated. †Official text-prompted pipeline (detector and tracker, all objects processed jointly), for reference.

Method	FPS (5 objects)	FPS per object	Memory (GB)
SAM3† (text prompt)	18.3	–	6.7
SAM3	8.1	40.3	5.0
Ours	4.1	20.7	5.2

Table 14: Inference cost of SAM3 and our method over all objects of DAVIS-17 val (30 videos, 61 objects) and LVOSv2 val (140 videos, 239 objects) on a single NVIDIA H200 GPU (speed protocol in Section D). FPS: total frames divided by total time. Memory: peak GPU memory allocated, maximum over objects, including decoded frames kept on the GPU. Tokens: memory tokens attended per frame (mean). Graph nodes: the node cap (trigger of memory management) and the peak of deletable nodes, maximum over objects; the nodes of the $K_{\text{prot}} = 7$ protected latest observations are exempt from the cap.

	FPS ↑		Memory (GB)		Tokens / frame		Graph nodes	
Benchmark	SAM3	Ours	SAM3	Ours	SAM3	Ours	Cap	Peak
DAVIS-17 val	38.1	12.7	5.0	6.1	34.1k	42.8k	1,000	999
LVOSv2 val	35.4	8.2	30.0	30.8	36.0k	47.1k	1,000	999

of frames divided by the total time. As shown in Table 14, our method runs at 12.7 and 8.2 FPS, compared with 38.1 and 35.4 FPS for SAM3. It attends to 1.3× more memory tokens per frame and raises peak GPU memory by 1.1 and 0.8 GB, and the number of deletable graph nodes never exceeds the cap.

Memory over long videos. Figure 3 traces memory over the longest LVOSv2 val video (2,140 frames). For both methods, SAM3's per-frame outputs are kept on the CPU, as in our evaluation. Once the graph fills its budget, the GPU memory of our method stays nearly constant, at 1.30–1.45 GB above weights and frames versus 0.72 GB for SAM3, regardless of video length. Without the budget, the same graph memory keeps growing with no plateau: by the end of the video it stores 11M tokens and occupies about 62 GB above weights and frames.

E Proofs of Propositions

Setting. We track one target, prompted on frame 0 and propagated forward; objects are tracked independently. Each memory frame has $H_m W_m = 72 \times 72$ spatial tokens. SAM3 scores frame s by

$$q_s = \operatorname{mean}\big(\hat{o}_s \cdot \mathrm{IoU}_s\big), \qquad \hat{o}_s = \begin{cases} 2\sigma(\ell_s) - 1, & \ell_s > 0, \\ 0, & \text{otherwise}, \end{cases}$$

where IoU_s is the predicted IoU and ℓ_s the object-score logit. We use the defaults of the official SAM3 tracker, in which memory selection is enabled: the memory-selection threshold $\tau_{\text{mf}} = 0.01$, $K = 6$ non-conditioning memory slots, and $K_p = 15$ non-conditioning object pointers. For our method, symbols are as in Sections 3.3 to 3.6 and Table 12; $T_N = 1{,}000$ is the graph-node trigger and $I = 10$ the number of PPR iterations.

E.1 Proof of Proposition 1

(a) Native SAM3. At frame t, SAM3 scans the stored non-conditioning frames backwards from $t-1$, keeps each frame with $q_s > \tau_{\text{mf}}$ until $K'_p = \min(K_p, T-1)$ frames are kept, where T is the number of video frames, and appends $t-1$ if it was not kept. The resulting list is ordered in time and ends with $t-1$. It contains the K'_p most recent passing frames, or all of them if fewer exist. The K spatial-memory slots read its last K entries, or all entries if there are fewer. If $K \le K'_p$, these are

exactly $\mathrm{Top}_K(\mathcal{O}_t)$; otherwise the video has at most K frames besides frame 0, the list contains all of $\mathcal{O}_t$, and again the entries read are $\mathrm{Top}_K(\mathcal{O}_t) = \mathcal{O}_t$. With a single prompt, the conditioning-frame selection returns frame 0. Each frame's memory is encoded only from its own image features and predicted mask, and memory attention receives only the memories of the selected frames. Hence $\mathcal{S}_t = \{0\} \cup \mathrm{Top}_K(\mathcal{O}_t)$. If $s \in (0, t)$ and $|\mathcal{O}_t \cap (s, t)| \geq K$, all K frames of $\mathrm{Top}_K(\mathcal{O}_t)$ are later than s, so $s \notin \mathcal{S}_t$. □

Remarks.

- The object pointers read at t belong to frame 0 and to at most K_p recent frames of $\mathcal{O}_t$; they are global vectors without spatial appearance.
- A frame with $q_s \leq \tau_{\mathrm{mf}}$, e.g., one in which the target is predicted absent, is not read once it is no longer the previous frame.
- Frame s may still influence later frames indirectly, through masks and pointers computed while it was read; the proposition concerns direct access to its stored appearance.

