NeurIPS 2026

What Should a Streaming Video Model Remember?

SelectStream: selective latent memory for streaming video understanding

Haonan Ge[1,2] Yiwei Wang[2] Hang Wu[3] Yujun Cai[2†]

[1]UC Santa Barbara [2]Southeast University [3]UC Merced [†]Corresponding author

haonan-ge.github.io/selectstream github.com/Haonan-Ge/selectstream

Project page



THE SETTING

Answer any time, from the past, on a budget

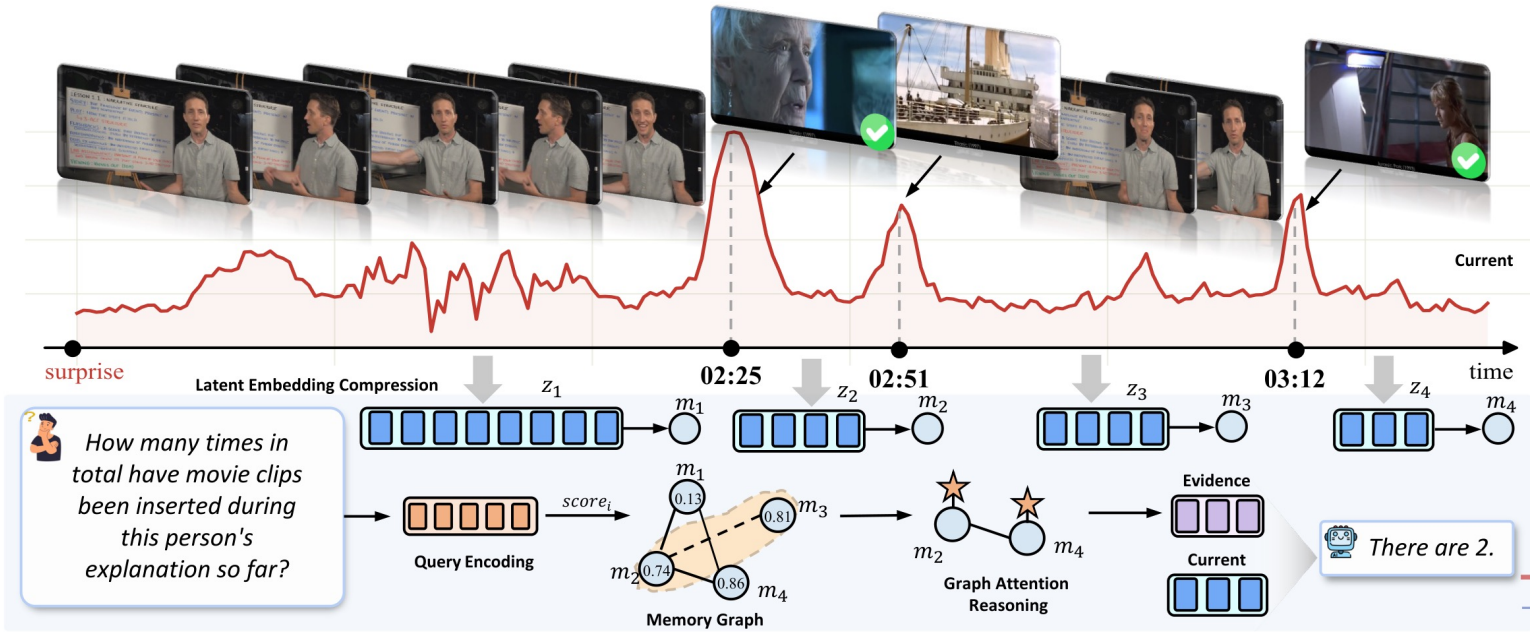


Figure 1 example. *To count inserted movie clips, the model needs three sparse surprise peaks (02:25, 02:51, 03:12), long after they leave any recent window.*

1 **Any-time queries**
A question can arrive at any moment of the stream.

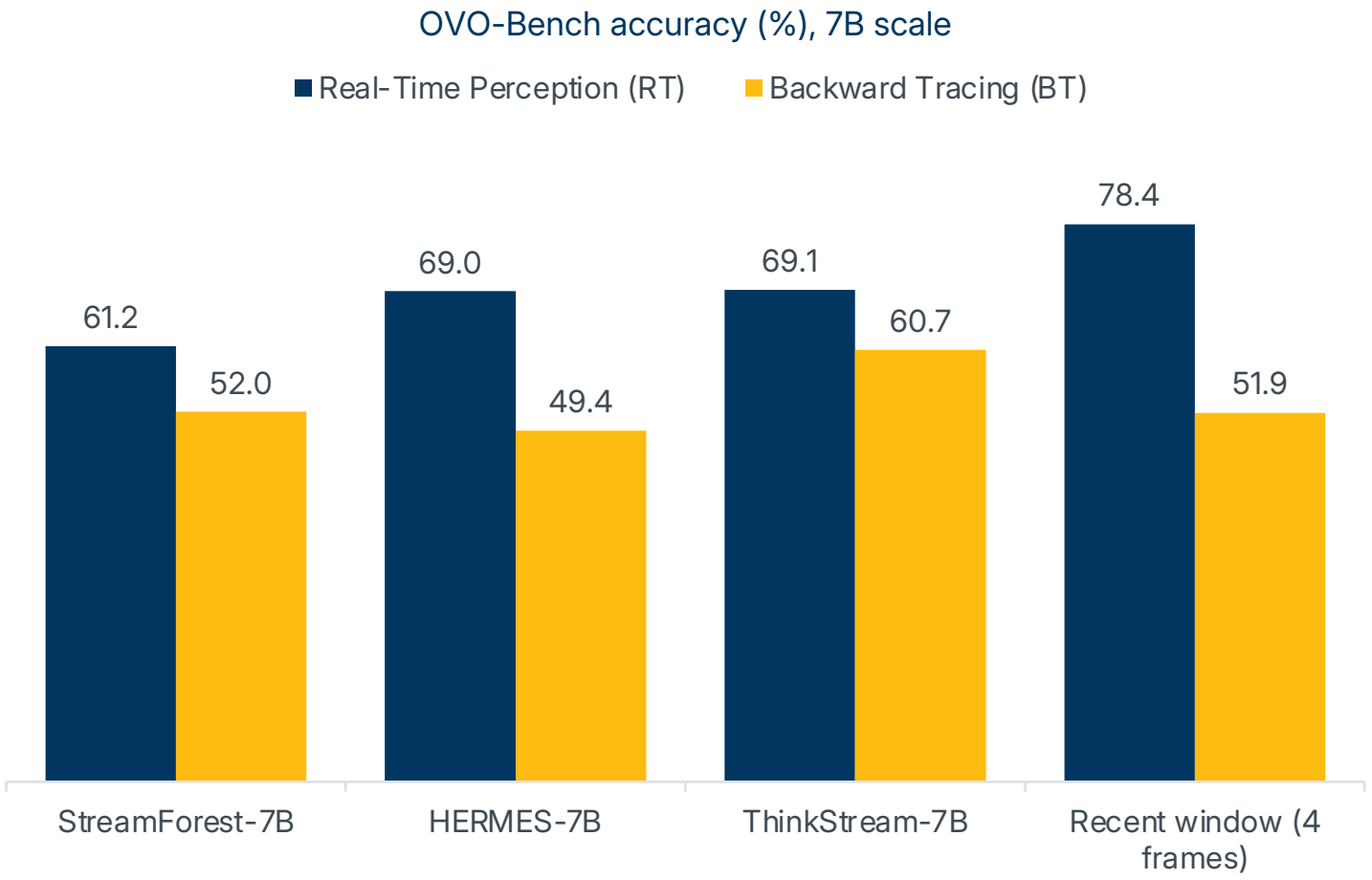
2 **Causal input**
Only frames observed so far; nothing from the future.

3 **Fixed budgets**
Memory, context, and compute stay bounded as the stream grows.

So what should the model keep, and what should it show the VLM?

THE PUZZLE

A recent window rivals memory-heavy models



Recent window

Sharpest present: best RT (78.4). But evidence that leaves the window is gone (BT 51.9).

Memory-heavy models

Can recover more history (BT up to 60.7), but indiscriminate history dilutes the present (RT ≤ 69.1).

Not whether to use memory, but how to allocate it selectively.

OUR VIEW

Memory as budgeted evidence allocation

Keep the present directly visible. Let history in only through a small, query-selected evidence budget.

When to write?

SAW

Surprise-driven Adaptive Windowing

Fine segments at surprises, coarse ones in stable stretches.

BUDGET
L_{min}, L_{max}, B_s

What to keep?

LVM

Latent Visual Memory

Gated writes into a graph; priority-aware merging when full.

BUDGET
N memory nodes

How to read?

GAR

Graph Attention Reasoning

Route each query through a small, relevant subgraph.

BUDGET
B subgraph nodes

How much to show?