(b) Our method. The construction in Sections 3.3 to 3.6 gives four properties.

(P1) *Admission.* When a frame leaves the short-term memory, FGAL partitions its grid into between 1 and $R_{\max}$ regions, and every region, foreground or background, is inserted into $\mathcal{G}$. No other admission test is applied.

(P2) *Removal.* A region leaves $\mathcal{G}$ only through one of three operations: (i) temporal consistency filtering, which removes it when its observation leaves the K_{prot} most recent observations without any accepted temporal association, and never removes a validated region; (ii) redundancy compaction, which keeps only the highest-utility unit of a trajectory segment whose cumulative path affinity stays at least τ_{cmp}; (iii) utility-guided eviction, which removes unprotected storage units, each trajectory offering only its oldest remaining unit and lower utility first, until at most B_N^{mem} unprotected regions remain. Operation (ii) is invoked only when the unprotected region count or the stored-token count exceeds its trigger, and (iii) only when the unprotected region count reaches T_N. Token downsampling reduces background tokens but removes no region.

(P3) *Accessibility.* At frame t, every region of $\mathcal{G}$ either belongs to an observation read in full as conditioning, short-term or selected middle-term memory, or belongs to the candidate set that retrieval scores against the current query.

(P4) *Bound.* Management runs after every insertion. When at least T_N unprotected regions exist, eviction reduces them to at most $B_N^{\mathrm{mem}} < T_N$, so at most $T_N - 1$ remain after each frame. Before the next management step, the observation leaving the protected interval adds at most $R_{\max}$ unprotected regions, and the K_{prot} protected observations hold at most $K_{\mathrm{prot}} R_{\max}$ regions. Hence

$$|\mathcal{V}_t| \leq T_N + (K_{\mathrm{prot}} + 1) R_{\max} = N_{\max} \quad \text{for all } t.$$

The bound on $|\mathcal{V}_t|$ is (P4). Let v be a temporally validated region of frame s. By (P2), rule (i) never removes v, so v can only be removed by redundancy compaction or utility-guided eviction, each of which is invoked only at a frame where a management trigger is exceeded. Before v enters $\mathcal{G}$, its tokens are read as short-term memory; afterwards, $v \in \mathcal{V}_t$ for every $t < t_v$ and is accessible by (P3). Hence v is accessible at every $t \in (s, t_v)$. □

Remark (storage). By (P1), every frame excluded in (a) is stored as its regions. An observation holds at most $H_m W_m$ tokens, since its regions partition the grid and downsampling only removes tokens. Evicted middle-term observations are kept only as attachments for middle-term readout, at most K_{mid} of them, and each trajectory has at most one anchor of at most $H_m W_m$ tokens, which is retired once the trajectory has no remaining storage unit. The stored memory therefore never exceeds $(2N_{\max} + K_{\mathrm{mid}}) H_m W_m$ tokens.

E.2 Proof of Proposition 2

All quantities other than N_t are constants of the model or the budget. Per frame, retrieval performs the following steps.

- *Routing*: the region partition of the query and its stability over the recent window $\mathcal{H}_t$; this depends only on $H_m W_m$, $R_{\max}$ and $|\mathcal{H}_t|$, not on N_t.
- *Scoring*: each region has at most N_{pool} pooled keys, recomputed only for new or modified regions, $O(N_t N_{\text{pool}} d^2)$; at most $H_m W_m$ routed query tokens are scored against all keys, $O(H_m W_m \, N_t N_{\text{pool}} d)$, and ranked, $O(N_t \log N_t)$.
- *Personalized PageRank* on the dense transition matrix: $O(N_t^2)$ to build and $O(I \, N_t^2)$ for I iterations.
- *Consolidation*: one pass over each trajectory, $O(N_t)$.
- *Readout*: at most K_{ret} regions and $B_T + H_m W_m$ tokens, since the last selected region may exceed the token budget by one region.

The dominant terms give, for every t,

$$\begin{aligned} C_t &= O\big(N_t N_{\text{pool}} d \, (H_m W_m + d) + I \, N_t^2\big) \\ &\le O\big(N_{\max} N_{\text{pool}} d \, (H_m W_m + d) + I \, N_{\max}^2\big), \end{aligned}$$

using $N_t \le N_{\max}$ from Proposition 1 (b). Taking the supremum over t gives the second claim; the right-hand side does not depend on the video length. □

Remarks.