Inject

Latent evidence injection

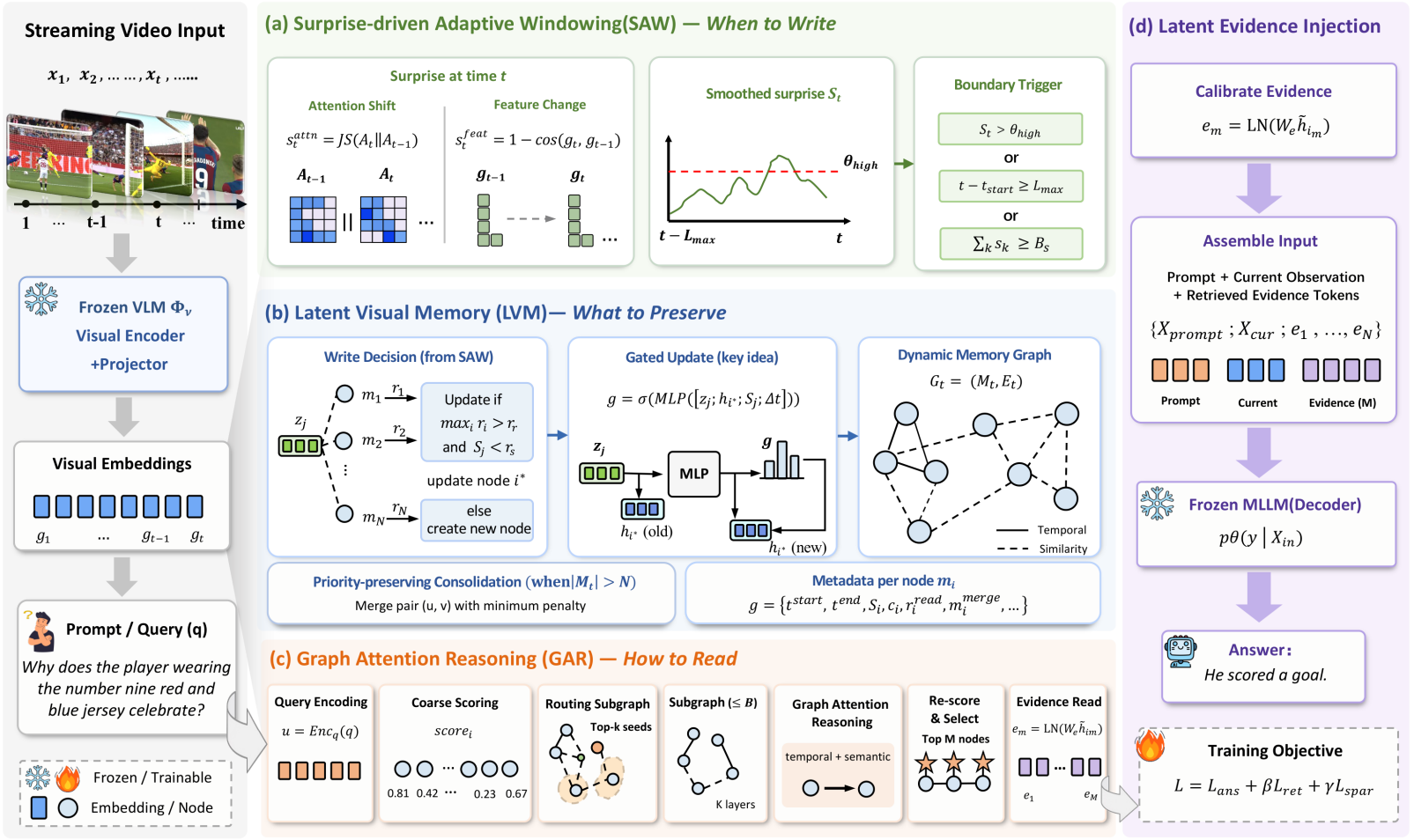
Calibrated latent tokens next to the prompt and current frame.

BUDGET
M evidence tokens

N, B, and M are separate user-facing knobs, and none of them grows with stream length.

METHOD

SelectStream at a glance



SAW

When to write

Cut segments at surprise spikes.

LVM

What to keep

Gated writes into ≤ N nodes.

GAR

How to read

Query routed through ≤ B nodes.

Inject

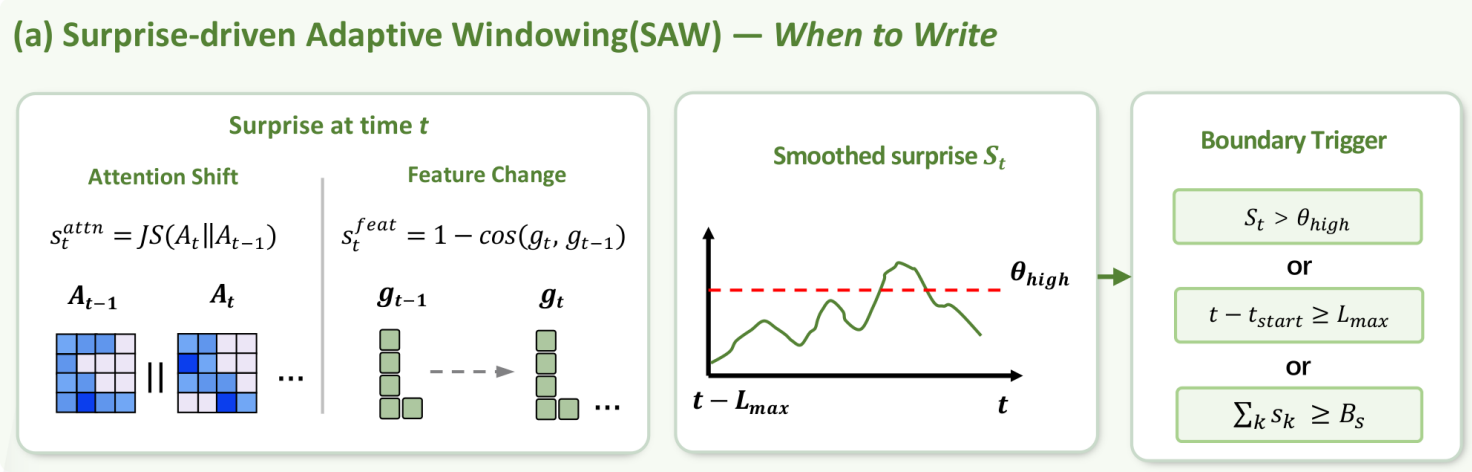
How much to show

Feed M calibrated latent tokens.

Backbone VLM frozen; only the SelectStream modules are trained.

METHOD · WHEN TO WRITE

Write finely where the stream surprises



Surprise signal. Attention shift (JS divergence) plus feature change between consecutive frames.

Adaptive cut. Close a segment on a spike, on accumulated surprise, or at the maximum length.

Effect. Stable stretches get coarse memory; transitions get fine memory.

$$s_t = \lambda \, \mathrm{JS}(A_t \,\|\, A_{t-1}) + (1-\lambda)\,(1 - \cos(g_t, g_{t-1}))$$

$$\text{close segment if } \quad \bar{s}_t > \theta_{\text{high}} \;\vee\; \sum_k \bar{s}_k > B_s \;\vee\; t - t_{\text{start}} \geq L_{\max}$$

Proposition 1

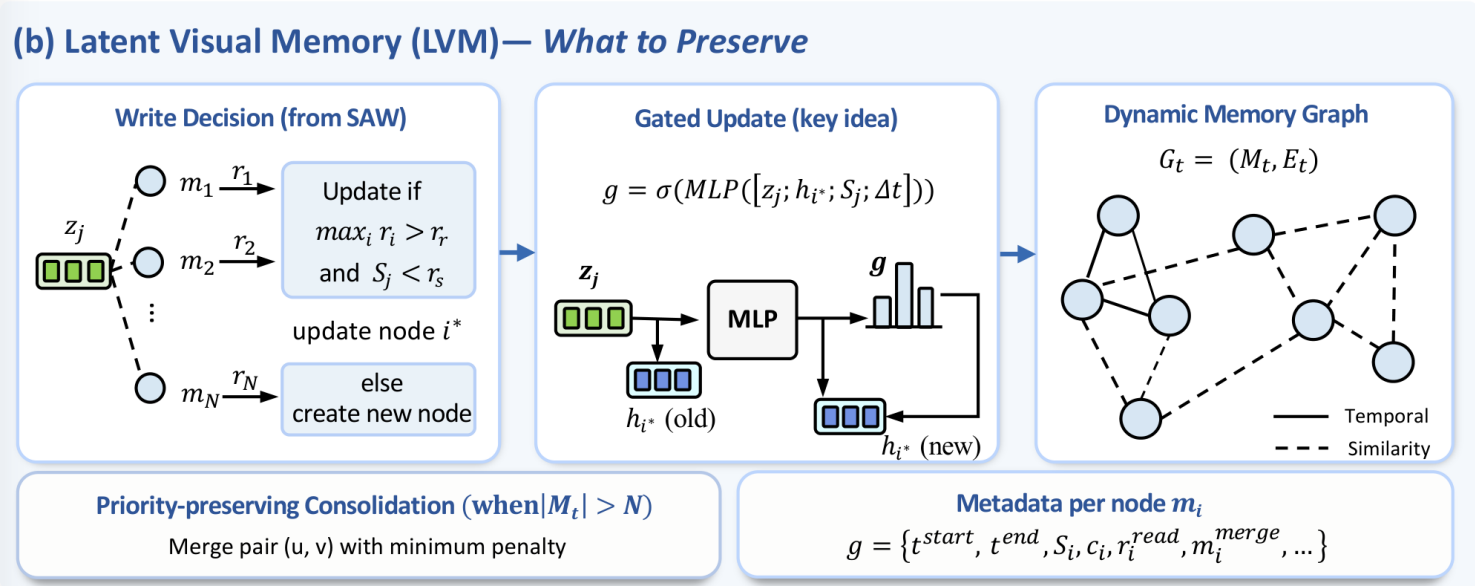
Writing stays budget-controlled

$$S_T \leq \frac{\sum_t \bar{s}_t}{B_s} + \frac{\sum_t \bar{s}_t}{\theta_{\min}} + \frac{T}{L_{\max}} + 1$$

and consolidation keeps $|M| \leq N$.

METHOD · WHAT TO KEEP

Keep useful history within N nodes



Gated write. A segment similar to an existing node updates it through a learned gate; otherwise it becomes a new node.

Graph. Temporal edges link events; similarity edges link recurrences.