- Graph construction compares the at most $R_{\max}$ new regions with each other and with the trajectory ends, $O(R_{\max}(N_t + R_{\max}) N_{\text{pool}}^2 d)$, and management costs $O(N_t \log N_t)$; both are bounded in the same way.
- Memory attention reads the conditioning memory, the short- and middle-term observations, the retrieved long-term regions, and the anchors, i.e., at most $(K_{\text{short}} + K_{\text{mid}} + 2) H_m W_m + B_T + B_T^{\text{compress}}$ memory tokens per frame, plus object pointers.
- Without the budget, compaction and eviction are never invoked, so every validated region is kept. If a constant fraction of each frame's regions is validated, $N_t = \Theta(t)$ and the PPR cost grows as $\Theta(I \, t^2)$. On `uXO4RUO3`, the unbounded graph grows by about 17 regions per frame and stores 10.8M tokens at the end (Fig. 3), and its time per frame rises from 0.16 s to 0.94 s.

F Extended Related Work

Memory-based VOS. Semi-supervised video object segmentation acquired its memory-based form with space-time memory networks, which segment the current frame by densely matching its features against stored past frames (Oh et al., 2019; Cheng et al., 2021), and later work has refined every part of that recipe (Yang et al., 2021; Yang & Yang, 2022; Seong et al., 2021; Cheng et al., 2024). On the organization of the store, XMem (Cheng & Schwing, 2022) maintains sensory, working, and long-term stores and consolidates frequently used working elements into the long-term store. The hierarchy follows storage timescale and usage, and the stored elements remain per-location features without explicit relations. On how the store is read, RMNet (Xie et al., 2021) matches locally against memorized regions to reduce the ambiguity of similar objects. On what the store retains, RMem (Zhou et al., 2024) bounds the memory bank to a fixed number of frames since redundant memory confuses feature decoding, QDMN (Liu et al., 2022) admits only accurately segmented frames and QDMN++ (Liu et al., 2025) refines the stored features, CSNet (Qiao et al., 2026) learns when and how to update the memory, and ASM (Dang et al., 2025) admits past frames sparsely over time. Across these designs, the unit of storage and retention remains the frame or its per-location features, and the relations among the target's partial appearances stay outside the model. MEMOIR retains the memory-based recipe while changing that unit: long-term history persists as regions linked by their relations, and management and retrieval operate on this structure rather than on whole frames.

Memory management in SAM-style models. This family meets the same retention question in a promptable streaming design. SAM2 (Ravi et al., 2024) set the template the family still follows, and SAM3 (Carion et al., 2025) leaves it largely intact while strengthening the surrounding model. Later work adjusts the template at different stages: SAM2Long (Ding et al., 2024) works at the propagation

stage, and DAM4SAM (Videnovic et al., 2024) and SENTRY (Alansari et al., 2026) screen the bank's entries against distractors and over time respectively. Beyond the stored frames, SeC (Zhang et al., 2025) supplies an object-level concept of the target, and SAAS (Hu et al., 2025) handles multi-shot scene transitions. Further variants refine the bank's writing rules with motion-aware selection (Yang et al., 2024), prototype calibration (Xu et al., 2025), and short-/long-term splits (Chen et al., 2025; Kim & Lim, 2026). A recent study surveys these strategies across the family (Alansari et al., 2025). In every case, the decision still takes an intact observation as its object, and the target's parts inside an observation never become separately manageable evidence. MEMOIR changes this object while retaining the native interface. Long-term history enters the native memory attention as related regions, and what persists and what reaches the current prediction are decided as two separate questions.

Structured memory and graph retrieval. Structured memory organizes stored units through explicit relations: within VOS, EGMN (Lu et al., 2020) builds a fully connected graph over stored frames and learns read and write controllers, and DTM-VOS (Fan et al., 2024) links local regions of neighboring frames in a high-order graph that refines features within a short temporal window rather than organizing the full memory for retrieval. Outside segmentation, graph-based retrieval establishes what relations contribute at query time: HippoRAG (Gutiérrez et al., 2024; 2025) propagates the query signal over a graph with Personalized PageRank, and SelectStream (Ge et al., 2026) conditions retrieval on the query over a bounded latent graph in a streaming setting. MEMOIR joins both lines in a segmenter that neither line addresses. Its nodes are spatially coherent regions drawn from the segmenter's own memory tokens, its edges are typed by spatial adjacency, feature similarity, and temporal correspondence, and no controller is learned. Seeds come from direct matching, relevance propagates over the typed edges, and retrieved regions re-enter the native memory attention as the segmenter's own tokens with their spatial positional encodings. The graph thus preserves within-observation distinctions that a frame-level graph cannot express, and its typed relations extend retrieval to evidence that direct matching alone would miss, all at no training cost to the base model.

G LIMITATIONS

MEMOIR trades speed for long-term evidence: the graph operations run at roughly one quarter to one third of SAM3's throughput (Table 14), although the retrieval cost stays bounded as videos grow. Its gains concentrate on long videos with prolonged absence, and on short benchmarks such as DAVIS 2017 it remains close to, and occasionally slightly below, the base model. The graph operations are governed by fixed thresholds, which are shared across all benchmarks and, apart from the partition threshold τ_g, across both base models; learning or adapting these thresholds is a natural direction for future work.