Priority merging. Over budget, merge the cheapest pair; surprising, often-read and recent nodes are protected.

$$g = \sigma(\mathrm{MLP}([\, z_j;\ h_{i^*};\ \bar{s}_j;\ \Delta t\,]))$$

$$h_{i^*} \leftarrow (1-g)\, h_{i^*} + g\, f_{\mathrm{write}}(z_j, h_{i^*})$$

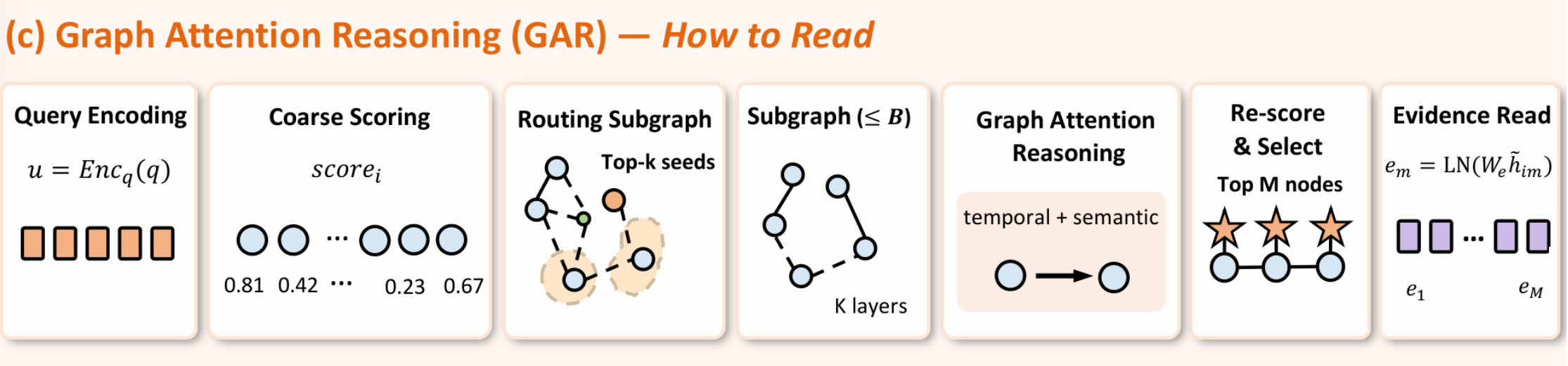
$$\pi_{uv} = p_{uv}^{\mathrm{sim}} + p_{uv}^{\mathrm{pri}} \quad \text{when } |\mathcal{M}_t| > N$$

+2.00

StreamingBench gain of priority merging over FIFO (81.42 vs. 79.42)

METHOD · HOW TO READ

Route each query through a small subgraph



1 Score & route

$$\mathrm{score}_i = \cos(u, h_i) + \eta\,\hat{s}_i - \xi_\ell\,\hat{\ell}_i - \xi_m\,\hat{m}_i^{\mathrm{merge}}$$

$$\psi_{ij}(q) = \mathrm{score}_j + \alpha_r\,w_{ij}$$

Top-k seeds expand along temporal and semantic edges; keep at most B nodes.

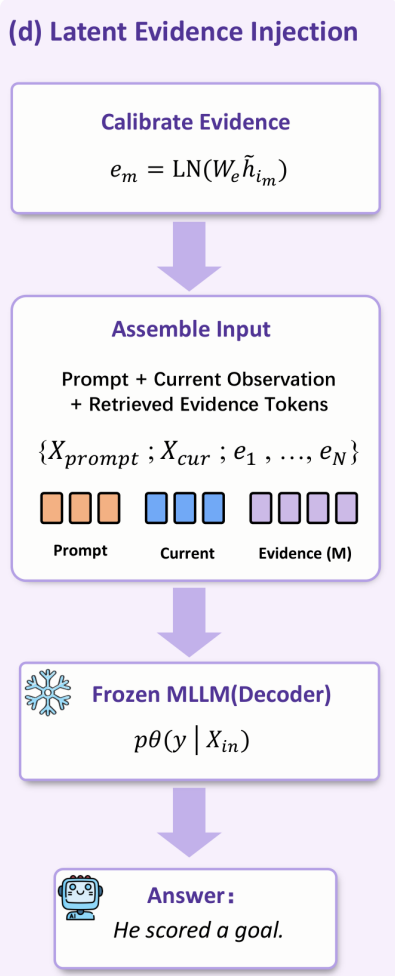
2 Reason & select

$$e_{ij} = q_i^{\top} k_j + b_{\mathrm{type}} + b_{\Delta t} + b_w$$

K layers of relational graph attention, re-score with the same terms, keep the top-M nodes as evidence.

METHOD · HOW MUCH TO SHOW

Show the VLM only M calibrated tokens



$$X_{\text{in}} = [\, X_{\text{prompt}};\ X_{\text{cur}};\ e_1, \ldots, e_M \,]$$

$$e_m = \mathrm{LN}(W_e\, \tilde{h}_{i_m})$$

Latent, not replayed. No frames re-encoded and no text summaries; history stays in latent space.

Calibrated. W_e corrects the drift from segment encoding, merging and graph reasoning.

Bounded. Context size is set by M, not by stream length.

Training

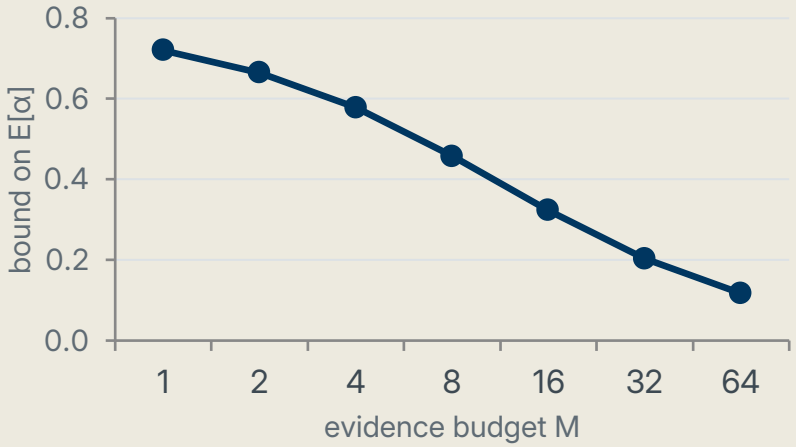
$$\mathcal{L} = \mathcal{L}_{\text{ans}} + \beta\, \mathcal{L}_{\text{ret}} + \gamma\, \mathcal{L}_{\text{spar}}$$

Backbone frozen · Streamo-Instruct-465K

Why a small M helps

Proposition 2 · evidence attention concentration

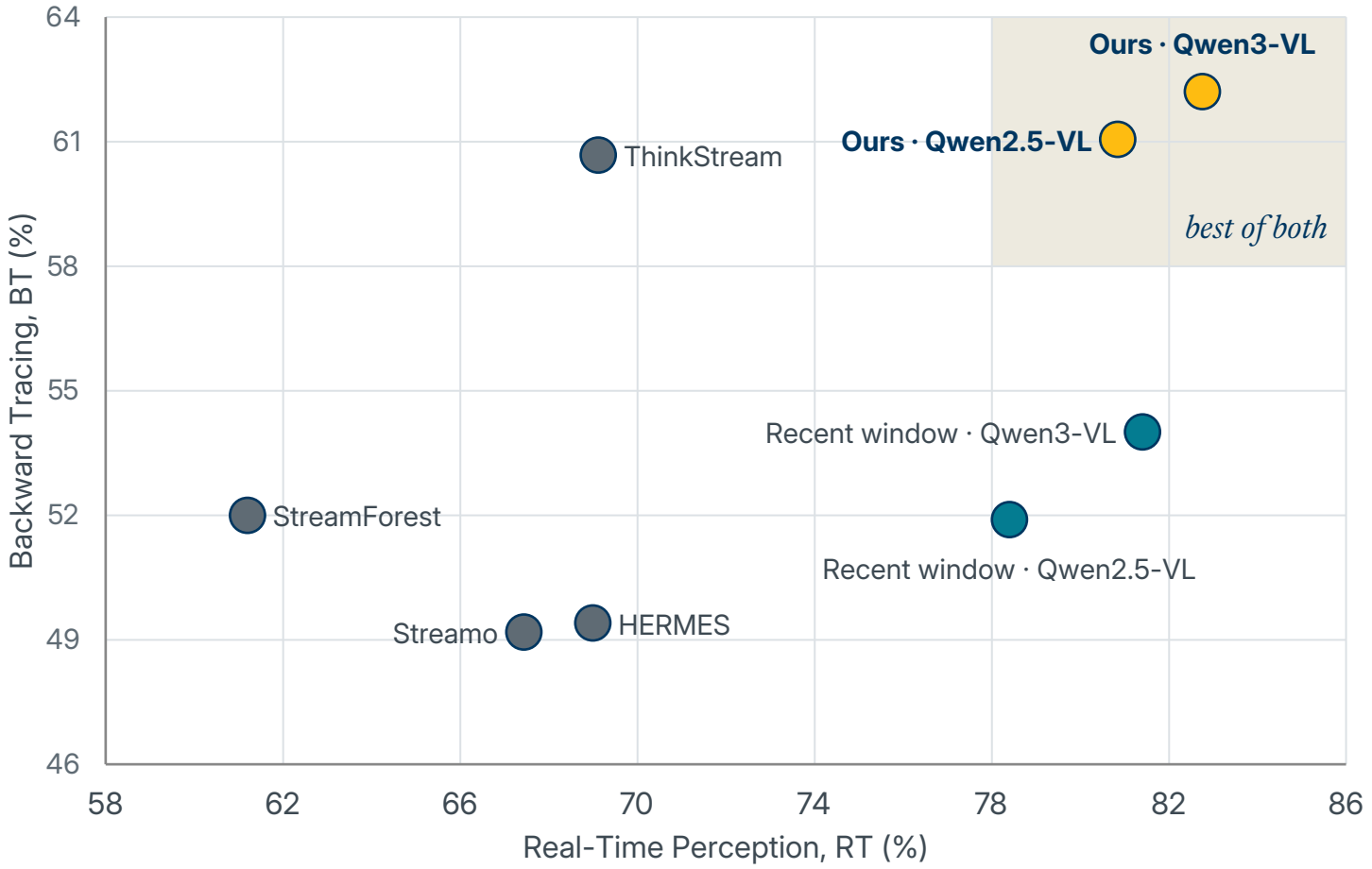
$$\mathbb{E}[\alpha_{h^*}] \;\geq\; r \cdot \frac{e^{\mu_{\text{sig}}}}{e^{\mu_{\text{sig}}} + (M-1)\, e^{\sigma^2/2}}$$



Fewer distractors leave more attention on the relevant evidence. *Curve: r = 0.72 (measured Recall@M), μ = 3, σ = 1 for illustration.*

RESULTS · STREAMING

Sharp present and long memory, together



82.67%

StreamingBench, best among all compared methods

+9.15 / +8.20

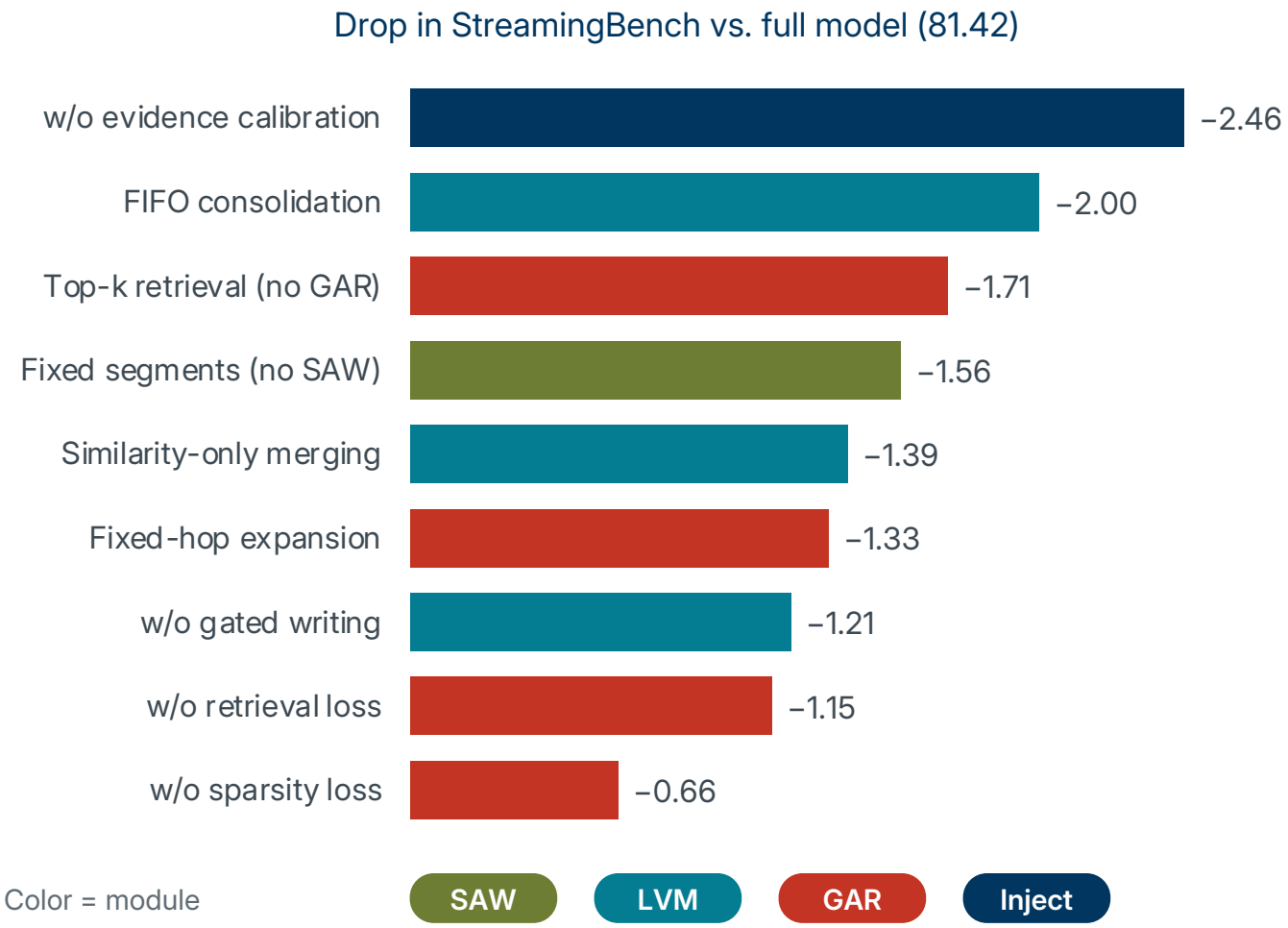
BT over the recent-window baselines (Qwen2.5-VL / Qwen3-VL)

67.03%

OVO-Bench overall (Qwen3-VL-8B)

RESULTS · ANALYSIS

Every component pulls its weight



Offline skills stay intact

Causal final-query protocol

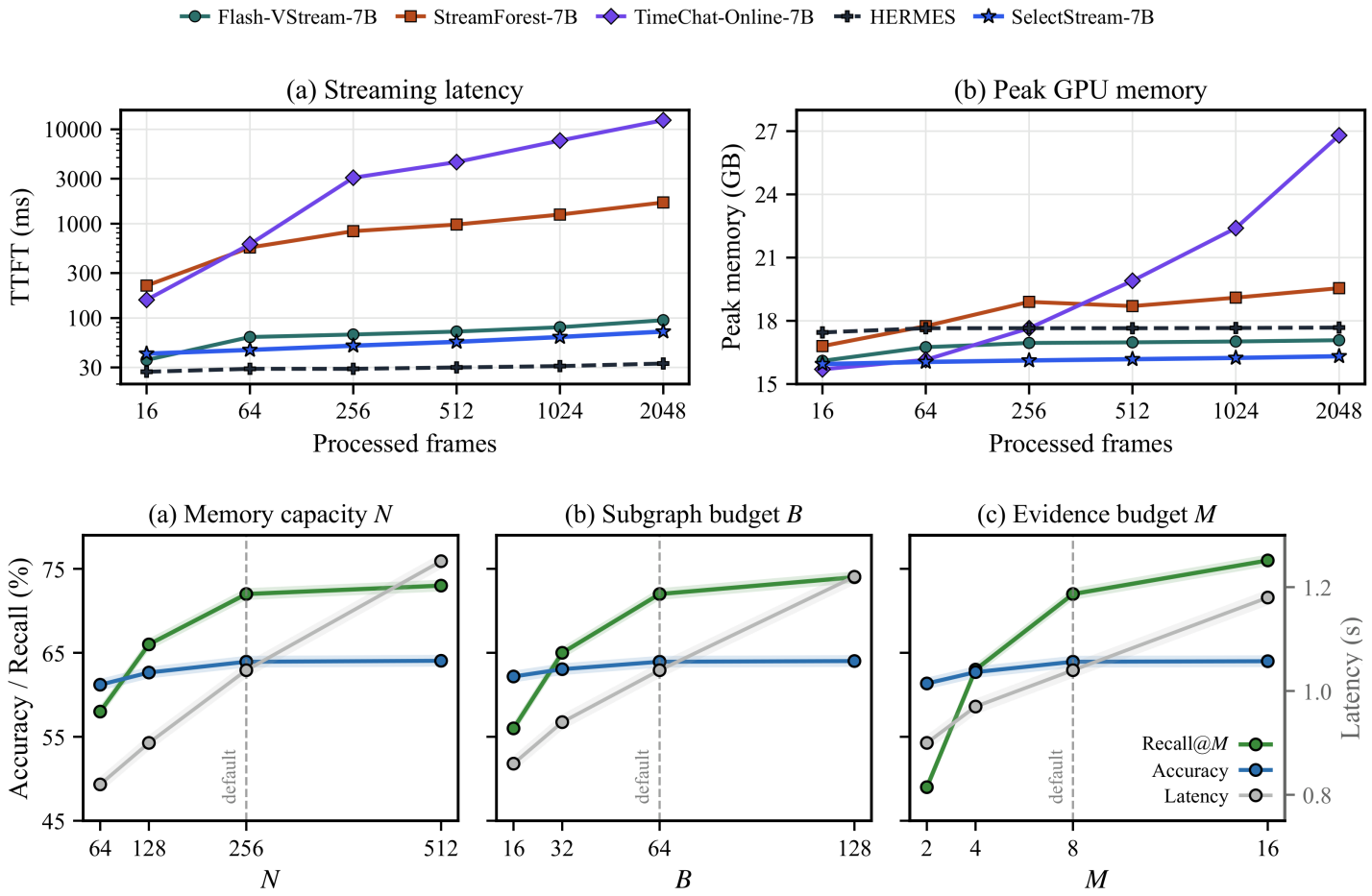
	Video-MME	MLVU	MVBench	Avg.
Qwen2.5-VL-7B	65.1	70.2	69.6	68.3
+ SelectStream	**67.8**	**73.0**	**70.4**	**70.4**
Qwen3-VL-8B	71.4	78.1	68.7	72.7
+ SelectStream	**73.2**	**80.0**	**69.9**	**74.4**

74.4%

average with Qwen3-VL-8B (+1.7). Gains are largest on longer videos (Video-MME, MLVU).

EFFICIENCY

Cost stays flat as the stream grows



Flat footprint. At most N active nodes; each query touches at most B nodes and injects M tokens. Active memory is O(1) in stream length.

Three separate knobs. N sets retention, B sets retrieval recall, M sets how much evidence (and latency) reaches the decoder.

Honest trade-off. HERMES has the lowest TTFT; SelectStream pays a small startup cost for scoring, routing and calibration, then grows slowly.

Takeaways

1 **Streaming memory is an allocation problem:** when to write, what to keep, how to read, and how much to show.

2 **Keep the present visible;** let history in only as M calibrated latent tokens, selected per query.

3 **82.67% on StreamingBench (#1),** +9.15 BT over recent windows, with nearly flat latency and memory.

Project page

haonan-ge.github.io/selectstream
github.com/Haonan-Ge/selectstream

Thank you!

